

GAME DEVELOPER MAGAZINE

FROM GAME EXPO
TO GAME JOB

Indie intro
to VR



GAME CAREER GUIDE



EDUCATION THAT **TOPS** THE CHARTS

Our awards don't stop with being one of the
★ **HIGHEST RANKED** ★ game programs in the world.

At SMU Guildhall, you'll learn from a faculty of award-winning industry pros and veterans, produce award-winning games, and you may even go on to be one of our many award-winning alumni.

Learn more at guildhall.smu.edu



#2 GRADUATE PROGRAM

ART CREATION • LEVEL DESIGN • PRODUCTION • PROGRAMMING



SMU GUILDHALL
GAME DEVELOPMENT EDUCATION

JULY 2016

CONTENTS



GAME CAREER GUIDE

DEPARTMENTS

4 EDITOR'S NOTE

IT'S ALL ABOUT TASTE!

96 FREE TOOLS

FREE DEVELOPMENT TOOLS 2016

104 ARRESTED DEVELOPMENT

INDIE DREAMIN'

STUDENT POSTMORTEM

32 BEGLITCHED

NYU Game Center students Alec Thomson and Jennu Jiao Hsia discuss their IGF Award-winning match three game about insecurity within computers, and within ourselves.

FEATURES

8 BREAKING IN THE EASY(ISH) WAY!

How attending expos can land you a job.

18 ZERO TO HERO

Hey! You want to learn low poly modeling but don't know where to start? Look no further! With this guide, we hope to provide a good introduction to not only the software, but also the concepts and theory at play.

34 DO YOU NEED A PUBLISHER?

In the old days, a publisher was the only way to get your game to market. In the current climate of self publishing, do you really need to work with one? This primer may help you decide!

53 GAME SCHOOL DIRECTORY

There are tons of options out there in terms of viable game schools, and this list is just the starting point to get you acquainted with the schools near you (or far from you, if that's what you prefer!).

72 VIRTUALLY DESIGNED

VR has quickly moved from buzzword, to prototype, to viable business. This guide will help you get started in VR development, avoiding some common pitfalls.

78 SOUNDS GOOD TO ME!

Advice for making audio (with or without) an audio specialist.

84 A SELLER'S MARKET

Marketing fundamentals for your first game.

90 INTRO TO GAME ENGINES

A brief discussion of some of the newest and most popular game engines.



IT'S ALL ABOUT TASTE!

YOUR INTRO TO THE CAREER GUIDE

HI FRIENDS! WELCOME TO ANOTHER EDITION OF THE ANNUAL CAREER GUIDE! The aim of this publication is to help increase your game dev know-how, whether it's through our primer on 3D modeling, helping you sort out whether you need a publisher, or advice on how to land a job by attending an expo.

I'd like to use these early pages to expand on two things, if you'll indulge me. The first is when to say no to a publisher, because I feel like almost nobody talks about that. Secondly, I want to talk to you about taste and critique. It's important, but a lot of games are made without it.

Publishers

I am, at any given time, pitching a game or two to publishers. As an indie game dev myself, with a studio called Necrosoft Games (www.necrosoftgames.com), we're always looking for that next source of funding, or a fancy way to market ourselves, or some other such thing that a publisher traditionally provides.

But when it comes down to it, not all publishers are actually useful beyond the money they provide. And if they're not useful beyond the money they provide, how much do you really need them?

Selling a game is more difficult than ever. There's no captive audience like there may have been in the Sega Genesis days. If that was the console you had—well, your grandma was buying you the new Batman

game for Genesis, because she heard you liked that movie and she knows you've got the console. It was in this sort of environment that traditional publishers thrived. They paid money for a game, and they knew they could sell it because they knew where the people were.

Now, more games are coming out than ever. You can't guarantee Grandma knows how to gift you a Steam code, and she certainly doesn't know which particular game you want, because it's not plastered all over the inside of a Toys "R" Us.

Some publishers combat this problem with innovative marketing. Companies like TinyBuild and Devolver use streamers, YouTubers, and social media to hype games before they launch. This actually works.

But most other publishers still use static press releases, a few game trailers, and an expo booth as their big marketing drive. This just doesn't work anymore. Friends of mine have brought their games to over a dozen expos, from PAX to E3, and it didn't affect their sales. Why? Because that's all they did,

nothing more. You've got to have a multipronged approach to marketing, and many publishers simply don't have the expertise to do it.

Another useful type of publisher is one with a built-in audience. Say you've got a JRPG. Atlus, Square Enix, or XSEED might be the publisher for you. Simply by putting their name on your game, you're likely to sell more, because they've got a fan base looking for the type of game you're making. In this case, it doesn't matter as much whether or not they do any aggressive marketing, because their fans are already looking for that kind of game from them.

But then you've got the other sort of publisher. The one I mentioned before, who gives you a bit of money, takes a recoup and a revenue share, and doesn't promote your game in any way that's going to help it sell. This is, unfortunately, most publishers out there today. These tend to stay in business with their bigger marquee titles and use indies as "try it and see" sorts of investments that they hope will hit—but will leave them to flop if they don't.

Marketing needs to be a

continuous push, even after launch. If a publisher can't or won't do that, it's not really going to help you do anything other than pay for exactly how much your game costs to make. And let's be clear, that is not enough money. The sales from that game have to pay you at least until you can get another deal, or ideally it makes you enough to fund your next game yourself.

If the only publishers who are interested in your game are this sort, you're might be better off releasing your game on your own. At least you get all the profits you earn and can live or die by your own skill. We've come to the conclusion that we can market our own game better than most publishers. Not nearly as well as the good ones, but... there just aren't that many good ones, so if none of those wind up being interested, might as well skip that whole thing.

So how the heck do you identify the good ones? Here's a quick primer.

- 1) Have you heard of them? If so, see what their fan base likes. If you can identify that and it lines up with what you're making, maybe it's a good match.
- 2) Read articles written by their staff, about marketing, publishing, and so forth. Are they trying new things? The fact that they're writing articles about marketing is already a good sign. But study what they're doing. If it seems interesting (read up on TinyBuild's Punch Club initiative for a good example), they're probably pretty good.
- 3) Do they have a good rep? Ask your developer pals whether they're good to work with. This is pretty important.

illustration by Juan Ramirez



Other than that, go with your gut! And be sure to read Adam Saltsman's guide to publishers for more about this.

Good taste

There are so many bland and flavorless games in the world! Let's not make too many more of them. This is a weird and controversial subject, but I believe that in order to make something good, you have to have taste. "Good" taste, "bad" taste, it doesn't matter; you've got to like certain things—and know why you like them—in order to make something that other people are going to like or dislike. Otherwise, they'll just pass you by.

To have taste, you have to be able to think about thinking. You have to know not only that you like things, but why you do. You've got to be able to be self-critical. This does not mean self-deprecation! I see lots of folks talk about how all

real artists are self-deprecating, but that is wrong-headed. You can think you're pretty much alright, as long as you're never complacent. Hot Topic goths being down on themselves don't necessarily make good artists. But nondestructive self-analyzers often do.

So really, it's about analyzing why you like the things you like. Let's say you're in your late 20s. You like *Symphony of the Night* because you grew up in the 1990s, and your older sibling passed it down to you. It feels like it was the best game ever and nobody ever made anything else like it, but really it was your best game ever, because of where you were in that time, what happened to you, what your life was like, and how this game slotted into it. You have to analyze that, not just accept the idea that "games were better back then."

I showed a bunch of folks a screenshot of a game I'm working on. Folks who were a bit older said it looked like a *Persona* game. Younger folks said it reminded them immediately of *Mega Man Battle Network*. The important thing is not the particular reference but that you understand what it is about the

image that resonates with you. What was it about that time that was meaningful, and elicited that excited response?

Really, our image is just an isometric school scene. It's not related particularly to any of those other games. But when you've invested time and energy into playing something like *Persona II* or *Mega Man Battle Network*, it sticks with you, and you start looking for more things like it. This is worth analyzing.

So here are some exercises to help cultivate your self-criticism. They may seem stupid, but they'll get you thinking.

- 1) Write three sentences about why you like your top ten songs or albums. What was going on in your life when you heard them? Why did they resonate with you? Do they resonate the same way now?
- 2) Watch a movie your friend likes but which you don't (or the reverse). Talk about your feelings while being respectful of the other person's views. Force yourself to break down why it doesn't work for one of you but does for the other.



An upcoming title from Necrosoft Games



Persona 2

- 3) Go into a thrift store and find three things you like. You don't have to buy them, just find some interesting things that appeal to you. What makes them interesting? Who do you think owned these things? Why did they give them away?
- 4) Here's a tougher one but valuable if you can swing it. If you travel to another city, another state, or another country, try staying in a friend's house, not in a hotel, Airbnb, or anything that isolates you from the way people live in that place. Next, avoid tourist hot spots and instead head to a specific restaurant, a thrift store, your friend's favorite bar, or somewhere else that's very local. Have a hard time getting there, see where it leads you. Don't take the easy road. You'll learn a lot about people, including yourself.

How do you apply this to games? It comes naturally, once you're in the business of self-analysis. Why should this character be blue? It could be red! Is there a reason for this person to be a man? What if it were a

woman? What if they had a dog? What are people you know who have dogs like? How would they react in this situation?

When you are practiced in self-analysis, these questions tend to come with answers built in, and will ultimately make your game a stronger work.

Make choices that matter in your games. Don't do something because it's popular, do something because it means something to you, or to someone on your team, or to your mother, or to your third grade teacher. When you make something for yourself, it's more likely to resonate with others, because it will be genuine and real.

People respect taste. They crave a connection to it and want to be complimented for having thoughts and feelings. Otherwise, there wouldn't be critics in the world. We wouldn't go on forums to voice our opinion about the latest Resident Evil trailer and how it compares to the historical canon.

If you make a game infused with your carefully considered thoughts, feelings, hopes, dreams, and taste, your game will be so much more interesting than the next Minecraft clone.

On with the show!

That's enough from Old Man Sheffield, but I do hope you enjoy this issue of the Game Career Guide and that you get something out of it. Whether you're trying to go to a game development school, get your first job in a big studio, or start your own indie game developer, know that we're all behind you.

We want to see more voices in the game industry, no matter who you are. We all crave new kinds of experiences, and you, the people reading this, are the ones who are going to make them. Thanks in advance!

Brandon Sheffield, editor
www.twitter.com/necrosofty



Mega Man Battle Network



Illustration by Del Northern

Breaking in the (easy-ish) way!

HOW ATTENDING EXPOS CAN LAND YOU A JOB

Brandon Sheffield

Let's say you're looking for a job in game development. Maybe you don't even want a full-time position at a big studio, you just want someone to throw some work your way so you can get your proverbial foot in the proverbial door. But everyone seems to want experience first, don't they? How the heck do you get that unless someone hires you?

There are lots of ways to do it, and all of them depend on diligence, intelligence, personality, and a bit of luck. One path that I've seen work multiple times is attending game conferences.

Simply getting to an event like the Game Developers Conference (or GDC—which, full disclosure, is run by this publication's parent company, UBM), E3, or even PAX can open up doors you never even knew about. You've just got to be ready and willing to be a nice and earnest person to capitalize on these opportunities when they arise.

To illustrate how this can work in the real world, I tracked down a few people who got jobs, or at least connections and the inspiration to become a game developer, because of their time at a conference. And they all attended with the cheapest (or most free) pass available to them, so you don't have to break the bank to do it either.

But getting in is the first step, isn't it? Some conferences have scholarships for students in need, as part of diversity initiatives, or in exchange for some work. GDC has their conference associate program, E3 has student scholarships, and PAX has their enforcer program. All of these can get you free admission in exchange for a bit of work. But once you get in there, it's all about what you do next.

LARA'S STORY

Lara Lunardi is an aspiring game dev, still in school. She got an internship at an Adult Swim-affiliated studio because of her time at E3. And she gets in for free. "Every year since 2013, I volunteer at E3," she says. "They run a program that allows students to get in if they are willing to work for four



Lara Lunardi

In the world of game development

EXPERIENCE MATTERS

Offering degree programs in the following subject areas:

- GAME DESIGN & DEVELOPMENT
- COMPUTER SCIENCE & ENGINEERING
- DIGITAL ART & ANIMATION
- MUSIC & SOUND DESIGN

SEE WHAT YOU CAN ACCOMPLISH

hours, which might be during the show or not. I worked in their international department, since I speak multiple languages. Every year they recruit me back for the same position. I work a bit and then enjoy the convention.”

So how did she turn this convention attendance into an internship? “As much as I love expos, the best way to meet people is definitely at after-parties,” she says. “At the expos, people are usually rushing around, trying to get to meetings or check out games. After-parties are more relaxed, and you can find people in their comfort zone so they are easier to approach.”

Her method works best for extroverts, who have no trouble jumping in and mingling. That’s not for everyone, but if that’s you, pay attention. “Usually I do go up and talk to people,” she says. “I also have the terrible habit of breaking into after-parties I wasn’t invited to. Not saying you should do that—in fact, you probably shouldn’t! But if you do, pretend you belong there and don’t go fangirling because you spotted the guy that writes for World of Warcraft or

something. You’d be surprised at how uncomfortable most of them feel about that!”

How do you turn these fun hangouts into something useful, then? “Some conversations result in mentoring, which is extremely valuable if you want to be successful in this industry,” Lunardi says. “Learning from the people that are doing [what you aspire to do] gives you new perspectives on your work and career. But other conversations may result in something more substantial, like a job. Recently I met one of the creators of Robot Chicken and we just instantly became friends.”

But nobody’s going to just give you a job because you’re personable. “This gig was definitely not handed to me,” she says. “I was lucky that I had [that creator] as a referral when I applied for their summer internship. Stoopid Buddy Stoodios was the first to give me a chance—which is surprisingly difficult to get when you’re in college. But it still takes the right portfolio and résumé and being in the right place at the right time to get a gig.”

Lunardi’s advice for anyone trying to break in at



Art by Lara Lunardi



Jarryd Huntley

JARRYD'S STORY

Jarryd Huntley is a game maker from Cleveland, Ohio. He works full-time as a coding bootcamp instructor at We Can Code IT, but he also teaches game development part-time at a local community college, has a newly founded studio, and is an organizer with the local game dev meetup group. They're currently working on a mobile game called Art Club Challenge. Phew!

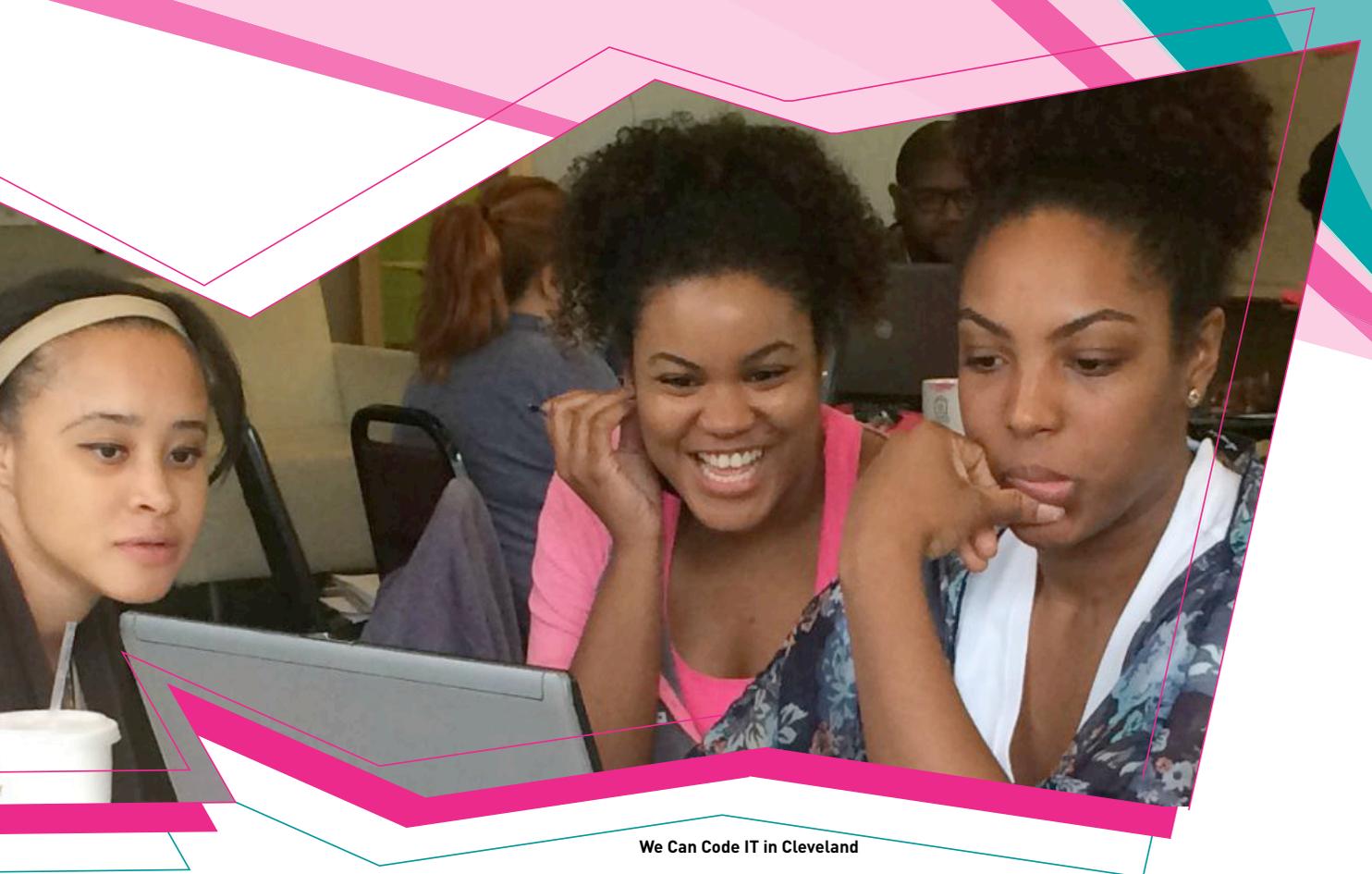
For Huntley, GDC was the turning point. He got to GDC through an expo pass program targeted toward women, people of color, and LGBTQ folks, which is an initiative I myself run every year. He had to scramble to get travel together in time, but once he got there, things started to come together.

"Some of my best and most valuable connections were entirely happy accidents," he says. "I ended up staying in a house with some friends and a few people I didn't know. This immediately gave everyone some sort of connection. If you get a chance to go, you should try to stay in the 'indie hostel' at GDC, find other indies in your hotel lobby at breakfast time, or even in the airport or on the plane."

This theme of new friends spans throughout

a conference? Build up a good portfolio. "The industry is very competitive, but never let that scare you or keep you from accomplishing your goals," she says. "Get your work out there, do game jams, make inspiring games, and make shitty ones too!"

But don't let your extroverted ways get the better of you. "Keep in mind that this is still a small industry and a lot of people know each other, so make sure you treat people nicely and leave egos aside," she says. "No one's work is perfect and everyone has room for improvement."



We Can Code IT in Cleveland

our narrative, but as with all human interactions, you should aim for quality over quantity. “If the conference spans multiple days, be sure to keep in touch with your new friends: They can help make connections and in turn introduce you to more friends,” says Huntley. “I followed my friends to a number of parties and networking events. As they ran into old friends, I was introduced to them.”

You don’t need to be an extrovert to draft along with some pals to an event. “Parties don’t necessarily need to be loud or crazy—oftentimes there are parties to play board games, playtest each other’s games, or are situated in a chill venue. These can be great if you’re a quiet person.”

“If you happen to be a shy or introverted person, try traveling with a friend or focusing on making one friend at a time,” he adds. “That can give you a person to shadow, which can be another way to approach networking. The important thing to remember is that industry friends, no matter where they are, were in your shoes at one point and can relate and help you when needed.”

For Huntley, attending GDC wasn’t about getting a game dev job; at first he just wanted to get involved. “At the time, I was working in IT managing servers and infrastructure for a large bank, but I wasn’t actively shopping for a job because I wasn’t sure I was qualified enough,” he says. “In talking with people at GDC, describing my job responsibilities led a few of them to ask if I was open to exploring job openings that they felt I was well-qualified for. I

feel like sometimes focusing on making a friend or connection before focusing on getting a job prospect can lead to opportunities you may have missed otherwise.”

Now, GDC didn’t lead directly to a job, but these conversations did set him on the path toward full-time game development. “I didn’t end up getting the two jobs I applied for at the time,” he says, “but those connections have been more than worth it. Also, not getting a job isn’t a statement about you or your skills at all; it just simply means the match wasn’t right at that time.”

As some parting advice, Huntley says you should focus more on hanging out with cool people than just hustling for work. “If the choice is between getting a résumé to a recruiter or making great friends who can call you when a position matching your skill set comes up, I’d choose the latter,” he says.

After GDC, he’s gotten a lot done, and seems to have caught the networking bug. “I travel a lot around the US and Canada to meet with other indies, learn about their local communities, and make connections where I can,” he says. “I’m currently head organizer and act as mentor in the local Cleveland Game Developers meetup group. I’ve consulted on design for a number of games and work closely with the Greater Cleveland Film Commission, a local nonprofit focused on job creation through entertainment, to help foster the game development community here in Cleveland.”

What more could you ask for?



**Jesse Harlin
in his natural
environment.**

JESSE'S STORY

Jesse Harlin has been in the industry for quite a while, and has worked on some big projects. He's currently a freelance composer at Dunderpate Music and spent many years at LucasArts as their in-house sound designer. He's been the composer on titles such as *Star Wars: The Old Republic*, *Mafia III*, *Marvel's Avengers Academy*, and *CounterSpy*.

But even for him, it started with a GDC expo pass, which was the cheapest one he had access to (a student pass is cheaper). He headed straight for the job fair, a section of the conference with recruiters trying to woo bright-eyed and bushy-tailed prospective game devs.

"I hit every booth I could," he says. "I didn't know a damn thing back then. I remember walking up to the guys at the Blizzard booth and asking them 'So... are you guys, like, a developer? Or a publisher? Or what?' They didn't even try to hide it when they rolled their eyes at me. Sixteen years later and I've still never come close to working for those guys. Anyway, I went from booth to booth asking companies if they outsourced the music for their games."

"Out of pure chance, LucasArts happened to be there and to be hiring for a staff composer," he adds. "We chatted, they liked me, and they asked me to do a phone interview. Long story short, I didn't get the gig. But they said they really liked me and that



they wanted me to keep in touch and let them know how I'm doing. So, I went to grad school instead. The next year, GDC rolled around and I went up again on another super-cheap expo pass."

This is where his persistence pays off. As you've no doubt noticed, staying in touch with the people you meet is a strong theme when it comes to getting jobs out of conferences. Harlin's story is no different. "I made it a point to go find the LucasArts booth, stop by, and just say 'Hey, guys. I'm still around. Here's what I've been up to with grad school and composing. Would still love to talk if you're looking to hire again.' And that was it. Eight months later, they called and said, 'We have another position and would love to chat.' I started at LucasArts about 12 months to the day after my second GDC Expo pass adventure."

Harlin's advice will be unsurprising to you but is

no less important for its familiarity.

Especially keep in mind the ways in which people like to maintain contact, and whether they want to at all. This requires some social skills and is a learned trait, but a very important one.

"I'd recommend to anyone attending something like GDC that they keep in mind that networking is a long game," he says. "It's not about who you meet there so much, because you'll meet a ton of people if you put yourself out there. It's about how you maintain the connection with those you meet. Don't pester. Don't make yourself an annoyance. Learn to read when someone is interested in talking to you or not. But if they're interested, don't be shy about staying in touch with them. If you can genuinely laugh with someone every time you run into them, you're going to find that you each want to find a way to work together."



Star Wars: The Old Republic, for which Jesse Harlin was composer.



Del Northern

CRYPT OF THE NECRODANCER



Del's cover for a Crypt of the Necrodancer OST.

DANNY BARANOWSKY

DEL'S STORY

Del Northern is a concept illustrator and environmental artist living in New York. So far she's done illustration work for a few titles and their soundtracks, such as *Crypt of the Necrodancer*, *Robo Puzzle Smash*, and *Rain World*.

Her expo story is similar to Jarryd Huntley's, in that she attended using a pass grant. And for her, as a shy person, it was all about being open, friendly, and distinctive. "I suppose my approach was just walking down the street with a Magfest hoodie on," she says, "since that sparked a completely spontaneous conversation with a stranger. That stranger happened to be one of my favorite composers ever, Jimmy Hinson [Threes, *Borderlands 2*!] And I have to be grateful that Jimmy is such a sweetheart to just off-the-cuff introduce me to his friends right then and there, who were also some of my favorite composers ever."

Her meeting was a bit more dramatic than some, as she admits. "Once we exchanged cards, out of GDC habit I guess, I recognized his name. Then there was a lot of crying and gushing, what with being so starstruck. But I suppose the 'artist' title on my card piqued his interest, and we started talking about that!"

For Northern, as with many others, this wound up turning into a hangout situation. "It was just that one night of hanging out in a hotel room with a bunch of game composers," she says. "There was a lot of friendly chatting and having fun as friends would! I was starstruck, showed them how much of all their music I had on my phone, and gosh, they were super humble and loved the fact that I listened to their work on a daily basis, which blew my mind. Either way, my being a visual artist in a group of composers made for a lot of art questions coming at me, and wanting to see my work, so it got shown, and I tried my best not to be shy."

"Long story short, a few days later, I was contacted to do some surprise birthday art for one of them, Danny Baronowsky, and that went so well that the next commission I got was for an album cover for *Crypt of the Necrodancer*. Once that got out, I had more people offering me work than I could handle."

As a shy person, ignoring the kind of fangirling advice given by Lunardi, this was a pretty big deal. "Essentially it broke me in, way harder and faster than I could have ever imagined," she says. "I'd made some incredible friends, and friends know friends who could use an artist!"

Her advice for those attending an expo for the

first time mirrors Huntley's as well. "Don't network, make friends," she says. "I mean, networking is fine and all, but especially in the indie game scene, a friend will get you farther than a person who has your name on a card, in a stack of a million other cards. Of course, if you're super good at networking then I guess have at it, but it's important to use your time at big conventions to meet people, and not just to toss a card at them. You should just try to have lots of fun, to the point where you can't wait to see them all again at the next GDC or PAX or what have you. It'll get you a lot farther and be worth a lot more to all parties involved."

MY STORY

I've had these sorts of experiences as well, time and time again. One of the first groups of people I met at a conference went on to become Capy Games, maker of *Sword & Sworcery* and *Below*. I managed to convince a Thai company with cool ideas but no English-language writer that I should do story work for them. A Japanese game developer I met at a conference wound up giving me a shot at narrative direction. Now, as the owner of a small game studio, I've gotten publishing deals by keeping in touch with people I met at conferences ten years ago.

There's no formula to this stuff. But if you have a chance to attend a conference, and you know someone with any kind of network, go hang out with them there. Meet some other people, and make some friends. If you like each other, maybe you'll work on something. Or if you're looking for that one big job, make sure you keep in touch with your recruiters, ask them about new listings, but try to keep your connections personal. You don't want to force the issue.

Above all, be friendly, be kind, and don't spend too much time on anyone who gives you bad vibes, or feels creepy, or like they've got ulterior motives. Likewise, don't be that way to anyone else. Get into conferences, follow these simple instructions, and who knows what could happen?

Brandon Sheffield is the director of [Necrosoft Games](#) and the editor of this publication right here. He's currently working on [Gunsport](#), which you could describe as 2v2 cyberpunk volleyball with guns. He advises for conferences and festivals such as GDC, Reboot Develop, and the Sense of Wonder Night and likes making connections for and with other game developers. Find him on Twitter at [@necrosofty](#), and give him a high five.



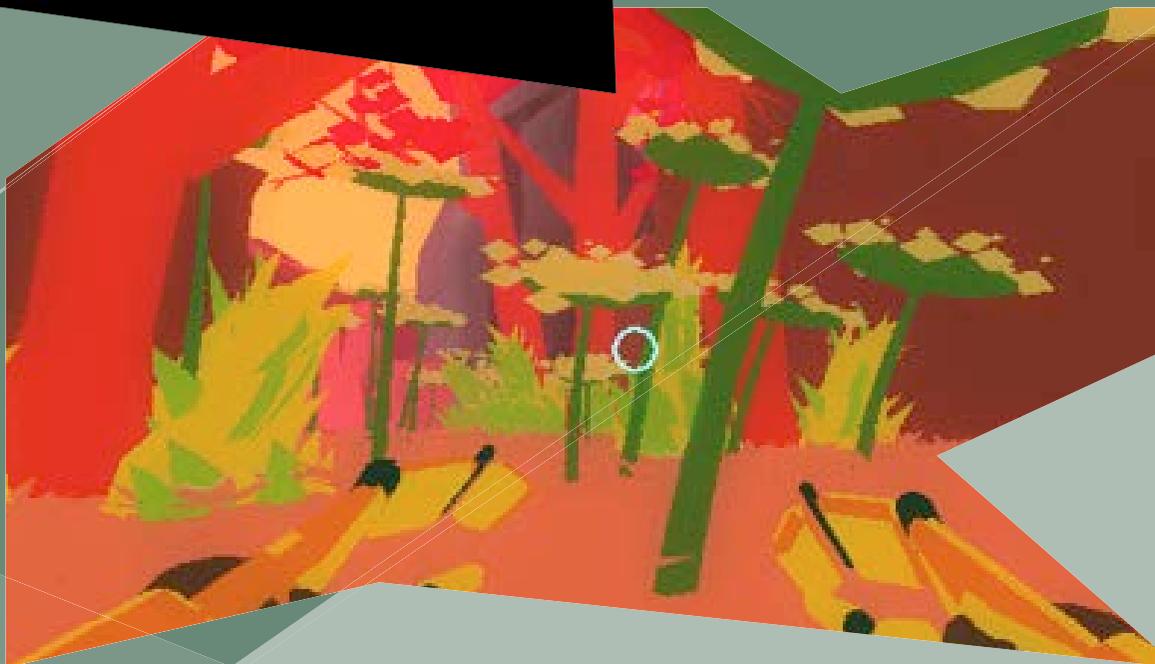
Zero to Hero

Ethan Redd

LOW POLY MODELING FOR GAME DEV

HEY! YOU WANT TO LEARN LOW POLY MODELING BUT DON'T KNOW WHERE TO START? LOOK NO FURTHER! WITH THIS GUIDE, I HOPE TO PROVIDE A GOOD INTRODUCTION TO NOT ONLY THE SOFTWARE, BUT ALSO THE CONCEPTS AND THEORY AT PLAY.

I draw heavy inspiration from the 90s sort of Saturn, PSX, and arcade era, and I try to inject my work with that same energy while staying fresh and original. I got into 3D at a young age and I'm pretty





much entirely self-taught, stealing techniques from my idols here and inventing tricks of my own there. When I started out, I had to learn on incredibly low-grade hardware and make the best of it. Over time, I learned to embrace this constraint and turned it into a style all my own. In this article, I want to share with you some of what I've picked up all these years and help you start along your own journey to radditude.

Disclaimer: While I intend to keep the information in here broad enough to serve as a general intro, this is by no means a replacement for a comprehensive survey of 3D, 3D math, or even modeling. My goal is just to have you up and running as quickly as possible; after I'm done with you, you'll have a solid enough base to go find the remaining details on your own.

A STEP BACK: THREE DEE

To simplify the hell out of it, 3D modeling is basically the process of plotting points in 3D. These points (or vertices), are then connected by edges into faces, ultimately describing the surface of a three dimensional figure. As you can tell by this summary, there's a lot of terminology and fancy math lingo involved. Since this is a practical and introductory guide, I'm not super concerned with giving pinpoint accurate definitions for the more math-y stuff, but

here's a quick rundown of the most necessary ones before we dive deeper:

Vertex [plural: vertices, slang: vert(s)]. A vertex is simply a point in space—3D space, in our case. Think of it as an infinitely small dot noting a certain spot in our little world. Vertices are what we're primarily concerned with in 3D modeling, and they are the basic unit we make stuff out of. Essentially, all we do is connect these points (forming edges, and then forming polygons from those edges) to describe our desired form.

Polygon. A polygon is simply a multi-sided figure, just like back in math class! Under the hood, all our models are just collections of triangles, but for ease of editing we usually will work with quadrangles.

Mesh. A collection of vertices, edges, and faces describing a 3D surface

Axis. Think of it as an imaginary line describing a direction we can move in or rotate around. When talking about dimensionality, we're referring to how many independent axes we're working with. E.g.: in 2D space, we're limited to moving horizontally and vertically; with 3D space, we're given the third axis of depth. We refer to these axes as X, Y, and Z, respectively.

