BREAKING INTO GAMEDEV

A Practical Game Development Manifesto and Comprehensive Guide on Staying in the Games Industry

Written By

STEVEN HARMON

BREAKING INTO GAMEDEV

WHY WE'VE SHARED THIS GUIDE

GaymerX's mission is to celebrate and uplift queer voices in gaming by fostering a safe and inclusive space. We do this through our conventions, our developer scholarships, our social media campaigns, and now by providing resources to help get more people into the games industry. A more diverse games industry is a better one!

WHO IS WRITING THIS GUIDE, AND CAN THEY BE TRUSTED?!?!

stevenharmongames.com

Steven Harmon, GaymerX's social media coordinator who also is an Independent Game Developer from Colorado with eight years of experience making games; six of those years self-taught and two years (so far) of college education in game design from the Princeton Review's No. 1 undergraduate game design program at USC. Steven has made and released over 50 games, some of which have been covered in PC Gamer, Polygon, Rock Paper Shotgun, and even gone off to get honorable mentions and other accolades at showcases like the Independent Game Festival. While by no means an expert, he'd like to share with you some of the things he's picked up in his time with games. Most books on game development are technical and become deprecated very quickly, and those that aren't, tend to be wrapped up in theory and academic jargon. Steven wanted to create a more comprehensive free guide to game development that would serve as a practical resource to revisit during the highs and lows of all stages of one's personal journey into gamedev.

ANYONE CAN MAKE A GAME

All you need to begin making your own (digital) games is a computer, an internet connection, a lot of patience with yourself when things get difficult, and passion to keep yourself motivated to bring your games to completion. Game development can be as frustrating as it is rewarding, but this guide is here to help those interested but hesitant in taking their first steps towards making games with practical resources, as well as more in-depth info and perspective. I do recommend taking this entire guide with a grain of salt, because it's opinionated and from my own point of view and experiences. Read with skepticism as the one universal truth of game development is: *NOBODY KNOWS WHAT THEY ARE DOING*. It's a new, unexplored frontier and artistic medium with no one right way to do it.

TABLE OF CONTENTS

0. STARTING OUT

An Idea

Everyone starts somewhere and brings something interesting to the table. So, step one is to come up with an idea for a game. It doesn't need to be super complex or original, it just needs to be small. Your first game shouldn't be the next open world procedurally-generated ESPORTS space RPG MMO. If that's what you want to make, then you've already set yourself up for failure – and not the good kind of learning failure that I'm about to set you up for momentarily.

Don't think too hard about it, really.

As Brendon Chung, developer of *Thirty Flights of Loving* and *Quadrilateral Cowboy*, described on his blog post "Hello World";

"Someone smarter than me once described game development as jumping out of an airplane with nothing but a needle and a silkworm [...] When the ground is rushing toward you at a million miles per hour, what's important? You make something. People can't play a design document. People can't play a grand vision. People can't play all the cool ideas you've planned out down the road. People can play the game you make."

Okay, so I have an Idea! What next?

Once you have an idea, you need tools to make that idea into a reality, and unless you're already a programming wizard, then don't try rolling out your own. No need reinventing the wheel at this point in your career. What I suggest is reading Chapter1 of this guide "The Big List of Game Engines" to choose a game engine that sounds right for you, and skimming each of the skill chapters (Programming, Art, Sound, etc.) "tools" section for what you'd like to use alongside it. Or you can use this handy interactive tool by Zoë Quinn to just walk you through that process of choosing a game engine, and then skim each of the chapters' "tools" sections.

Tools! Gotcha! Alright, what now?

Make a game, finish it, come back, and read the rest of this guide.

Are you serious? I thought this was supposed to be the most comprehensive guide to gamedev out there?!?! You can't just skip from part A to Z!

Well this video might help.

Just consider:

1. What's it gonna look like?
2. What's it gonna sound like?
3. What's it gonna play like?

And then start prototyping the game! Whether it be greyboxing out levels with cubes and other basic geometry in the 3D editor view of your engine, sketching out

2D levels onto grid paper, or coding a box (your player) moving around, colliding with other things, triggering events, and so forth until you have a basic playable game.

Well that isn't very helpful.

Read the rest of this guide and I promise you'll find helpful additional resources that go more in-depth on specific parts that may be confusing to you. Other than that, games are an artform and there's no one way to make them. Games are such a new medium and uncharted ground that there's a lot of freedom, so embrace it! Lean into what you're already good at and challenge yourself where you aren't. Don't expect to make your masterpiece or dream game right away, just save it for when the time is right and when you have enough work and resources under your belt to take it on. Focus on what you can do in the moment and try creative solutions to make bugs look like features. Fake it till you make it, trust me – everyone doing creative work does this whether they admit to it or not.

How do I find resources on my own?

Google or any search engine will do. Odds are your simple game mechanic idea for your first game will have already been done before and have documentation online on a forum or blog post. Sites like Stack Overflow are where you can find and ask questions to get where you want to go. But before posting a new question, ask yourself these questions.

1. Has someone already asked this question or a question similar to it previously?
2. If so, was that question already answered?
3. Am I expecting someone to write code or do work for me for free?

If you answered "YES" to any of those questions, then rethink your question before posting. People are only going to be willing to help you if you're willing to help yourself. Showing them that you've already attempted and done your own research into the error codes and bugs will increase your chances of getting a response. Be nice. The community wants you to succeed, but make sure you are asking the right question to get the right response.

1. THE BIG LIST OF GAME ENGINES

Picking the right tool for the job

Every game runs on an “engine” which basically determines what a game made in that engine can and cannot do. All engines have things they do well and other things that they do poorly. The tools of each creator shape the work they make, but at the end of the day there’s no perfect tool. There have been many instances where developers stretch the bounds of the tools they’re using. With that in mind, here’s an in-depth list of the most popular engines for indies, as well as a more general table of some of the more obscure ones.

Full Disclosure: I’m a Unity Student Ambassador so I am biased. However, I’ll do my best to give you an impartial overview of all the engines I’ve had exposure to or experience in.

UNITY

https://unity3d.com/

Price:
Personal Edition (Free), Plus & Pro (Paid)

Available for:
Windows, Mac, Linux (Experimental)

Exports to:
Desktop, Console, Mobile, Web, TV, etc.

Learning Curve:
A little steep, but it’s flexible with editor tools in the Unity Asset Store, has a fantastic community, and there’s tons of free learning resources out there.

Resources:
Unity Documentation – Great examples of code, but lacking in some parts.

Unity Answers – Ask questions and get help, super responsive & helpful community.

Brackeys – Top-notch tutorials, discord community, up to date with latest changes

Sebastian Lague – Wonderful tutorials on not just Unity, but Blender and more…

Making Stuff Look Good with Unity – Shader programming & technical art tutorials

Official Tutorials – Tutorials by Unity with open source downloadable projects

Board To Bits Games – Non-gamedev tutorials (menus, saving, ui, etc.)

Devin Curry – Mobile-focused tutorial series, miscellaneous Unity tutorials

Quill 18 Creates – Great multiplayer focused Unity tutorials

UNREAL

https://www.unrealengine.com/

Price:

Free (5% royalty on gross revenue after the first $3,000 per product, per quarter)

Available for:

Windows, Mac

Exports to:

Desktop, console, mobile, etc.

Learning Curve:

Similar to Unity, perhaps a little easier with blueprints, the visual scripting solution Unreal uses. Plenty of online resources and tutorials with an active community.

Resources:

Unreal Docs – Good documentation with quick starts for different specializations

Unreal Forum – Threads with active community

Official Unreal Tutorials – Video tutorials covering a variety of topics

VirtusEdu – Multiple playlists of top-tier tutorials for Unreal

Raywenderlich (Tommy Tran) – Written step-by-step tutorials for Unreal

GAMEMAKER STUDIO 2

https://www.yoyogames.com/

Price:
$40 - $1,500 (Different plans, sometimes on sale)

Available for:
Windows, Mac

Exports to:
Desktop, console, mobile, etc.

Learning Curve:
Very easy. Takes an afternoon to get an understanding of the basics to make a game. Whether you use GameMaker's visual scripting or GML language, it'll be easy to learn. Well-documented with tons of tutorials.

Notes:
I've only used, the now deprecated GameMakerStudio 1.4. However, at a glance both editors are similar, and the new GameMakerStudio 2 comes with some great new features.