(Of course, it's a little more complicated than that.

Since Blender, the software we'll be working with, uses a right-handed coordinate system, we'll think of Y as depth and Z as height. This is inconsistent, like a lot of common conventions for modeling programs/game engines—but hey, I don't make the rules.)

Normals. "Normal" basically means perpendicular. When talking about modeling, we're most interested in surface normals, which are vectors pointing perpendicular to the faces of our mesh. These are used for a ton of cool stuff, but shaders and lighting especially depend on these. Don't worry too much about them yet, but keep the idea in the back of your head.

MOVING ON: "LOW POLY?"

Now that we think we know what we're talking about with 3D modeling (even if you don't yet, it's chill), we can get a little deeper into the specifics of what makes low poly modeling special and interesting.

"Low poly" modeling is the practice of intentionally constructing geometrically efficient meshes. Low poly started out as just the way things were done; back in the olden days of realtime CG, computers weren't very good at or fast when rendering 3D models. Artists of the time had to be careful when plotting out their creations lest they tank the performance of their projects with excessive render time. Nowadays, when we can chuck hundreds of thousands, or even millions of polygons on-screen at a time on even basic consumer hardware, you might find yourself asking, "Why should I even care?"

But there are a ton of reasons to love it. Here are just a few that I find most compelling:

It's fast. Similar to pixel art, low poly practices can leverage detail economy to shorten iteration times, lessen workload for artists on small teams, and generally make asset creation less of a pain in the bahookie. For me, this is a huge plus to the low poly discipline. Though this varies from person to person and everybody's workflow is different, generally speaking, low poly makes 3D work less time intensive.

Constraints lead to creativity. Your mileage will definitely vary on this, but it's my personal view that constraints generally have a stimulative effect on the imagination. When you're trapped in a box it's hard not to think about what's outside of it right? For those who like a challenge—or just value the visual consistency of set limits—having a poly cap can really get the old creative juices flowing.

Performance. Years ago, before it was a thing, low poly was just the way 3D was handled to accommodate

the available technology. Now that we have the tech to push verts like no one's business, we can still leverage these techniques to milk more performance out of new hardware. Whether you want to render hordes of creatures at high framerates, or sweeping shots into the distance showcasing a detailed landscape, or even just run your game on shitty hardware (this is how I got into it), low poly techniques and smart planning can help you get the most out of whatever hardware your game is running on.

It's just rad as hell. Because I say so.

GETTING STARTED: BLENDER BASICS

For this article, we'll be using my personal weapon of choice for all my 3D and vector needs: Blender3D (www.blender.org). Blender's a wonderful software with an even more amazing community of users and developers—and best of all, it's free, open source, and extendable via Python scripting. Although the interface can be byzantine, and the learning curve is steep (even for 3D pros switching from other software packages), once you get the get the basic interactions down you can

start making stuff fairly quickly and export to most common 3D formats.

INTERFACE OVERVIEW

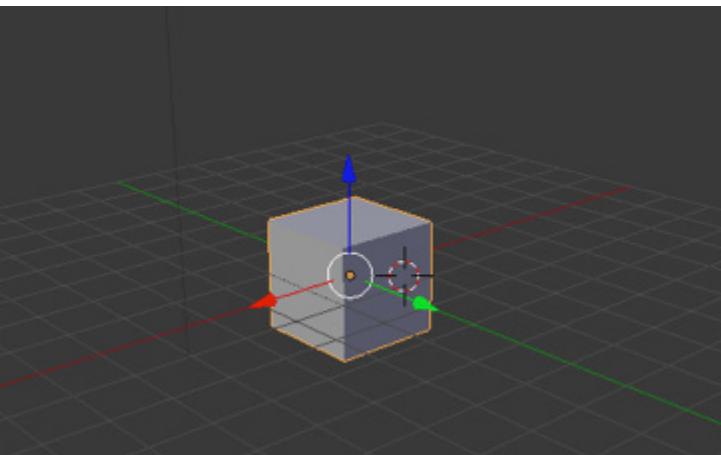
Not even gonna lie; at first glance, Blender is not a welcoming environment. If Blender were a person, it'd be that friend that you thought was a total jackass for the longest but is actually a pretty chill person once you got to know them. Don't let the screw-face intimidate you!

BASIC OPERATIONS

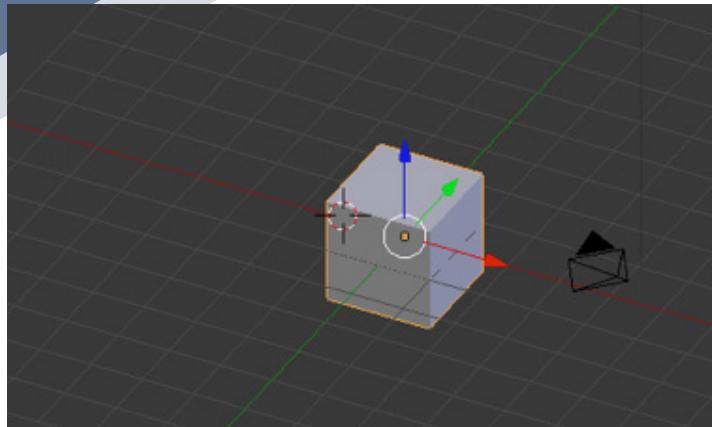
For our intents and purposes, these are the most common interactions you'll have during your stay in Blenderland:

Selection. To pick ("select") what object you'd like to manipulate, simply hover your mouse over it and **RClick**. If you'd like to select multiple objects, hold **LShift** and select each object individually. Lastly, press **A** to select all available objects, or to deselect everything currently selected.

Orbiting/Panning/Setting the Viewing Angle. Use the middle mouse button (**MMB**) to "orbit" around your currently set pivot point (which defaults to the origin, but you can set it to a currently selected object with the period button on your number pad). Hold **LShift** while orbiting to pan instead.



You can also use the **Num1**, **Num3**, **Num7**, and **Num0** keys to set the viewport to the front, side, top-down, and active camera viewpoints, respectively. Additionally, you can orbit incrementally with **Num4**, **Num8**, **Num6**, and **Num2**. Lastly, you can use **Num5**



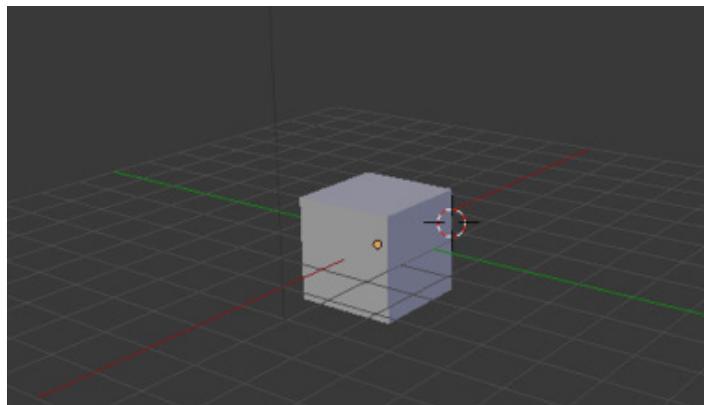
to toggle between orthographic and perspective viewing. It's weird to explain; try it out and it'll make immediate sense.

One of the hardest parts about getting into 3D for a lot of people is adjusting to moving around the digital space. When you're just getting started, take your time and move around slowly, especially if you're prone to motion sickness in real life.

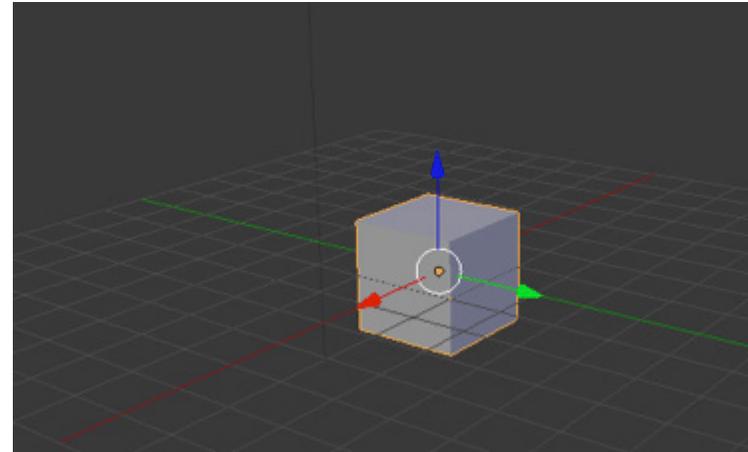
Undo. Mistakes happen; use **Cmd/CTRL + Z** to undo them one at a time. Blender has "unlimited" undos so roll back to your heart's content. If you need to undo an undo, use **Cmd/CTRL + LShift + Z** to reverse it.

For the following actions, press their respective shortcut key and move the mouse to control its application: Hit the left mouse button (**LMB** for short) to lock in your changes, right mouse button (**RMB**) to cancel. Additionally, all these operations can be constrained along an axis by hitting **X**, **Y**, or **Z** mid operation. If you really want to get fancy, you can also enter a decimal number for even more precision.

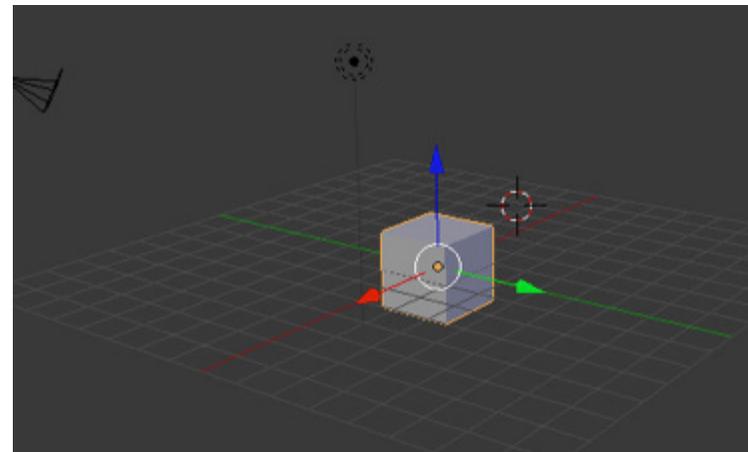
Translation. Translation is just fancy talk for sliding stuff around. Hit the **G** key and move the mouse around to reposition whatever you have selected. You can also hit **X**, **Y**, or **Z** to constrain movement to one of the current axes; if you need precision, you can type in a decimal number for the number of units to move as well.



Rotation. Exactly what it says on the tin—this operation rotates whatever is selected around the current pivot point. Hit **R** and move the mouse to start rotating your selection.

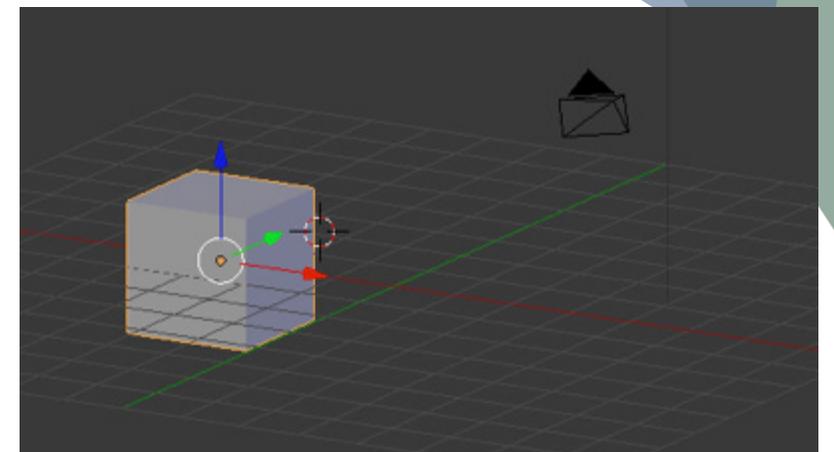


Scaling. Scaling (**S**) resizes whatever is currently selected from the current pivot outward.

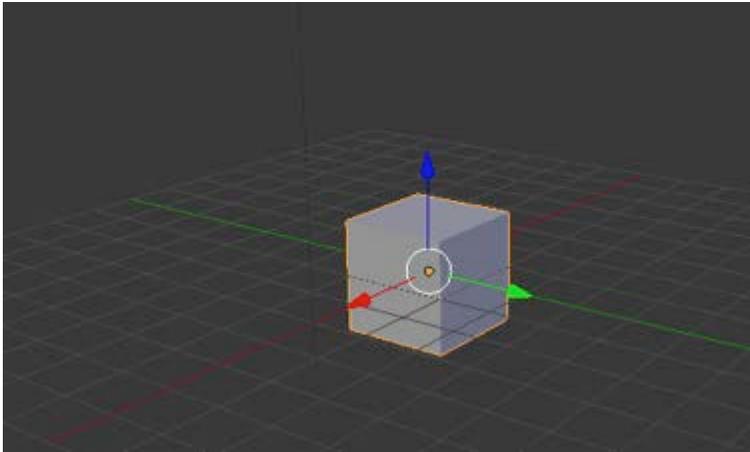


Duplication. To make a duplicate of your current selection, hit **LShift + D** then hit **LMB** after moving it. In Object Mode, this will create an unlinked copy of the object—meaning that editing the original won't have any effect on the copy.

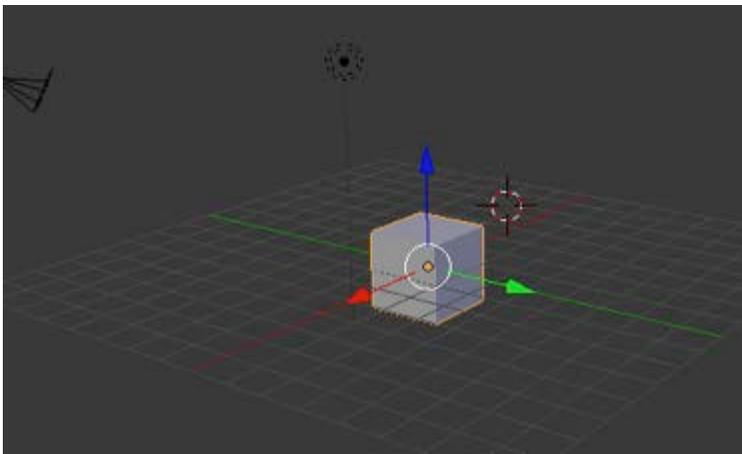
If you'd like to keep the duplicate linked to the original, use **LAlt + D** instead. This is useful for things like plants, bystanders, and anything else you'll need a ton of similar copies of.



Rotation. Exactly what it says on the tin—this operation rotates whatever is selected around the current pivot point. Hit **R** and move the mouse to start rotating your selection.

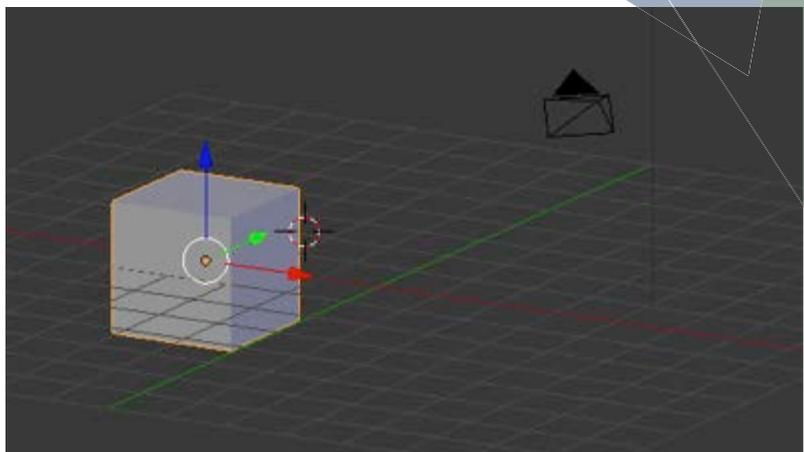


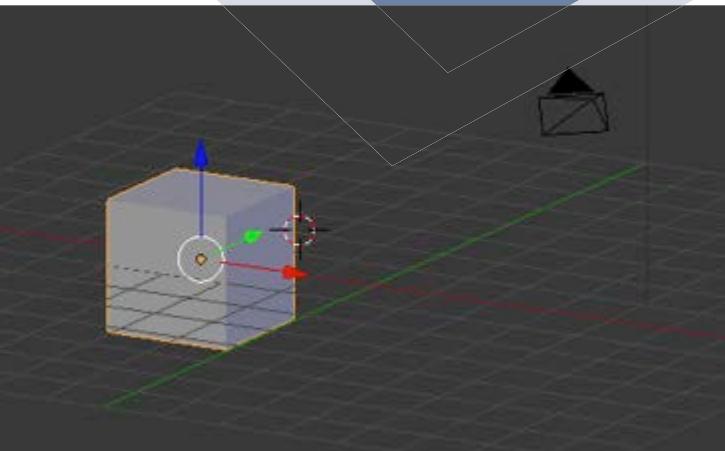
Scaling. Scaling (**S**) resizes whatever is currently selected from the current pivot outward.



Duplication. To make a duplicate of your current selection, hit **LShift + D** then hit **LMB** after moving it. In Object Mode, this will create an unlinked copy of the object—meaning that editing the original won't have any effect on the copy.

If you'd like to keep the duplicate linked to the original, use **LAlt + D** instead. This is useful for things like plants, bystanders, and anything else you'll need a ton of similar copies of.

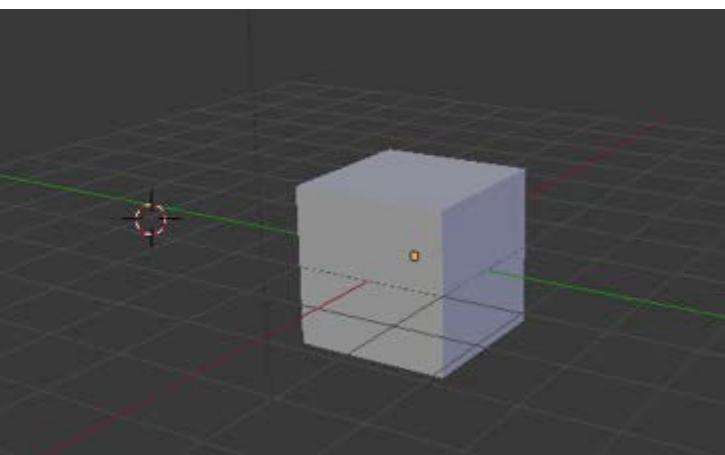




Clearing. While in Object Mode (there are others—more on that later), you may sometimes want to reset the position/rotation/scale of an object; use **Cmd/CTRL + G**, **Cmd/CTRL + R**, and **Cmd/CTRL + S** to reset them respectively.

EDIT MODE

Select the mesh object you'd like to edit, and hit Tab. You'll see something like this:



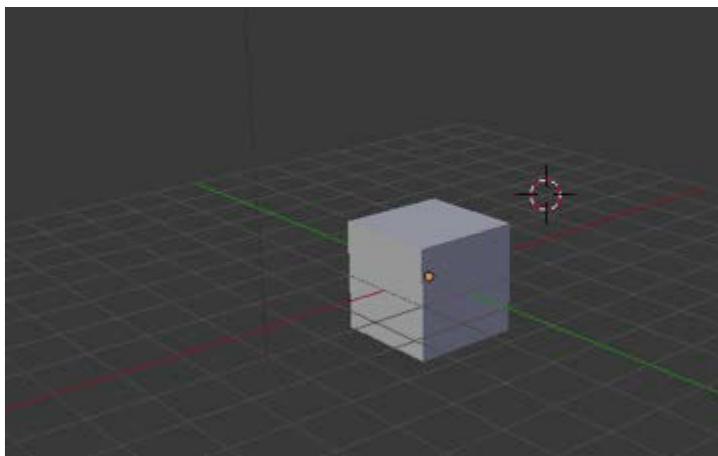
The little dots you see on the corners of the mesh are the vertices that we talked about earlier. Just like objects, you can select, translate, scale, rotate, and duplicate these verts to create the radical shapes you have in mind. Even cooler, we use **LCmd/CTRL+Tab** to switch between editing the vertices, faces, and edges of our mesh for even more control.

When in edit mode, we edit the vertices of our meshes by default: **RClick** on any vert and use the previously mentioned operations to begin deforming it. Additionally, you can link two or more vertices into edges/faces by selecting them and hitting F. Finally, if you'd like to select the faces or edges of the mesh instead of verts, simply hit **Cmd/CTRL+Tab** and select

“Edges” or “Faces” in the pop-up.

Now we're ready to get into some special Edit Mode operations! First up:

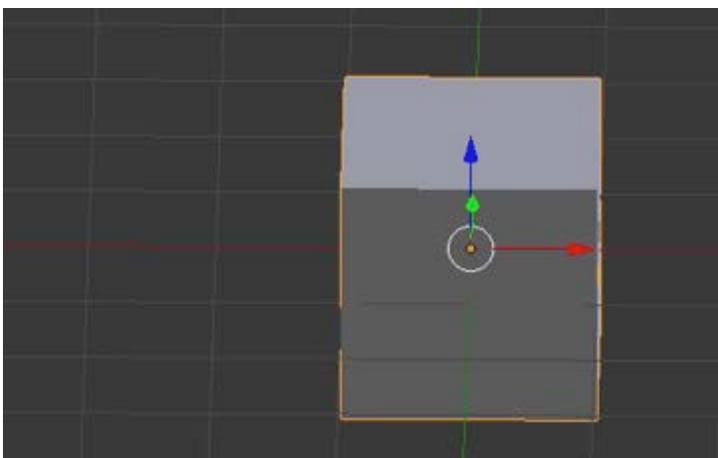
EXTRUSION



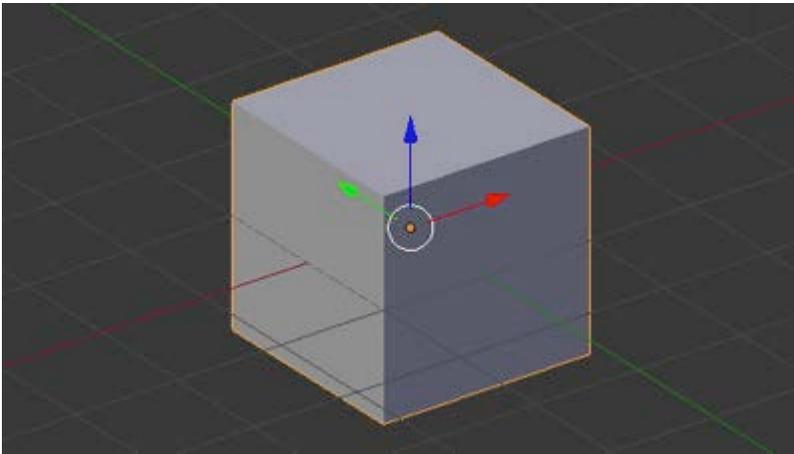
Extrusion (**E**) takes your current selection and pulls new faces/edges/verts along with your mouse. Extruding verts creates an edge from the old vert to a new vert; performing the operation on edges creates a new edge and a face connecting it to the old one; and doing it to faces creates a new face connected to the old face by faces along each outer edge of the extruded face. This is one of the most common ways to block out a shape.

DIVVY IT UP: LOOPS, KNIVES, SUBDIVISION

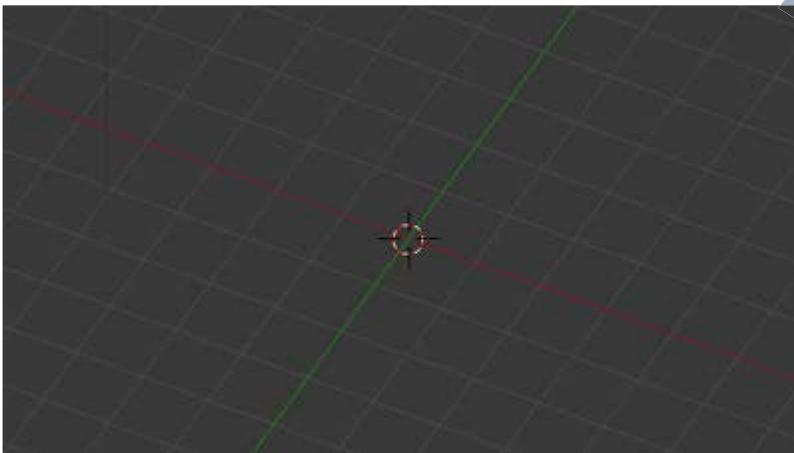
Subdivision is exactly what it sounds like: We can take our selection and divide it across all its edges to get a little more granularity in our mesh. To subdivide your selected edges/faces, hit the **W** key in Edit Mode to bring up the specials menu. There's a lot of cool stuff in the specials menu that you should check out on your own, but right now just click the first option, **Subdivide**.



The **Knife Tool** is another way to chisel some detail into your mesh. Click where you would like to start cutting, and then click where you'd like to stop cutting; the line(s) formed by the 2 (or more) will be cut into the faces they overlap.



WHEW! That's a lot to take in all at once, but believe it or not, if you practice just these basics you already know enough to start creating simple 3D art! But we're not content with just the basics, are we? No, we're diving even deeper into this rabbit hole and getting into some *pro strats*. If this is your first exposure to 3D, and especially Blender, I'd recommend taking some time to practice what you've learned so far, take a breather, and come back once you've got these basics more or less down.



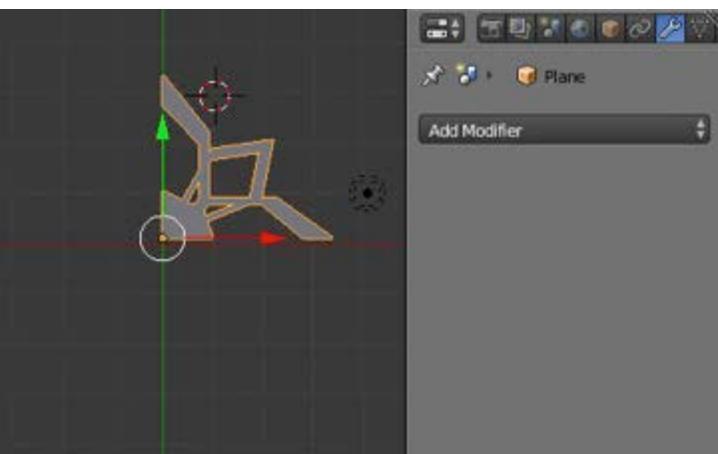
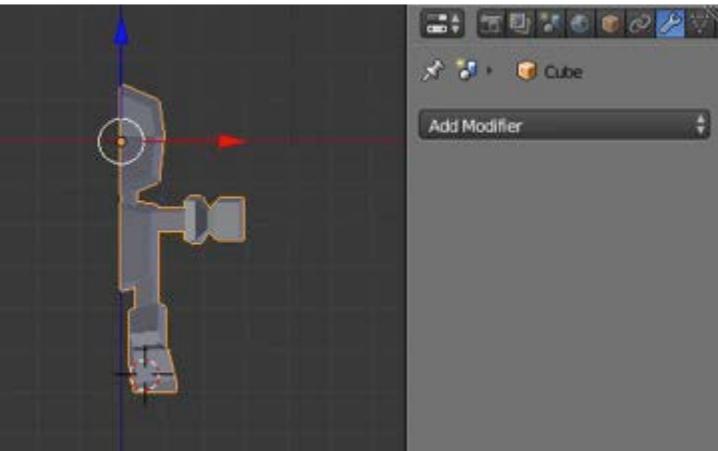
GETTING TRICKY: MODIFIERS, UVS, MATERIALS

If you've followed along this far, you're well on your way! Let's up the ante a bit and get into some Advanced Manuevers™.

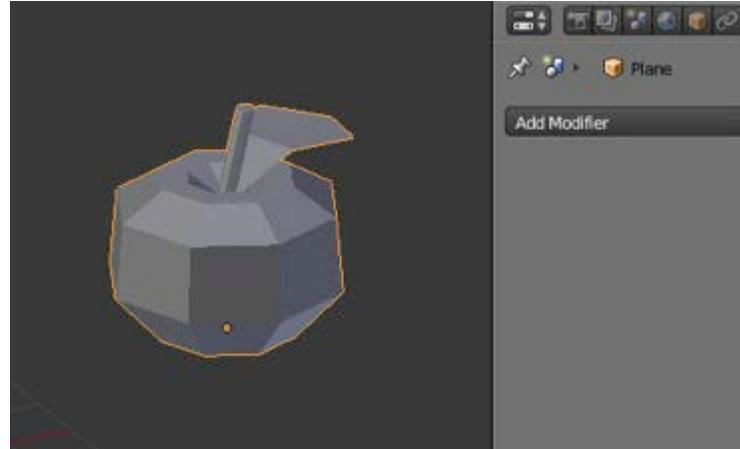
MODIFIERS

Blender has a bunch of handy features wrapped up in modules called modifiers. Think of them like this:

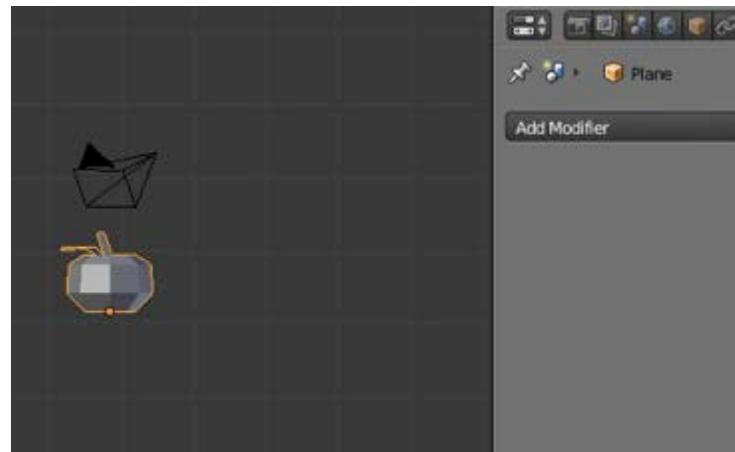
Mirror. More often than not, the base shape of what you'll be modeling will have symmetry along some axis. Instead of modeling everything twice, the mirror modifier lets you model one side, and automagically duplicate and flip your geometry across the axis of your choosing. This will save you a ton of time and effort on things like animals, furniture, and vehicles.



Subdivision Surface (Subsurf). Subsurf is a lot like the subdivision operation we talked about earlier. Basically, it takes each face of your mesh and divides it along the edges, creating 4 new faces that have their vert positions averaged to smooth everything out. Generally, we won't be using this for low poly (unless you want an easy "smooth" version of your model), but it's nice to keep in your back pocket.

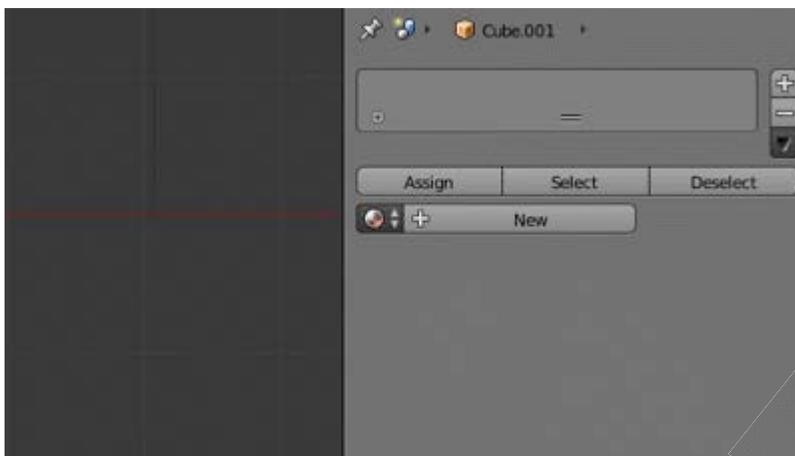
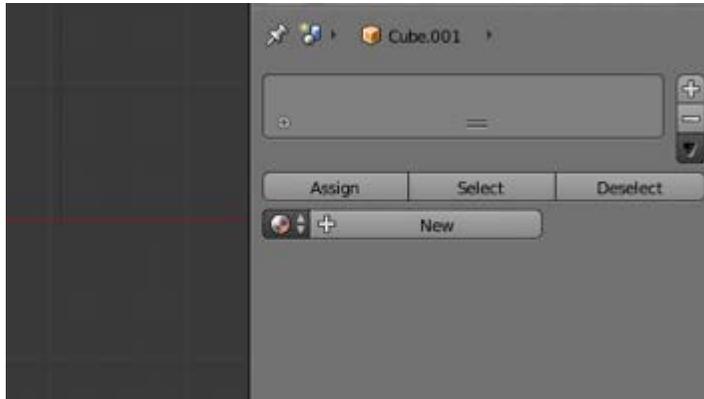


Array. The array modifier is also a nice addition to your growing toolkit. Similar to linked copies, the array modifier generates duplicate copies of your mesh in specified quantities/intervals. If you need rows and columns of something, this might be a nice easy option to get it done.