Resources:
GameMaker Docs – Includes informative pictures which are helpful

GameMaker Learn – Official tutorials by YoYo Games

GameMaker Forum – Active community with threads for more resources

Shaun Spalding – Up-to-date, high quality GameMaker tutorials

TWINE

http://twinery.org

Price:

Free (Open Source)

Available for:
Windows, Mac, Linux, Web

Exports to:
Web (HTML), offline executable with Node WebKit

Learning Curve:
Syntax is close to English, very simple, takes minutes to learn. The easiest tool for making games out there.

Note:
Mostly used for choose-your-own-adventure hypertext games, visual novels, and role-playing games (with some programming).

Resources:
Twine Cookbook – Git repo with examples for all story formats

Twine Q&A – Ask questions, get answers. What isn't to love?

Twine Wiki – Wiki page with everything Twine from Twine 1.4 to 2 complete guides

Twine Discord – Talk directly to other Twine developers and get help and advice

Twine Stylesheet Examples – Make your Twine games look pretty with CSS

THE BIG LIST

And an even bigger list

	Price	Available for	Exports to	Notable Games	Learning Curve	Notes
Unity	Free & Paid	Windows, Mac, Linux	Desktop, console, mobile, web, etc.	So many...	A little steep	Very flexible, multiplatform made easy
Unreal	Free + 5% royalty	Windows, Mac	Desktop, console, mobile, web, etc.	Hellblade, Fortnight, Rime, a lot.	Similar to Unity's	Needs good hardware, tons of tools inside
GameMaker	Paid	Windows, Mac	Desktop, console, mobile, web, etc.	Hotline Miami, HLD, many more!	Easy to learn, fun to master	Great 2D tools, drag & drop programming & normal scripting
Twine	Free	Windows, Mac, Linux, Web	Web, desktop (with addon)	Queers in Love at the End of the World, Temple of No	The easiest to learn, no tutorial even necessary	[[Text | PassageTitle]] It's that simple.
Ren'Py	Free	Windows, Mac, Linux, Android	Desktop, mobile, web	**Butterfly Soup**	Slightly more difficult than Twine	Quickstart guide and comprehensive documentation
RPGMaker	Paid	Windows	Windows	Always Sometimes Monsters, Mainichi	Average learning curve	Genre specific to top down RPGs
Adventure Game Studio	Free	Windows	Windows	Ben There, Dan That, The Cat Lady	Similar to RPG Maker	Genre specific to point & click games
Amazon Lumberyard	Free & Paid	Windows	Desktop, console, web	Star Citizen	Difficult just because it's new	First Twitch native engine, multiplayer with aws
Godot	Free	Windows, Mac, Linux	Desktop, mobile, web	... It's new	Nice docs, but new	Open source engine with some promise

2. COMMON PITFALLS OF DEVELOPMENT (AND HOW TO AVOID THEM)

My big design doc

Some people are idea people, and those people aren't game developers. People who make games are game developers. Game developers can have ideas and people with ideas can make games, but no matter how much you plan on paper, that design document isn't a playable game. Not to say that design docs are useless, they're incredibly important for teams to get on the same page; consider that the game itself is the most important living, breathing, and constantly evolving design document.

My homemade game engine

You don't need a deep understanding of how engines work from the inside to be able to make a game within one. I've seen talented aspiring game developers quit on game development with having no finished games to show after attempting to write their own engine at the get go. It's not the 90's anymore and we don't need to roll our own tech if we don't have to. There are plenty of amazing 3rd party open source or free-to-use tools out there, so there's no reason to try reinventing the wheel.

Prototyping!

To be a game developer, you need to have made and finished a game. Not a demo or proof-of-concept, but a finished product (commercial or not) that people can play. It's great to prototype all your ideas, but save those ideas for the next game, jot them down in a journal, and finish your current game before moving onto another one. It's okay to be working on multiple projects at once, but do so when you've gotten a few finished projects to your name. Practice finishing games, because it is a valuable skill that separates hobbyists from professionals.

Feature Creep

Once you're able to make just about anything you can think of, you'll want to make it – all of it — and you will never stop. So, save it for the sequel! Always consider the scope of your game. If the game is not good in its purest form, then adding set dressing and polish won't fix the inherent underlying issues nor make it much better. Perfection rarely comes from adding more stuff to something.

I've learned so much, time for a redo!

Your code or art will never be perfect. If you're growing during a project, you're doing something right. Long-term multi-year projects are incredibly prone to restarts but resist the urge. If your game is so complex or disorganized that it's slowing production to a crawl, then refactor your code and move on – but don't completely restart! Trust me, I've been there and done that… multiple times. "Duct tape" code is working code! It's fine, seriously.

Too many cooks

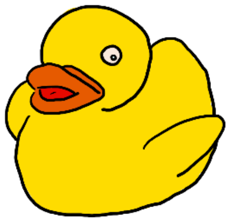
Everyone is enthusiastic at the start. However, enthusiasm is expendable. What you want in a teammate is grit. Someone who sticks with it to the end, even when things get tough. Finding a good team member is like finding a soul mate. You want someone who works as hard as you do, is as invested in the project as you are, is honest with you, and has skill that compliments yours. You want teammates who have experience with consistently finishing games, and likewise no professional will want to work with someone who has never finished anything. Making a game doesn't require a team, and assembling one isn't a guarantee of a finished game.

Commonly given advice is to find "T-Shaped" people, people who have a vast knowledge but specialize in one aspect of development, but for smaller teams it's important to be able to wear many hats and be able to switch them around for the task at hand.

My advice is to be able to make a game entirely by yourself starting out, then shop around for team members you trust and are comfortable with. If you're the lead of a project, front the code yourself! So, if people leave the show will go on. If you must start a team, have an uncomfortable conversation about IP (intellectual property) ownership, money split (don't get your hopes too high), and all the unfun things of game development at the start. What you don't want is a messy divorce that ends with the game never being released due to legal reasons.

The wall

When running, marathons runners mention a point in the marathon where they hit "a wall". Not a literal wall, mind you, but a problem where you stop and can't progress. This wall can be a bug, a platform specific API with incomplete documentation, or publishing contracts with tons of legal jargon and no explanation or reference point. Regardless, it needs to be dealt with and the longer you try avoiding it, the more you will resent the project and yourself. So just do it! Don't let your dreams only be dreams!



Other solutions include: taking a small break to gain a fresh perspective, talking to a rubber ducky about your problems, and (if possible) scoping down and removing troublesome features.

3. PROGRAMMING

Without programming nothing happens

Programming, whether or not you consider yourself a programmer, is necessary. Yes, you can make games without programming (in the traditional sense). There are tools that require minimal coding, especially for visual novels and hypertext fiction. However, the more you use these tools the more you will realize how limited they are and want more functionality out of them. You don't need to be a math genius to be a decent programmer, but if you'd like to touch up on your math skills, here are some places to do that.

Khan Academy – Tutorial videos and interactive quizzes to test your knowledge

Immersive Linear Algebra – Interactive online textbook

Math for game developers – Fantastic series on practical application of math

TOOLS

Most all code is written in something called an IDE (Integrated development environment) and all it essentially does is process the code you write and convert it to binary (1's and 0's aka computer language) to run. There's also a suite of features such as a debugger, which is pretty much a very advanced version of Spell Check to let you know exactly where you went wrong in writing your code. Don't worry about choosing an IDE, as most engines come with one pre-packaged. However, here are a few of my favorites.

	Description
Visual Studio 2017	The golden standard, but perhaps a tad bloated and overkill depending on the scale of your project. Free version called Visual Studio Code.
Notepad++	Open-source text editor with basic features, clean, and fast. Should be installed on every computer.
MonoDevelop	C# multiplatform IDE. No frills, but it works.
Eclipse	Java specific, but good for what it is.

For more text editors with programming functionality, click here.

EVERYTHING THEY TEACH YOU AT COLLEGE

Okay, so not everything, but as far as games go. C++ is what is taught. However, I do not recommend learning C++ as your first programming language. You can always revisit C++ later when you're ready and more comfortable with general game scripting, because C++ is more "low level", meaning it's closer to the hardware which also means more things can go wrong if you're not careful. Also, things in C++ like dynamic memory allocation are confusing for beginners and not required in making a game. If you're just starting out, C# with Unity and Blueprints in Unreal are great choices. The tools or languages don't matter, just focus on the logic and algorithms of programming that will carry over when you eventually switch languages for a new project.