MATERIALS

Up until now, our models have been boring and gray. Let's fix that with materials! Materials define how your model will react to light, shading, and generally control how it will get rendered. Rendering works somewhat differently in Blender than most modern game engines (which use a physical-based system), so we'll just gloss over this.

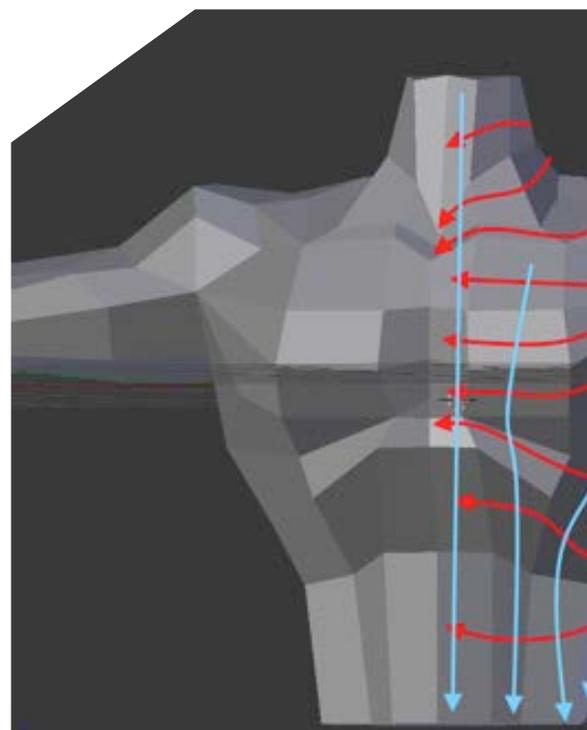
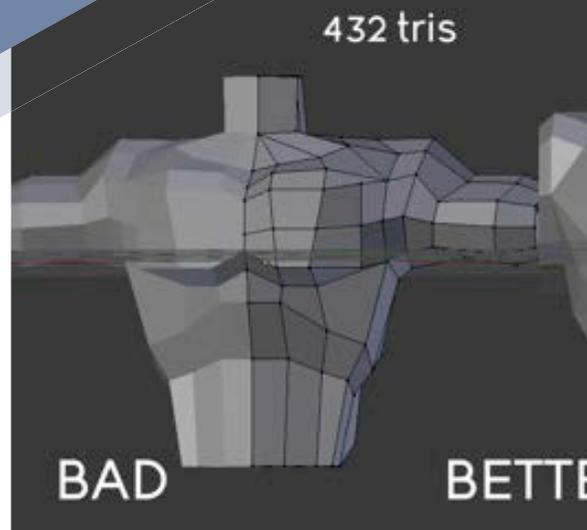
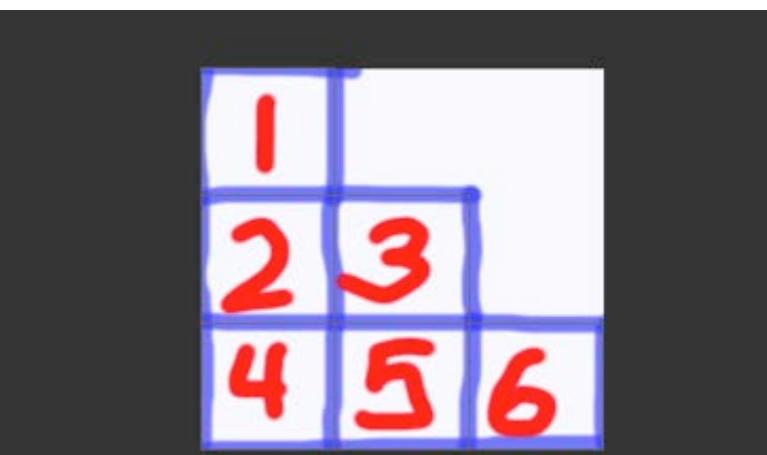
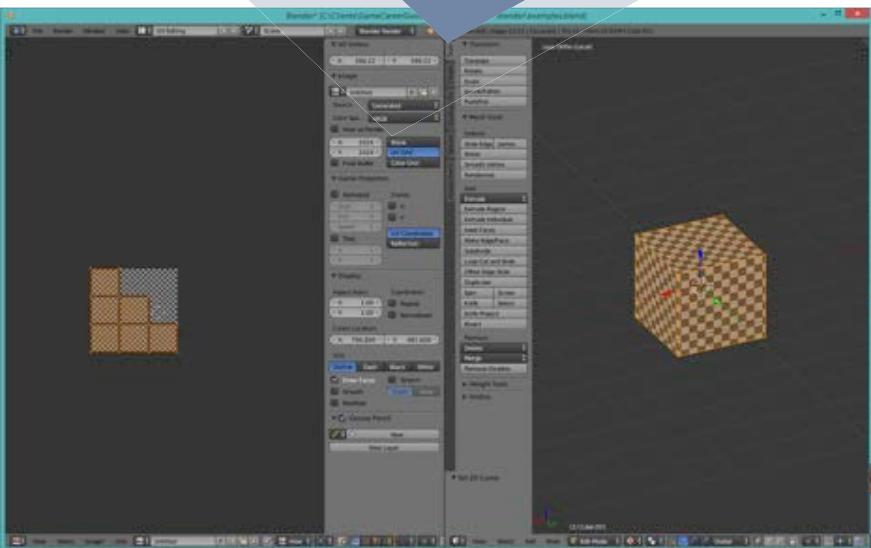


UV UNWRAPPING

Some call it a can of worms; others, Satan himself; others still, “a great way to add detail to your model!” Jokes aside, UV wrapping is essentially the process of projecting your 3D model into 2D space so you can paint on it in a process called texturing your model. If that sounds weird, think of the last map you saw of the Earth; it’s a 2D picture on a piece of paper, describing the look of our favorite 3D spheroid. Another great metaphor is papercraft; when you take the flat images and fold them properly, you end up with a 3D statuette.

A full write-up on UV techniques is beyond the scope of this primer, but it should be enlightening to at least touch on the subject.

First, let’s change our window layout to the UV editing preset (yup, Blender gets even more complicated). Then, in Edit Mode, select the faces you’d like to unwrap and hit the U key. This will bring up the unwrapping menu and show you a bunch of options for projecting your model. To keep it simple, I’ll show what a cube looks like when unwrapped using “smart projection.” Notice how each face of the cube has a 2D counterpart on the left viewport



BEYOND THE BASICS: FOOD FOR THOUGHT

Assuming you've followed along up until this point, now you sorta know what you're doing! Even so, there's still more to learn and think about.

TOPOLOGY: WHAT AND WHY?

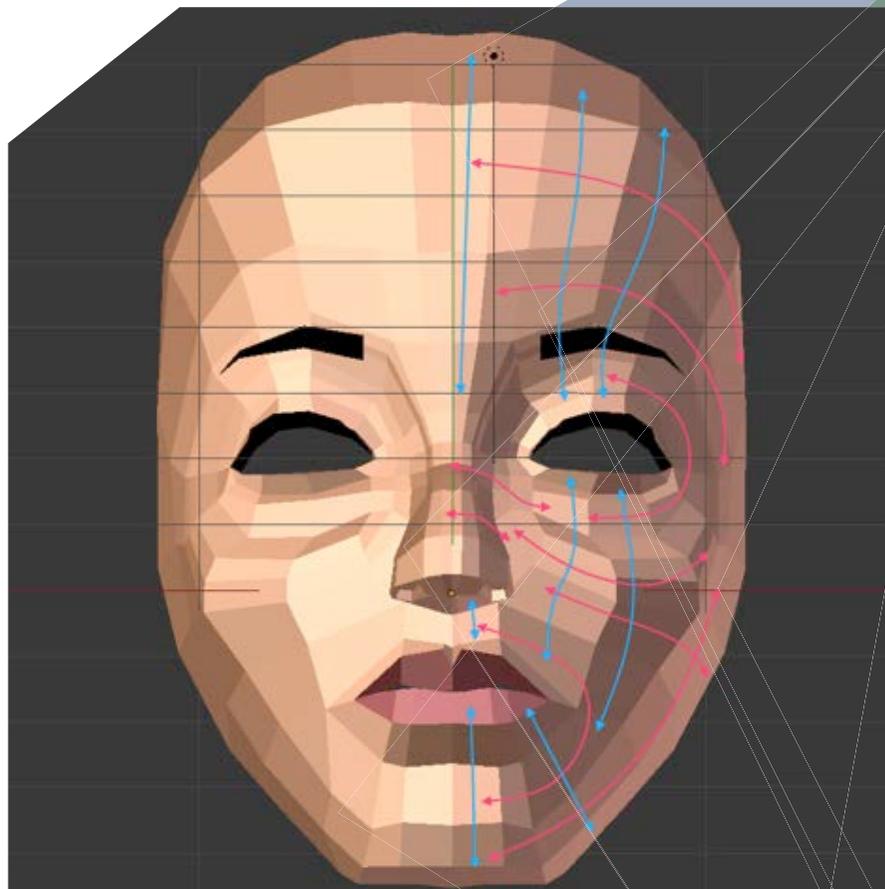
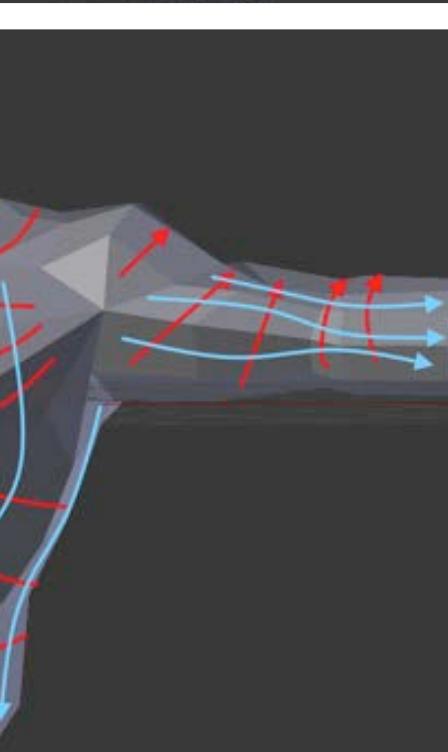
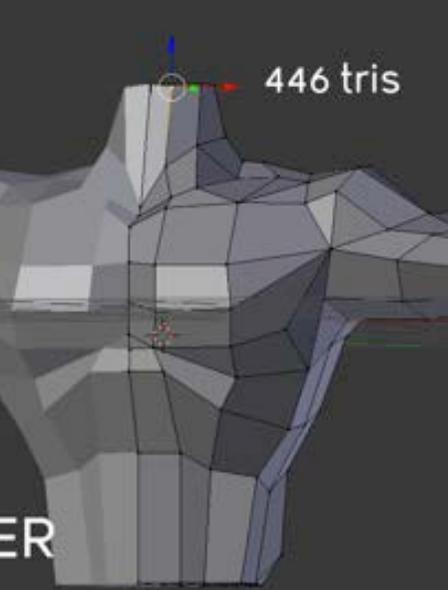
As you begin your journey into the wonderful world of 3D modeling, a great concept to get in your noggin early on is the concept of "topology." Put plainly, topology describes the flow of the surface of your object. While it's easy to push and prod verts in any which way and close faces all willy-nilly, there are many benefits to being conscious, such as:

- Ease of revision
- Better shading—Especially when using low poly models or "toon" shaders, if your loop flow sucks you're gonna have a bad time. Save yourself the hassle.

Although I usually hate telling people what to do, here's a list of dos and don'ts to consider when plotting your geometries:

DO (IF IT WORKS FOR YOU)

- **DO:** Try to keep your (smooth surface) meshes organized into neat quad strips where possible.
- **DO:** Consider the flow of the surface of your object. Try to preserve and reflect this in your mesh.
- **DO:** Use triangles (sparingly!) to terminate loops where appropriate, or to add jaggedness to your mesh.
- **DO:** Take some time to plan out your loops if necessary. Sketch out your object in as many/few angles as you want, and trace the "lines" of the ebb and flow in the surface.
- **DO:** Throw all of this out the window when appropriate! Some surfaces are craggy and



don't need clean lines, some styles like/need jankiness, and so forth. Above all else, do what works and discard the rest.

DON'T (OR WHATEVER, IT'S YOUR LIFE)

- **DON'T:** Use excessive triangles in your (smooth surface/organic) meshes. Quads are (generally) easier to manage, especially when well organized into loops/strips. Leverage this!
- **DON'T:** Cut excessive loops into your mesh to add spot-details to a specific area. In my personal experience, if you're cutting a ton of loops around a large area to add some pizzazz to one small one, it's time for a retopo (refactoring your mesh).
- **DON'T:** Be afraid of starting over. Happens to the best of us; keep it moving!
- **DON'T:** Blindly follow the "dos." Consider your application and decide for yourself.

ECONOMY: TECHNIQUES FOR REDUCTION

TEXTURE VS. GEOMETRY: WHAT AND WHERE

There are several instances in which you should add detail to your mesh. The first is when that detailing will have an impact on the overall silhouette of the piece. A great illustration of this point is the excellent player model from Mario 64.

Notice how many precious polygons were spent detailing the shape of Mario's hair, yet his hands remain amorphous spheroids. When viewed from the side, Mario's hair protrudes backward, as is proper; if it were flat against his head, it'd just look weird. On the other hand (heh), his fists don't really need much definition, since no matter the angle, from 15 or so feet away a fist is just a sorta-ball-looking-thingy.



Mario has blobby hands in Mario 64, but pointy hair.

Another quality example is every character in Sonic R. Hate on it all you want, Sonic R is a perennial favorite of mine and an aesthetic masterpiece, and if you can't dig it we can't be friends. Notice that the greebling on Robotnik's hovercraft is all drawn on, and the hoods over Amy's tires are also simply pixelated

textures. In contrast, Sonic and Tails' ears are both done as a triangular pyramid so they won't look strange from off angles. These sorts of conscious decisions allowed poly-pinching devs to streamline their art while focusing visuals in one fell swoop. Take heed!



Next, if the detailing is essentially 2D in nature, consider texturing it onto a billboard sprite. You can't beat using literally 1 polygon! You see examples of this even in higher fidelity

modeling today, with trees and foliage. Often, you'll see groups of leaves simplified to a plane with an alpha-cutout leaf texture. Depending on the amount of layering you need, this is much cheaper to render than thousands upon thousands of tiny polygons.

Sonic R does a great job streamlining 3D art



The Bob-omb boss is mostly a series of 2D planes.

Depending on the aesthetic of your project, you may be able to use this even more aggressively and replace entire meshes with billboards. Yet again, we return to Mario 64. (They optimized the hell out of the visuals of this game in general. It's a great case study if you have some time to just sit and take notes.

The boss at the end of Bob-omb Battlefield is simply 5 billboards (one for his body, one for both hands, and more for both ankles), two feet, and a cylindrical crown with an alpha cut-out texture. This is genius, ruthless optimization; instead of wasting

valuable geometry on a geometric surface defined by literally looking the same from every angle, they were able to not only save some tris on the spheres but also improve the overall look of the character with some prerendered shading.

Again, your mileage may vary. In my games, I often break every single one of the "rules" in this section consciously because my style and workflow can be a bit unorthodox, but regardless, these are important things to consider and weigh on a case by case basis.

ASK YOURSELF: DO I REALLY NEED THIS VERT/FACE/LOOP?

Every element of your model should either make it more legible, easier to edit, or descriptive of the desired shape. If you have an element that's not, get rid of it.

ROUNDNESS IS RELATIVE

This is a related yet specific enough case of the previous point to deserve its own section. When working on anything cylindrical, it's easy to go a little overboard on the verts describing the circumference. Don't slip into that. For something to appear "round," all that we really need to do is ensure that the silhouette of the "rounded" area appears to occupy roughly the same visual space around the rounded axis; we can fake the hell out of this if we know some specifics about the presentation of the piece. The trick is approximation. Let's look at some examples.

First, if the piece doesn't rotate/move much relative to viewing angle and isn't somewhere that's going to draw attention, we can just use a rectangular prism rotated 45 degrees, and it's good enough. Done deal!

Also, if you're working on something that's a bit more mobile but still not in focus, you can use 6-8 equidistant verts and it'll more or less look "round." Again, what we're after is approximation, not accuracy.

Now these specific numbers and solutions may not fit your specific style or use-case, but my main point is that I want you thinking in terms of approximation and effect.

CUT EVERYTHING

Expanding on the theme of approximation over accuracy, something else to consider when creating your meshes is potentially dividing it into separate, distinct sections, even if it's visually all one piece. "Why?" I metaphorically hear you ask. "What difference will that make?" you proverbially inquire. Consider this example from *Virtua Fighter*:

Pay careful attention to his "joints." You'll notice he actually doesn't have any! The major sections of his body are disconnected, and simply intersecting where you'd expect them to connect, resulting in almost an action figure-like look. This technique can be applied to reduce the geometry required to describe his arm, as the artist didn't have to worry about any "stitching" between the upper arm and the forearm; additionally, this makes rigging so much simpler, as each "vertex island" can be weighted



entirely to its respective bone without having to worry about weight painting or artifacting (side-note: these exact situations are two major factors in my decision to adapt this technique to my in-development game *PUSH HANDS!*).

This is a very versatile tool in your toolkit that can be applied to all sorts of shapes and surfaces, not just humanoids. Be imaginative with it—it can drastically reduce both your poly counts and iteration times!

ANOTHER DON'T

Lastly, never ever ever use the Decimate modifier for a focal element. It's called decimate for a reason, and 99% of the time it will royally screw up your carefully crafted topology. Decimate is great for menial tasks such as quickly generating some level of detail resolution-defined meshes, but in my opinion it's not suitable for creating "low poly versions" of more granular meshes. If you need to de-res a high poly model, consider either carefully hand reducing the geometry to preserve/strengthen the silhouette, or using an old model as reference and starting anew. Seriously, if I ever catch you on Twitter posting decimated geo, so help me.

WHERE TO GO FROM HERE

Dig deeper into Blender! That's right folks, Blender's even bigger and more complicated



Virtua Fighter ignored joints, saving a lot of polygons.

than I've shown you! There are tons of additional modifiers, hotkeys, operations, and completely unrelated portions of the software to explore. The best place to start is the official manual (www.blender.org/manual), which is surprisingly accessible and well written. Consider getting involved at the [BlenderArtists forum](#), joining Facebook groups, posting to Twitter, and anything else to connect you to the community.

Find some artists who inspire you. Study their techniques, try to pick apart their work, try to understand how/why they do things the way they do. [Here are some of my personal favorites!](#)

Practice, practice, practice! Practice, practice, practice, practice, practice, practice.

Thanks for reading! Modeling and low poly have brought a great deal of joy, expression, and opportunity to my life, and I honestly hope that you too can have some fun pushing verts. Practice regularly, study hard, and improve daily!

Ethan Redd is a freelance interactive/multimedia designer based out of Buffalo/NYC who is into bright, bold, fast, intense experiences. You can check him out on [Twitter @EthanRedd](#), catch one of his streams at twitch.tv/kiddraddical, and play a couple of his games at kiddraddical.itch.io.

Do you need a

A PRIMER ON WORKING WITH OTHERS TO FINISH YOUR GAME

By Adam Saltsman

What is a publisher even?

A valid question. In ye olde times (well, it depends on which olde times), a publisher distributed your game. Self-publishing wasn't really an option, or at least a particularly viable one, because there was no internet. Someone had to print your game onto a thing and put it in a box, put that box on a shelf somewhere, then argue with the person that owned the shelf about what things should go where, and so on. They had to print a paper manual because wikis didn't exist yet. They had to take customer support calls because your customer put the floppy disk into the wrong port and now the game is literally physically broken. Or, I dunno, a triceratops ate part of the box or something. It was a grim age.

Now, anyone with access to an internet café can upload their game to something, somewhere, and eventually PayPal will unfreeze their account and send them some money. This is what we call "distribution," and while publishers still help with this sort of thing, it's something that "online stores" and "exciting new platforms" do too.

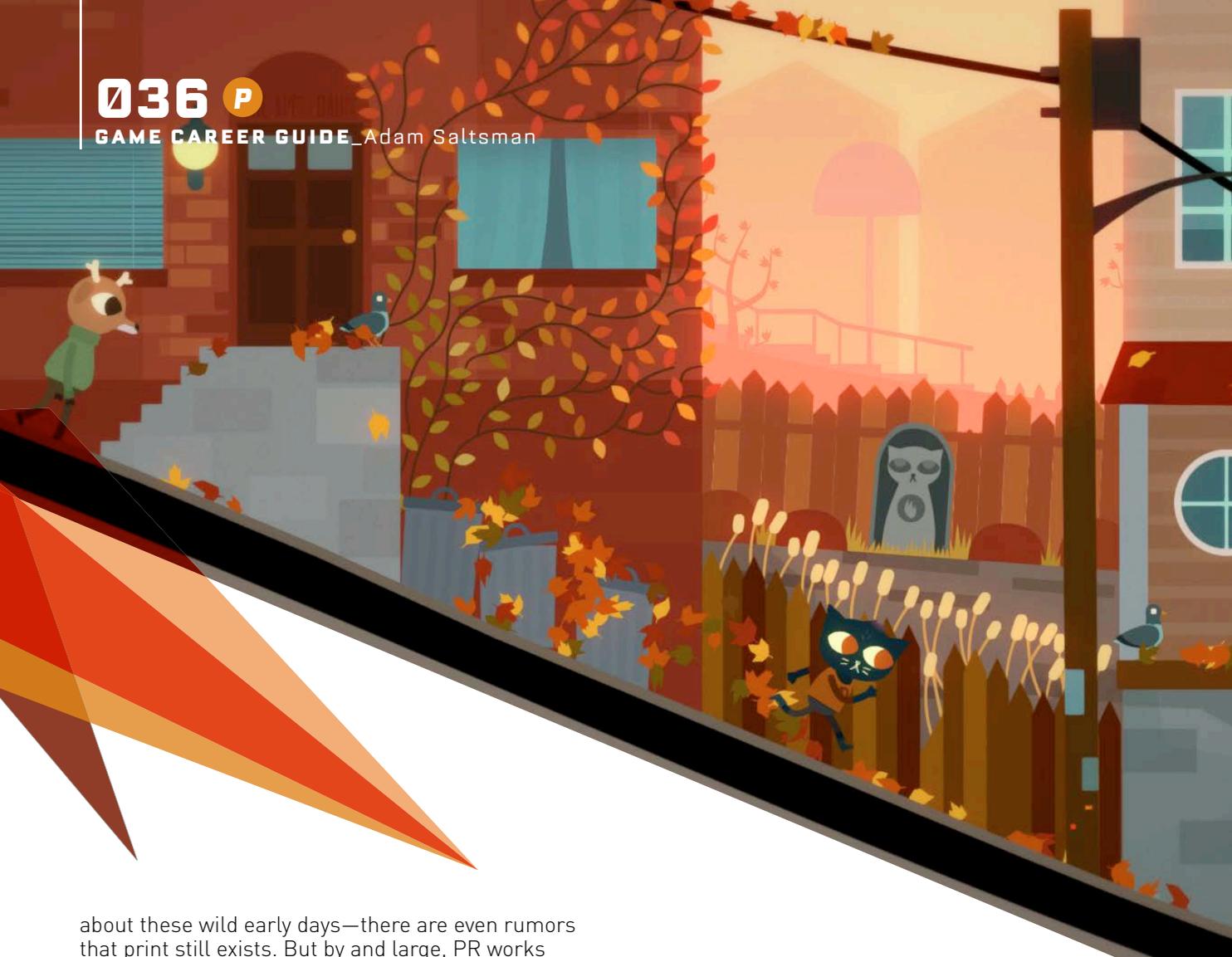
Publishers also used to be in charge of press and marketing. They would send champagne or name-brand perforated cheese-style crackers to some "journalists" in San Francisco, who would then write nice things about your game in a "paper magazine" that they would mail to grocery stores. We've all heard stories



DISCLAIMER: MY COMPANY FINJI SOMETIMES "PUBLISHES" GAMES. IT'S NOT SOMETHING WE DO FOR PROFIT, AND WE'RE NOT COMPLETELY SURE YET WHETHER WE ARE EVEN GOOD AT IT. BUT IT IS A THING THAT HAPPENS!

publisher?





about these wild early days—there are even rumors that print still exists. But by and large, PR works completely differently now: YouTubers, Let's Players, streamers, and their ilk essentially constitute a fractured press of varying independence. Publishers can still help maintain these relationships, but there are also PR consultants and marketing firms that can manage and maintain these relationships.

Publishers also used to fund and finance (most of) a game's development. They would pay everyone's bills, rent, and health insurance while the developers worked themselves to death in a basement somewhere. Then they would fire the team that built the game and hire some cheaper studio in a worse basement to make the sequel, keeping 90% of all the revenue from every version. I personally feel that this was a great system that definitely should have continued indefinitely. Unfortunately, it too works differently now, thanks to Indie Fund, Kickstarter, and Patreon.

When it comes to financing these days, at least among independent developers, it's even more common for people to fund development by alternate sources of income, like a spouse with a day job, and/or working a day job themselves. Or, coming into a substantial inheritance by questionable means—we know what you did, Brad. We're watching you. Always watching.

So what does a publisher do in these strange

modern times? A lot of publishers, especially in the indie space, function almost like surrogate companies, addressing a variety of project-dependent needs. This might include corporate paperwork, legal paperwork, QA and testing, managing platform relationships, accounting, the aforementioned assistance with press relationships, and yes, stepping up to fill funding gaps where other alternatives have left the developers wanting.

don't like to read, just tell me if I should work with a publisher!

Yikes, okay, okay! Unfortunately, the answer is a relatively complicated "maybe." The main factors in the decision here are:

- You
- Your team
- The game
- The publisher

The first, most basic, most important step is to stop thinking of this question at all. "Should I work



Adam Saltsman

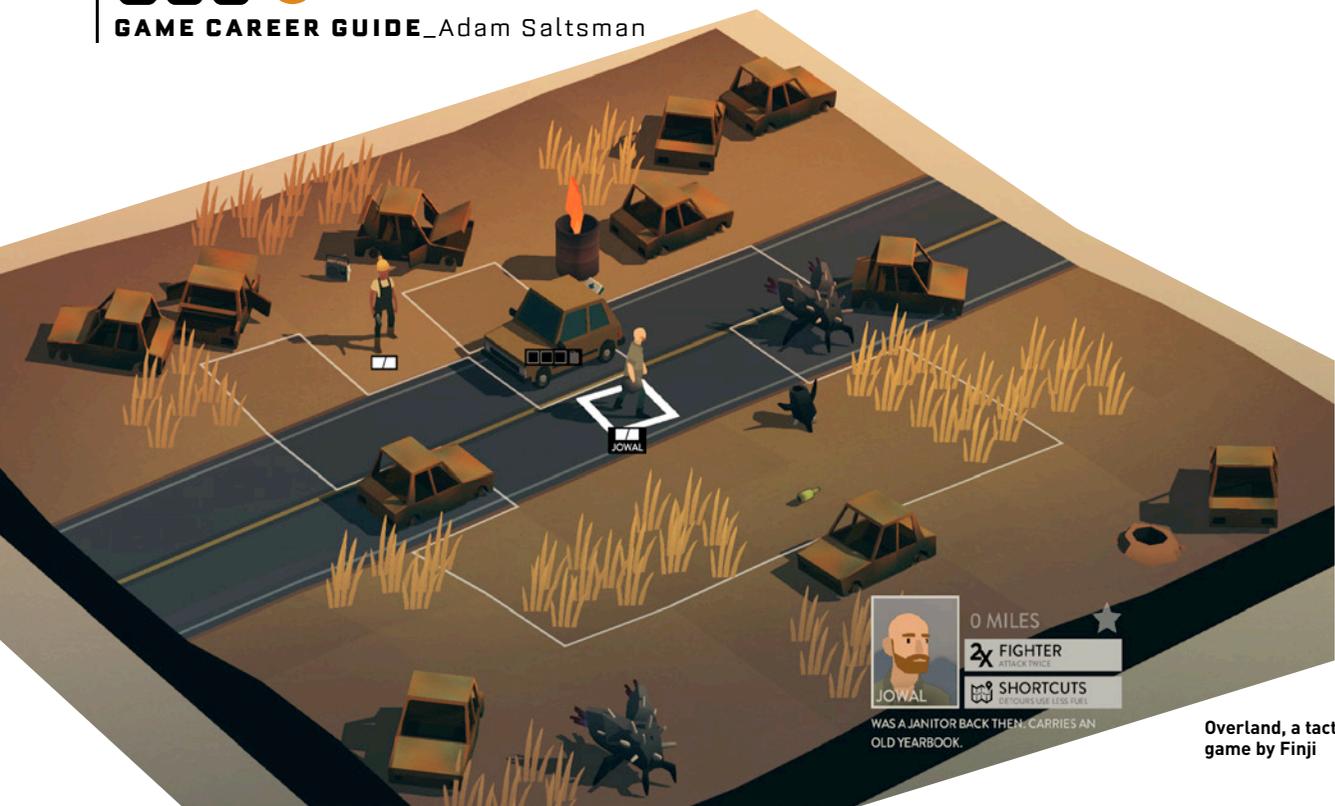
with a publisher?” is kind of a bad starting point. Instead, start here: “What does this project need?”

Is it a commercial project that you intend to launch on Steam and/or consoles? If the answer is no, then probably the things a publisher can offer are almost completely irrelevant to you. Carry on being happy and emotionally satisfied. If for some reason you are trying to make Kind Of A Big Game or something, that’s fine too. Nobody’s perfect! But you do need to parse this goal into components first. For example:

S HUT UP! I want to do a lot of marketing so my game sells a lot of copies!!

I know, trust me, I feel you on this. But what is it that you built/are building? Who’s the audience? Are you doing a paid alpha type of thing? Does that suit the game? What’s the essential appeal? How do you want people to talk about it? Does the game’s design support all those needs?

Assuming you’re interested in producing some sort of commercial project, the goal is probably to raise



Overland, a tactics game by Finji

awareness among folks you think might buy it. We're trying to find people that don't even need hype or recommendations—they just need to know it exists. That should be enough for them.

But even that is a challenging goal! Communication is difficult in person, and it's ten times more difficult through advertising or articles or other weird media. So for different games, this goal of communicating the game's existence means different strategies. Maybe it needs playable demos or supplementals, or sweet trailers, or an LP series, video previews, comic books, a fully choreographed

musical production, and so on.

The first step is just figuring out the most effective way to communicate the existence of the project to total strangers. Doing that will give you a big head start on figuring out whether a publisher is right for you, and which publisher that might be. For an extreme example, hardcore strategy games might do better partnering with, say, Paradox, instead of Adult Swim. Both companies have good reputations right now, but Paradox has a big audience that is already aligned with that hardcore strategy vibe. That shouldn't be the only factor considered, but it would be crazy to ignore it.

Maybe you don't need a publisher at all. Maybe your team just needs a super good video editor. Maybe you need a PR consultant to connect you with streamers. Maybe you need a web designer to make the perfect landing page. If you are making a commercial project, I urge you to do your best to figure out how to expose the things that make that project special. Maybe there is a publisher out there that can help you do that. Maybe there isn't. It depends on all these very specific things.

Righteous! Damn the man!
Self-publishing all the way!

I couldn't agree more! That's what we do, and we love it. It only consumes three to six months a year in

development time, too; it's really great, no downsides at all, no indeed. Seriously though, I've seen other devs call this the "non-game-dev" part of a project, and that's sort of true but sort of misleading too. If you're making a commercial game, helping the game find its audience is a part of making it. Sorry!

And somebody's got to do it. This is a big job that takes thought and care and time, just like any other design problem. This one just involves a little less math (sometimes) and a little more empathy (most of the time). I find it helpful to look at a publisher as essentially a work for hire or other basic pro/con situation. Do I want to put in the time and effort and investment to do all this stuff myself? Or do I want help? And how much do I want to pay in the long and short term for that help?

For a lot of developers, giving up a bunch of their revenue hurts, but getting to focus mainly on the internal parts of the game is worth it. Or the fit is so good that whatever revenue they are giving up, it's worth it for the audience they're gaining through that relationship. For others, it's just not the right fit. Maybe they want to build their own studio, or are hopelessly neurotic control freaks (like us). For still other developers, there's a kind of hybrid approach—get some marketing from over here, get some funding from over here, make some trailers in-house, team up with this porting studio. That works well for some companies, too.

But again, the key here is for your decisions to be driven by the needs of the project. There's no one-size-fits-all approach here (unless you have a million-dollar marketing budget laying around, in which case, why are you reading this?). Even though revenue targets might be a part of your goals, looking at marketing primarily through a lens of "how do I sell more copies" is counterproductive, in our experience. Focus on doing right by your project and finding the people that will love it. If you don't take this part seriously, it is probably going to be very hard to hit whatever revenue goals you set anyway.

Okay. Assess our needs, and look for one or more solutions that can address them. I feel like I knew that already. How do I actually work with these outside "partners" or whatever, if I decide they are right for my project?

That depends on the partner and the need and so on, but the basic idea is to practice being empathetic. This is not the same thing as being sympathetic or emotional; I'm just talking about being able to put yourself in your prospective partner's shoes.

Let's look at funding as a first example. Say you need \$20,000 to finish your game. There are a few ways you could try to source those funds, and they all have different trade-offs. What does that \$20K get you? How do you structure repayment? How do you support the folks that lent you those funds? For example, multi-tiered repayments (when lenders get lower rev share after they recoup their initial investment) are a handy way of managing some of that risk for all parties, but they are more complicated to negotiate and structure in a contract.

And these things vary wildly depending on where the funding comes from (parents, investors, backers, animal shelters, supervillains, so on). If investors are loaning you money, is the thing they're asking for in return proportional to their risk? If they give you a little, are they expecting a lot in return? Or are you expecting your investors to take on all of your risk? This can change over time as well. If you have to return to your investors to borrow more money, that can change the so-called "risk profile" of your project quite a bit. A bigger investment means a bigger risk, so they might need to change the repayment terms to accommodate the new state of the project. A little empathy goes a long way here (on both sides).

Actually, this idea of risk and reward applies to marketing, PR, QA, localization, and all of that. Does localization actually benefit your game? Is it enough to localize the store descriptions and then fan-source the in-game text later? Or maybe you just need to localize to Russian and Brazilian Portuguese? QA is more important than it seems, especially on hardware-fractured environments like PC and Android, but the impact of QA on games that are doing public alphas versus launching cold is really different. Maybe the money you were going to use on a PAX Prime booth could be used to hire someone to set up some in-office appointments with games press for more dedicated, one-on-one coverage. Or maybe your press-tour budget should be spent on a booth! It depends a lot on what you're making and how you're making it.

And I've almost entirely glossed over production, which in retrospect is pretty typical for a programmer/designer like me. But production can involve a whole lot, like having to step in to straighten out development bottlenecks, planning check-ins, basic

meat-shielding, translation efforts, or just creating your whole schedule in the first place. These things usually mean the difference between shipping and not. But again, it depends on the size of the team, timeline, and so on.

So: what are your needs, and how can you address them? What parts do you want to work on? What parts don't you want to work on? If you can figure this stuff out, you will be in much better shape when you start talking to anyone, anywhere, publisher or otherwise, about helping you ship.

Are you done yet?

I want to end with a list of things that in our experience are worth taking very, very seriously anytime you collaborate with anyone, be it a publisher, another developer, an audio engineer, a florist, whatever:

- Contracts won't prevent someone really malicious from ruining your life, but they sure won't hurt.
- Be honest—blunt, if necessary—about your needs and the needs of the project.
- You can hope that your game will “speak for itself” but it's just that—hope.
- That said, it is easier to raise awareness for a game that wears its heart on its sleeve.
- Look for ways your interests align with platforms and publishers.
- Empathy is so important—try to remember to put yourself in their shoe

Above all, remember: always keep both of your middle fingers ready when the vultures come for you.

Adam Saltsman is an independent game developer based in Grand Rapids, Michigan. His past projects include Canabalt, and the flash development platform Fixel. He is currently working hard on his tiny tactics game Overland. Find him on twitter at [@ADAMATOMIC](https://twitter.com/ADAMATOMIC).



IMAGINE U STUDY AT THE #1 GAME DEV PROGRAM IN THE WORLD

THE UNIVERSITY OF UTAH'S ENTERTAINMENT ARTS & ENGINEERING PROGRAM



LEARN FROM INDUSTRY PROS

GAMES FOR RESEARCH

TRULY INTERDISCIPLINARY

A woman with long, wavy red hair is seen from behind, sitting at a desk in a computer lab. She is focused on her work, with her hands on a keyboard. The desk is cluttered with a computer monitor displaying a game development interface, a smartphone, and some papers. In the background, other students are working at their desks, some looking at their screens, others talking. The environment is a modern, well-lit studio or classroom.

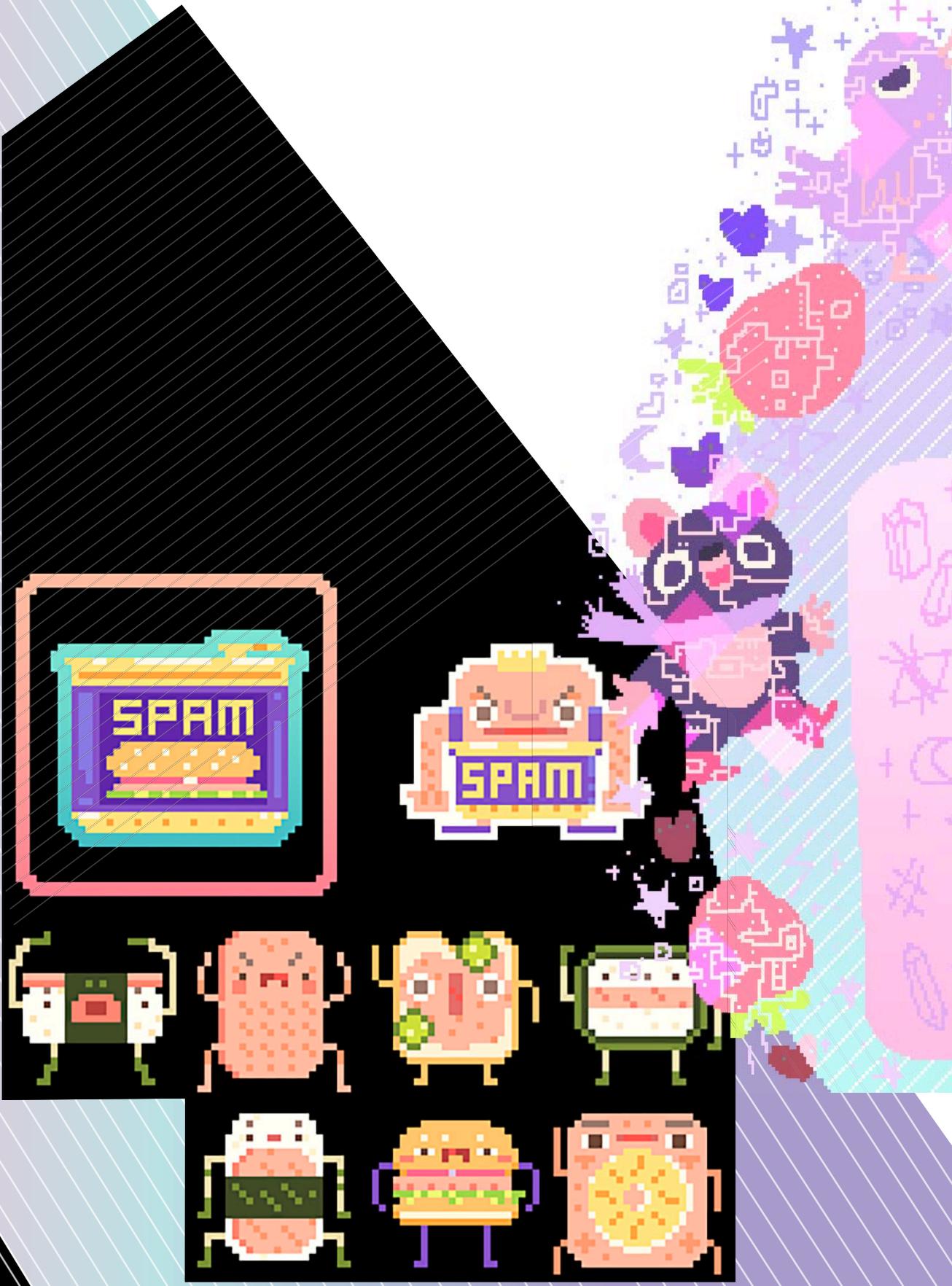
Go from playing games to designing them.

Honours Bachelor of Game Design

Our diverse programs, and passionate, experienced professors will give you the skills to succeed. As a grad, you'll join a vibrant Sheridan alumni community who work as animators, filmmakers, designers and creative professionals around the world.

gaming.sheridancollege.ca

Sheridan | Get
Creative





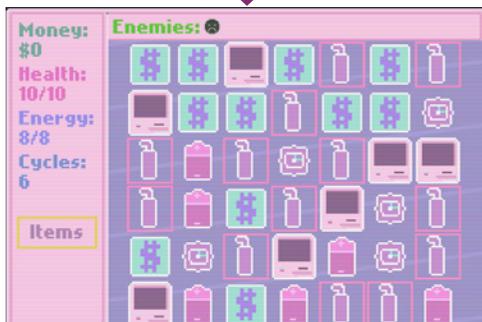
Beglitched_

pm 043

STUDENT POSTMORTEM_Alec Thomson

Beglitched is a game about insecurity in our computers and in ourselves. Players discover a mysterious computer belonging to a legendary hacker, and they gradually uncover her secrets through a combination of network navigation and Match-3-inspired hacking.

Beglitched began as my MFA thesis at the NYU Game Center. After a semester of solo work, I invited Jenny Jiao Hsia to join the project, and we eventually brought it to the NYU Game Center Incubator with the intention of releasing it as a commercial product after finishing school. While the game is still in development and pending a commercial release, we've been fortunate that it received quite a bit of attention and acclaim during its time as a student project—culminating in its win for Best Student Game at the 2016 Independent Games Festival. Producing a project of this scale was a huge learning experience for both of us, so we wanted to share a few things that went well and a few that could have gone better during its development.



WHAT WENT RIGHT

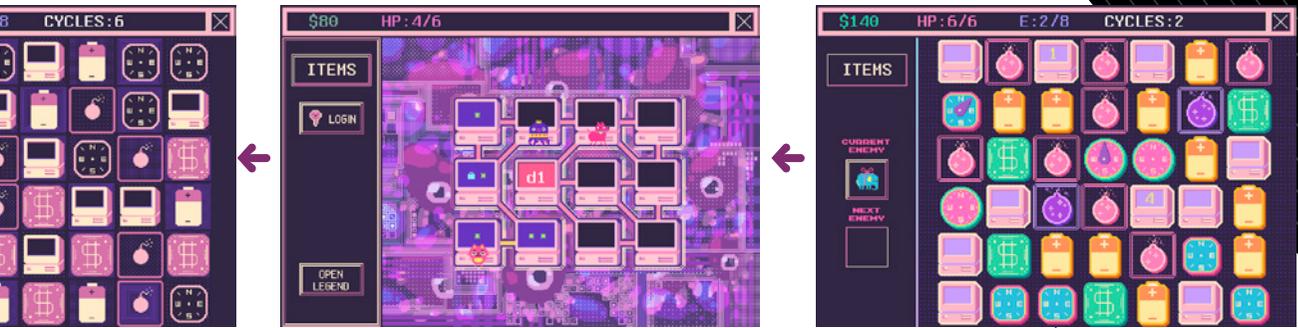
1. Iterating on the core design.

The initial game idea was simple: Try to make debugging computer programs a fun and accessible experience. It's no secret that many programmers find debugging one of the most frustrating parts of development, but I firmly believed that under the right circumstances (i.e., with a strong knowledge of the underlying system and good tools to probe the system for information), it could be cathartic and empowering.

With this idea in mind, I started prototyping a game about solving procedural crimes in an autonomous city. The initial prototype was an obvious failure, ending up even more frustrating to interact with than actual debugging. My next attempt resulted in a slightly more interesting prototype that involved solving deduction puzzles to capture cybercriminals on a computer network. For a while, it felt like I'd "found" my thesis, and I spent months trying to expand this network puzzle into something more substantial.

During that period, I made six or seven prototypes that were all variations on the original network puzzle (including one ill-conceived "investment" game). Ultimately, none of them felt like they added enough depth to the game, and I made the controversial decision to scrap everything I had up to that point (almost halfway into the thesis period) and start working on Match-3 prototypes instead. After three or four attempts at producing a computer-themed deduction-focused Match-3, I had a small system that I found satisfying. Then someone suggested combining my





original network prototype with the new Match-3 prototype, and suddenly the core gameplay of Beglitched emerged. Whew! That took a while.

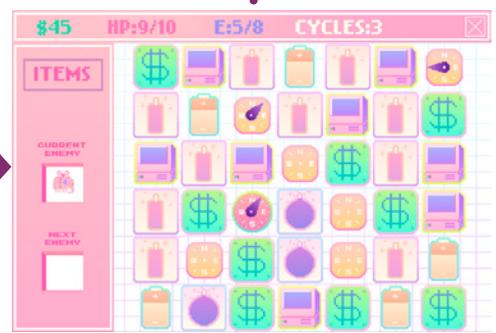
Ultimately, the lesson I took away here was to not be afraid to throw away existing work if that leads to a stronger game, even if each iteration takes you farther from your original vision.

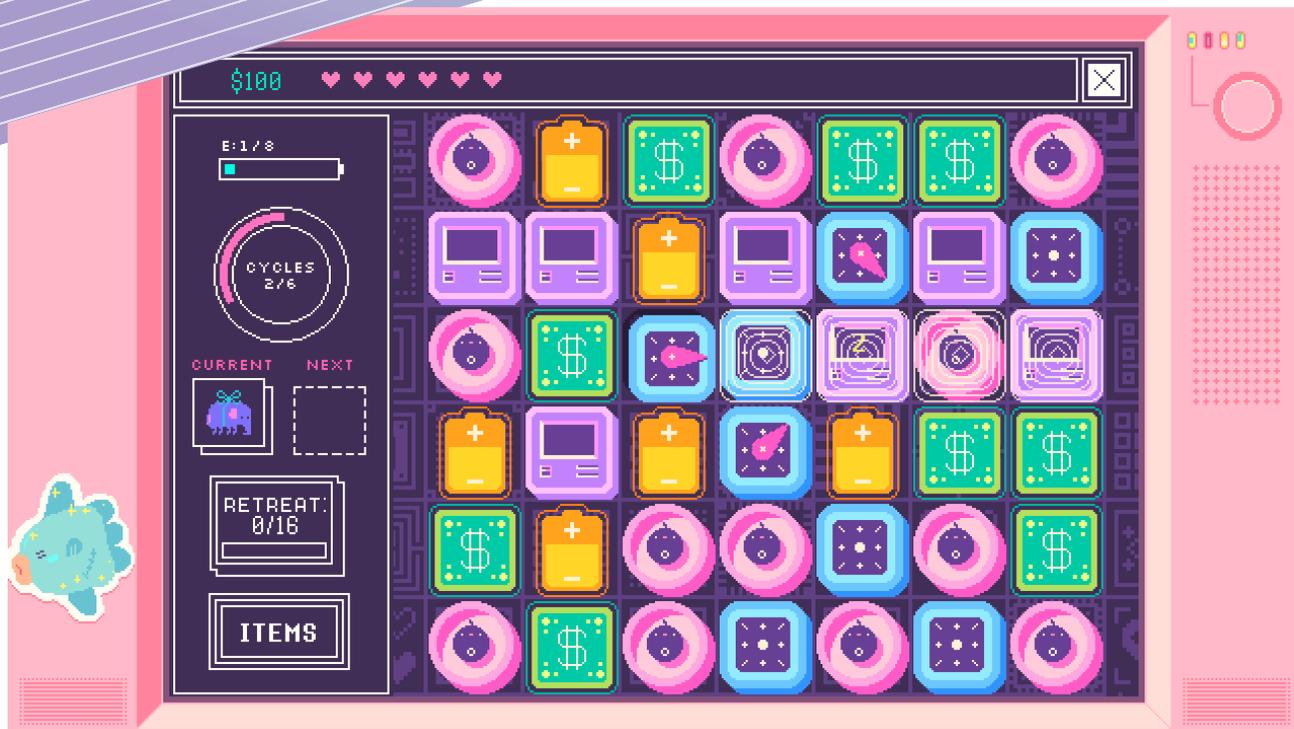
2. Creating stronger visuals.

I originally planned to simply keep going with the visual direction I had created for the initial network prototype. My thesis mood boards were a collection of very traditional cyberpunk images: green text on black backgrounds, dark cityscapes, geometric patterns, circuit boards, etc.

I didn't question this art style until Jenny joined the project after the first semester. One of the first things she did was show me a secret Tumblr she had of bright pastel images, many of which depicted computers and game consoles. Since the two of us tend to heavily prefer cute visuals (see our first project together: Stellar Smooch), it became pretty clear that the only real reason I had been holding on to traditional cyberpunk imagery was because the game was about hackers. With this realization, we threw the old cyberpunk aesthetic out the window and charged full speed into making the game "cyberpink" (a term you can blame entirely on NYU Game Center Program Director Frank Lantz).

This new visual direction not only made the game look unique and interesting, it also opened the door for a





completely different mood and narrative that Jenny and I found a lot more exciting.

3. Spending time on the tutorial.

Bglitched's game mechanics are difficult to describe. The game intentionally takes different genre conventions (from Match-3 games like Bejeweled and deduction games like Minesweeper) and mashes them together in unfamiliar ways.

Early on in the development of the fully realized game (i.e., after the long period of prototyping), it became obvious that teaching people how to play it would be one of our greatest challenges. Even with an incredibly precise and detailed tutorial, we found that many players simply tried to fall back on conventions they were familiar with. Players who had experience with Match-3s would play the game exclusively like a Match-3, unaware of other goals they should be pursuing. To combat this, we deemphasized the game's Match-3 elements, which had the side effect of alienating players who were unfamiliar with Match-3 mechanics. Yikes!

Ultimately, we had to iterate on our tutorial sequence many times before we found something that worked. This process included removing a lot of text, splitting ideas into multiple encounters (e.g., the first battle requires no deduction because the enemy hacker conveniently forgets to hide), and narrativizing the learning process (e.g., a lot of what the player learns is due to the incompetence of

the computer opponents, rather than from a tooltip telling them how to play).

While it took a lot of time and energy to come up with a learning curve that actually works, it was absolutely necessary. I doubt the game would have gotten much traction if players had a harder time getting started.

4. Letting the narrative reveal itself.

We began work on the game without a clear idea of our narrative, and I think that ultimately worked in our favor. When Jenny joined the project, my only narrative goal was something along the lines of "I feel like a lot of kids have this fantasy that they'll accidentally find the old spell book of a powerful wizard tucked away in the back of a dusty bookstore. Well, let's roll with that fantasy, but in a modern setting involving hackers and computers."

From there, the mood of our game became pretty clear: Hackers are wizards, computers are magic, nobody (especially the player) has any idea what they're doing. Just that rough idea allowed us to have a lot of fun coming up with goofy characters, silly events, and a playful environment. Eventually, the level progression and characters started to form the thread of a plot that explored additional themes, but I think if we'd married ourselves to a specific story line earlier on, we would have ran the risk of losing the playful mood that ultimately defines the game.



Get
Certified

Validate your Unity knowledge and skills.

Earn a credential to help you build your résumé.

Stand out from the crowd.

Get Unity Certified

Unlock the ultimate perk for
your career in games.

For more information and to register visit getunitycertified.com/gamasutra





5. Making our own music.

Something we saw a lot of our fellow students struggling with during development was licensing music or hiring external musicians. This created a lot of overhead; they had to negotiate contracts, provide upfront funds, or navigate integrating an additional team member late in the development process.

We were able to avoid these issues by composing and producing our own music for the game. Even though our music production skills don't quite match those of a professional, we ultimately created music that fits our vision faster and cheaper than our peers, while keeping our decision-making team to the very manageable size of two people.

(I also improved my music skills in the process, so it was a double win!)

WHAT WENT WRONG

1. Too much confusion in the core interactions.

As I mentioned earlier, teaching people how to play our game was difficult. Looking back on our design process, one reason for this is that the game's core interactions are a bit unfocused and require a certain level of mental commitment to learn. For instance, the Match-3 battle system involves both dragging tiles and clicking on tiles (with different effects in both cases), which continually caused confusion for our players.

2. Overreliance on "over the shoulder" play-testing.

Much of our time testing the game in its early stages involved the two of us standing behind a new player and silently taking notes as they struggled to learn the game. It wasn't until we began testing the game in different settings (such as sending a build to testers and asking for written feedback when they could get around to it) that we realized the "over the shoulder" approach was producing unintended consequences in the feedback we received.

For instance, we found that most players in the "over

DISRUPTOR BEAM

WOULD YOU RATHER KICK ASS OR TAKE NAMES?

If you're an ass-kicking Mobile or Platform Engineer, UI/UX pro or Lead Game Designer, you belong at Disruptor Beam.

Help our diverse and talented crew create games based around some of your favorite IPs. We're a company where your healthy disregard for the rules is wholly embraced.

Learn more about us and our industry-disrupting benefits (like unlimited paid time off) at DisruptorBeam.com.

VFS GAME DESIGN

INDUSTRY-LED **INDUSTRY-LEADING**

Vancouver Film School's acclaimed one-year Game Design program doesn't just teach you the in-demand skills to break into the gaming industry... **it IS the industry.**

- » Exclusive education partner with Crytek in VR First Initiative to create Canada's first virtual reality lab and curriculum
- » World-class Unity game engine education with official certification
- » Instructors + Mentors with industry experience at top studios
- » Industry-endorsed + recognized curriculum

If you're really serious about a career as a game designer or developer — you need to learn more about VFS today!

“ Vancouver Film School has stood out year over year as an outstanding academic institution forming professionals with relevant and high quality market skills. **Their hands on curriculum and vast faculty knowledge of the entertainment and media industry has made them one of most desirable programs worldwide.**

- Unity Technologies

“ Vancouver Film School generates some of the best graduates in the world right now. Some of the most brilliant, dedicated systems designers I've ever worked with have emerged from this program. We are always interested in visiting with new VFS grads and are consistently impressed with their passion and drive to grow as game-makers. VFS is a terrific program.

- Tim Fields, Vice President of Kabam, Inc

VFS

VANCOUVER FILM SCHOOL
LEARN MORE AT **VFS.EDU/GAMECAREER**



the shoulder” setting felt an intense pressure to perform or complete the game quickly. It didn’t help that this normally occurred at larger play-testing events involving multiple games. This pressure caused them to skip or miss hints or feel judged/inadequate when they inevitably messed up. Testers who had the opportunity to play the game privately on their own time learned the game more easily. Additionally, since we had multiple hours of game content, the limited “over the shoulder” sessions could never produce feedback on the later portions of the game.

In retrospect, we now recognize that all forms of testing can be useful, and it’s important to not over-rely on a single type of testing or you run the risk of skewing feedback toward a specific environment—which, in our case, wasn’t even the environment we expected most players to be in when playing the final game release.

3. Spending too much time on some prototypes. I mentioned earlier that Beglitched went through a very long prototyping process before we settled on a design. While this process was really important to finding the game we wanted to make, I’ll be the first to admit there were prototypes that received more attention than they deserved.

For instance, at one point during the prototyping phase, I was only focused on the prototypes that featured the early “network game” concept as their core. I spent about a month trying to hammer one of the least compelling prototypes into something worthwhile before realizing that the entire direction was flawed. It’s easy to criticize that decision in hindsight, but I do feel like more planning and careful thought during this phase could have revealed the flaws of some of these prototypes earlier.

4. Focusing too much on procedural generation for content.

Beglitched as it exists now contains a mixture of procedurally generated and fully designed levels. Early in development, we were very focused on making procedural

generation a core element of the design, so we refused to even consider making hand-designed levels. Fortunately, we eventually did give hand-designed levels a try and were pleasantly surprised at how well they meshed with the game world and complemented the existing procedural levels.

It’s possible that if we had given up our “procedural only” requirement earlier, we would have had more room for experimentation (in terms of puzzles, mechanics, and even narrative) during the student phase of the project’s development. Fortunately for us, the project continued beyond the student phase so that experimentation was still able to occur.

5. Letting the game become too serious. Occasionally we let the game become too serious, something we struggle with even as we continue development. This was a problem with the early work I did before Jenny joined the project (Cyberpunk! Green! Evil corporations! Boring!). As the final narrative and darker themes of the game start to take shape, we have to constantly remind ourselves of this early misstep and ensure that we aren’t betraying our prime directive of keeping the overall mood of the game playful and silly.

Overall, working on Beglitched as a student project was a pretty great experience. We look forward to releasing the full game in the near future. Wish us luck!

Alec was an MFA student at the NYU Game Center. Then he graduated, and now he teaches there while also making games! Games he’s released include Stellar Smooch (also made with Jenny Jiao Hsia) and Stair: Slide the Blocks to Ascend. He is currently working on Beglitched and other projects. Find him on twitter at @bad_tetris and Jenny at @q_dork.





VANCOUVER'S MASTER OF DIGITAL MEDIA PROGRAM

Enhance Your Undergrad With This Innovative Graduate Degree

**CANADA'S 1ST
PROFESSIONAL
GRADUATE DEGREE IN
DIGITAL MEDIA**

- ▶ Industry-Focused Master's Program
- ▶ Competitive Scholarship Opportunities
- ▶ Startup Business Support
- ▶ Graduate from 4 Major Canadian Universities

Housed at the Centre for Digital Media in Vancouver, BC, this 1-year plus internship program engages students in real-world projects and industry connections. Students graduate with the skills to work in the top jobs in the digital media industry as creators, producers, designers, managers, art directors and entrepreneurs.

a collaboration between



SFU

emily carr
university of art+design

BCIT

CENTRE FOR
DIGITAL MEDIA

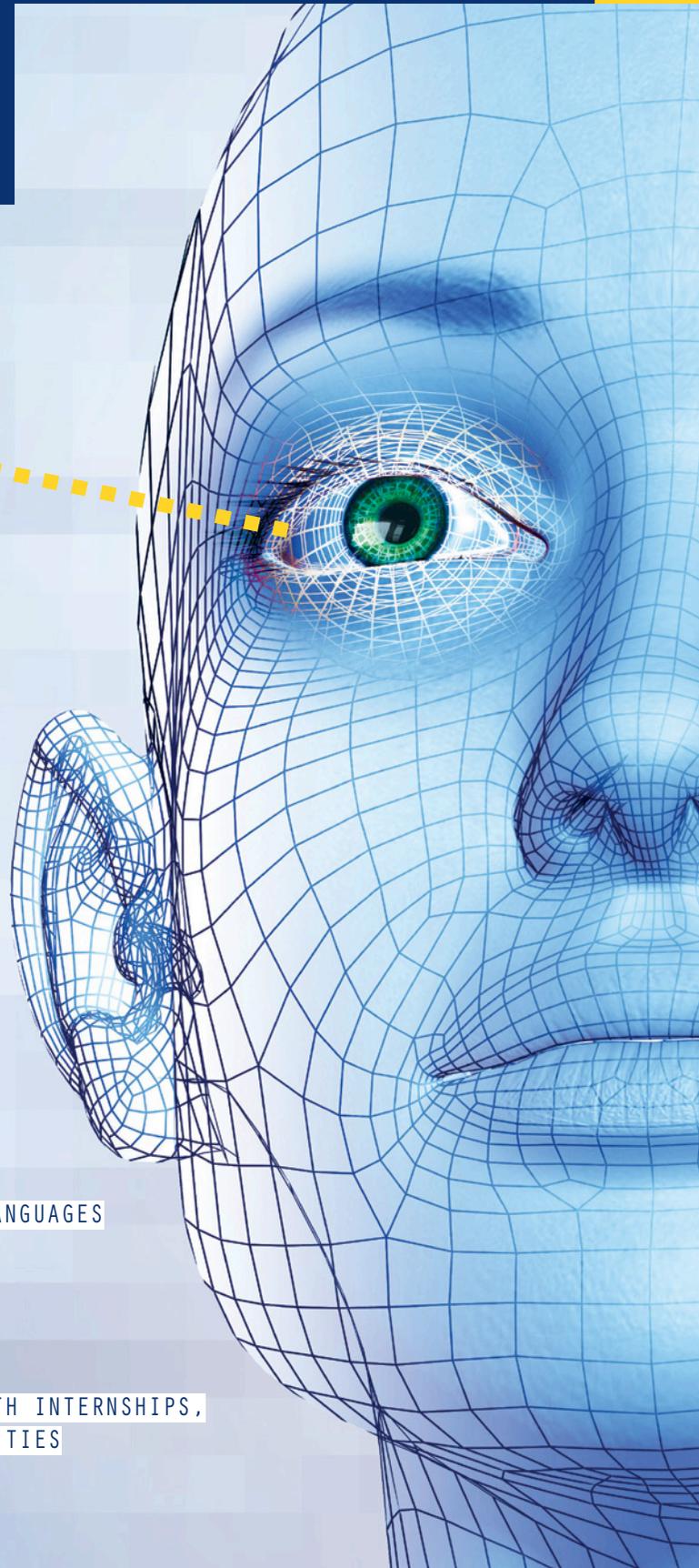
LEARN MORE ABOUT THE MDM PROGRAM ▶ THECDM.CA

BRING YOUR CHARACTERS TO

LIFE

SEE YOURSELF
SUCCEED
IN SOUTHERN
NEW HAMPSHIRE
UNIVERSITY'S
CAMPUS AND ONLINE
VIDEO GAME
DEGREE PROGRAMS

- LEARN C#, C++ AND JAVA PROGRAMMING LANGUAGES
- MASTER UNREAL AND UNITY GAME ENGINES
- GET THE INSIDE TRACK FROM FACULTY WITH INDUSTRY EXPERTISE
- TALK THE TALK (AND WALK THE WALK) WITH INTERNSHIPS, COMPETITIONS AND NETWORKING OPPORTUNITIES



[CLICK HERE TO FIND YOUR VIDEO GAME DEGREE PROGRAM TODAY.](#)

snhu.edu
See Yourself Succeed

Southern
New Hampshire
University

Game School Directory

There are tons of options out there in terms of viable game schools, and this list is just the starting point to get you acquainted with the schools near you (or far from you, if that's what you prefer!). No list could possibly tell you all you need to know, so you should use this directory to find schools you're interested in, and then research them with our comprehensive directory at GameCareerGuide.com/Schools. Over there, you'll find all sorts of information, including:

- In-depth school profiles
- Listings of specific game development programs and degrees offered
- Tuition and financial aid information
- Student-to-faculty ratio
- Online class offerings
- Actual playable student games

While you're there, you can use our Digital Counselor app (GameCareerGuide.com/digital_counselor) to search through the full database of schools by location, degree offerings, online study options, and more to find the game development school that's right for you!