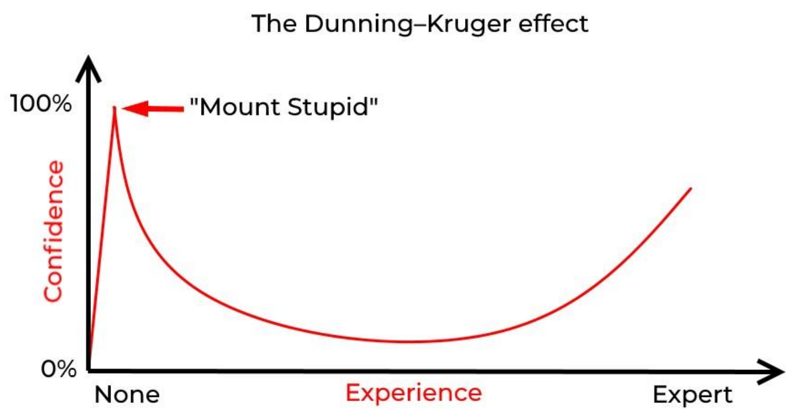
EVERYTHING THEY DON'T TEACH YOU AT COLLEGE

You don't need to go to college to learn how to program. In fact, learning how to program may be easier and more effective if you're doing it on your own at your own pace. Only trouble is there's one very bad habit that will prevent you from learning anything.

DO NOT COPY AND PASTE CODE BEFORE YOU UNDERSTAND IT!

You are only doing yourself a disservice if you don't know how to read and interpret each character and command in the language you are programming in. Open-source projects and online forum code are great resources and should be studied, but not used as a crutch. Get to know the language by reading up on it, watching tutorials online, and trial and error with debugging by printing out messages like "I'm working" to see if that part of the code was called or not.

The most important truth of programming is that you'll feel like the dumbest and smartest person in the whole world, and you'll alternate between the two quickly. It's a humbling practice and anyone who thinks they're an expert at it – probably isn't. Just be honest with yourself and your abilities because you can't learn what you think you already know.



CONGRATULATIONS! YOU NOW HAVE THE SUM TOTAL OF ALL HUMAN KNOWLEDGE

Now here are some great resources to learning cs and programming!
Computerphile – Great online informative videos, just enjoyable to watch

The New Boston – The Holy Grail of programming tutorial series by Bucky

Brackey's C# series – C#, but simple and beginner-friendly

4. ART

TOOLS

MAYA

https://www.autodesk.com/products/maya

Price:
Student Edition (Free), Professional Edition (Paid – EXPENSIVE)

Available for:
Windows, Mac, Linux

Notes:
Industry standard, intuitive UI and navigation, most commonly used 3D software package out there for games and animation.

ALLEGORITHMIC SUITE

https://www.substance3d.com/

Price:

Student Trial (Free), Professional Edition (see notes below)

Available for:

Windows, Mac, Linux

Notes:

From creating procedural materials and PBR maps to 3D painting. No other program comes close... only downside is Adobe recently acquired Allegorithmic so perpetual licenses may become void and the software may be moved to a subscription model.

BLENDER

https://www.blender.org/

Price:

Free

Available for:

Windows, Mac, Linux

Notes:

Open-source, free to use for commercial or non-commercial projects, great online community, original UI clunky and weird... but the recent 2.8 update remedies that. Some features behind the industry standard, others are ahead of the industry.

SKETCHUP

https://www.sketchup.com/

Price:

Free

Available for:

Windows, Mac, Web

Notes:

Intended for architectural planning and visualization, but the easiest tool for 3D modeling with a wide array of plugins to make it a viable tool for game development. While not well-optimized in terms of output for final models, it's the best software for greyboxing 3D level geometry for design.

ZBRUSH & SCULPTRIS

http://pixologic.com/

Price:
Z Brush: Paid 40/month – Sculptris: Free

Available for:
Windows, Mac

Notes:
Digital sculpting standard suites for high poly detailing and retopology

MAKE HUMAN

http://www.makehumancommunity.org/

Price:
Free

Available for:
Windows, Mac

Notes:
If character modeling isn't your thing, this is a useful tool in generating some realistic looking naked people.

AUTODESK FBX CONVERTER

https://www.autodesk.com/developer-network/platform-technologies/fbx-converter-archives

Price:
Free

Available for:
Windows, Mac

Notes:
If you're using SketchUp or any other software, the export options may not be compatible with certain game engines. Autodesk's FBX converter converts models from various file extensions to more widely-used ones.

3DF ZEPHYR

https://www.3dflow.net/

Price:

Free Version (limited) & Paid Version

Available for:

Windows

Notes:

Photogrammetry tool. Photogrammetry is high-quality 3D models captured from the real world by taking a lot of photos from slightly different angles around said objects. Models need manual optimization afterward in external software.

PAINT.NET

https://www.getpaint.net/

Price:

Donationware

Available for:

Windows

Notes:

The best photo editing software with the ease of use of paint, but with the features of Photoshop (with addons & extensions). A must-install for everyone.

GIMP

https://www.gimp.org

Price:

Free

Available for:

Windows, Mac, Linux

Notes:

High quality and professional open-source suite of photo editing and digital art tools with a high learning curve but great results.

INKSCAPE

https://inkscape.org/en/

Price:

Free

Available for:
Windows, Mac, Linux

Notes:
Fantastic vector art open-source suite. Vector art is images that don't lose quality when scaled, unlike the more common raster images which have a set resolution. Great for graphic design, print, and user interface elements.

ASEPRITE

https://www.aseprite.org/

Price:
$15

Available for:
Windows, Mac, Linux

Notes:
Fantastic pixel art suite with great animation features.

PYXELEDIT

https://pyxeledit.com

Price:
Free & Paid Version ($10)

Available for:
Windows, Mac, Linux

Notes:
Great tileset features & intuitive UI. Free version to use and if you like it, spend the $10 to get the updated version.

PUREREF

https://www.pureref.com/index.php

Price:
Donationware

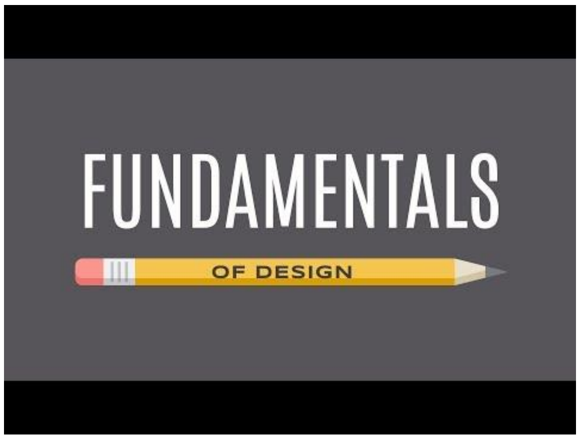
Available for:
Windows, Mac, Linux

Notes:
Easily organize and view reference images. Goes great with a second monitor!

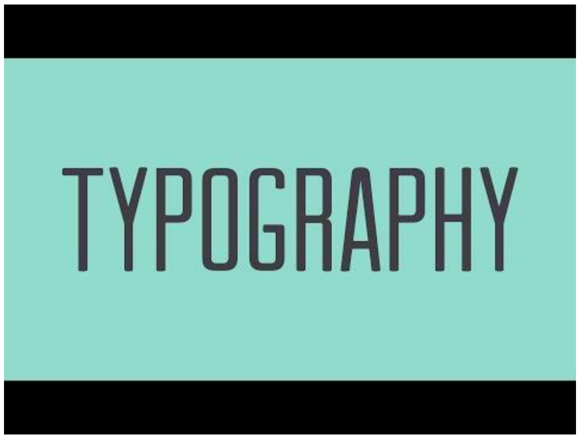
GRAPHIC DESIGN IS MY PASSION

The Basics

The use of images in games is to convey information and the way we do that is through design. Life is chaos and art is order. We must be selective and intentional in our artistic choices. However, to do that you need a vocabulary. A basic understanding of the fundamentals of art – how to make things look pretty, readable, and coherent. Here are some videos that cover the basics and spark the imagination. Most important thing is that you practice – art is a muscle, exercise it!



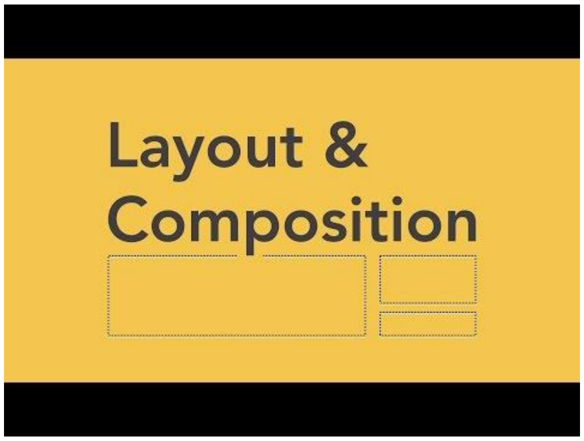
https://youtu.be/YqQx75OPRa0



https://youtu.be/sByzHoiYFX0

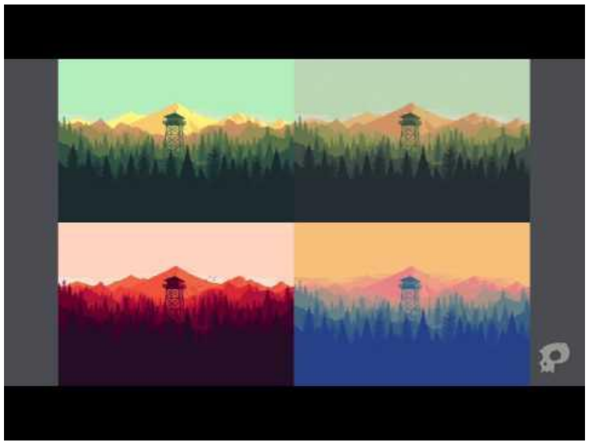
https://youtu.be/_2LLXnUdUIc

https://youtu.be/aXgFcNUWqX0

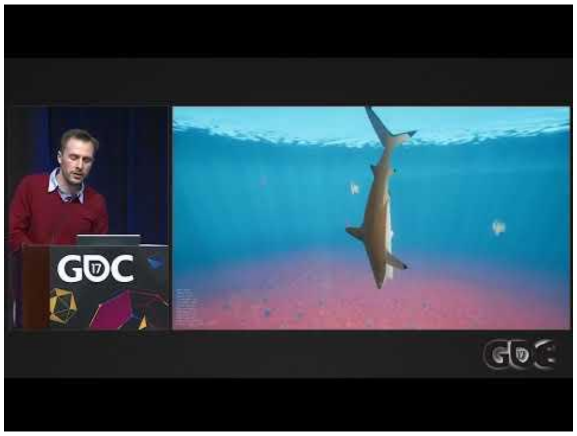


https://youtu.be/a5KYlHNKQB8

https://youtu.be/CvLQJReDhic



https://youtu.be/ZYnS3kKTcGg



https://youtu.be/I9NX06mvp2E

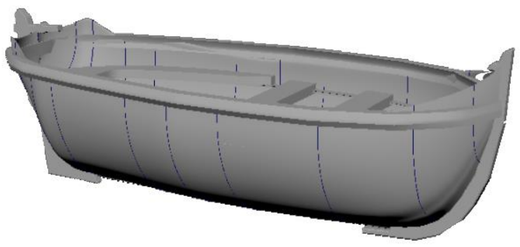
AESTHETICS

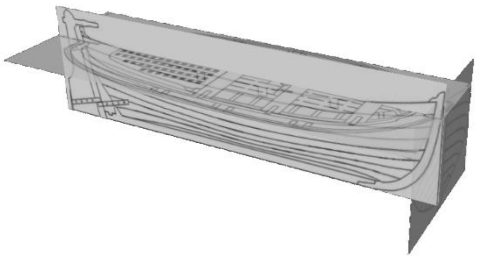
What is beautiful to you?

Take a moment out of your day every day to appreciate the beauty that surrounds you. Breath slowly, forget everything, and just take it all in. Your surroundings are an endless wealth of visual inspiration, so don't waste it. Take photos, play with color correction and filters, hone in on your aesthetic taste and channel it in your art!

HOW TO IMPROVE WITHOUT REALLY TRYING

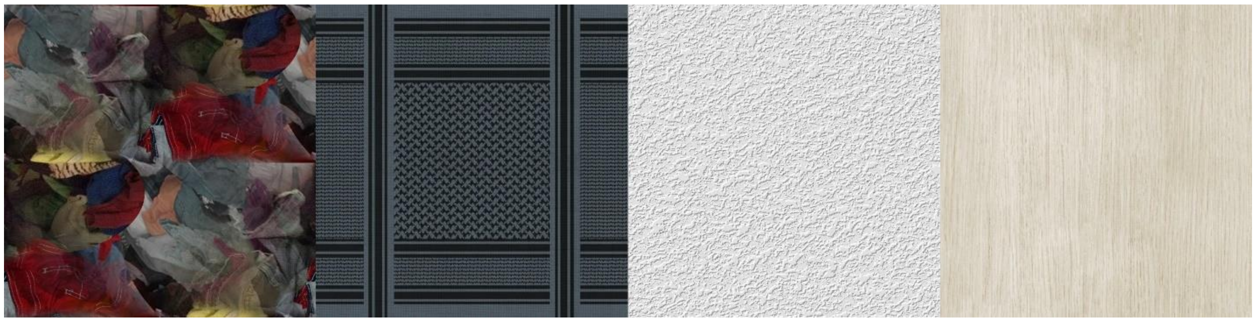
Use References





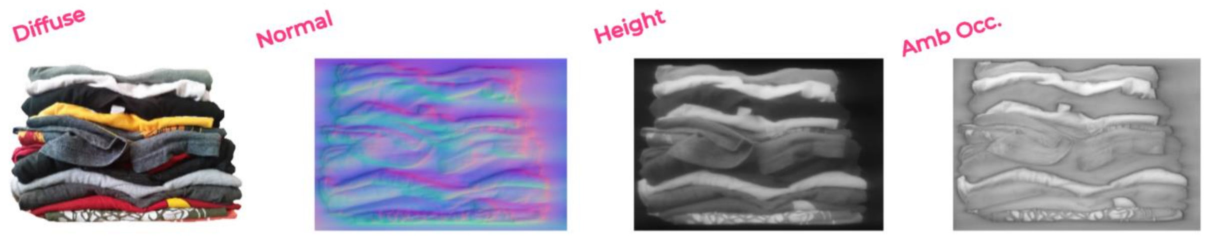
It's not cheating if you use a reference image when creating art. It's especially helpful when you're creating something you have little-to-no experience with to understand the shape and features of what you're modeling/drawing. Model/draw with a reference starting off, and once you've worked on that object long enough, you'll be able to create it from memory. If you have more than one monitor, there's a handy donationware program called PureRef for displaying reference images. When modeling in 3D, create intersecting image planes and use the orthographic views to construct the right shape of the model then add detail by hand. By using image planes in your pipeline, you'll speed up your process for modeling objects you have little working knowledge of.

Good Textures



A good seamless image texture properly UV wrapped around the model can make a good 3D model into a great one. Older-generation games with less-capable hardware weren't able to handle as many polygons as newer tech is today. However, artists back then used textures to make up the difference. A well textured scene has a better chance of not looking outdated.

Detail Maps

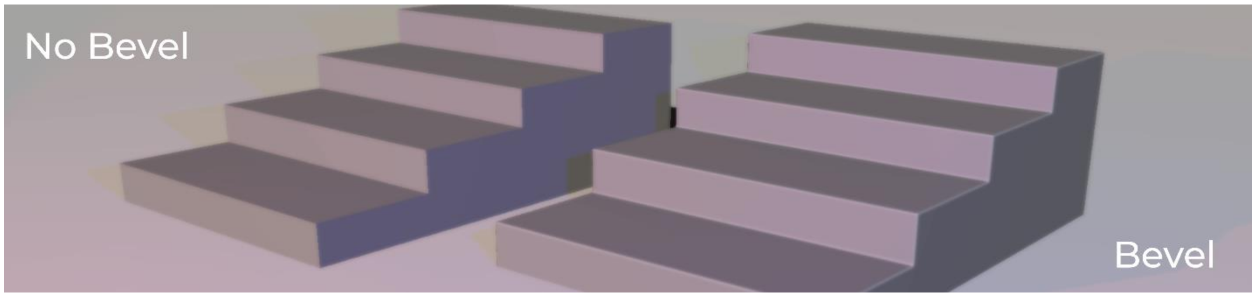


With the advancements of Physically Based Rendering materials with high-quality light bouncing and attributes such as roughness, metallic, and specular, we're able to achieve uncanny levels of realism and polish to our art. While applications like Substance Designer and Substance Painter are quite expensive for beginners, similar results can be achieved with more inexpensive and/or free software, such as MindTex 2 and Gimp.