TITLE	URL	LOCATION	PROGRAM OFFERED
3D Training Institute	www.3dtraining.com	New York, NY	3D Foundation Workshop , 3D Project-Based Source
3dmx Digital Design University	www.3d.com.mx	Zapopan, Jalisco, MX	Videogame Development
3D-Online	www.3D-Online.com	Manhattan Beach, CA	Game Engine Fundamentals, Introduction to 3D Graphics Programming, Game Development and 3D for Web and Wireless Devices
Abilene Christian University	www.acu.edu/academics/sitc/index.html	Abilene, TX	Digital Entertainment
Academy of Art University	www.academyart.edu	San Francisco, CA	Maya, Game Design, Character Animation, Computer Graphics, 3D Modeling, Modeling, 2D, Character/Tech, VFX/ Compositing, Storyboard, Game Environments, Background Painting, Visual Effects, 3D, Web Design 1, Visual Development, 3D Animation
Academy of Interactive Entertainment Sydney	www.aie.edu.au/sydney	Ultimo, NSW, AU	Advanced Diploma of Professional Game Development (Software Development)
Academy of Interactive Entertainment, Lafayette	www.theaie.us	Lafayette, LA	Advanced Diploma of Professional Game Development (Software Development)
Academy of Interactive Entertainment, Melbourne	www.aie.edu.au	Melbourne, VIC, AU	Advanced Diploma of Professional Game Development (Software Development)
Academy of Interactive Entertainment, Seattle	www.theaie.us	Seattle, WA	Advanced Diploma of Professional Game Development: Game Programming, Advanced Diploma of Professional Game Development: Game Art & Animation, Advanced Diploma in Screen & Media: 3D Animation & VFX for Film
Academy of Interactive Entertainment, Watson	www.aie.edu.au/	Watson, ACT, AU	Advanced Diploma of Professional Game Development (Software Development)

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
Acadia University	www.cs.acadiau.ca	Wolfville, Nova Scotia, CA	BCSS Game Development
Alberta College of Art & Design	www.acad.ab.ca	Calgary, Alberta, CA	Visual Communications Design Program: Character Design Stream
Algoma University	www.mastersingaming.com/algoma	Sault Ste. Marie, Ontario, CA	Master of Science (Computer Games Technology)
Algonquin College	www3.algonquincollege.com/medi-aanddesign/program/game-development/	Ottawa, Ontario, CA	Game Development
American University	www.american.edu/gamelab/	Washington, DC	MA in Game Design
Angelo State University	www.angelo.edu	San Angelo, Texas	Computer Science
Animation Gurgaon	www.frameboxl.in	Gurgaon, Haryana, IN	Redboxx-I
Animation Mentor	www.animationmentor.com	Emeryville, CA	Diploma in Advanced Character Animation Production, Diploma in VFX Fundamentals, Diploma in Advanced Animal and Creature Animation Production, Diploma in Animation Fundamentals
AnimSchool	www.animschool.com	Orem, Utah	3D Character Program, 3D Animation Program
Anne Arundel Community College	www.aacc.edu	Arnold, MD	Game Art and Design Transfer AS Degree, Game Interface Design AAS Degree, Interactive Technologies AAS Degree
Art Center Design College - Albuquerque	www.theartcenter.edu	Albuquerque, NM	Animation
Art Center Design College - Tucson	www.theartcenter.edu	Tucson, AZ	Animation
Art Institute of California - Los Angeles	www.artinstitutes.edu/losangeles	Santa Monica, CA	Game Art & Design
Art Institute of California - Orange County	www.artinstitutes.edu/orangecounty	Santa Ana, CA	Media Arts & Animation, Game Art & Design, Visual & Game Programming
Art Institute of California - San Diego	www.artinstitutes.edu/sandiego	San Diego, CA	Game Art & Design
Art Institute of California - San Francisco	www.aicasf.aii.edu	San Francisco, CA	Game Art & Design, Visual & Game Programming
Art Institute of Charlotte	www.artinstitutes.edu/charlotte	Charlotte, NC	Digital Filmmaking & Video Production, Web Design & Interactive Media, Graphic Design
Art Institute of Las Vegas	www.artinstitutes.edu/lasvegas	Henderson, NV	Game Art & Design
Art Institute of Pittsburgh	www.artinstitutes.edu/pittsburgh/	Pittsburgh, PA	Entertainment Design, Media Arts & Animation, Game Art & Design, Interactive Media Design
Art Institute of Seattle	www.artinstitutes.edu/seattle	Seattle, WA	Media Arts & Animation, Web Design & Interactive Media, Audio Production, Game Art & Design, Animation Art & Design, Audio Design Technology
Art Institute Online	www.aionline.edu/information/programs/animation/game_art_design/?cid=GCARG_121508_Profile_Lin	Pittsburgh, PA	Game Art & Design
Art Institutes International Minnesota	www.artinstitutes.edu/minneapolis	Minneapolis, Minnesota	Design Management, Media Arts & Animation, Photography (BFA), Visual Effects & Motion Graphics, Interactive Media Design
Artcode Game Academy	www.artcode.la/	San Salvador, San Salvador, SV	Game Programming and Development

TITLE	URL	LOCATION	PROGRAM OFFERED
Asian Institute of Gaming and Animation - AIGA	www.aiga.in	Bangalore, Karnataka, IN	Game Programming, Game Art
Attend iD Tech Camps	www.iDTech.com	Various	Tech Programs for Kids
Austin Community College	www.viscom.austincc.edu/	Austin, TX	Visual Communication, Game Development Institute with specialization in Programming, Art or Design
Backstage Pass, School of gaming	www.backstagepass.in	Hyderabad, Andhra Pradesh, IN	Game Design and Development
Baker College Online	www.baker.edu/	Flint, Michigan	Bachelor of Computer Science, Game Software Development
Becker College	www.becker.edu/gamedev	Worcester, MA	Game Development and Programming, Game Design
Bellecour Ecole	www.bellecour.fr	LYON, , FR	Bachelor in Concept Art, Bachelor in 3D Art and Animation, Bachelor in Game Design, Master in Concept Art, Masters in 3D Art and Animation, Master in Game Design
BES La Salle - Universitat Ramon Llull	www.salle.url.edu/	Barcelona, Catalonia, ES	Game Design and Development
Blekinge Institute of Technology	www.gamescience.bth.se	Karlshamn, Blekinge, SE	Digital Game Development, Master of Game Design
Bloomfield College	http://www.bloomfield.edu/academics/degrees-programs/creative-arts-technology/bachelors-programs/game-design	Bloomfield, NJ	Game Programming, Graphics, Game Design, Animation, Music Technology
Boston University Center for Digital Imaging Arts	www.cdiabu.com	Waltham, MA	Animated Short, 3D Animation + Interactive Media, Game Art & Character Animation
Boston University Center for Digital Imaging Arts	www.cdiabu.com/	Washington , DC	Animated Short, 3D Animation + Interactive Media, Game Art & Character Animation
Bradley University	www.bradley.edu/academic/departments/interactive/	Peoria, Illinois	Animation, Game Design
British Columbia Institute of Technology, School of Computing	www.bcit.ca/study/programs/B25hbtech	Burnaby, British Columbia, CA	Bachelor of Technology - Computer Systems Technology - Games Development Option
Broadview Entrainment Arts University	www.beau.broadviewuniversity.edu/	Salt Lake City, Utah	Sequential Imaging, Digital Video and Media Production, Media Business, Entertainment Design, Music Business, Game Art
Brown College	www.contact.browncollege.edu/game-design-development.aspx?src=60246	Mendota Heights, Minnesota	Bachelor of Science in Visual Communications - Multimedia Emphasis, Bachelor of Science in Visual Communications - Graphic Design Emphasis
Brown University	www.cs.brown.edu	Providence, RI	Computer Science
Brunel University - School of Arts	www.brunel.ac.uk/about/acad/sa/artsub/filmtv/videogames	Uxbridge, Middx, GB	MA Digital Games: Theory & Design
Bryan College	www.bryancolleges.edu/gaming-robotics-specialist.asp	Springfield, Missouri	Gaming and Robotics , Gaming and Robotics Specialist
C.W. Post Campus of Long Island University	www.liu.edu/cwpost/game	Brookville, NY	Master of Arts in Digital Game Design and Development
California Institute of the Arts	www.calarts.edu	Valencia, CA	Character Animation, Experimental Animation
California State University Channel Islands	www.cs.csuci.edu/	Camarillo, CA	Master of Science in Computer Science, Bachelor of Computer Science, Interdisciplinary Minor in Game Design and Development
California State University East Bay	www.multimedia.csueastbay.edu	Hayward, California	MA in Multimedia and Interactive Content

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
California State University, Fullerton	www.fullerton.edu	Fullerton, CA	Entertainment Art/Animation
Cambridge Coding Academy	www.cambridgecoding.com/	Cambridge, , GB	Build a Game Workshop
Camden County College	www.camdencc.edu	Blackwood, NJ	Game Design & Development, Computer Science, Computer Graphics
Canada College - Redwood City - CA	www.canadacollege.edu/multi-media	Redwood City , CA	Multimedia Art & Technology
Capilano University	www.gradshow.com/	North Vancouver, British Columbia, CA	Digital Animation
Carnegie Mellon University, Entertainment Technology Center	www.etc.cmu.edu	Pittsburgh, Pennsylvania	Entertainment Technology Center
Centennial College	www.bccc.com	Toronto, ON, CA	Game Design & Development
Center for Distance Education	www.cd-ed.com	Sydney, Nova Scotia, CA	3D Advanced: Character Animation, 3D Animation, 3D Game Artist
Centre for Arts and Technology - Fredericton Campus	www.digitalartschool.com	Fredericton, New Brunswick, CA	Audio Engineering & Production, Animation for Game, Film & Visual Effects, Digital Filmmaking, Graphic Design & Web Development
Centre for Arts and Technology - Halifax Campus	www.digitalartschool.com	Halifax, Nova Scotia, CA	Digital Filmmaking, Animation for Game, Film & Visual Effects, Audio Engineering & Production , Event & Promotions Management, Graphic Design & Web Development
Centre for Arts and Technology - Kelowna Campus	www.digitalartschool.com	Kelowna, British Columbia, CA	Animation for Game, Film & Visual Effects , Digital Filmmaking, Graphic Design & Web Development, Audio Engineering & Production
Centre for Digital Media	www.thecdm.ca	Vancouver, BC, CA	Master of Digital Media program
CENTRE NAD - National Animation and Design Centre	www.centrenad.com	Montreal, Quebec, CA	Bachelor in 3D animation and digital design (Accredited undergraduate program)
Centro de Entrenamiento Alcance Digital	www.alcancedigital.com	Leon, Guanajuato, MX	Production, Visual Effects
Centro de Informatica - Universidade Federal de Pernambuco	www.cin.ufpe.br	Recife, Pernambuco, BR	Game Design and Development
Champlain College	www.champlain.edu and gamestudio.champlain.edu/	Burlington, VT	Game Programming, Game Design, Game Art and Animation, Management of Creative Media
Champlain College	www.gamestudio.champlain.edu	Burlington, Vermont	Game Design, Game Art and Animation, Production, Game Programming
Charles Sturt University	www.csu.edu.au	Bathurst, NSW, AU	Bachelor of Computer Science (Games Technology), Bachelor of Arts (Animation and Visual Effects)
Cloud Cursos	www.cursocloud.com/	Blumenau, SC, BR	Game Design and Development
Clover Park Technical College	http://www.cptc.edu/	Lakewood, WA.	Media Design and Production - 3D Art and Animation
Cogswell College	www.cogswell.edu/info/game.php	Sunnyvale, CA	Game Design & Development, Digital Media Management, Digital Art & Animation, Entrepreneurship & Innovation , Software Engineering, Digital Audio Technology, Digital Arts Engineering
Colegio Universitario IES21	www.ies21.edu.ar	Cordoba, AR	Desarrollo de Simulaciones Virtuales y videojuegos
Coleman University	www.coleman.edu	San Diego, CA	Game Programming Development and Design

TITLE	URL	LOCATION	PROGRAM OFFERED
College for Creative Studies	www.collegeforcreativestudies.edu	Detroit, MI	Illustration, Entertainment Arts, Illustration, Advertising Design
College of Lake County	www.clcillinois.edu/programs/cit/index.asp?gaming	Grayslake, Illinois	Computer Information Technology (Game Development)
Collins College	www.collinscollege.edu/	Phoenix, AZ	Game Art, Game Design
Cologne Game Lab	www.colognegamelab.de	Cologne, NRW, DE	Game Development & Research
Columbia College Chicago	www.colum.edu/game	Chicago, IL	Game Design (Game Art), Game Design (Game Sound), Game Design (Game Development), Game Programming (BA), Game Programming (BSc)
Concordia University Centre for Continuing Education	www.carina.concordia.ca/conted/Index.taf	Montreal, Quebec, CA	Graphics & Visualization, Game Programming Certificate
Confetti ICT	www.confetti-ict.com	Nottingham, Nottinghamshire, GB	Interactive Gaming
Conservatory of Recording Arts & Sciences	www.audiorecordingschool.com	Gilbert, AZ	Audio Engineering
Cornell University	www.gdiac.cis.cornell.edu	Ithaca, NY	Game Design Minor
Creajeux	www.creajeux.fr	Nimes, FR	Programmeur , Concepteur 3D
DADIU	www.dadiu.dk/english	1437 Copenhagen K, Denmark, DK	Game Development and Design
Dakota State University	www.dsu.edu	Madison, South Dakota	Computer Game Design
Daniel Webster College	www.dwc.edu	Nashua, NH	Video Game Programming
DAVE School (The Digital Animation & Visual Effects School)	www.daveschool.com/	Orlando, FL	Game Production, Visual Effects
De Montfort University	www.dmu.ac.uk/faculties/art_and_design/ug_courses/game_art.jsp	Leicester, Leicestershire, GB	Game Art Design
Department of Digital Technology and Game Design, Shu-Te University	www.dgd.stu.edu.tw/en/index.htm	Kaohsiung , Taiwan , TW	Multimedia Program Design, Information Science and Design, Digital Game Design
DePaul University - GameDev.DePaul.edu	www.CDM.DePaul.edu	Chicago, IL	Computer Science, Game Development - Programming, Animation, Digital Cinema, Animation - Game Art, Animation - Technical Art, Game Development - Game Design
Derby University - School of Computing and Mathematics	www.derby.ac.uk/computing/games	Derby, UK, GB	Computer Games Modelling and Animation, Computer Games Programming, Computer Graphics Production
DeSales University	www.desales.edu/	Center Valley, PA	Computer Science - Game Programming Track
design3	www.design3.com/	Santa Cruz, CA	Game Development and Design
DeVry University	www.devry.edu	Downers Grove, IL; Irving, TX; Oakbrook Terrace, IL	Game & Simulation Programming
DigiPen Institute of Technology	www.digipen.edu	Redmond, WA	B.S. in Engineering and Sound Design, B.S. in Computer Science in Real-Time Interactive Simulation, B.S. in Computer Science, B.A. in Game Design, MFA in Digital Arts, M.S. in Computer Science, B.S. in Computer Science and Game Design, B.A. in Music and Sound Design, BFA in Digital Art and Animation, B.S. in Computer Engineering

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
DigiPen Institute of Technology Europe-Bilbao	www.digipen.es	Zierbena, Bizkaia, ES	Digital Arts and Animation, Computer Science in Real-Time Interactive Simulation
DigiPen Institute of Technology Singapore	https://singapore.digipen.edu/	Singapore, , SG	Computer Science and Game Design, Game Design, Computer Science in Real-Time Interactive Simulation, Digital Arts and Animation
Digital Asia School of Animation	www.dasathai.com	Bangkok, MK, TH	CGI Animation
Digital Media Academy	www.digitalmediaacademy.org/	Campbell, CA	Pro Series Training Courses, Educator Series
Digital Media Arts College	www.dmac.edu/computer-animation-gcg	Boca Raton, Florida	Bachelor of Fine Arts - Computer Animation Game Art Concentration, Bachelor of Fine Arts - Computer Animation, Master of Fine Arts - Visual Effects Animation
Drexel University, Antoinette Westphal College of Media Arts and Design	www.drexel.edu/westphal/academics/undergraduate/digitalmedia	Philadelphia, PA	Digital Media, Graphic Design, Photography, Film & Video, Screenwriting & Playwriting
Edinboro University of Pennsylvania	www.edinboro.edu	Edinboro, Pennsylvania	Game and Virtual World Development, Computer Animation
Edison Community College	www.edisonohio.edu/index.php?page=computer-games-and-simulation-programming-and-design-option	Piqua, OH	Computer Games and Simulation Programming and Design
Edith Cowan University	www.ecu.edu.au/	Perth, Western Australia, AU	Bachelor of Science (Digital Media), Bachelor of Computer Science (Games Programming), Bachelor of Creative Industries (Game Design and Culture), Master of Games and Simulation Programming, Graduate Diploma of Games Programming
Edmonds Community College	www.edcc.edu/	Lynnwood, WA	Game Development and Design
Electronic Gaming Education	www.electronicgamingeducation.com	Online	Game Development
Emily Carr University of Art + Design	www.ecuad.ca	Vancouver, BC, CA	Bachelor of Fine Arts , Master of Applied Arts - Media Stream
ENJMIN (National School of Video Game and Interactive Media)	www.enjmin.net	Angoulême, FR	Graphics, project management, computer science, production, game design, sound design
Escuela Da Vinci	www.escueladavinci.net/	Buenos Aires, Buenos Aires, AR	Game Development and Design
euNoia Animation School	www.eunoia.tv	Mexico City, Mexico, MX	Traditional and 3d Animation
Expression College for Digital Arts	www.expression.edu	Emeryville, CA	Animation and Visual Effects, Motion Graphic Design, Sound Arts, Interaction Design, Game Art and Design, Interactive Audio, Digital Filmmaking
EzetriX-Gaming and Animation Institute	www.ezetrix.com	Pune, Maharashtra, IN	Diploma in Game Art & Design
Facultad de Ingeniería - Universidad Nacional Autónoma de México	www.ingenieria.unam.mx/	Mexico D.F.	Computer Engineering and Graphics
Faculty of Informatics/ Complutense University of Madrid	www.videojuegos-ucm.es/	Madrid, Madrid, ES	Game Development and Design
Fanshawe College	www.fanshawec.ca/programs-courses/full-time-programs/gdp1	London, Ontario, CA	Game Development - Advanced Programming
Fatec Sao Caetano do Sul	www.fatecsaocaetano.edu.br	Sao Caetano do Sul, SP, BR	Game Development and Design

TITLE	URL	LOCATION	PROGRAM OFFERED
Ferris State University	www.ferris.edu/dagd	Big Rapids, MI	Digital Animation and Game Design
Fingerlakes Community College	www.flcc.edu/academics/gameprogramming/index.cfm	Canandaigua, NY	Game Programming and Design
Florida Interactive Entertainment Academy at UCF	www.fiea.ucf.edu	Orlando, Florida	Art Track, Programming Track, Production Track
Foothill College	www.foothill.edu	Los Altos Hills, CA	Music Technology
Friends of Design - Academy of Digital Arts	www.friendsofdesign.net	Cape Town, Western Province, ZA	Game Technology and Multimedia Entertainment
Full Sail University	www.fullsail.edu/gamecareerguide	Winter Park, FL	Game Art, Game Development, Game Design
FuturePoly	www.futurepoly.com	Bellevue, WA	3D Modeling for Games, Digital Painting, Character Modeling with ZBrush, Character Animation for Games, Texturing for Games, Concept Art
FZD School of Design	www.fzdschool.com/index.html	Singapore, Singapore, SG	Game Design
Game Academy	www.gamea.com.cn	Shanghai, , CN	Game Design, NextGen Game Art
Game Character Academy	www.gc-academy.net	Online	Character Rigging/Technical Art, Character Art
Game Institute, Inc.	www.gameinstitute.com/	New York, NY	Game engine, 3D graphics development
Games Academy	www.games-academy.com	Berlin, Germany, DE	Game Production, Game Design, Film Art & Animation, Game Art & Animation, Game Programming, Interactive Audio Design
Gemini School of Visual Arts & Communication	www.gemini-school.com	Cedar Park, TX	Diploma in Visual Arts & Communication
Generando IT	www.generandoit.com/	Buenos Aires, Buenos Aires, AR	Game Development and Design
George Brown College	www.georgebrown.ca/Marketing/FTCal/design/G405.aspx	Toronto, Ontario, CA	Game Design
George Mason University, ACS	www.cs.gmu.edu/~acsgame/	Fairfax, VA	ACS Game Design
George Mason University, BFA	www.game.gmu.edu	Fairfax, VA	Computer Game Design
Georgia Institute of Technology	www.gatech.edu	Atlanta, GA	Graphics & Visualization, Artificial Intelligence
Glamorgan Centre for Art & Design Technology	www.glamfest.net	Pontypridd, RCT, GB	Computer Animation
Glasgow Caledonian University	www.gcu.ac.uk/	Glasgow, GB	BA Computer Games (Art & Animation), BSc Audio Technology with Multimedia, BSc Computer Games (Design), BA 3D Computer Animation, BSc Computer Games (Software Development)
Glasgow Caledonian University, School of Engineering and Computing	www.gcu.ac.uk/sec/study/	Glasgow, Scotland, GB	Computer Games (Design), Computer Games (Software Development)
Glendale Community college	www.gc.maricopa.edu/	Glendale, AZ	AAS Multimedia Animation
Glyndwr University	www.glyndwr.ac.uk	Wrexham, Wrexham Borough, GB	Computer Game Development, Digital Art for Computer Games
Gnomon School of Visual Effects	www.gnomonschool.com	Hollywood, CA	Maya Fast Track Program, Highend Computer Graphics Certificate Program
Goldsmiths, University of London	www.gamesgoldsmiths.com	London, England, GB	MSc Computer Games & Entertainment
Great Eastern Technology	www.get3dtraining.com	Salem, NH	Game Development and Design
Great Northern Way Campus	www.mdm.gnwc.ca	Vancouver, British Columbia (Canada), CA	Masters of Digital Media Program

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
Guildhall at SMU	www.smu.edu/guildhall	Plano, TX	Level Design, Production, Art Creation, Programming
Hagerstown Community College	www.hagerstowncc.edu/	Hagerstown, MD	Simulation and Digital Entertainment
HAL Institute of Computer Technology - Nagoya	www.hal.ac.jp/english.html	Nagoya, Aichi-ku, JP	Multimedia, Music and General information processing
HAL Institute of Computer Technology - Osaka	www.hal.ac.jp/english.html	Osaka, , JP	Multimedia, Music and General information processing
Hamburg University of Applied Sciences (HAW)	www.gamesmaster-hamburg.de/en/	Hamburg, Hamburg, DE	Time-based Media / Sound - Vision - Games
Hamidrasha Art College, Beit Berl Academic College, Kalmaniya	www.beitberl.ac.il	Beit Berl, Near Kfar Sava, IL	Software Development , Animator, Copywriting , Graphic Designer, Content Design
Harper College CE	goforward.harpercollege.edu/page.cfm?p=2469	Schaumburg, IL	Flash Game Designer CE Certificate
Harvard Extension School	www.extension.harvard.edu/	Cambridge, MA	Understanding and Developing Multimedia, Introduction to Computer Graphics, Digital Multimedia Art
Herzing University - Madison WI	https://www.herzing.edu/	Madison; Milwaukee, WI	Graphic Design, Game Development
Hidden Edge 3D	www.hiddenedge3d.com	Tacoma, WA.	3DS MAX 2009 Training
High Point University	www.highpoint.edu/	High Point, NC	Game and Interactive Media Design
Hochschule Darmstadt University of Applied Sciences	www.h-da.de	Darmstadt, Hessen, DE	Media, Media Direction, Digital Media
Hochschule für Technik und Wirtschaft Berlin	www.gamedesign.htw-berlin.de/	Berlin, Berlin, DE	Interaction Design / Game Design
Hochschule Mittweida (FH) University Of Applied Sciences	www.hs-mittweida.de/	Mittweida, Sachsen, DE	Media Informatics and Interactive Entertainment
Hollywood CG School of Digital Arts at Shepherd University	www.shepherduniversity.edu/digitalarts/	Los Angeles, CA	Game Art & Design , VFX & Animation
Homer College	www.omiros.gr	Athens, Attiki, GR	MultiMedia Computing
Hong Kong Polytechnic University - Multimedia Innovation Centre (MIC)	www.mic.polyu.edu.hk	Hong Kong, Kowloon, HK	Multimedia & Entertainment Technology, Traditional & Interactive Media, Animation & Games
Howest - University College West-Flanders	www.digitalartsandentertainment.com	Kortrijk, West-Flanders, BE	Digital Arts & Entertainment
Humber College - School of Media Studies	www.humber.ca	Toronto, ON, CA	Game Development and Design
Id Animation & Arts	www.idanimations.com	Pune, Maharashtra, IN	Specialization in Game Art, Modelling, Texturing, 2D and 3D Animation
iD Game Design & Development Academy for Teens	www.idtech.com/teens/game-design-development/	Held at Stanford, Harvard, Emory, the University of Washington, Vassar, the University of Denver, Villanova, TCU, and Lake Forest College	Game Development for Android with Unity and Javascript, Minecraft 3D level Design, Unreal Engine, Maya
Illinois Institute of Art - Chicago	www.ilic.aii.edu	Chicago, IL	Game Art & Design, Digital Media Production, Visual Effects & Motion Graphics, Media Arts & Animation
Image College of Arts, Animation & Technology (ICAT)	www.icat.ac.in	Chennai, TN, IN	3D Animation, Game Development, Game Design, Visual Effects

TITLE	URL	LOCATION	PROGRAM OFFERED
Indiana University	www.games.indiana.edu/	Bloomington, IN	Game Design, Game Design - Programming, Game Design - Management, Game Design - Audio, Game Design - Art
Informatics Professional Development Centre	www.informaticsgroup.com.sg/ipdc/	Singapore, SG	Game Development and Design
Institute Desgraff	www.institutdesgraff.com	Sherbrooke, Quebec, CA	3D digital arts and animation for video games
Institute of Technology Carlow	www.itcarlow.ie	Carlow, Carlow, IE	Computer Games Development
Institute of Virtual Reality	www.institutevr.com	Peora, Uttaranchal, IN	Game Developer Program , 3D Animation & Vfx, Artificial Digital Life, Game Design and Art, Virtual Reality Art
Instituto Tecnológico y de Estudios Superiores de Monterrey - Campus Estado de Mexico	www.cem.itesm.mx	Atizapan de Zaragoza	Computer Graphics Animation
Inter-Dec college	www.interdeccollege.com	Montreal, Quebec, CA	3D animation, Video games
iPlay.design	www.iplayd.com	Jogjakarta, DIY, ID	Professional Extension Game Development
ISART Digital	www.isartdigital.com/	Paris, Île-de-France, FR	Game Design, Game Programming , CG Animation, Production, Visual Effects, Game Art, Game Design and Programming , Sound Design
Istanbul Yildiz Technical University - Communications Design And Multimedia	www.ilet.yildiz.edu.tr/english/index.html	Istanbul, Marmara, TR	Digital-interactive media
Istituto Europeo di Design	www.ied.it	Rome, RM, IT	Videogame Design and Development
IT University of Copenhagen	www.itu.dk/game	Copenhagen, , DK	Center for Computer Games Research, Media Technology and Games
IT University of Copenhagen	www.itu.dk	Copenhagen, , DK	Media, Technology and Games
ITP at New York University	www.itp.nyu.edu	New York, NY	Interaction Design
ITT Tech - Green Bay Campus	www.newwavegamedev.org	Green Bay, WI	Digital Entertainment and Game Design
ITT Tech - Greenville, South Carolina	www.itt-tech.edu/	Greenville, , SC	Digital Entertainment and Game Design Bachelor of Science Degree, Visual Communications Associate of Applied Science Degree
ITT Technical Institute - Arnold, Missouri	www.itt-tech.edu/campus/school.cfm?loc_num=91	Arnold, MO	Digital Entertainment and Game Design
Kajaani University of Applied Sciences	www.kajak.fi/?depid=13937	Kajaani, , FI	Game Programming
Karelia University of Applied Sciences	www.tiko.ncp.fi/gameprogramming/index_en.html	Joensuu, Finland, FI	Game Programmer
Keiser University	www.keiseruniversity.edu	Fort Lauderdale, FL	Video Game Design
Laguna College of Art + Design	www.lcad.edu/site/art-of-game-design-mfa/	Laguna Beach, CA	Art of Game Design
Lake Washington Technical College	www.lwtc.edu	Kirkland, WA	Print/Web, Digital Games and Media, Video Production
Lancaster University UK	www.dcs.lancs.ac.uk/admissions/postgraduate_courses.php?course_id=008623	Lancaster, Lancashire, GB	Mobile games development
Langara College Continuing Studies	www.langara.bc.ca/cs	Vancouver, BC, CA	Business & Computer Technology

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
Living Arts College	www.living-arts-college.edu	Raleigh, NC	Animation & Game Design, Digital Photography, Digital Audio Production & Design, Digital Filmmaking, Interactive Media Arts
Long Island University	www.liu.edu/post/gamedesign	Brookville, NY	Digital Game Design and Development
Los Angeles Film School - Game Production	www.lafilm.edu	Hollywood, California	Game Production
Lost Boys Learning	www.lostboys-learning.com	Courtenay, BC, CA	Visual Effects
Louisiana State University Digital Media Arts & Engineering	www.dmae.lsu.edu	Baton Rouge, LA	Digital Media Minor (DMART) - Undergraduate, Digital Media Minor (DMTEC) - Undergraduate
LSC - Kingwood	www.kwctechology.com	Kingwood, TX	Game Development and Design
Mad Academy	www.madacademy.com.au	Adelaide, SA, AU	3D Character Animators and Artists this program
Madeira Interactive Technologies Institute/University of Madeira	www.m-iti.org	Funchal, R. A. Madeira, PT	Masters of Entertainment Technology, partnership with ETC/Carnegie Mellon University - USA
Madison Media Institute College of Media Arts	www.mediainstitute.edu	Madison, WI	Game Art & Animation, Recording & Music Technology, Video And Motion Graphics
Massachusetts Institute of Technology (MIT)	www.web.mit.edu/admissions/	Cambridge, MA	Electrical Engineering and Computer Science, Sloan School of Management, Comparative Media Studies
Massasoit Community College	massasoit.edu/	Brockton, MA	Computer Information Systems - Programming Option
Max the Mutt Animation School	www.maxthemutt.com	Toronto, Ontario, CA	Concept Art, Classical & Computer Animation and Production, Illustration for Sequential Arts: Comic Books and Graphic Novels
McGill University, School of Computer Science	www.cs.mcgill.ca/	Montreal, QC, CA	Major in Computer Science Computer Games Option
MD.H Mediadesign-Hochschule	www.mediadesign.de/	Berlin, , DE	Game Design
Media Arts & Game Development	www.uww.edu/games	Whitewater, Wisconsin	Game Development
Media Design School	www.mediadesignschool.com	Auckland, , NZ	Bachelor of Art and Design (3D & VFX), Bachelor of Media Design, Bachelor of Software Engineering (Game Programming), Graduate Diploma in Creative Technologies, Bachelor of Creative Technologies (Game Art)
Mesa Community College	www.mc.maricopa.edu/~lybbert/studio180/degrees.html	Mesa, AZ	Associate of Applied Science in Game Technology
Michigan State University	www.seriousgames.msu.edu	East Lansing, Michigan	Serious Game Design MA, Game Design & Development Specialization
Middlesex University	www.md.ac.uk	London, , GB	Computing, Graphics and Games, 3D Animation and Games, Multimedia Computing, Animation
Mildred Elley - School of Digital Media Arts	www.mildred-elley.edu	Albany, NY	Digital Graphics & Multimedia Design, Ditial Graphics & Multimedia Design, Game Design & 3D Animation
Milwaukee Area Technical College	www.matc.edu	Milwaukee, WI	Animation
Minneapolis Media Institute	www.minneapolismediainstitute.com/	Edina, MN	Game Art & Animation
Missouri State University - West Plains	wp.missouristate.edu/Academics/default.htm	West Plains, MO	Computer Graphics & Programming

TITLE	URL	LOCATION	PROGRAM OFFERED
Mohawk Valley Community College	www.mvcc.edu/academics/departments/art/dgtlmtn.cfm	Utica, NY	Game Development and Design
Monash University - Berwick School of Information Technology	www.infotech.monash.edu.au/berwick	Berwick, Victoria, AU	Bachelor of Information Technology and Systems - Multimedia Applications, Bachelor of Information Technology and Systems - Games Development
Montgomery College	www.studygaming.com	Rockville, MD	Computer Gaming and Simulation - Production & Design Track, Computer Gaming and Simulation - Art & Animation Track, Computer Gaming and Simulation - Programming Track, Internet Gaming and Simulation - Certificate
Montgomery County Community College	www.mc3.edu	Blue Bell, Pennsylvania	Digital Design: Computer Graphics, Electronic Game and Simulation Design, Digital Design: Multimedia, Digital Audio Production, Software Engineering
Motherwell College	www.motherwell.ac.uk/	Motherwell, North Lanarkshire, GB	National Certificate/Diploma
Mount Ida College	www.mountida.edu	Newton, MA	BS in Game Art and Animation
Mt. Sierra College	www.mtsierra.edu	Monrovia, CA	Game Arts & Design, Media Arts & Design
Murdoch University	www.games.murdoch.edu.au/	Perth, Western Australia, AU	Games Technology, Games Software Design and Production
Musitechnic	musitechnic.com/en/	Montreal, Quebec, CA	Audio production technique
Napier University	www.i-media.soc.napier.ac.uk/bsc_imd/index.html	Edinburgh, , GB	Interactive Media Design, Interactive Media Design
National Centre for Computer Animation, Bournemouth University	www.ncca.bournemouth.ac.uk	Poole, Dorset, GB	Computer Visualization & Animation, BA (Hons), Computer Animation & Visual Effects, MSc, 3D Computer Animation, MA/PD, Computer Animation Arts, BA (Hons), Digital Effects, MA/PD, Software Development for Animation, Games and Effects, BSc (Hons)
National Film and Television School	www.nfts.co.uk/courses/games-design-and-development	Beaconsfield, Buckinghamshire, GB	MA Games Design and Development
National University - School of Media & Communication	www.nu.edu/Academics/Schools/SOMC.html	La Jolla, CA	Video Game Production and Design, Digital Entertainment and Interactive Arts
NBCC Miramichi	www.nbcc.ca/miramichi	Miramichi, NB, CA	Applied Arts - Electronic Game - Design (programming), Applied Arts Diploma of Advanced Studies, Applied Arts - Animation and Graphics, Applied Arts - Electronic Game - 3D Graphics, Applied Arts - Media Studies (Art Fundamentals)
Nescot - National Diploma Game Development	www.nescot.ac.uk/	Epsom, Surrey, GB	National Diploma Game Development
Neumont University	www.neumont.edu	South Jordan, UT	Computer Science
New England Institute of Technology	www.neit.edu	East Greenwich, RI	Digital Recording Arts, Graphics, Multimedia and Web Design, Software Engineering Technology, Game Development and Simulation Programming Technology, Video and Audio Production Technology, Video Game Design
New Jersey Institute of Technology	www.njit.edu	Newark, New Jersey	Information Technology
New Media Campus	www.newmediacampus.com	Kelowna, B.C., CA	Multimedia Professional Studies, 3D Animation & Game Design