Visible Detail

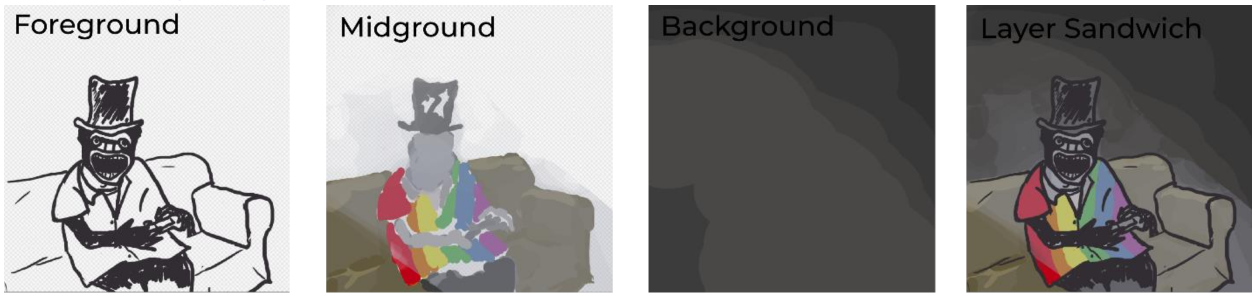
Art takes time, and if you're spending a lot of time on something that not many will be able to see or appreciate, you should probably shift your focus on what the big picture. However, that isn't to say you shouldn't leave Easter eggs for your players to incentivize the completionists and make their playthrough something special. You only need to add detail for what the camera will see.

Bevel & Smooth, But Not Too Early!!!



Perfect straight edges rarely exist in nature, therefore rounded edges go a far way to making something look more believable. However, it's smart to develop the shape and silhouette of the model first before going higher poly. Likewise, it's not a good idea to jump from high to low poly as seen in Ethan Redd's talk "Low Poly Modeling: Style Through Economy". This goes for 2D art as well, start with little detail and add more as you go along.

Edit History / Layers



If you're working in a primarily digital medium it's important to make sure your creation process isn't destructive, meaning that work isn't overwritten and lost. For 2D art, be sure to work within layers (background, foreground, detail, etc.), as making a mistake only needs to be corrected on one aspect of the piece as opposed to the whole.

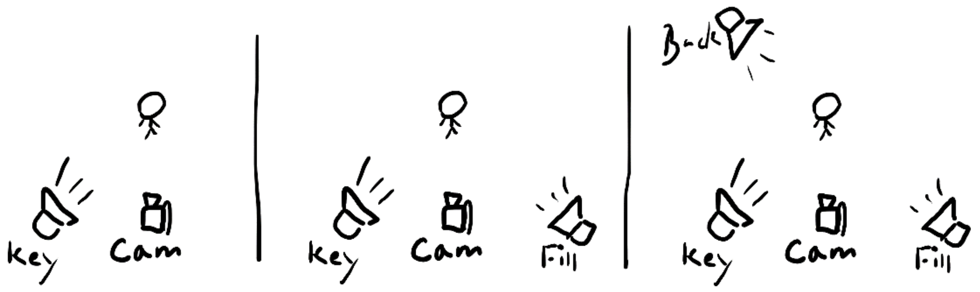
Toilet.mb	2/22/2018 12:57 PM
Toiletv2.mb	2/22/2018 1:56 PM
Toiletv3.mb	2/22/2018 4:26 PM

For 3D modeling and computer graphics, it's important to save often and keep your revisions, guidelines, and reference images in the scene until the final render. Often if you're working with a team or for a client, you'll get feedback and need to make major revisions to the model, so it's helpful to be able to do so from a checkpoint and iterate fast instead of starting over from scratch.

Also, modern art software, as advanced as it may be, can crash and lose data often. It's important to keep a revision history in case your project file becomes corrupt and unusable. This is less likely to happen if you open the software package first, then open the project file through the file/open tab, as opposed to opening it directly by double clicking the file itself in your file explorer. It takes a few extra seconds, but better safe than sorry.

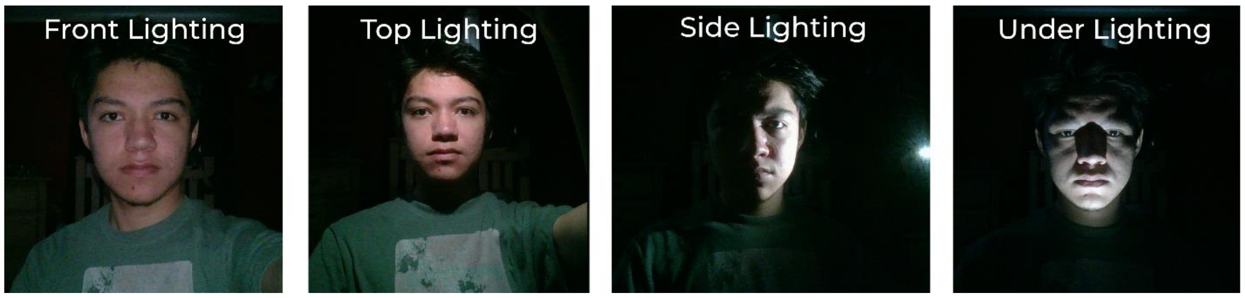
Lighting & Fog

The best lighting design is not noticed. The job of light is to make sure that the overall picture is visible, reinforcing the dramatic elements (challenge, play, story, character arcs, etc.), mise en scene (setting the scene in terms of ambience – the surroundings and mood – how the player feels about their surroundings), and guiding the player's attention to what's important, both navigationally and narratively.



Generally, the most effective lighting is 3-point lighting. The "key light" is the primary and strongest light of the three. The "fill light" is there to soften and fill in shadows from the key light. And finally, the "back light" is there to provide highlight to the subject(s) being lit and separate them from their background for depth.

Here's a quick overview of how the placement of a single light can change the context of the image.

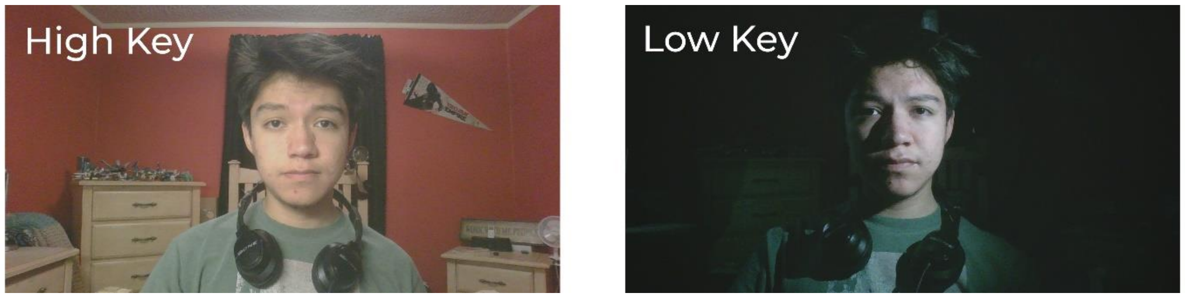


Front Lighting – Removes shadows, but bleaches out the subject

Top Lighting – For glamour, brings out cheekbones but hides eyes

Side Lighting – Brings out structure and sculpts features; mystery

Under Lighting – Horror; realistic light source



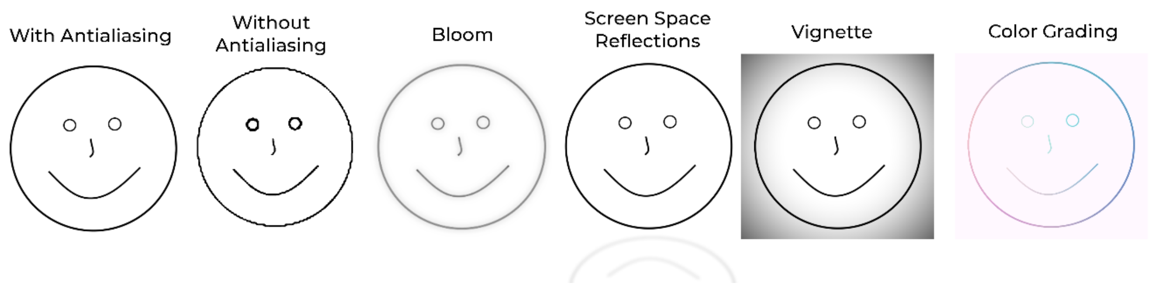
Low Key – High contrast, lots of key light not much fill (if any) OR lots of fill and not much of key light; shadows focused. Used for suspense, horror, mystery, and noir.

High Key Lighting – Low contrast, lots of key and fill light; balanced lighting, everything is well lit. Used for most situations.

While good lighting can transform a dull environment to a moody or atmospheric one, performance must also be considered. Too many real-time lights are costly and can slow down your game. As a general principle, it's wise to limit yourself to a single real-time light with shadows and bake the rest into the model's UV textures, or use projectors to fake shadows and lights.

Fog can also be used to save on performance in regard to saving on rendering things far in the distance, but also can be applied stylistically to evoke suspense as seen in *Silent Hill* or for beauty as in *Firewatch*.

Post Processing Effects



Finally, post-processing effects such as anti-aliasing, ambient occlusion, bloom, vignette, screen space reflections, and color grading can improve the visual fidelity of the image quality when used correctly and in moderation. A big mistake many beginners make is overdoing it on the effects without considering performance and optimization. Less is best.

HOW TO ACTUALLY IMPROVE

Only through repeated practice and time will you develop your skills as an artist. While some have innate talent, talent will never beat hard work and persistence. Your art skills are a muscle that must be worked out on a regular basis to develop the muscle memory needed to not to only imitate but originate.

Carry a sketchbook around

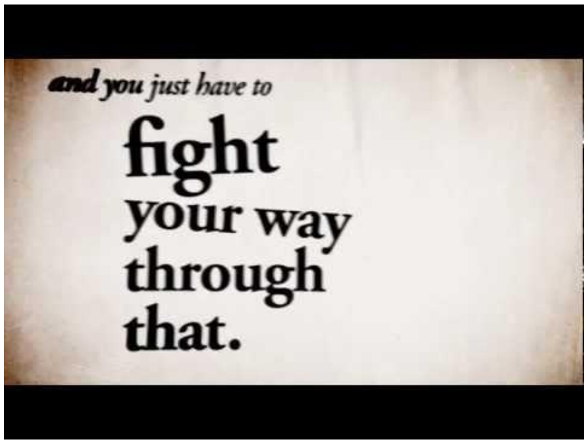
It doesn't need to be a fancy sketchbook or journal, loose-leaf printer paper and a flat surface or an average notebook will do perfectly fine. All you need to do is just draw and draw a lot. Here are some helpful videos on how to improve your art.

https://youtu.be/Bu3uIVhO3z4

https://youtu.be/orgVs2csqbo

Practice makes permanent

While these videos focus on how to practice and improve your 2D art, you can still apply the same practicing tips and techniques in your three-dimensional work as well. All you need to do is practice, practice, practice! So, here are some inspirational videos to keep you motivated.



https://vimeo.com/24715531

https://youtu.be/H14bBuluwB8

https://youtu.be/h48hGHALFC4

5. ANIMATION

TOOLS

Most of the 3D modeling suites such as Maya or Blender have animation already built in, but here are some more niche animation software and hardware.

MOTION BUILDER

https://www.autodesk.com/products/motionbuilder/overview

Price:
Student Trial (Free), Paid (Expensive)

Available for:
Windows, Mac, Linux

Notes:
Can characterize and generate IK and FK rigs to retarget and animate characters with or without motion capture data. A powerful story timeline editor to create cinematics and looping animated cycles.

FIRE ALPACA

http://firealpaca.com/

Price:
Free Version (with ads), Paid $40 (no ads, faster)

Available for:
Windows, Mac

Notes:
Reliable and bug-free drawing software, tablet-friendly, easy to learn, with animation features such as onion skin, and plenty of online tutorials. Export gifs

AVI4BMP

http://www.bottomap.com/Software/A4B/A4B.html

Price:
Free

Available for:
Windows

Notes:
Imports gifs and video and exports an optimized game engine ready sprite sheet of rows and columns. Great pairing with Fire Alpaca!

PAINT OF PERSIA

https://dunin.itch.io/ptop
Price:
Pay What You Want

Available for:
Windows, Mac

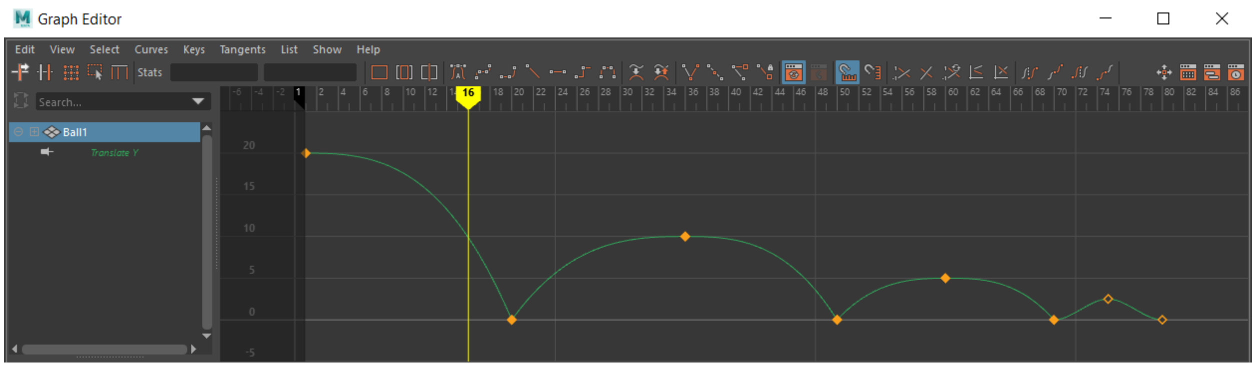
Notes:
Rotoscoping pixel art tool where you can draw over video frame by frame. Can export single frames or sprite sheet. Takes time but produces nice results.

KINECT V2, ADAPTER, AND CAMERA TRIPOD (HOMEMADE MOCAP)

It won't beat a professional multi camera motion-capture studio and body suits, but it works well for indies and hobbyists. It's expensive, but... mocap!!! Here's a tutorial on how to set up your Kinect for mocap in Blender.

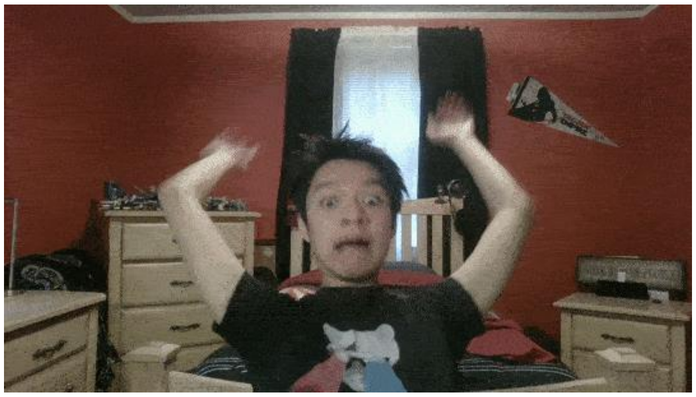
THE 12 PRINCIPLES

Frank Thomas and Ollie Johnston's 12 principles of animation from the 1981 book *The Illusion of Life: Disney Animation* is pretty much the bible for animation. While I could provide examples of each myself, AlanBeckerTutorials has a series of tutorials that showcase each principle perfectly. Additionally, Issara Willenskomer wrote a great piece on Medium showing how these principles can apply to UX design. The most up-to-date animation resource is a YouTube channel *The AniMates*, which is a fantastic resource for newcomer 3D animators, with talks and live animation from industry experts on critique, staging, motion capture, motivation, and much more!



After watching the Alan Becker video try practicing by animating bouncing balls, each with different physical properties, or a looping walk cycle for a character. When using references, try to only use real-life ones as opposed to another animator's work. Animation is an exaggeration and characterization of real-life motion so always pull straight from the source – pull from real life.

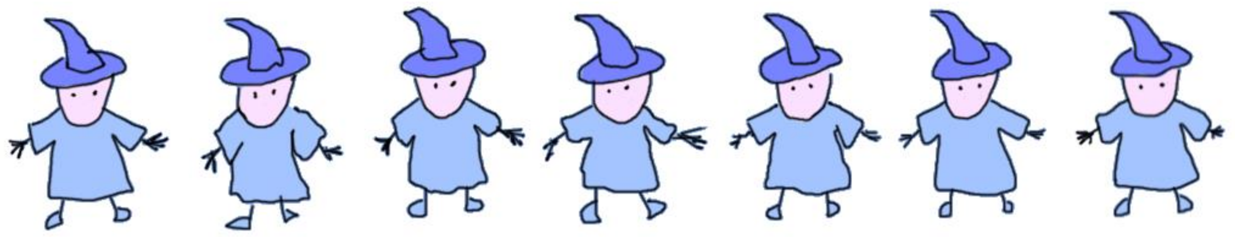
Don’t be afraid to look stupid



A great tool in your roster as an animator is your own body. It may feel silly but try to replicate the motion you’re trying to animate. So much of animation is understanding of how bodies move… so move. Use your webcam to record yourself acting out the animation and if you have another monitor, play it on loop in slow motion while you recreate the motion.