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
New York Film Academy	www.nyfa.edu/game-design-school/	New York, NY	Game Design
New York University	www.gamecenter.nyu.edu/	Brooklyn, New York	Game Design B.F.A, Game Design M.FA
New York University - School of Continuing and Professional Studies	www.scps.nyu.edu/dcom	New York, NY	Digital Communications and Media
NHTI Concord Community College	www.nhti.edu/academics/academicprograms/degaggo.html	Concord, New Hampshire	Animation and Graphic Game Programming
NHTV Breda University of Applied Science	www.nhtv.nl/made	Breda, , NL	International Game Architecture and Design
North Carolina State University - College of Design	www.ncsdesign.org	Raleigh, NC	3D Animation, Interactive Design, Ideation & Illustration, Game Design, Multimedia & Digital Imaging
Northeastern University - College of Professional Studies	www.northeastern.edu/cps/digitalmedia	Boston, MA	Interactive Design, Digital Video, Game Design, 3D Animation
Northeastern University - Creative Industries [Game Design & Interactive Media]	www.northeastern.edu/ci/	Boston, MA	Game Design and Computer Science, Interactive Media and Graphic Design, Interactive Media and Computer Science, Game Design and Graphic Design, Interactive Media and Digital Art, Interactive Media and Music Technologies, Creative Industries Minor, Game Design and Digital Art
Northern Oklahoma College	www.north-ok.edu/ics	Tonkawa, OK	3D Animation and Post-production
Northumbria University	www.northumbria.ac.uk/sd/academic/ceis/	Newcastle upon Tyne, Tyne & Wear, GB	Games Programming, Computer Games Design & Production
Norwegian School of Information Technology	www.nith.no	Oslo, , NO	Game design, Game programming
Norwich University College of the Arts	www.nuca.ac.uk	Norwich, Norfolk, GB	Games Art and Design
Nova Scotia Community College	www.nsc.ca	Truro, NS, CA	Interactive & Motion Graphics - Visual Effects, Interactive & Motion Graphics - 3D Modelling & Motion Capture, Interactive & Motion Graphics - Game Design
NTI Birmingham	www.bcu.ac.uk/pme/nti/gamercamp/courses/pro	Birmingham, , GB	Gamer Camp: Pro, Video Games Development - MA / MSc
Ohio University	www.tcomschool.ohiou.edu/	Athens, OH	Digital Media: Special Effects, Games, and Animation
Oklahoma Christian University	www.oc.edu/art	Edmond, OK	Oklahoma Christian U. Gaming + Animation
Oklahoma City Community College	www.occ.edu	Oklahoma City, Oklahoma	Computer Aided Technology- Game Design Option
Oklahoma Panhandle State University	www.opsu.edu	Goodwell, OK	BTEC Game Art Design, BFA Computer Graphics
Örebro University	www.oru.se	Örebro, SE	Simulation and Game Technology
Otis College of Art and Design - Digital Media Department	www.otis.edu	Los Angeles, CA	Game design, animation, interactive design, motion graphics, visual effects
Parsons the New School of Design: School of Art, Media and Technology	www.cdt.parsons.edu	New York, NY	Design & Technology
Pennsylvania College of Technology	www.pct.edu	Williamsport, Pennsylvania	Gaming & Simulation
Pennsylvania State University	www.ist.psu.edu	University Park, PA	Game Development
Pensacola State College	www.pjc.edu	Pensacola, FL	Simulation and Game Design

TITLE	URL	LOCATION	PROGRAM OFFERED
Piedmont Community College	www.piedmontcc.edu	Roxboro/Yanceyville, NC	Digital Effects and Animation Technology
Pinnacle College	www.pinnaclecollege.edu	Alhambra, CA	Audio for Games and Interactive Media
Platt College - Digital Media Design	www.platt.edu	San Diego, CA	Media Arts (Video Production), Media Arts (Visual Effects & Compositing), Multimedia/Animation, Media Arts (3D Animation), Graphic Design, Web Design, Media Arts (Web Design), Media Arts
PlaygroundSquad	www.playgroundsquad.com	Falun, SE	Programming, Game Design, 3D Graphics
PowerUp Games	www.powerupgames.com/game-tester-course.html?cid=78pid=5	Jacksonville, FL	Game Testing Certification
Pyramid: the Institute for Advanced Digital Audio Training	www.Pyramid.com	San Francisco, CA	Sound Design
QANTM College - Australia	www.qantmcollege.edu.au	Brisbane, QLD, AU	Bachelor of Interactive Entertainment (with a Major in Games Programming), Bachelor of Interactive Entertainment (with a Major in Animation), Bachelor of Interactive Entertainment (with a Major in Games Design), Diploma of Screen and Media (Animation)
QANTM College - Munich	www.qantm.de	Munich, 81737 DE, Bavaria, DE	Bachelor of Arts, Interactive Animation, Bachelor of Science, Games Programming
QANTM College London	www.qantm.co.uk	London, England	Game Development
Queen's University Belfast - School of EECS	www.qub.ac.uk/eeecs	Belfast, Antrim, GB	Computer Games Design and Development
Queensland University of Technology	www.creativeindustries.qut.edu.au	Kelvin Grove, QLD, AU	Bachelor of Games and Interactive Entertainment, Master of Information Technology - Games Design/Production
Queensland University of Technology - Brisbane City	www.qut.edu.au/scitech	Brisbane, QLD, AU	Bachelor of Games and Interactive Entertainment, Master of Information Technology - Games Design/Production
Quinnipiac University	www.quinnipiac.edu/	Hamden, CT	Game Development and Design
Rasmussen College	www.rasmussen.edu/degrees/technology-design/game-design/	Currently Offered Online and at 25 campuses in 5 states., Multiple States	Information Systems Management, Multimedia Technologies, Game & Simulation Programming, Digital Design and Animation
Ravensbourne College Of Design & Communication	www.rave.ac.uk	Chislehurst, kent, GB	Interactive Digital Media, Computer Visualisation and Animation, Animation
Recording Arts Canada	www.recordingarts.com/programs/game-design	Toronto, Ontario, CA	Game Design
Rensselaer Polytechnic Institute	www.gsas.rpi.edu	Troy, NY	Game and Simulation Arts & Sciences
Ringling College of Art and Design	www.ringling.edu	Sarasota, Florida	Computer Animation, Game Art & Design
Rochester Institute of Technology	www.rit.edu	Rochester, NY	Computer Science, Visual Communications Design, Game Design & Development, New Media Interactive Development, Information Technology, Software Engineering, 3-D Digital Design, Film and Animation, New Media Design
S4G School for Games	www.school4games.net	Berlin, Berlin, DE	Online Game Graphics, Online Game Development, Online Game Engineering
Sacred Heart University	www.sacredheart.edu/pages/35_computer_science_information_technology.cfm	Fairfield, CT	Game Design and Development

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
SAE Institute (Singapore Campus)	www.sae.edu.sg	Singapore, Singapore, SG	Diploma of IT, Bachelor of Science (Hons) in Games Programming
SAE Institute Amsterdam	www.amsterdam.sae.edu	Amsterdam, Noord-Holland, NL	Audio Engineering Course
SAGE	www.sageinfolabs.com	New Delhi, New Delhi, IN	Game Programming
Sam Houston State University	www.shsu.edu/~animate	Huntsville, TX	Computer Animation
San Jacinto College Central	www.sjcd.cc.tx.us	Houston, TX	3D Animation courses (I,II,and III)
Santa Ana College	www.ext.sac.edu/academic_progs/art/3D/	Santa Ana College, ca	3D Animation Certificate
Santa Monica College - Academy of Entertainment Technology	www.academy.smc.edu	Santa Monica, CA	Animation, Game Development, Post Production, Visual Effects
SCAD	www.scad.edu	Savannah, GA	Animation, Interactive Game Design & Motion Graphics, Visual Effects, Sound Design
School of Animation, Communication University of China	www.animation.cuc.edu.cn/	Beijing, , CN	Game Programming direction of Game Design, Animation Writing and Directing direction of Animation, Digital Media Art, Digital Animation direction of Animation, Game Design, Media Art Online direction of Digital Media Art, Animation, Game Art direction of Game Design, Animation Design direction of Animation, Digital Film and Television Production direction of Digital Media Art
School of Arts and Humanities, University Campus Suffolk	www.ucs.ac.uk	Ipswich, Suffolk, GB	Computer Games Design
School of Communication-American University	www.american.edu/soc/	Washington, DC	M.A. in Game Design
School of Computer Science, University of Windsor	www.cs.uwindsor.ca	Windsor, Ontario, CA	Game Development Specialization
School of Computing & Mathematical Sciences, Liverpool John Moores University	www.cms.livjm.ac.uk/	Liverpool, , GB	Computer Games Technology, Interactive Media Design, Animation for Film and Games
School of Computing and Intelligent Systems, Faculty of Computing and Engineering, University of Ulster	www.ulster.ac.uk	Derry, County Derry, GB	BSc (Hons) Multimedia Computer Games, BEng (Hons) Computer Games Development
School of Multimedia Systems	www.multimedia.monash.edu.au	Berwick, Victoria, AU	Games Development
School of Video Game Audio	www.School.VideoGameAudio.com	Vancouver, , CA	Wwise Demo Reel, FMOD Demo Reel
School of Visual Arts	www.sva.edu	New York, NY	BFA Computer Art, Computer Animation and VFX, MFA Computer Art, BFA Animation
Seamedu - Media School	www.seamedu.com	Pune, Maharashtra, , IN	BTEC Level 5 HND Creative Media Production (Computer Game Design), Diploma in Game Development
Seattle Central Community College	www.learnatcentral.org	Seattle, WA	3D Animation / Design & Gaming Program
Seattle Recording Arts	www.seattlerecordingarts.com/	Seattle, WA	Game Audio
Seneca College - Game Art & Animation	www.senecagaming.ca	Toronto, Ontario, CA	3D Animation Graduate Certificate Program (dan), 3D Gaming Graduate Certificate Program (GAM), Animation Advanced Diploma Program
Serious Game Design Institute	www.sgd.i.sbccc.edu	Santa Barbara, CA	Serious Game and Simulation Design
Sessions College for Professional Design	www.sessions.edu/certificate-programs/game-art	Tempe, AZ	Accredited Game Art Certificate

TITLE	URL	LOCATION	PROGRAM OFFERED
Shawnee State University	www.shawnee.edu	Portsmouth, OH	Game Programming, Game Graphics
Sheffield Hallam University	www.shu.ac.uk/multimedia/games/	Sheffield, South Yorkshire, GB	BSc Game Software Development, MSc Game Software Development, MA Animation for Computer Games
Sheridan College	www.gaming.sheridancollege.ca	Oakville, Ontario, CA	Bachelor of Game Design, Game Level Design, Bachelor of Interaction Design, Game Development - Advanced Programming
Sierra College	www.cs.sierracollege.edu	Rocklin, CA	Computer Science
Simon Fraser University - School of Interactive Arts and Technology (SIAT)	www.siat.sfu.ca	Surrey, British Columbia, CA	Graduate Program, Undergraduate Program: Media Arts, Design & Informatics Concentrations
Southbank Institute of Technology	www.southbank.edu.au	Brisbane, Queensland, AU	Diploma of Interactive Digital Media - Game Design
Southern Adventist University - School of Visual Art and Design	www.art.southern.edu/	Collegedale, TN	Animation, Fine Art, Interactive Media, Film Production, Graphic Design
Southern New Hampshire University - SNHU	www.snhu.edu	Manchester, NH	Information Technologies, Information Technology - Game Design & Dev, Game Programming & Development
Southern Polytechnic State University	www.games.spsu.edu/	Marietta, GA	Computer Science, Information Technology, Software Engineering, Computer Game Design and Development
Springfield College	www.spfldcol.edu/	Springfield, MA	Game Design
Sprott-Shaw Community College	www.sprottshaw.com	Calgary, AB, CA	Game Development
St. Edward's University	www.stedwards.edu/business/graduate/mbad/index.htm	Austin, TX	Digital Media Management
St. Petersburg College-Seminole Campus	www.spcollege.edu/se/DigitalMedia/index.htm	Seminole, FL	Digital Media Production, Video Game Foundations, Digital Media Video production
Staffordshire University	www.staffs.ac.uk	Stafford, GB	Computer Games Programming, Multiplayer Online Games Programming, Computing: Games Development
Stony Brook University	www.cs.stonybrook.edu	Stony Brook, NY	Computer Science Specialization in Game Programming
Stroud College in Gloucestershire	www.stroud.ac.uk/	Stroud, Gloucestershire, GB	Multimedia - BTEC National Diploma
Swansea Metropolitan University - School of Digital Media	www.she.ac.uk/	Swansea, Swansea, GB	Digital Media
TAFE - Tea Tree Gully	www.tafesa.edu.au/game-art-studies.aspx	Modbury, SA, AU	Game Art Program, Game Art VET
TAFE N.S.W. Hornsby	www.infotech.hornsby.tafensw.edu.au/	Hornsby, New South Wales, AU	Programming, Game Development, Digital Media
TAFE NSW - Wollongong West	www.illawarra.tafensw.edu.au	Wollongong, NSW, AU	Game Development
Tallahassee Community College	www.tcc.fl.edu/about_tcc/academic_affairs/division_of_technology_and_professional_programs/ar	Tallahassee, Florida	Computer Game Design
Temasek Polytechnic (Temasek InfoTech School)	www.it.tp.edu.sg/	Singapore, Singapore, SG	Game & Entertainment Technologies
Texas State Technical College	www.waco.tstc.edu	Waco, Texas	Game Programming and Design, Game & Simulation

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
The Academy of Entertainment and Technology at Santa Monica College	www.academy.smc.edu	Santa Monica, CA	Game Design, Animation
The Art Institute of Vancouver	www.artinstitutes.edu/vancouver/	Vancouver, British Columbia, CA	Game Art & Design, Visual & Game Programming
The Art Institutes System of Schools	https://www.artinstitutes.edu/areasofstudy/game-design-and-programming/detail/50	Pittsburgh, PA	Animation Art & Design, Visual & Game Programming, Game Art & Design, VFX & Motion Graphics, VFX for Film & Television, Computer Animation, Programming, 3D Modeling for Animation & Games
The College of Westchester	www.cw.edu	White Plains, New York	Multimedia Development & Management
The Game Assembly - Game Art	www.thegameassembly.com	Malmö, SE	Game Art, Programming, Design
The Rydan Workshop	www.therydanworkshop.com	North Vancouver, BC, CA	Introduction to ZBrush, Intro Digital Concept Art, 2D Matte Painting, Digital Character and Creature Design, Intro to Maya Rigging, Game Character Pipeline, Real-Time Environments with Unreal Engine, Photoshop Intro for the Entertainment Industry
Toronto Film School	www.torontofilmschool.ca	Toronto, Ontario, CA	Video Game Design & Development, Video Game Design & Animation
Train2Game	www.train2game.com	Nationwide, United Kingdom, GB	TIGA Diploma in Games Design, TIGA Diploma in Games Development
Training Center Alcance Digital	www.alcancedigital.com	León, Guanajuato, MX	Digital Musical Composition, 3D Animation, Digital Design, Visual Effects, Digital Cinematography, Video Games Programming
Tribeca Flashpoint Media Arts Academy	www.tfa.edu	Chicago, IL	Recording Arts, Game & Interactive Media, Film + Broadcast, Design + Visual Communications, Animation + Visual Effects
Trinity College Dublin	www.cs.tcd.ie/courses/msciet/5	Dublin 2, IE	MSc in Interactive Entertainment Technology
triOS College	www.getintothegame.ca	Available at these campus locations: London, Kitchener, Hamilton, Mississauga and Toronto, Ontario, CA	Video Game Design and Development + Internship
Tulsa Tech	www.tulsatech.com/Programs/it.aspx	Tulsa, Oklahoma	Information Technology-3d Design & Animation
Tyler Junior College	www.tjc.edu	Tyler, TX	Gaming and Simulation Development--Programming, Gaming and Simulation Development--Graphics
Universidad Iberoamericana Leon	www.leon.uia.mx	León, Guanajuato, MX	Digital Design
Universidade Anhembi Morumbi	www.anhembi.br	Sao Paulo, BR	Game Design
Universidade do Vale do Rio dos Sinos	www.unisinos.br/jogos	Sao Leopoldo, Rio Grande do Sul, BR	Digital Games
University Campus Oldham	www.uco.oldham.ac.uk/	Oldham, Greater Manchester, GB	Digital Arts Practice (Games Art)
University for the Creative Arts at Farnham	www.ucreative.ac.uk/index.cfm?articleid=22622	Farnham, Surrey, GB	BA Computer Games Arts

TITLE	URL	LOCATION	PROGRAM OFFERED
University of Abertay Dundee	www.abertay.ac.uk	Dundee, Scotland, GB	Computer Games Technology, Games Development, Game Design & Production Management, Computer Game Applications Development, Computer Arts, Game Art & Animation, Creative Sound Production
University of Advancing Technology	www.uat.edu	Tempe, AZ	Serious Game and Simulation , Game Design, Game Programming, Game Art and Animation, Game Production and Management
University of Applied Arts and Design	www.interaction.hgkz.ch	Zurich, ZH, CH	Game Design
University of applied sciences Salzburg / FH Salzburg	www.fh-salzburg.ac.at	Puch bei Hallein , Salzburg, AT	Master MultimediaTechnology, Master MultiMediaArt / Producing, Master MultiMediaArt / visual content creation, Master MultiMediaArt / audio - content creation
University of Applied Sciences Technikum Wien	www.technikum-wien.at/en/	Vienna, Wien, AT	Computer Science, Game Engineering and Simulation Technology
University of Baltimore	www.ubalt.edu/games	Baltimore, MD	Simulation and Digital Entertainment
University of Birmingham	www.cs.bham.ac.uk	Birmingham, UK, GB	Game Design and Development
University of Bradford - School of Informatics	www.inf.brad.ac.uk/	Bradford, West Yorkshire, GB	Design for Computer Games, Interactive Systems and Video Games Design, Artificial Intelligence for Games
University of British Columbia Continuing Studies	https://cstudies.ubc.ca/programs/game-writing-academy	Vancouver, BC, CA	Game Writing Academy
University of California , Santa Cruz	www.games.soe.ucsc.edu	Santa Cruz, CA	Digital Arts & New Media / Playable Media, Computer Game Design, Computer Game Design Ph.D.
University of California San Diego Extension Digital Arts Center	www.dac.ucsd.edu	San Diego, CA	Video & Editing, Graphic & Web Design, Mobile Applications Development
University of California, Irvine	www.ics.uci.edu	Irvine, CA	Digital Arts, Computer Science, Computer Game Science
University of Central Florida (FIEA)	www.fiea.ucf.edu	Orlando, FL	Production Track, Art Track, Programming Track
University of Central Lancashire	www.uclan.ac.uk	Preston, Lancashire, GB	Games Design, Computer Games Development
University of Colorado, Colorado Springs	www.cs.uccs.edu/~chamillard/GameOptions/GameOptions.htm	Colorado Springs, CO	Bachelor of Innovation in Game Design and Development, Game Design and Development Minor
University of Denver	www.gamedev.cs.du.edu	Denver, CO	Animation and Game Development, Computer Science, Electronic Media Arts Design, Studio Art, Digital Media Studies
University of East London	www.uel.ac.uk/ssmcs/programmes/undergraduate/computer-gamesdesignstory.htm	London , GB	Computer Games Design (Story Development)
University of Glamorgan	www.glam.ac.uk/course-details/685/53	Pontypridd, RCT, GB	Computer Game Development
University of Gotland	www.mainweb.hgo.se/index.nsf/index?OpenForm , www.hgo.se/game/index4thyear.htm	Visby, Gotland, SE	International Game Production Studies II, International Game Production Studies I, International Game Production Studies Degree
University of Houston	www.games.cs.uh.edu	Houston, TX	Game Development
University of Houston-Victoria	www.uhv.edu/asa	Victoria, Texas	BS Computer Science-Digital Gaming and Simulation, BAAS Digital Gaming Simulation

Game School Directory

TITLE	URL	LOCATION	PROGRAM OFFERED
University of Hull - Department of Computer Science	www.net.dcs.hull.ac.uk/	Hull, E Yorkshire, GB	Games Programming
University of Lincoln	www.lincoln.ac.uk/dci/	Lincoln, UK, GB	Games Computing, Advanced Games Programming
University of Maryland, Baltimore County	www.gaim.umbc.edu/	Baltimore, Maryland	Visual Arts + animation and interactive media, Computer Science + game development track
University of Miami - Music Engineering	www.music.miami.edu/programs/mue/	Coral Gables, FL	Music Engineering
University of Michigan - EECS Department	www.eecs.umich.edu	Ann Arbor, MI	Computer Engineering, Computer Science
University of Montevallo	www.montevallo.edu/games	Montevallo, AL	Game Studies and Design
University of Montreal	www.dej.umontreal.ca	Montreal, Quebec, CA	Graduate Diploma in Game Design
University of Ontario Institute of Technology	www.uoit.ca/	Oshawa, Ontario, CA	Game Development and Entrepreneurship
University of Otago	www.otago.ac.nz	Dunedin, Otago, NZ	Computer Game Design, Artificial Intelligence, Computer Graphics, Software Engineering
University of Pennsylvania	www.cis.upenn.edu/grad/cggt/	Philadelphia, PA	Computer Graphics & Game Technology, Digital Media and Design, Human Modeling and Simulation
University of Portsmouth - School of Creative Technologies	www.port.ac.uk/games	Portsmouth, Hampshire, GB	Computer Games Technology
University of Prince Edward Island	www.upei.ca/csit/	Charlottetown, PE, PEI, CA	BSc CS with Specialization in Video Game Programming
University of Skövde	www.his.se	Skövde, SE	Design, Sound/Music, Media, Aesthetics and Narrative, Graphics, Programming, Serious Games, Game Writing
University of Southern California - Interactive Media Division	www.cinema.usc.edu	Los Angeles, California	Interactive Media, Interactive Entertainment
University of Technology Sydney (UTS)	www.uts.edu.au/	Sydney, NSW, AU	Master of Animation, Bachelor of Science in Games Development, Master of Interactive Multimedia
University of Teesside	www.scm.tees.ac.uk/html/undergraduate_computer_games_courses.html	Middlesbrough, Cleveland, GB	Computer Games Design, Computer Games Art, Virtual Reality, Visualisation, Computer Animation, Interactive Computer Entertainment
University of Texas at Austin	www.gammaprogram.utexas.edu	Austin, TX	Game Development Program
University of Texas at Dallas - Arts and Technology program	www.atec.utdallas.edu	Richardson, TX	Arts and Technology
University of the Philippines IT Training Center	itc.up.edu.ph	Quezon City, NCR, PH	Game Development Track
University of the Witwatersrand	wsoa.wits.ac.za/digital-arts/	Johannesburg, Gauteng, ZA	BA PVA Game Design, BEngSc Digital Arts, MA Dissertation / Research Report and Practical Project, BA Hons Game Design
University of Utah: Entertainment Arts and Engineering Master Games Studio	www.eae.utah.edu/	Salt Lake City, UT	Entertainment Art and Engineering Master Games Studio
University of Utrecht	www.uu.nl/	Utrecht, , NL	Game and Media Technology
University of Verona	www.mastergamedev.it/	Verona, Italy, IT	Master in Computer Game Development
University of Wales, Newport	www.newport.ac.uk/	Newport, South Wales, GB	Computer Games Design

TITLE	URL	LOCATION	PROGRAM OFFERED
University of Washington - Bothell	www.bothell.washington.edu	Bothell, WA	Game Design and Development
University of Washington - Professional and Continuing Education	www.pce.uw.edu	Seattle, WA	Game Development, Virtual Worlds, 3D Animation for Games & Digital Media
University of Waterloo	www.cs.uwaterloo.ca/	Waterloo, Ontario, CA	Computer Science and Computer Engineering
University of West Scotland	www.uws.ac.uk/	Paisley, Renfrewshire, GB	Computer Game Technology
University of Western Ontario	www.csd.uwo.ca	London, Ontario, CA	Computer Science with Minor in Game Development
University of Winnipeg - Professional, Applied and Continuing Education	www.UWinnipegCourses.ca	Winnipeg, Manitoba, CA	Game Design and Development
University of Wisconsin-Stout	www.uwstout.edu/programs/bsgdd/index.cfm	Menomonie, WI	Game Design & Development-Computer Science, Game Design & Development-Art
UPC School of Professional and Executive Development	www.talent.upc.edu/professionals/presentacio/codi/20801200/	Barcelona, Barcelona (Spain), ES	Videogame Design and Creation
Utrecht School of the Arts/ Faculty of Art, Media & Technology	www.hku.nl/	Hilversum, CL, NL	Game Design & Development
Vancouver Animation School	www.vanas.ca	Vancouver, British Columbia, CA	3D Computer Character Animation, Concept Art
Vancouver Film School	www.vfs.com	Vancouver, BC, CA	Animation Concept Art, Game Design, 3D Animation, Classical Animation, Sound Design, Programming for Games, Web & Mobile
Vancouver Institute of Media Arts (VanArts)	www.vanarts.com	Vancouver, BC, CA	Web Development & Interactive Design, Game Art & Design, 2D/3D Character Animation, Visual Effects
Victoria University	www.computergraphics.ac.nz	Wellington, , NZ	Computer Graphics
Virtual Technology & Design	www.caa.uidaho.edu/vtd/	Moscow, Idaho	Virtual Technology and Design
Visual College of Art & Design	www.vcad.ca	Vancouver, British Columbia, CA	3D Modelling Animation Art & Design
Wake Technical Community College	www.cet.waketech.edu	Raleigh, NC	Simulation & Game Development - Programming Track, Production Certificate, Mobile Game Development, Simulation & Game Development - Art Track, Modeling and Animation Certificate
Westwood College	www.westwood.edu	Various Locations and Online	Software Development: Major in Game Software Development
Wits University	www.wits.ac.za/academic/humanities/wsoa/digitalart/13408/game_design:_faq.html	Johannesburg, Gauteng, ZA	Game Design
Worcester Polytechnic Institute	www.imgd.wpi.edu/	Worcester, MA	Interactive Media and Game Development
Yoobee School of Design	www.yoobee.ac.nz	Christchurch, Canterbury, NZ	Diploma of Computer Graphic Design, Diploma of Web Development, Diploma in Advanced 3D Graphics, Diploma of Interactive Design, Diploma of Animation & Digital Video
Zurich University of the Arts	www.zhdk.ch	Zurich, ZH, CH	Interaction Design, Game Design

Virtually

A BEGINNER'S GUIDE TO DEVELOPING GAMES IN VIRTUAL REALITY

Chris Pruett

The speed at which virtual reality has gone from buzzword to high-profile consumer product is astounding. Scrappy, duct-taped prototypes have matured into full-fledged headsets in less time than it takes many games to ship.

But VR is a challenging platform to develop for. It has stringent technical requirements and is incompatible with many tried-and-true game design patterns. Still, developers are flocking to virtual reality because it offers something that no other platform can provide: A way to step into worlds of your own design, with full immersion, no distractions.

If you are an indie developer thinking of dabbling in VR, I have good news for you: VR customers are hungry for games and experiences, and most of the best software available today comes from small, independent teams. But VR development is new and different, so before you jump in it's worth considering the challenges. This guide will get you started.



designed

THE TECH

At a technical level, virtual reality software is much like a traditional 3D game's. You've got a real-time game engine rendering a 3D scene to a display. Actually, you're rendering to two displays, or at least to two different textures—one for each eye. The two eye textures contain the same scene from slightly different perspectives to create a stereoscopic image. These textures are then warped to match the curvature of the lens in the VR headset before display.

For many engines, stereoscopy requires rendering your game's scene twice for every frame. Double the culling, double the draw calls, double the passes, and double the number of polygons sent to the GPU for every frame. Naturally, there are clever ways to reduce the amount of work needed to produce a stereo image, but first-pass VR implementations usually involve a doubling of almost every step in the rendering pipeline.

VR frame-rate requirements are also strict compared to traditional games. While it's reasonable for PC or console games to swing between 30 and 60 frames per second, VR requires a high, locked frame rate. On PC, this means you must produce your scene—for both eyes—at 90 fps. On mobile VR platforms, the rule is 60 fps. Frame rate drops are not acceptable in VR because they tend to make players sick.

This means that performance and optimization are a big part of most VR projects. Though existing game engine technology works well for VR, strict rendering requirements mean that VR games must be exceptionally efficient. The good news is that Unity, Unreal, CryEngine, and others already provide

VR-specific rendering optimizations, and more are on the way. These engines all have great support for VR, and I recommend them over trying to roll your own, at least for early experiments. Most current VR developers are using Unity or Unreal.

COMFORT AND GAME DESIGN

Though the nuts and bolts of performance optimization can be difficult, by far the most challenging part of VR game development is design. You will spend a lot of time on your VR project trying to find new ways to interact with the environment. Some developers see this as a limitation, but I think this is the most exciting aspect of VR game design. You have the opportunity to invent brand-new design patterns for core systems. Precedent is so established on traditional platforms that few games get to experiment with basic mechanics like first-person locomotion. In VR, almost nothing is set in stone. VR design today looks like smartphone game design before *Angry Birds*, or console first-person shooter design before *Halo*. The design space is wide open.

That said, you can't just take a first- or third-person game designed for screens and plunk down a VR camera. Everybody tries this once, and the result is almost always sickening. Many common approaches to camera-motion design do not work in VR because certain types of movement can induce nausea. Right-stick rotation? Terrible. FPS walk bouncing? Unbearable. Spline or animated camera curves? Never, ever. Your standard first-person shooter control code is a recipe for gastro disaster in VR.

It's critical that you take your players' comfort seriously. Motion sickness is caused by many factors and affects individuals in different ways. Even if your rocket-powered bobsled game feels great to you, others may be more sensitive. The more comfortable you can make your game, the larger the slice of this nascent audience you have the opportunity to capture. Before you get too deep into development, it's worth reading up about the sources of motion sickness in VR, the feeling ofvection, and the basic function of the human vestibular system. Though there may not be many VR design rules yet, we know a lot of base principles about motion sickness.

It is also important to test your game with as wide and diverse an audience as possible. As you develop for VR you will become less sensitive to sickness, and over time you will lose the ability to judge what feels good and what does not. Find a friend who feels queasy on a swing set, the kind of person who gets sick just thinking about reading a book in a moving car. If you can make something that these folks can play, your work will be broadly accessible.

Plan to build something new. It's possible to rework an existing game for VR, but it isn't a simple



Dead Secret, a survival horror VR game by Robot Invader.

port; you'll end up ripping bits of the game apart and gluing them back together to fit a new, VR-specific design. It's almost always better to start a VR project from scratch.

THE MARKET

There are several competing VR headsets today, and before you go off to build your holodeck opus, it's worth considering where your game fits best. The industry can be roughly divided between high-end cabled headsets [like the Oculus Rift, HTC Vive, or Sony PlayStation VR. -ed.] and mobile phone-based devices [like Samsung Gear VR, Google Cardboard. -ed]. The cabled headsets are more expensive, have higher visual fidelity, and support positional tracking (the ability to track head motion in addition to rotation). Mobile devices have a lower barrier to entry, and are growing very quickly. Most people will experience VR for the first time with a mobile phone-based VR device.