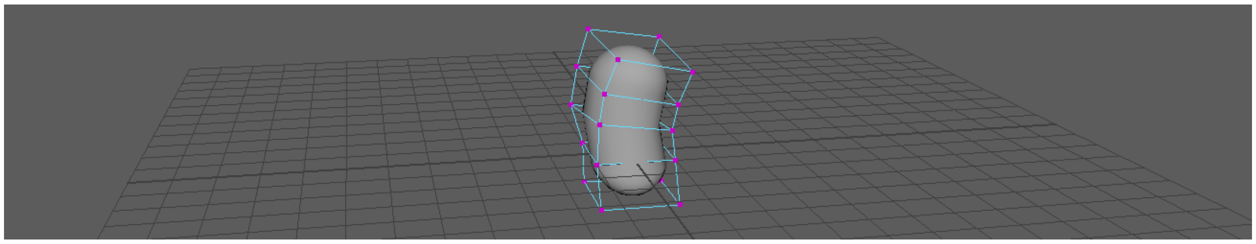
TOOLS & TECHNIQUES

Idle loops



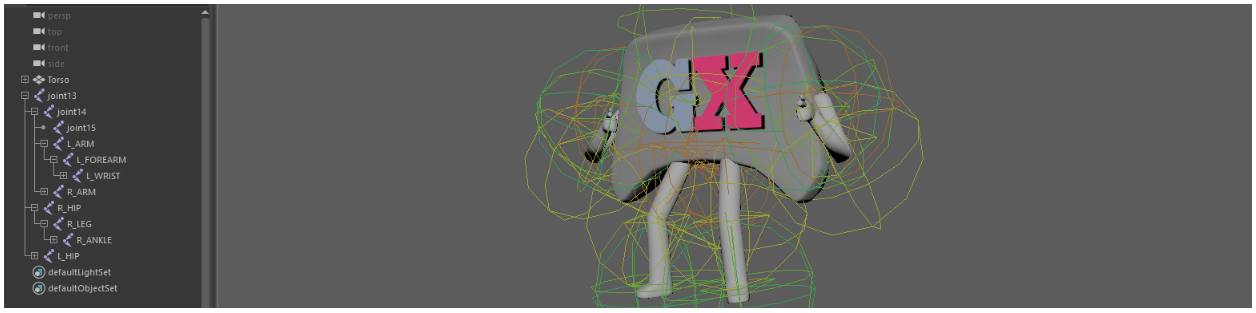
Not the best animator or artist, but still want to breathe life into your characters or UI? Draw your subject over and over again, but don’t trace. Eventually when you have enough frames, you’ll have a naturalistic animation. Why is this? It’s difficult to perfectly copy a drawing, therefore the subtle differences among the frames are the motion. Sure, the proportions may be a tad off-scale, but the cuteness factor makes up for it!

Deformers



Most major modeling software packages have a feature called “deformers”, which allow you to translate, manipulate, and warp a mesh (without the need of a skeletal rig) for animations. It’s very helpful for squashing and stretching. This tool has been used in the past for entire feature animations such as *Veggie Tales*.

Skeletal Animations & Rigging



If your character is a robot with joints, it’s probably best to just parent objects to one another. However, if you want a character whose skin bends when their limbs are rotated, you’ll need to create a skeletal rig and have it skinned to the mesh. While there’s plenty of tutorials out there with a quick Google search, rigging is quite difficult to get perfect. There’s also an extra level to it with weight painting, but when you’re just starting out with simple rigs, that isn’t very necessary. Rigging is such a complex job that there’s a specific job position in the animation, film, and games industry for just that. Don’t be scared away though! After you make a few rigs you’ll fall into a groove with it and be comfortable with the entire process.

FK & IK

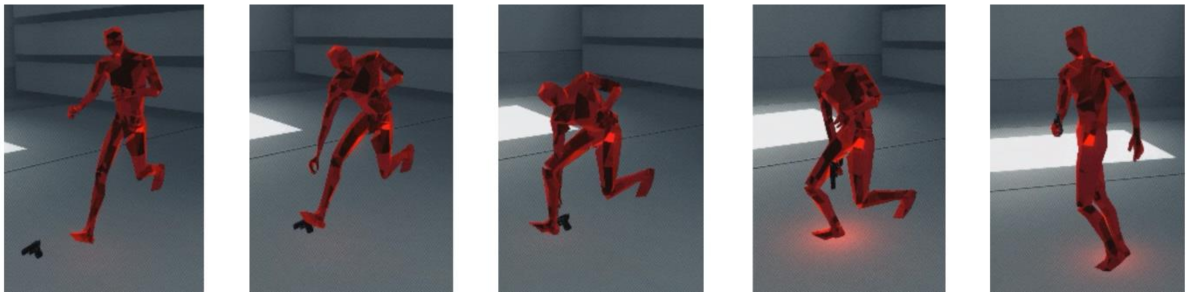
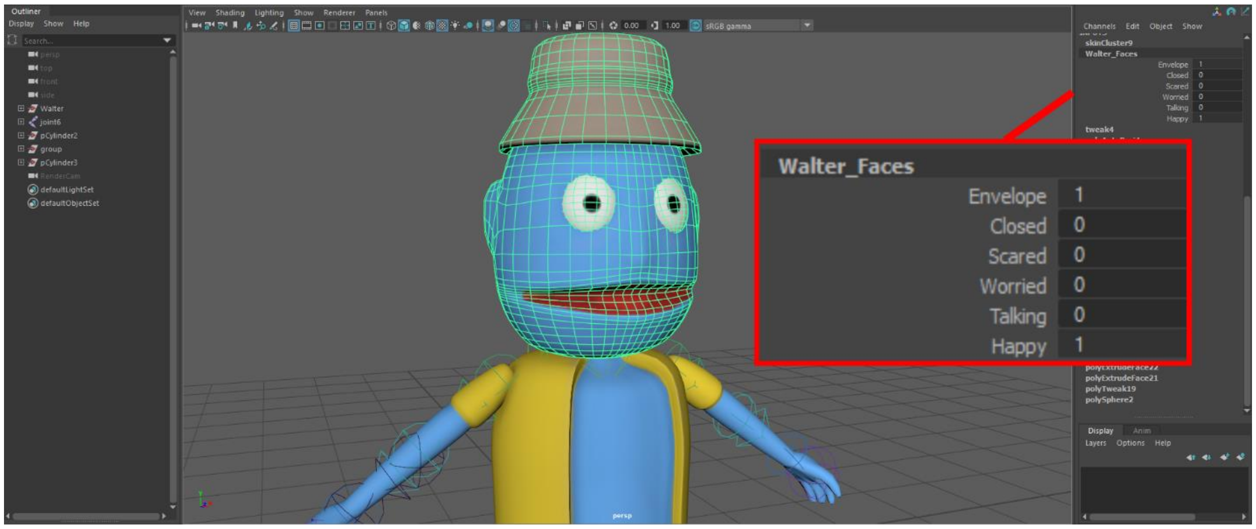


Image credits: SUPERHOT (2016) by SUPERHOT Team

In mathematical terms, Forward Kinematics (FK) is mapping from joint space to cartesian space and Inverse Kinematics (IK) is mapping from cartesian space to joint space. However, in animation terms FK is animating by rotating and translating joints into poses for animation key frames whereas IK uses draggable handles that force the limbs to reach for or stay grounded to surfaces. For example, IK is helpful for animating legs, whereas FK is best used for arms (with the exception of characters in games dynamically reaching out for objects in the scene). A good example of IK in games can be found in *SUPERHOT*, when enemies reach for a fresh weapon on the ground and the rest of their body follows. IK can be implemented fairly easy into your project, and Alan Zucconi has a wonderful guide covering IK and procedural animation within Unity.