Another interesting split relates to input and play method. The cabled headsets I mentioned will have support for motion controllers this year. Motion controllers allow the player to move one or both hands in space, similar in concept to Nintendo's Wii Remote but with dramatically better tracking accuracy. Will you make a game that is played with a game pad while seated? A game designed for walking around a large virtual space with motion controllers? Something in the middle? These decisions will define which headset can support your game.

To reach the largest audience, including customers in countries where high-end PCs are not pervasive, you should develop for a mobile VR platform like the Gear VR (full disclosure: the Gear VR's tech is powered by Oculus, my employer). Mobile is also a fantastic target if you are considering 360-degree

video. If your target is a core audience who is willing to spend money on high-fidelity VR experiences, the cabled devices may be your best bet.

[Also consider the differences between the storefronts on each of these platforms. Oculus, Sony, and Google have their own digital storefronts, while HTC has partnered with Valve to distribute VR games via Steam. Each of these platforms has its own rules and processes for accepting and publishing game content, as well as their own set of platform-specific APIs for things like in-app purchases and achievements. -ed.]

BUT IS THERE A THERE, THERE?

At this point you're probably asking, "If I pick the right market, do the R&D, and optimize all the polygons, is it actually possible to make money in VR?" The answer is yes (with caveats) today and probably yes (without caveats) tomorrow.

My studio, Robot Invader, released *Dead Secret* for Gear VR in November 2015. After about eight months on the market it has about 130,000 players on Gear VR alone, and has since shipped for other platforms. Gear VR sales have remained steadily strong. As the number of people who own this device grows, so does our customer base. Oculus recently announced that active monthly users for Gear VR surpassed one million people in April.

The cabled headset market is smaller and newer: HTC and Oculus shipped their devices only a few months ago, and Sony's headset doesn't reach customers until later this year. With that said, the titles designed for those platforms tend to sell at a higher price point, and I know several developers who are already profitable.

It is still early days, though, and customer behavior in these markets is hard to read at this point. There is evidence that customers like high-production value games that use VR in innovative ways, like *The Climb* (no surprises there). Indie titles like *Windlands* and *Job Simulator* have garnered wide praise. There are a few examples of non-game experiential software, like *Apollo 11* and *Titans of Space*, finding a strong audience. Stereoscopic 360 video applications also appear to be quite popular, particularly on mobile devices. On PC, game pricing tends to match non-VR equivalents pretty closely. On mobile, almost all software available today is premium, paid-up-front experiences rather than free-to-play.

One clear trend is that the easy, low-hanging fruit games have already reached saturation. If your idea is to create a virtual rollercoaster or a virtual haunted house simulator, your competition will be considerable and your customers will be bored. There's a lively community around small games, but the era of players spending money on tech demos is

over. Do not embark on a VR project if your plan is to make something that ends in 15 minutes.

There are also several VR-centric funds out there looking for developers to invest in. Oculus, HTC, GREE, Colopl, and a few other game companies and venture firms have announced funds designed to help spur VR software development. This money comes in various amounts and with various strings attached. Some deals look like grants while others are investments designed to buy stock in upcoming VR developers. There are a few traditional publishers working in VR today, but these are, at least for the moment, a little harder to find. It is likely that publishing deals will become more common as the number of VR headsets in the world increases.

Regardless of whether you work with an external group to fund your project, it's smart to see VR development today as an investment in the future. As headsets reach customers in volume over the next year or two, the market will mature quickly. Because VR is technically challenging and breaks many common design rules, developers who learn how to address their audience in VR early will be at a significant advantage over latecomer competition.

SENSE OF WONDER

A virtual reality headset is not a peripheral. It's not a funky display that augments the games you already have. It is a doorway, designed to take you to places you've never been before. The best VR software is written with this key understanding woven deeply into its design.

VR development is hard. The tech is hard, the game design is hard, and the market is still young. And yet, the experience VR affords is irresistible. It's easy to see why so many people think VR will change the world. The pioneers in this space are laying the groundwork for a generation of software unlike anything we have seen before. Slipping the headset over your eyes and finding yourself standing in another world is magical. I guess that makes us magicians.

Chris Pruett works at Oculus to help developers make better VR software. He is the cofounder of Robot Invader, which shipped its first VR game, Dead Secret, in 2015. Find him on Twitter at [@c_pruett](#).

VR games must bend their perspective to fit eyes or lenses, as seen in this Google Play game *Hardcode*.



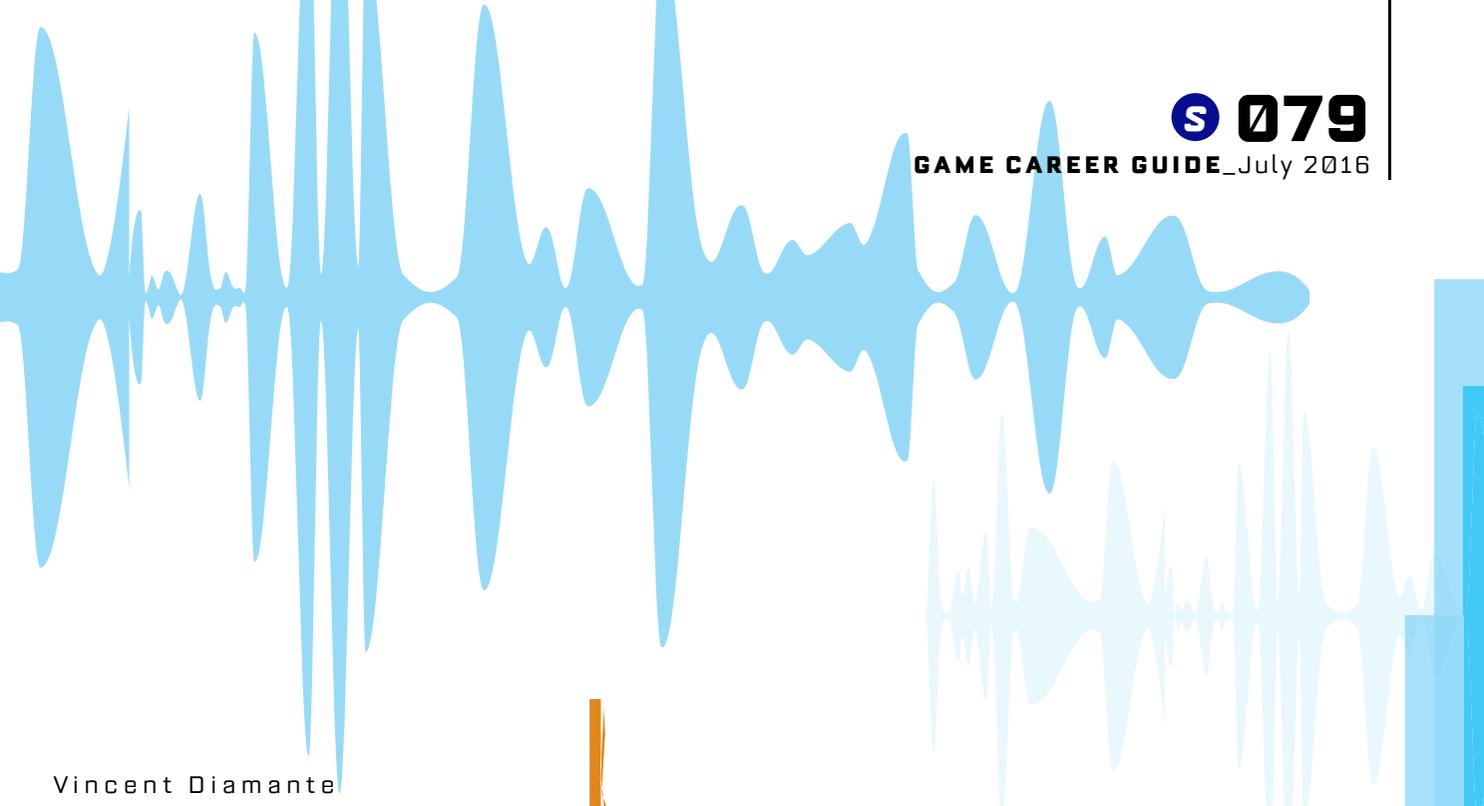
Swery, developer of *Deadly Premonition*, helps his partner Sharapova experience VR.

Sounds good

ADVICE FOR MAKING AUDIO (WITH

Game audio is a huge part of a game's identity, something that players will take away with them after the game ends. It's really important to get right. Luckily, sound libraries and software make audio creation easier than ever. So why not just do it yourself?

That's a perfectly reasonable path to take, but it's only one of your options as a game developer. Game audio runs a huge gamut, from synthesized sound cues to professionally recorded vocal artists, from stock library sounds to bespoke foley, and



Vincent Diamante

to me!

OR WITHOUT) AN AUDIO SPECIALIST

from minimal aural structures to thick sonic landscapes.

And then of course there's music, comprising all instruments and genres. Thanks to the evolution of software tools and the ever-declining cost of entry, good sounds and good music are easier than ever to not just acquire, but to create with minimal training and experience.

With all these options, though, it's easier than ever to have your game audio go astray. Beyond bad sounds and music, bad decisions—small and large—can impact gameplay in ways you don't intend. These audio missteps can diminish an otherwise enjoyable experience, teach players the wrong way to play the game, or even repel players from playing at all.

Here are some things to think about as you give aural life to your games.

WHAT DOES THE PLAYER NEED TO KNOW?

This may seem like a very basic tenet of game development, but even for the most seasoned developers it bears repeating. Does the player need to know what the sound is communicating in the game? What, in fact, do you want to communicate?

Not too long ago I would toil away without asking this question, both while working on my own games and while working as an audio designer for other projects. Back then, I was extremely asset-focused, working in spreadsheets filled with EVENT ID and ANIM ID tags, dutifully

making sure there was a corresponding audio effect for every thing that existed in the game. Footsteps, screen wipes, rustling leaves, street lamps, everything.

Because everything can and should make a sound... right?

But let's think about that. Does the player need to know about the crunchy autumn leaves on the ground or the birds flying in the background? If your visual novel is set in the heart of a bustling downtown, does the player need to know about the car traffic? Should your side-scrolling stealth shooter set in a cruising 747 jumbo jet have the engines constantly droning through the speakers? Must your first-person shooter carefully play every footstep your unseen player character takes?

I may have baited you into thinking the answer is an emphatic no. Well, it's not, but it depends on what your game needs to do. Perhaps your visual novel sets up a conversation where the sound of traffic is incredibly important, as the honking cars interrupt the spoken dialogue, creating confusion and misunderstanding. Maybe that 747-based stealth shooter encourages the player to take advantage of the engine noise to mask their in-game

actions, creating a difficulty progression as the player makes their way from the cargo hold to the more muffled passenger area. Audio can certainly play an important and essential role in your game, but you have to be honest with yourself about whether that audio is truly necessary for the player's enjoyment and progression.

EVALUATE SOUND ON THE GAME'S TERMS

It's very easy to consider sound its own separate world, but that does your soundscape a disservice. Instead, realize and act on the idea that sound does the same thing that all other game components do. It creates atmosphere, establishes lore, communicates a game state, and provides rewards and punishment. Sound can do these things wonderfully, but so do game mechanics, dynamics, art assets, and narrative.

Is the sound working to further a concept that's already established elsewhere in the game? Is the sound by itself creating an essential component of the game loop? If a particular part of the game is already well served by art and animation, does it also need to be accompanied by sound?

Let's say your game is set in a forest. How much of that forest needs



Flower, a game with sound design by Vincent Diamante, uses adaptive sound to support the game state.

to be present in the sound? Do you need leaves rustling in the night wind and crickets and coyotes and owls constantly making their presence known? Maybe you just need a smooth ambient forest wind. Or maybe you don't need any sort of ambient sound. Perhaps getting the forest creatures to let themselves be heard at the very beginning of the level's start is enough, as the player just doesn't need or care to have forest sounds constantly assaulting their ears.

What if your game's player character is notable for jumping? How much does the audio sell the idea of jumping? Perhaps your game is rather abstracted, and the player is just a few pixels or cubes. Maybe in this case the sound needs to sell the idea of jumping far more, with a bold moving FM synth or a pitch-bending sample. But if the player character is impeccably modeled, beautifully textured, and artfully mo-capped, perhaps the sound of jumping doesn't need to be a jump sound at all; a little shuffle of feet synced up with the animation data could be all that's needed.

Though we try to make our sound decisions with great forethought and care, the nature of games and interactive systems means we inevitably wind up piling sounds upon sounds. This leads us to the world of mixing.

DON'T LET YOUR AUDIO DO TOO MUCH

Curiously, it's far easier to do too much in the game audio world, compared to too little. You wind up with a game that's too loud, with too many sounds, and that feels too busy.

Larger screen sizes and resolutions mean we're bombarded with increasingly more visual stimuli, but sound still has to stay workable in a small space. Our ears and brains have not evolved beyond the folks who played Pong at the local bar, and our devices are actually worse in terms of audio presentation than those from the 1980s and 1990s.

Our 24-inch all-in-one PCs might be able to do pixels and texels aplenty, sending particles flying across the screen while maintaining legibility, but

the speakers can't even represent the same range of sounds a lowly 20-inch CRT-based television can. We can't count on surround sound... heck, we can't even count on stereo sound, with the proliferation of mobile devices.

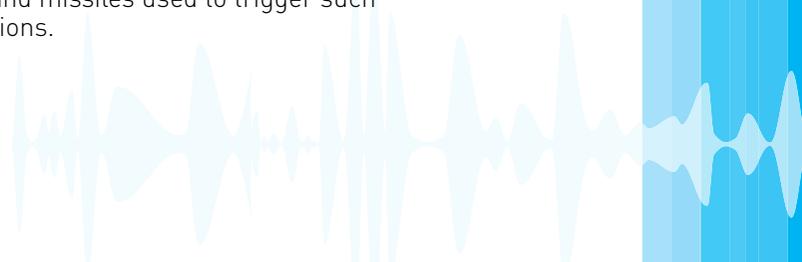
It's a reasonable bet that you're reading this article on something you also consider a portable gaming device. It's likely capable of HD-resolution graphics yet can't handle stereo sound or decent bass without headphones—and you can't rely on a player having those.

Moving from sound curation to prioritization is a big step. While curation encompasses all that should be played, prioritization is all about how. How should the sound actually sound, how should the sound be turned on, and how should the sound be turned off?

Let's say we have a shoot-'em-up game with exploding spaceships. Lots of exploding spaceships. Perhaps you've decided that you've got the best explosion sound already... and it's a wonderful explosion sound: big and rich and full of circumstance. It perfectly matches with the spaceship destruction animation, so you put it in the game.

How does that explosion sound when you hear it by itself? What if you hear it five times in a row? Eight times in a row? At a rate of once per second? At a rate of five per second? What if it's called five separate times within the space of 100 milliseconds? What if three explosions are called on the exact same frame?

And keep in mind, this is all before you even consider how this explosion will work with the rest of the soundscape, complete with background music, ambient noise, and other foreground elements like the guns and missiles used to trigger such explosions.





The Bioshock series has a lot of complex audio queues, and requires an advanced understanding of audio mixing.

Simply put: sound can get quickly out of hand. If your game is more complex, it might be more than you can handle on your own.

SHOULD I ASK FOR HELP FROM A SOUND SPECIALIST?

While sound can achieve the same goals within your game as other components like art and mechanics, that doesn't necessarily mean the metaphors and techniques translate as well. Visual minimalism, for example, differs markedly from musical minimalism, which is totally different from audio minimalism.

Time plays a huge role in audio, which puts it at odds with the state-based worlds of video games. The immediacy of reaction to audio compared to visual response times makes audio incredibly powerful in game design.

But there are many ways of interpreting audio cues, and audio can take much longer to convey the information contained in a simple picture. This is part of why you see fewer games driven by audio than by visuals. Audio mixing approaches the nebulous world of interaction design.

More sound design means connecting more audio assets (WAV and MP3 files) to the game state and player input in a complicated dance of code and envelopes. Because audio needs can sometimes seem to be separate from game needs, specialized tools like Fmod, Wwise, and Fabric have come about to help audio designers do complicated audio maneuvers in connection to the game without requiring a programmer as an intermediary.

In the 1990s, making the sound of a car engine required a bit of programming know-how. Back then, the sound designer didn't just make cool sounds, they also had to know the ins and outs of the particular FM sound chip on that particular arcade board. Nowadays, a good sound designer can serve car engine audio needs easily... so long as the game is properly reporting the engine's RPM to the audio middleware.

And car engines are hardly the only thing that needs advanced or dynamic audio. Anything that can be clearly conveyed by the game to the audio designer, courtesy of a friendly audio API, is potential grounds for dynamic

audio and music. A few examples include player character hit points falling below a certain threshold. An enemy army taking a certain percentage of total territory. An NPC falling in love with the player. A non-Euclidean geometric figure being uncovered in a puzzle game.

Given today's high-quality audio tools and accelerated pace of video game development, it's ideal to have direct access to the game, and iterate the audio in step with the game itself. With greater ease of development, quality increases, and it's easier than ever to find sound designers who can drive an Fmod or Wwise project.

Still, it's tough to find brilliant sound designers. They need to have that *je ne sais quoi* that makes audio sound great, not just serviceable. The best video game sound designers are as good at understanding game design as they are at audio. They're able to intuit not just how the audio should sound, but also the numerous ways that sounds can potentially be driven by the machinations of game design.

Sound specialists are a weird and wonderful brand of game developer. The fact is, all the pitfalls and examples in this article are typically considered in that first second after being posed the question: "What do you think this should sound like?" The world of sound design for games is filled with heuristics and statistics, probability and possibility.

While there are certainly sound designers out there that are better than others, skill level matters less than the quality of your working relationship, and communication is key. Feeding the sound designer information is a healthy start to good game sound design. Ideally, constant connection to the game itself, inside and out, can engage a sound designer and get them to create great content that is well matched to the game.



GAME AUDIO IS REALLY TOUGH (SOMETIMES).

With so many game audio options, it's crucial to consider your time, budget, and skill set. If you are thinking about doing the audio yourself, start by looking at the potential audio needs from a player perspective rather than a production perspective. After that, remind yourself that the things audio does in your game are the same things that are achieved elsewhere with other types of assets. This can help you maintain perspective regardless of your familiarity with audio or audio production. Finally, when you decide to tackle the audio component, keep yourself reined in with the knowledge that a little goes a long way (along with the converse: too much can destroy all of your hard work).

And if, after evaluating your audio needs, you decide that you want to go with an audio specialist, remember that there's amazing potential to be had from a great working relationship... so long as you recognize the power of the modern audio workflow. It should be motivated by knowledge of both sound design and game design, working together!

Vincent Diamante is an audio designer, composer, and engineer, currently working at ThatGameCompany. He previously worked on Cloud, Flower, and Skullgirls.

Game audio pipelines have come a long way since the days of Outrun.



A seller's market



er's
et



By Michael Cox

MARKETING FUNDAMENTALS FOR YOUR FIRST GAME

I'm Michael Cox, Marketing Director for Crows Crows Crows and many Kickstarter projects. And I'm going to tell you one way you can use your game's metrics and turn them into fans—and hopefully dollars.

I started learning about advertising by watching my older brother run a web site that offered free sites in exchange for ad space. My brother and I were concerned with what made our web site appealing to advertisers, who in turn were concerned with views, clicks, and impressions. The reason they were interested in these factors, of course, was that they wanted to know how many users were likely to convert into customers.

I wondered: if these companies value information on views, clicks, and conversions, why aren't many indie developers tracking these basic metrics too? In comparison to other industries, indie developers are allocating far fewer resources to marketing.

The indie game industry still hasn't seriously started to include marketing in the development process. Even when I had only been in advertising for a few years, I was able to achieve early success by taking the elements advertisers value and applying them to indie games. Understanding and implementing these fundamental principles can guide new developers to make educated decisions when marketing their games.



Crows Crows Crows

WHAT IS MARKETING?

I commonly hear developers say something along the lines of: "If I failed, it had to be because of the macroeconomic state of indie games and for sure not me."

There are numerous fallacies circulating in the indie game industry that discourage developers from believing they can influence the fate of their work. People who preach that an indie game's success is based on luck are perpetuating their own destruction; this conclusion just reflects a lack of information.

In meteorology, a weatherman is not considered unlucky when predicted forecasts are incorrect. Instead, we recognize that meteorologists are working with imperfect information. Likewise, it is not purely luck if a journalist writes an article on your game or if your post gets upvoted to the top of Reddit. To think like a marketer, ask yourself: What information am I missing, and how do I obtain it?

Marketing is a way to multiply your product's value, creating more value than it costs. Most industries put more money into marketing than the product because it is the most effective use of the budget. Think of it like this: since increased product views multiplicatively increase potential sales of that product, then investment in advertising will theoretically provide more value than improving the product could.

This is difficult for many people to accept creatively, but pragmatically the idea is sound, and it is something you should consider. At a certain point, it becomes more difficult and more expensive to make only marginal improvements.

Marketing as a term is very broad, encompassing hundreds of disciplines, but you don't need to understand all of marketing to increase your game's success.

THE CONVERSION SYSTEM

By figuring out how you can track views, clicks, impressions, and conversions, you will form a library of information that you can use to determine whether

your actions are providing more benefit than their expense. The conversion system is a method of organizing these scattered data points and making them functional. A conversion system is complete if the output is more than the input. But, even an incomplete system is valuable because it identifies potential risks.

There are five points to a conversion system:

- 1. COST** includes everything you put into the project, including money, labor, time, and investment in your team's morale.
- 2. VIEWS** are the number of people who will see the action. Think about views as the number of people you can possibly influence.
- 3. ENGAGEMENT** is how many people saw the material and sought more information. A percentage of the views will become engaged.
- 4. CONVERSION** is what happens when someone cares about what you are doing enough to contribute something back. A percentage of those who engage will convert.
- 5. VALUE** is the contribution that converted people will provide. In order to obtain value, you need a way for people who want to contribute to give something back.



This five point conversion system runs in a repeated cycle.

NEWSLETTERS

To measure views and engagement, try attracting views to a website through press articles, ads, and social media. You can then use software such as Google Analytics or built-in social media analytics to measure the number of views and the rates of engagement. Even something as simple as someone enlarging a photo shows a level of engagement.

Now that people are engaged, a percentage of them will convert, but what do you convert them to? Indie games don't normally have a way to capture this value until the product is released, but by allowing people to subscribe to a newsletter through the web page, you can start building your audience long before your product's launch date.

There are many benefits to having a newsletter. For one, they are low effort. There is no production involved (aside from writing), and subscribers are reached directly. Sendy and MailChimp are inexpensive mailing list options, costing only pennies to ship newsletters to thousands of people.

A newsletter also makes it easy to track information. Through Sendy or Mailchimp, you can track how many people opened the email, how consistently they opened newsletters, and whether or not they clicked through your links. The newsletter

allows you to use the conversion system to recruit people before you launch the game, giving you more information to predict how many sales you might obtain.

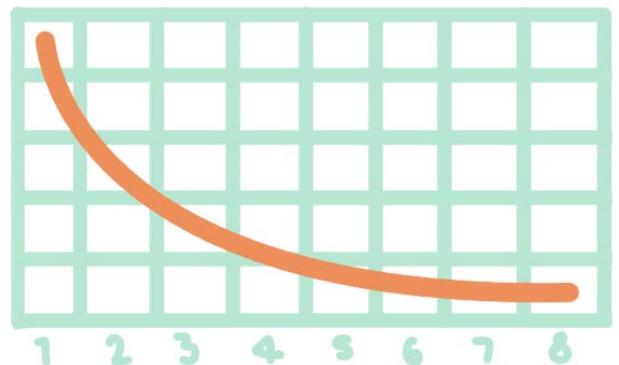
THE COHORT ANALYSIS

The more hurdles someone is willing to jump through, the more they like you. The more someone likes you, the more likely they are going to be there for you.

The concept of conversion is simple: The more someone engages with your content, the more likely they are to become a potential sale. A conversion by itself isn't very interesting, but the reason behind someone actually buying something is. Engagements typically lead to conversions because your subscribers are excited about what you have to share.

The Cohort Analysis predicts how many conversions will become sales. To perform a Cohort Analysis, you take a group of people and separate them into smaller groups based on how much they engaged with whatever you sent them. The more consistent the viewer is, the more they like you—and the more they like you, the more likely they are to purchase your game.

Although you cannot measure exactly how many people will buy the game, you can estimate which subscribers will be extremely happy, very happy, or moderately happy with your product. Knowing that a percentage of subscribers will most likely convert to sales, we can make a rough estimate of the possible number of sales.



Once you've filtered down to the bottom of the graph, you've got people who will definitely buy your game.





HOW DO YOU GET STARTED?

Using this system, you can now measure whether an effort—such as going to a convention like GDC or PAX—will be valuable. First figure out how much it will cost, including flight, booth, hotel, and so forth. This number will be the input. Then you will need to calculate how many views you might get, based on the views of articles on somewhere like Kotaku, which handily shares their viewership information alongside each article. Although the exact number of people in attendance is variable, you know that views to engagements to conversions leads to sales, and from there, you can work backward to figure out how many views you need to make attending viable.

estimated required views and determine whether the potential exposure even remotely justifies the expense. (Keep in mind, press will sometimes flake on you and not show up.)

This is the same process for any potential marketing endeavor. Complete the equation as best as you can and with as much detailed information as possible, then identify which elements you might be missing. The more complete you make the conversion system, the less reliant on luck your decision will be.

NOT ONE SIZE FITS ALL

How can it be that one equation gives you the answers to every question? Well, it doesn't. A conversion

$$\text{VIEWS NEEDED} = \frac{(\text{COST OF ATTENDANCE})}{(\text{AVG. CONVERTED USER VALUE})} / (\text{ENGAGEMENT RATE} * \text{CONVERSION RATE})$$

A formula to determine whether attending an event is worth it.

To figure out the engagement and conversion rates before attending the event, take the average rates from articles published on the game in the past along with the cohort and plug them in. If you've had no articles, views, or other coverage yet, you'll probably need to start by trying to get some of that, which is beyond the scope of this article. But suffice it to say, find something interesting to talk about—a trailer, a new feature, or a character model, and send it off to some sites for potential publication.

This equation predicts how many sign-ups will become potential customers. It's alright if the math seems sketchy—it doesn't need to be extremely accurate. We're just looking for something realistic and achievable.

The real problem of purchasing an expensive booth at an event is that even if you had perfect information about the value of each sign-up and their likely conversion rate, there is no guarantee you will ever receive enough coverage to make attending worthwhile. In an attempt to maximize my chances, I contact publications, schedule interviews, pitch articles to the press beforehand, track how many interviews and articles I was promised, and tally up the estimated views they might provide. Then I compare that to my

system isn't some mystical crystal ball of precision. It's meant to provide a more grounded approach to making better decisions. Although it is inaccurate, it's a start, and the more experienced you become with analytics, the more accurate your predictions will be.

It's important to recognize that marketing can't be summed up in an equation, and it might even take decades to hone your skill with it. But by having a basic understanding of what your marketing plan is missing, you can start taking the necessary steps to refine it.

Michael Cox is Marketing Director at Crows Crows Crows, a new studio from one of the creators of the Stanley Parable. Prior to that, he helped manage and create over a dozen successful kickstarters as a freelance marketer. You may reach him on Twitter at [@DevMicco](https://twitter.com/DevMicco).

GAMASUTRA JOBS

YOUR GAME INDUSTRY CAREER RESOURCE



FIND THE MOST SOUGHT AFTER POSITIONS

with top companies around the world

PUT YOUR RESUME IN FRONT OF INDUSTRY

leading employers via the resumé database

SOURCE TALENT ON GAMASUTRA.COM

the game industry's thought leader

gamasutra.com/jobs



Intro to game engines

**A BRIEF DISCUSSION OF SOME OF THE NEWEST
AND MOST POPULAR GAME ENGINES**

Adam Rippon

A few months ago, an old friend of mine from high school asked me what her son ought to focus on if he wanted to get into games. My immediate thought was, "Isn't your son like six?" to which she responded, "Sixteen, actually," and I realized how amazingly old I am now. The crushing realization of my biological decay aside, I realized I had no idea how I would approach getting into game development if I were starting out today.

I got my start in the year 2000, and my first professional projects were Game Boy Color games written in Z80 assembly (well, technically the Z80 workalike Sharp LR35902... but I digress). My work has changed an enormous amount since then, and I have changed platforms and languages multiple times. 8-bit assembly won't get one very far professionally today (although if you slid a GBC cart across my desk in an interview, who knows!). Luckily, in modern times, there





CTRL+L-Mouse: Pick Layout

are tons of helpful game engines to get you started.

It used to be that game development meant learning an esoteric instruction set and the ins and outs of bespoke hardware. It was more fun than it sounds, but there are much easier ways to create video games now. Arguably the three most important game engines available for aspiring commercial game developments are Unreal, Unity, and Lumberyard. Just for fun, let's compare these three engines to the three big operating systems.

UNREAL ENGINE 4

Unreal is somewhat Mac-like in the sense that the editor is immensely polished and refined and has very strong tech behind it, but it's a little harder to find good advice for. Searching forums for how to do something in Unreal is often a chore, as there is a ton of bad advice out there from people who speak confidently but know very little.

Unreal officially supports two major language choices—C++ and Blueprints. I love C++, and I think everyone should at least learn it. However, C++ has a huge hobgoblin in the spectre of memory management. Manual memory management is a chore, but once you understand it, you are a professional programmer. I'm serious. Knowing how to manually manage memory means you are part of a secret society of nerds who can control the destiny of software and, by extension, the world. Learning how to manage memory is like graduating from Hogwarts—yer a wizard, 'Arry.

Plain and simple—in my estimation, someone who knows how to manage memory knows how to manage hardware, and programming is by definition controlling hardware with software. So C++ is good, and if you learn it, you will benefit from it. But Unreal is unique in that it also supports a visual scripting language called Blueprint. Blueprints let you connect logic and arithmetic functions visually, and for a lot of people, it is an absolute godsend.

I have written an entire game prototype with exactly zero C++ code using Blueprints, and it worked amazingly well. If you're a visual learner, you will

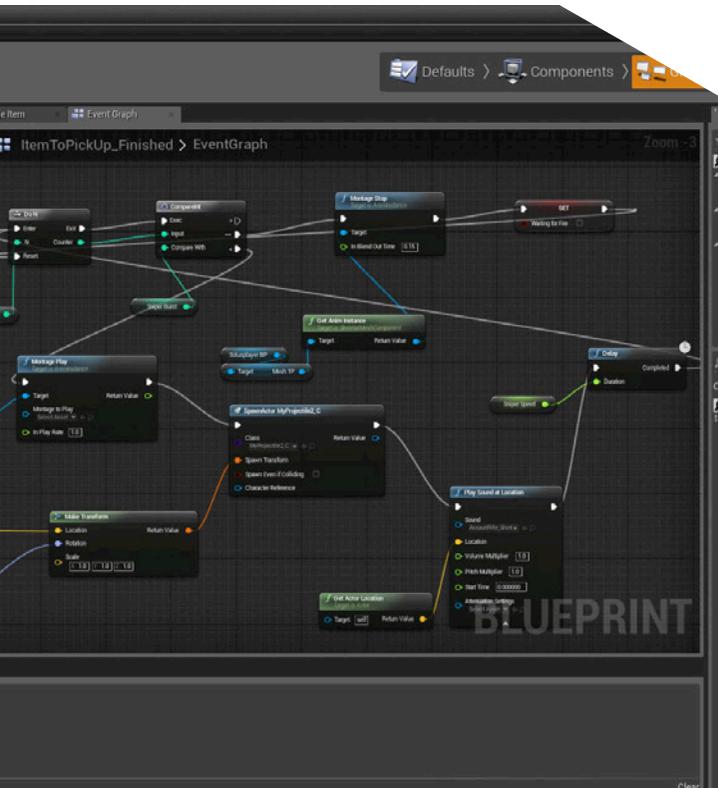


A Minecraft clone made using only UE4 Blueprints, (not created by the author).

probably love Unreal for Blueprints. This makes the engine attractive to designers, who don't want to fuss about with memory management but do want to script things like "when the player steps up to the console, open the monster closet behind 'em and spawn 3 imps."

Being able to create games in a combination of the two languages gives you incredible flexibility in your own work. Using C++ for the core bits, and Blueprints for behaviors and scripting is an excellent way to divide work. But Unreal is a smaller ecosystem in terms of raw numbers. There are fewer plugins that support Unreal, and so you'll be doing a lot of your own work.

Artists, however, will be in heaven. The pipelines for importing and exporting models and animations are top-notch, and the in-editor tools for manipulating the environment make it a number one favorite amongst level architects. That said, Unreal targets a more professional market, and while they've made strides towards

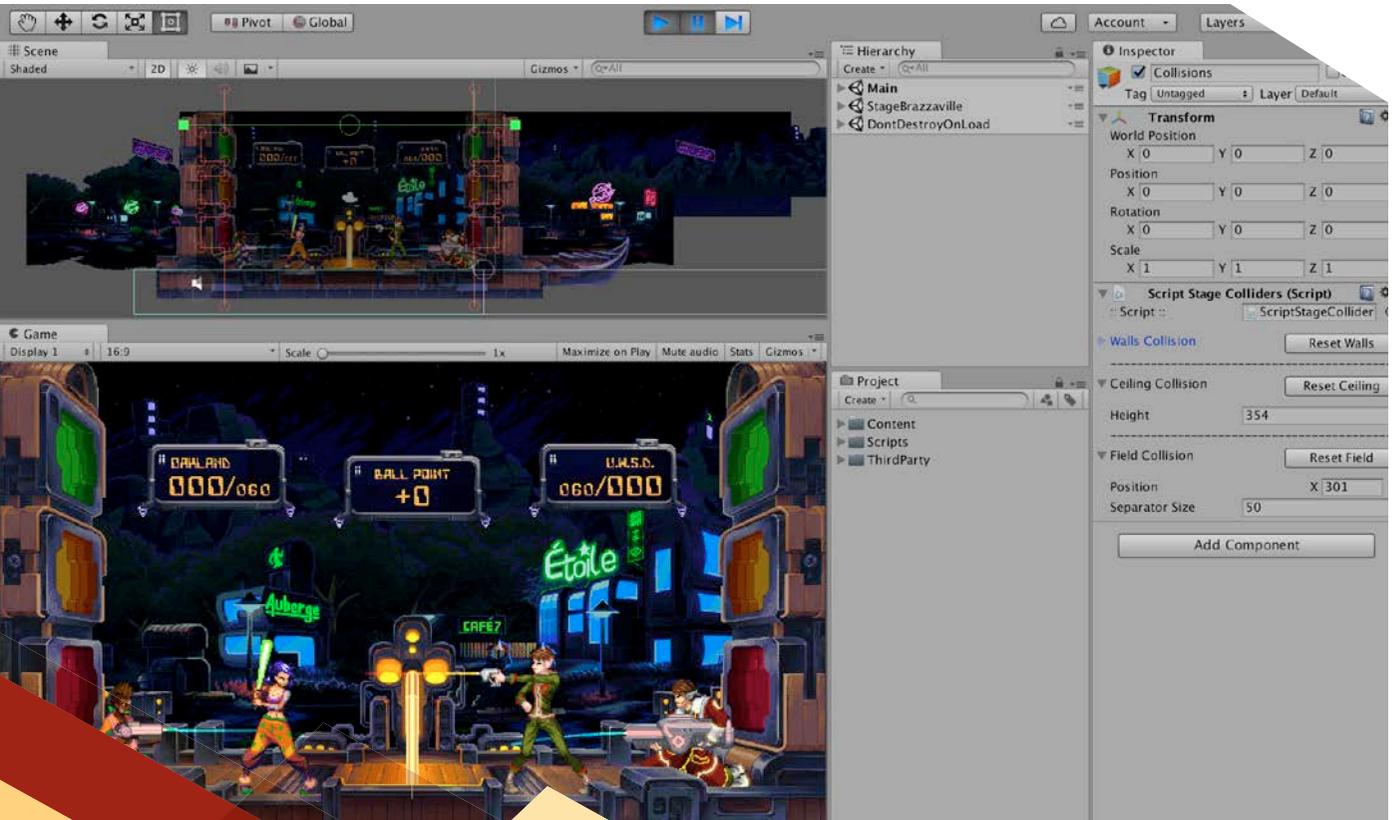


making it accessible to smaller teams, there's a reason that a lot of the big studios make this their de-facto choice.

UNITY 5

Unity is a lot like Windows: it's immensely popular and very well understood by the community, but many of its best practices are opaque, and it sometimes crashes at extremely inopportune moments. Still, while Unity has many warts, it also has many salves. Pretty much every difficulty point in Unity has spawned a cottage industry of plugins to fix or work around it. As a result, a lot of the work is already done for you if you want to strike out along an already-beaten path. It's also one of the few engines to have robust Mac editor support.

When you go down this road, know that you'll be using C# to program. C# is a more modern language than C++, and as such it's had support for features like automatic memory management, reflection, and closures for a longer time than C++. I think it's important to note that on many C# projects I see, weird performance issues can and do pop up. Most often, it's that telltale sudden frame rate drop when automatic memory management decides to clean up.



The Unity interface.

This issue is what most consider the downside of programming in C#. As a developer, you're constantly either fighting the collector to control when it collects, or you're trying to figure out the inconsistencies of memory allocation across different platforms. For instance, using C# with Mono versus C# with Visual Studio will cause memory to allocate differently and even leak memory in some cases.

That's not to say you can't fix these leaks, but with great power comes great responsibility. It's your job to figure out what's killing performance under the hood—and then to solve it without actually being able to look under said hood.

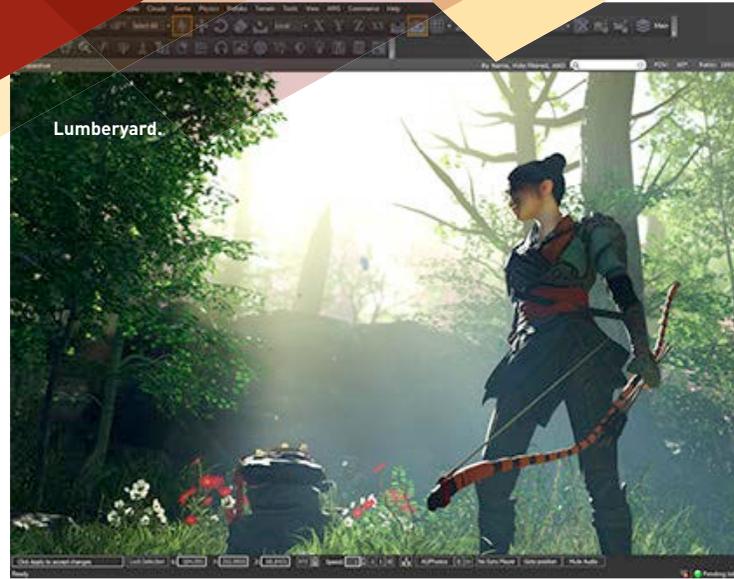
Even so, Unity is probably the best bet for people just starting out in the industry. The editor is friendly, compilation times are just about nonexistent, and you can get something up on screen quickly and easily. It's probably the most artist- and designer-friendly engine around, and it lets you twiddle with things while the game is actually running. Programmers can quickly add functionality for content creators, making this an ideal choice for a small team (especially if you like wearing a lot of hats).

LUMBERYARD

Finally, there's Amazon's Lumberyard. Even though the editor is only available on Windows for now, the engine is so Linux it hurts. I have spent the least amount of time with Lumberyard, but my experience has been that it is the most difficult to get started with.

Designers beware! You may need to get acquainted with Visual Studio (Microsoft's C++ compiler) in order to get started with the editor, and the setup process is a bear! That said, it's based on Crytek's CryEngine, which means it's very, very powerful once you learn how to play nice with it. I just don't know many people who know how to play nice with it yet.

Much like Android is a fork of Linux, Lumberyard is what happened when Amazon took a version of CryEngine and began making their own modifications to it. They're attempting to make it more modern by adopting the latest C++ coding standards and adding new functionality. And while it's slowly drifting away



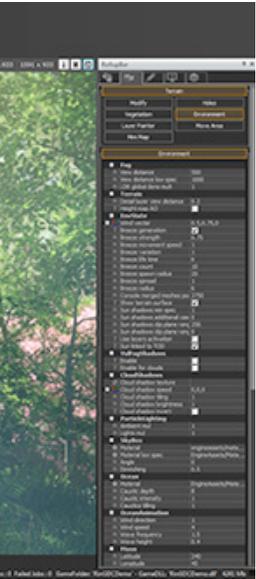
from the codebase it was spawned from, a lot of the original CryEngine tutorials will still work with Lumberyard, meaning it comes with something of a built-in knowledge base.

Designers use a visual scripting language called Flow Graph. While Flow Graph has its own share of warts, it will get you up and functional quickly. It lets you lay out your game logic like a Visio flow chart, connecting outputs into inputs and putting a lot of power in the hands of the designer. Amazon has invested a lot of time into making the art pipeline as friendly as possible, and while it hasn't quite caught up to Unreal, it's definitely giving them a run for their money. There are also a lot of cool experimental pieces—such as talking to the cloud directly from within your game—which have a lot of potential.

When I evaluated the beta, I felt like it needed a bit more time in the oven before I would want to work with it professionally. They're clearly targeting the Unreal market, looking for larger teams, but if you're not planning to ship anything any time soon, and you're willing to grow with a new technology, Lumberyard might be fantastic for you! Being at the forefront of a new technology carries great risk if it doesn't pan out, but the reward of being one of the first experts is immense.

ROLLING YOUR OWN

And then there's my favorite and least favorite option: rolling your own engine. Everyone will tell you that if your goal is to make a game, you should not make your own engine. It's absolutely true—if you're making a large game, you will spend years tinkering before you have a game worth shipping. But if you're making something smaller, this can actually be the right move. Look at games like Minecraft, or the even smaller Oh, Deer! by this



Oh, Deer! by Necrosoft Games is a small pseudo3D racer, and thus doesn't need a big engine to power it.

publication's editor and team. It's small and contained, so it doesn't need a lot of bloat around the edges.

There's also the fact that if your goal is to get a systems programming job in the game industry and be the kind of person people call up when they are desperate for help, rolling your own engine is a legitimate choice. I have maintained my own engine since 2009, and between myself and my coworkers, we've shipped at least one game a year and several apps with the engine. Because of that I have the experience necessary to command a pretty decent contracting rate when someone needs my help. Even better, since last year I've been integrating my engine with Unreal, so I am now able to do all the nice things

Unreal does while still enjoying the benefits of my own code. Of course, I'm 35 and bald now, so there's definitely a downside!

AND THE WINNER IS...

If you're looking to make a career in video games—a twenty-plus-year odyssey of blood, sweat, and tears—I believe you should look at Unreal Engine 4. It has everything you need in a AAA game engine, along with enough niceties to make one heck of an indie game. But it's not for everyone.

If your goal is to make a game right now, Unity 5 is probably the quicker path to releasing a video game. Many more people know Unity, and you'll have the benefit of a huge assortment of plugins.

Lumberyard is an excellent choice for tinkerers, presenting the possibility of gaining expertise in a new field that might be huge someday.

But don't make your own engine. ...unless you want to.

Adam Rippon makes games with his company Muteki Corporation. His best known series is probably Dragon Fantasy, but he's hard at work on new stuff. Find him on twitter at @TheAdamRippon.

Dragon Fantasy II by Muteki Corp. is a relatively large game, but also uses its own engine.



FREE DEVELOPMENT TOOLS 2016

KICKSTART YOUR TOOLCHAIN WITHOUT BREAKING THE BANK

Shane Marks

Game development tools have long been a big business.

Game development tools have long been a big business. The more complex games get, the more external software can help us develop faster. Time was, these were all pretty dang expensive, but nowadays there are scads of tools that are either totally free, or charge a revenue share percentage at the end of your project. These range from very specific art and production tools to full-on engines. So this year we've once again updated our list of best free development tools of 2016. This is not a comprehensive list, of course! But it can be a good jumping-off point for those looking to get into game development on the cheap.

Development Tools/engines

This section is devoted to tools for people who want to implement game systems via code or editors.

MonoGame

monogame.net

MonoGame is a C# framework that allows developers to target desktop, handhelds, mobile, and consoles using a single codebase.



The framework is written on top of Mono, an open source implementation of the .NET framework. The framework was originally designed to implement Microsoft's XNA 4.0 API, but Microsoft's decision to abandon XNA has meant that the

MonoGame project has started to grow beyond the stagnated XNA implementation.

Developers can choose to use MonoDevelop or Visual Studio as their IDE (integrated development environment). To date, the framework has been used

in multiple successful titles such as Bastion, FEZ, Mercenary Kings, TowerFall Ascension, and Transistor.

You should also be aware that Ethan Lee, the developer behind the ports of games such as Escape Goat 2, FEZ, Rogue Legacy, and TowerFall Ascension, has been working on a fork of the project known as FNA, which aims to reimplement MonoGame using SDL2 (see below), and can be found here: <https://fna-xna.github.io/>.

Simple DirectMedia Layer (SDL)

libsdl.org

SDL is a cross-platform library written in C that is designed to act as a wrapper around operating system-specific functionality. Some of this functionality includes audio, input devices, graphics hardware, file access, timing, and threading. SDL allows developers to write more generalized code rather than worrying as much about the underlining specifics of the system they're targeting.

Beyond this, extra functionality can be added through libraries that sit on top of SDL. Some of these libraries include support for multiple image formats, audio mixing, networking, TrueType font, Rich Text Format support, and many more via third party libraries.

SDL provides bindings for several languages such as C#, Lua, Python, Lisp, Haskell, and others. It's also commonly used as the base code for other game libraries such as LÖVE, PyGame, ScummVM, and even CryEngine, as well as emulators such as MAME and ZSNES.

SDL is probably the most commonly used game



library around—you'll see it in games such as *Amnesia: The Dark Descent*, *FTL*, *Portal*, *Psychonauts*, *Team Fortress 2*, *The Cave*, *The Walking Dead* and many, many more.

Unity

unity3d.com

Unity is a game engine developed by Unity Technologies that allows developers to target desktop, web, mobile, handhelds, and consoles using a single codebase. It comes with a fully-fledged game editor as well as the MonoDevelop IDE, which provides code auto-completion and debugging.



The engine provides scripting through Mono and allows developers to write games in C#, a JavaScript-like language that most people call UnityScript, or a Python-like language called Boo. While some developers new to Unity start off using UnityScript or Boo due to its perceived lower barrier of entry, the majority of developers use or eventually switch to C#.

Unity comes with full support for 3D and 2D games, and anything that you feel is lacking can usually be found through their online Asset Store or through an online search. Games like *Kentucky Route Zero*, *Night in the Woods*, *Superhot*, *Shadowrun Returns*, and *Gone Home* have all been made in Unity, showing the engine's versatility across multiple genres.

Unity has recently changed their subscription model so that regardless of the version you use, there's no difference in engine features provided to the developer. You can visit <https://unity3d.com/get-unity> to see what benefits a paid subscription will offer you.

Unreal Engine

unrealengine.com

Unreal Engine is probably the most well-known game engine in the industry. Unreal 3 powered the majority of the triple-A titles on the PlayStation 3 and Xbox 360—with the release of Unreal 4, Epic Games looks set to keep that going. Titles such as *Street Fighter V*, *Tekken 7*, and *Kingdom Hearts III* are all being powered by the engine, not to mention *Gears of War 4*.

But Epic Games, developer of the Unreal Engine, has been trying to reach out to the indie community as well. The engine has now become totally royalty-based. This change has made the engine a viable option for people looking to keep development costs down, with

royalties set at 5% after the first \$3,000 of revenue per product per quarter.

Unreal comes with a powerful editor that you can use to configure every aspect of your game. Developers can write their games using C++ or use a visual scripting language called Blueprint. While visual scripting might sound restrictive—and counter to what most other frameworks or engines do—it actually can be quite expressive. I've found on several occasions that it was quicker to implement or test a feature in Blueprint, and jump into C++ afterward. My only criticism of Blueprint would be that if you're not very conscious of space and clean-up, your Blueprints can quickly end up looking like a mess of spaghetti, and when you come back later you might think, "What the heck was I doing here!?"

One of the biggest advantages Unreal has over other big engines is that the full source code is available. That means you don't have to wait weeks or more for bugs to be fixed or even spotted, because you can just do it yourself!

Unreal currently provides support for Mac, Linux, Windows, PS4, Xbox One, iOS, Android, and WebGL. The engine comes with so many features that I couldn't possibly go over them here, but if you're interested in seeing where the engine shines, check out its rendering, materials editor, particle editor, networking, and level streaming functionality.



**UNREAL
ENGINE**

Top Alternative Libraries/ Frameworks

CryEngine

cryengine.com

CryEngine is one of the most well-known game engines in the industry and has been used in games such as *Evolve*, *Farcry*, *Crysis*, and *Everyone's Gone to the Rapture*. Like Unreal and Unity, CryEngine supports all the features you'd expect from a modern game engine. You'll need to become familiar with C++ and Lua, but with the latest version of the engine, you also have the option of using C#. CryEngine relatively recently changed to a pay-what-you-want model, so now is a good time to give it a shot.

Lumberyard

aws.amazon.com/lumberyard

Amazon Lumberyard is a licensed version of the CryEngine that has been customized to work with Amazon Web Services, Twitch, and other Amazon services. Some parts of the engine have been changed from the original source, such as the networking system, editor, and several other elements. The engine has support for C++ and Lua but is missing CryEngine's latest addition of C# support. To me, Lumberyard seems more geared toward developers who see Amazon services or Twitch as an integral part of their game.

Cocos2d

cocos2d.org

Cocos2d is a 2D framework designed for mobile and desktop, but the majority of games made with it are for mobile. It's divided into multiple branches, the most popular being cocos2d-x, cocos2d-XNA, and cocos2d-objc. Each branch is written in a different language, though they do try to keep some consistency.

LÖVE

love2d.org

LÖVE is an awesome framework written in Lua designed for making 2D games. The nice thing about LÖVE is that it's well-documented and has loads of tutorials, an active community, and there are a lot of cool libraries built on top of LÖVE to help you add features you feel are missing. The fact that LÖVE is written in Lua is also a big plus since it's an extremely popular scripting language in the game industry. So even if you later decide LÖVE isn't for you, you can still apply that Lua knowledge somewhere else.



OpenFL

openfl.org

OpenFL is a Flash API-compatible library designed for desktop, mobile and web. OpenFL is built on top of a library called Haxe, so if you feel like you want lower level access you should check that out. Also, since it's compatible with Flash, you can use popular Flash libraries such as Flixel and FlashPunk if you feel more

comfortable with them. OpenFL has been used in such titles as *rymdkapsel* and the IGF winner *Papers, Please*.

Phaser

phaser.io

Phaser is a HTML5 framework designed for desktop and mobile built on top of Pixi.js. Although Phaser is relatively new, it supports a lot of features—such as preloading, animations, tilemaps, sounds, physics, and WebGL rendering—that are not commonly found in most competing JavaScript frameworks.

SFML

sfml-dev.org

SFML is a C++ library similar to SDL but with additional features such as networking within the core library. Like SDL, SFML comes with additional third party libraries and also supports bindings to multiple other programming languages.

Polycode

polycode.org

Polycode is a free and open source framework created by Ivan Safrin. While it's been around for a long time, it only recently had its first official release. It's written in C++ and supports scripting via Lua and Javascript. It currently exports to Mac, Linux, and Windows, with Android, iOS, and Web support in the works.

The framework comes with an IDE which you can use to edit your scene, modify your game assets, or write Lua/Javascript code. The degree to which you utilize this IDE depends on your configuration of the framework.

If you're interested in a framework that gives you something similar to the functionality of Unity but also has the benefit of access to the source code, then Polycode might be for you.

Godot Engine

godotengine.org

Godot is a free and open source game engine created by Juan Linietsky and Ariel Manzur. The engine comes with an editor and a custom Python-like scripting language called GDScript. The scripting language was built from the ground up to work efficiently with the engine. Developers can also use C++ for performance critical features.

The Godot team have recently hit 2.0, so definitely check out their demos to see the latest features. It currently exports to Mac, Linux, Windows, Android, iOS, and Web.

Art/Graphics

Here are some free art tools that can be used to create various types of visual content for games.

Blender

blender.org

Blender is a cross-platform open source 3D graphics suite developed by the non-profit Blender Foundation in collaboration with hundreds of people around the world.

Blender supports modeling, animation, rigging, texture mapping, UV wrapping, compositing, morph targets, sculpting, simulations such as clothing and hair physics, and several other features. Any features you feel are missing can be added using Blender's Python API, and generally any custom features that become popular enough in the community end up being part of future Blender releases. The most recent well-known addition has been Cycles, providing Blender users a powerful new rendering engine.



There are plenty of books and online tutorials about Blender which show off some of its newer features. You can also check out projects made using Blender in the showcase reel on their website. (And see our article on low-poly 3D for more info as well.)

GIMP

gimp.org

GIMP is an open source raster graphics editor originally started by Spencer Kimball and Peter Mattis in 1995.

Today, it is developed by The GIMP Development Team and has gone to become one of the most popular graphics editors in the world.



GIMP supports various paint tools common to most raster graphics editors such as Photoshop. It also has support for animation, layers and channels, path tools, and quick masks, exports to many formats,

and can be enhanced via scripts and plugins. GIMP also supports Photoshop brushes, so you don't have to worry about looking for GIMP-specific brushes.

One of the most interesting things about GIMP is that because it's open source, it can be repackaged by its users with different default settings and plugins, which allows it to be configured for a more specific kind of task. This has led to the popularization of configurations like GIMP Paint Studio.

Inkscape

inkscape.org

Inkscape is a cross-platform open source vector graphics tool similar to Adobe Illustrator. It's designed to be used for the creation of icons, logos, diagrams, maps, web graphics, and much more. It does this through scalable vector graphics (SVGs), implementing an open standard set by the World Wide Web Consortium.



Inkscape supports advanced drawing tools, object manipulation, styling, text manipulation, as well as many other features. It can also export to several formats such as SVGZ, AI, PDF and PNG.

Probably one of the most useful things I've done with Inkscape is generate game terrain. This is extremely useful when creating the kind of terrain you see in games like World of Goo or Worms.

Top Alternatives for Art/Graphics

Krita

krita.org

Krita is an open source graphics editor originally started in 2005. Krita is similar in some respects to Paint Tool SAI, but thanks to their recent Kickstarter campaign, the latest 3.0 release also brings a new advanced animation system, interface improvements, as well as a host of other improvements.

Graphics Scale

humanbalance.net/gale/us/index.html

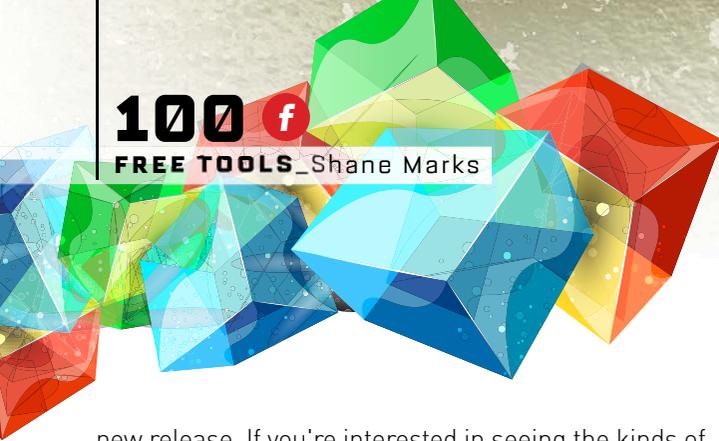
This is a popular tool designed specifically for pixel art. It supports all the features commonly found in pixel art tools, and while there is a pro version, chances are you won't need it since the free version supports all the commonly needed features.

Hexels

marmoset.co/hexels

Hexels is a grid-based art program that lets users paint with shapes. The specialised nature of this tool means that artists may have an easier time making their game look distinctive. Hexels has a newly-added animation system, layer duplication, and more, in its





new release. If you're interested in seeing the kinds of things possible with Hexels, check out madeinhexels.tumblr.com.

Piskel

piskelapp.com

This is an open source web-based pixel art tool. It supports layers, frames, previews, imports, and all the other basic functionality you'd expect from a pixel art program. One of the coolest things about Piskel is that because it's web-based, it allows users to create accounts and save their progress online. The most obvious advantage of this is being able to pick up where you left off without carrying the files around with you or worrying about your machine having the software installed.

Spriter

brashmonkey.com

Spriter is an animation tool created through Kickstarter backing. It allows you to set up advanced animation sequences using an editor without having to worry about which development environment you are using. It also allows an artist to create, preview, and tweak animations without the need of a programmer, which can remove a lot of the overhead in getting a game to look right. Spriter was used on Gunhouse by Necrosoft Games (run by this publication's editor Brandon Sheffield).

Audio

While there are plenty of free tools for development and art, for whatever reason I've always found it hardest to find good free audio tools. Here are some tools that I think are pretty useful for editing audio or creating music and sound effects.

Audacity

audacity.sourceforge.net

Audacity is a cross-platform open source audio editor and recording tool created by Dominic Mazzoni and Roger Dannenberg in 1999. Several years later, it's still being developed, with new improvements every release.



You can use Audacity to record live audio, record

computer playback, perform noise removal, add audio effects, convert, edit or mix multiple types of audio formats, as well as export to multiple formats. Audacity also supports plugins and allows custom effects to be made using a programming language called Nyquist.

A basic use for this software would be to remove lag in podcasts where the speakers are in different locations. This would be done by every participant recording their own audio stream using Audacity, and afterward stitching every audio stream together in the editor while accounting for the lag that happened while recording.

Another simple use for this software would be to create sound effects or splice up an audio track in case you need to remove parts of it to make it better loop in your game.

Bfxr

bfxr.net

Bfxr is a browser-based synth program used for generating sound effects. Bfxr is based on Sfxr, a tool that was written during the 10th Ludum Dare but has been extended over time with additional features.



More recently, Bfxr has added new waveforms and filters, expanded the pitch and jump parameters, as well as added a mixer and visualizer. It also allows for users to lock specific parameters during randomization and mutation for better control. Once you've created the sound effects you want, you can then download them as .WAV files.

Top Alternative Audio Tools

Audio.audiotool.com

Audio is a web-based audio workstation that can be used to create music for your game. It also supports midi devices, and since it works within a browser you don't have to worry about installing any kind of software. The only downside you might find is you have to publish your tracks to a public page. You still retain the copyright, but it does mean the tracks will be seen by everyone.

LLMS

lmms.sourceforge.net

LLMS is an open source cross-platform audio workstation that allows you to produce music. It supports midi devices, as well as a plugin system to extend the software.

Famitracker

famitracker.com

This is a tracker specially designed for creating NES/Famicom audio. This can be useful if you're interested in creating a game that's trying to be a throwback to that era, or if you're simply interested in seeing what you can produce with a more limited set of options.



MilkyTracker

milkytracker.org

Milky is a popular cross-platform tracker used to create sound effects and music tracks for games. Milky exports to the .mod format, which is supported by popular game engines such as Unity.

OpenMPT

openmpt.org

OpenMPT is a tracker for Windows which allows you to create music and effects for your game. OpenMPT also supports a plugin system which allows it to extend its functionality.

Misc

Finally, I've decided to throw in a few pieces of software I use on a day-to-day basis or have found to be useful at various times.

Document/Spreadsheets/Press

Google Docs

docs.google.com

This comes in pretty handy if you're trying to put together documents for your project and want to collaborate with someone else.

Libre Office

libreoffice.org

Libre Office is a useful free word processing software suite for when you'd rather store all your documents locally, or you know you're going to be without a connection and need to work on those spreadsheets!

presskit()

dopresskit.com

presskit() is a piece of software used to quickly and easily show off information and media about your game to press who might be interesting in finding out about your game! The only requirement is PHP on your server.

Differently

necrosoftgames.com/differently

Differently is another press kit (made by me!), which is similar in layout to presskit(), but the main difference is it doesn't require PHP and can be hosted just about anywhere since it's only serving static content. It also allows more flexibility in terms of changing subheads and moving images around, using plain HTML.



Source Control

Bitbucket

bitbucket.org

BitBucket provides Git and Mercurial source control, a wiki, and issue tracker. Free accounts are provided for up to five users, but after that you'll need to start paying.

Github

github.com

Github provides source control, a wiki, and issue tracking. Unfortunately, unlike BitBucket, all free accounts are publicly viewable unless you pay for a private account. However, a pretty neat feature in Github is Github Pages. These allow users to create a static website for the projects they host there. A lot of users use this feature to host their own blogs. You could do the same thing for your game website and save yourself some money.

Gitlab

gitlab.com

Gitlab is an open source project that provides Git source control as well as the other features provided by Bitbucket and Github.



GitLab

The main advantages of Gitlab are that it offers free unlimited private repositories and users, and allows you to install the software on your own private server if need be. This is the source control option I use for my own projects.

Project Management

There are many project management tools out there, but unfortunately most of them are bloated or filled with junk I don't think is particularly useful. Here are some free ones I've found myself going back to.



Slack

slack.com

This software is used for team communication. It allows users to separate out conversations into different channels or topics to help keep track of conversations and issues. It also allows users to hook in different pieces of software to send different kinds of notifications to teams. To those who have used IRC, it's essentially an IRC web front end with a helpful bot.



Trello

trello.com

Trello is used for creating, organizing, and tracking tasks. This really comes in handy when collaborating with other people.

Freedcamp

freedcamp.com

Freedcamp is designed for task management. The core features allow you to assign tasks and set up milestones and other significant events. It also provides a team the ability to discuss and iterate on a task and set up timelines for tasks on a shared calendar. The free plan can be extended with paid add-ons.

Shane Marks is a game developer from Ireland, who has worked on titles for console, desktop, and mobile. He is currently working at Necrosoft Games on a variety of projects. Contact him on twitter at [@d92008](https://twitter.com/d92008).

ANYONE
CAN
MAKE
GAMES

AT THE AU GAME LAB



**WE
MAKE
GAMES MATTER**



WANT TO MAKE
YOUR GAME
MATTER?

GAMELAB.AMERICAN.EDU

What makes them matter?

Certificate and master's programs that teach industry-standard coding skills backed by an emphasis on creating change.

4:1 student to faculty ratio means you work side-by-side with internationally recognized leaders.

Diversity that leads to innovation: Student age range 22 - 42; 58 percent female compared to 89 percent male for the industry.

Access to unusual and high-powered D.C.-based networks, from think tanks using games to government agencies studying play.



AMERICAN UNIVERSITY
WASHINGTON, D.C.

GAMASUTRA JOBS

YOUR GAME INDUSTRY CAREER RESOURCE



FIND THE MOST SOUGHT AFTER POSITIONS

with top companies around the world

PUT YOUR RESUME IN FRONT OF INDUSTRY

leading employers via the resumé database

SOURCE TALENT ON GAMASUTRA.COM

the game industry's thought leader

gamasutra.com/jobs



up because it'll be so obviously better than all of the other games there. Oops! That's a little embarrassing!

But I'll be gracious when I accept the prizes. I'll thank the judges and look the audience in the eye and say, "Hey, you too can be like me—you just gotta try!"

Keep the Fans Happy

Unlike a lot of other game developers, I know how to handle the online world. As long as I'm nice to my fans, they'll be nice back! I'll have a subreddit too, and that'll be a calm, rational place for everyone to share their anticipation for the game with insightful posts and reasoned discussion.

I'll make sure to attend lots of conventions, and I'll give away lots of swag because I love the community so much. When you give people stuff, guess what? They automatically love you back! I can just picture the long lines of people waiting to play my game and maybe even meet me in person and get my autograph. That is, if I decide to make a public appearance at all. I might get tired of all the constant attention and go incognito, as they say—posing as just another fan of my awesome game. Then, if I overhear someone bad-mouthing me, I could dramatically reveal I'm actually

right there, shocking them and making them realize the error of their ways! "Forgive us!" they'll shout. Man, it'll be so sweet when that happens.

Fame and Fortune!

Anyway, after that, I'll probably take a break for a bit and let others just sort of bask in the warm glow of my success. I'm very generous by nature and always happy to set an example, so when a prizewinning film crew contacts me to inquire about making a documentary about my life, I'll participate in a totally earnest and sincere way. I'm not completely sure exactly how I'll enjoy all that money just yet—I doubt I'd buy an opulent mansion in Hollywood or anything gauche like that. Maybe I'd consider an old castle in Scotland, though.

Greatness Awaits™

Well, that's enough talking. It's time to start doing the real hard work of making my indie game dreams come true! See you on Twitter!

Matthew Wasteland is the author of "Surviving the Game Industry: A Wasteland Guide." Find more of his work at magicalwasteland.com.

THE GAME STUDIO at CHAMPLAIN COLLEGE

STAY AHEAD OF THE GAME

Recognized by Princeton Review as one of the "Top 25 Video Game Schools and Programs," Champlain College brings game development education to the next level. Champlain College students work together in our Game Studio—a collaborative environment that mirrors the tight-knit teams of the game industry.

At Champlain, experience matters.
gamestudio.champlain.edu



LET US DARE