Blendshapes & Facial Animation

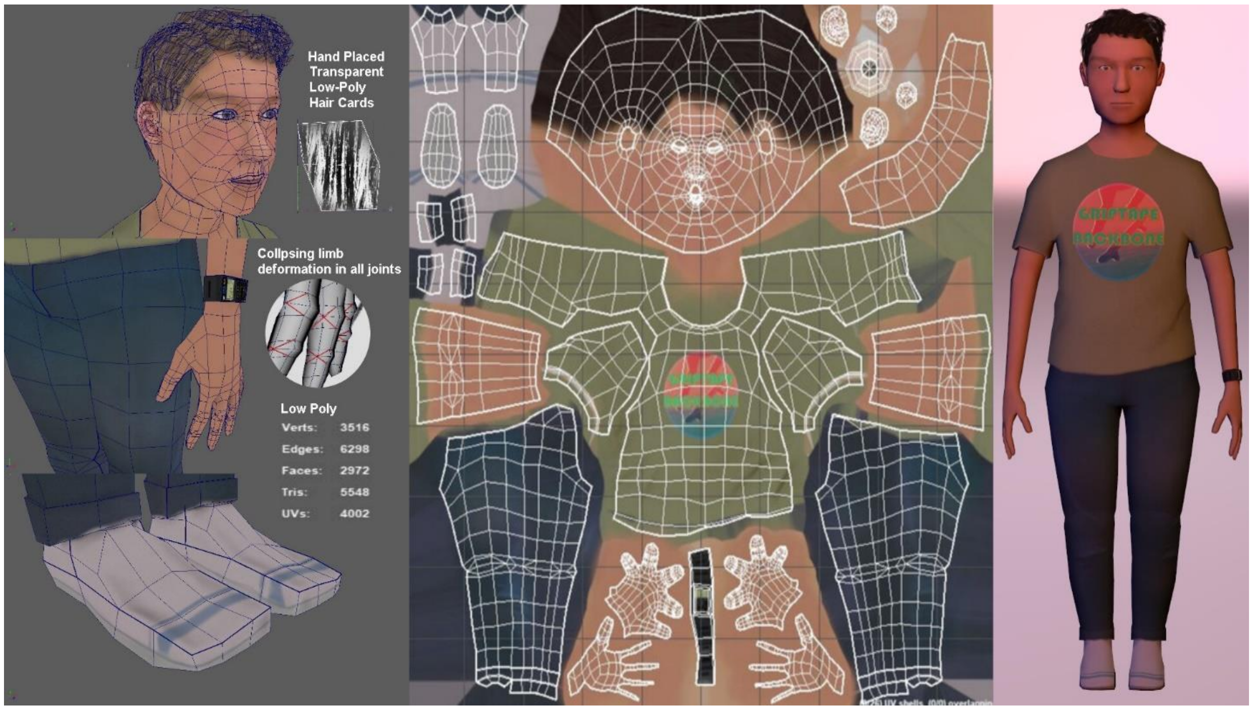


Walter_Faces

Envelope	1
Closed	0
Scared	0
Worried	0
Talking	0
Happy	1

Blendshapes, a tool available in many 3D animation programs, is creating different versions of your model side-by-side and morphing between them with sliders to create even more nuanced results. Handy for facial expressions and animation.

Good topology


Hand Placed Transparent Low-Poly Hair Cards
Collpsing limb deformation in all joints
Low Poly
Verts: 3516
Edges: 6298
Faces: 2972
Tris: 5548
UVs: 4002

N-Gons (surfaces with more than 4 sides) are bad! Symmetry is your friend and quads deform well when bent. Yes, everything will eventually be triangulated, but if you do so yourself ahead of time you can chose how it will deform. Good topology can be simplified and added to easily with more divisions or fewer edge loops. Additionally, UV mapping becomes much easier to stitch when you know what you're looking at.

6. SOUND

Don't leave it till the last moment

Sound communicates so much necessary information and is arguably one of the most important and underappreciated aspects of game development. In most major studios sound is usually paid the least amount of resources towards which is a shame. What would the *Star Wars* films be like without the breathtaking mastering by Skywalker Sound?

TOOLS

AUDACITY

https://www.audacityteam.org/

Price:
Free

Available for:
Windows, Mac, Linux

Notes:
A must-install for anyone with a computer. Reliable multitrack audio editor with tons of effects and extensions.

BFXR

https://www.bfxr.net/

Price:
Free

Available for:
Windows, Mac, Web

Notes:
Make beeps and boops with sliders and buttons.

FREE AUDIO CONVERTER

https://www.dvdvideosoft.com/products/dvd/Free-Audio-Converter.htm

Price:
Free & Paid Version (ad-free without audio watermark)

Available for:
Windows, Mac

Notes:
Audacity can't import advancing audio coding files such as M4As (the default recording extension for older android phones), so if you use your phone for sound effect recording then this works fine to convert your recordings to a format usable by Audacity.

FMOD

https://www.fmod.com/

Price:
Free Version & Paid $5K

Available for:
Windows, Mac

Notes:
Professional audio engine with visual interface similar to a DAW built for teams; makes interactive audio scripting easy. Built-in sound effects library store.

WWISE

https://www.audiokinetic.com/products/wwise/

Price:
Free Version & Paid $750

Available for:
Windows, Mac

Notes:
Professional cross-platform audio workstation with tweakable audio during play and plenty of features to keep both indies and large studios busy.

STEAM AUDIO

https://valvesoftware.github.io/steam-audio/index.html

Price:
Free

Available for:
Unity plugin, Unreal plugin, FMOD Studio plugin, C API

Notes:
Spatial audio and HRTF to make immersive sound environments for games and VR.

REAPER

https://www.reaper.fm/index.php
Price:
60 Day Free Trial & Paid $60 Personal / $225 Commercial

Available for:
Windows, Mac, Linux

Notes:
Professional DAW in the vein of Avid Pro Tools with great plugins.

FOLEY & EXPERIMENTATION

Sound is everywhere

In most film scenes there are at least nine different sound effects playing at once, and for action films that average is more around 50. Sound can inform the player how to feel, what to think, and what to do all without being noticed. Sound is a designer's subliminal secret weapon, and it's fun! Editing and creating sound effects provides so much instant feedback that you can literally hear your work coming together; good sound editing, when done well, is play.

Public domain sound effects are all good and great, but if you want something specific you'll spend more time searching for the perfect sound effect online than it'll take to create and record it – maybe not explosions and gunshots though. While it's fantastic to record the actual thing on location, sometimes that's not always practical and we must rely on Foley. Foley is layered sound effects of other sounds combined to produce a desired sound.

There's no one perfect recipe or way to get the sounds you want, it's all about experimentation and listening for sounds that give you the texture you want to make a sound more complete and believable. So here are some tips.

1. Listen for sounds everywhere, from background ambience to specifics, and record them. Record on the go with a portable mic or your phone and keep an organized sound library to pull from later.
2. Study ASMR. While fantastic for relaxation and treating insomnia. it's also an endless source of satisfying sound effects that you can reproduce at home

and mix in your work. The layering, repetition, and patterns of sound in these videos will never fail to inspire you.
3. Practice by deconstructing a sound effect into each of its separate parts and reproducing it to match the original.
4. When placing ambience in your games make sure it's seamless and not just one looping track. Good ambience is many specific tracks scattered around an environment, all correctly leveled in volume in a hierarchy of importance. For example, if there's a dog barking sound in the distance, you should be able to walk in that direction and find an animated dog. Every sound should be attached to a real source in the world to not shatter the illusion.

ADDITIONAL RESOURCES

I don't claim to be an expert, only an enthusiast with some practice. For me sound design is mostly intuition so here are some videos from folk who can articulate and demonstrate sound design and Foley better than I can.

Akash Thakkar – GDC Talk "Sound of Hyper Light Drifter" & Playlist "5 Minute Sound Designer"

Marshall McGee – "Waveform" video series on YouTube

The Secret World of Foley by Daniel Jewel

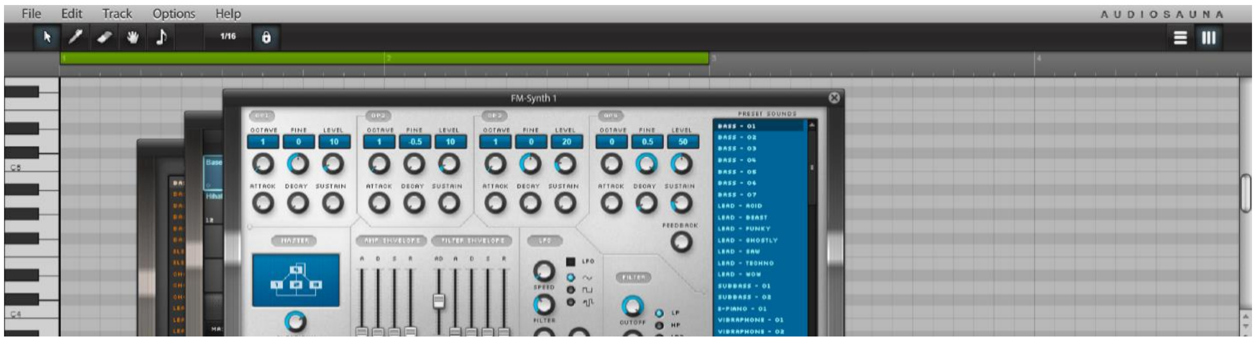
The Music & Sound of Bethesda Game Studios by Noclip

7. MUSIC (FOR NON-MUSICIANS)

TOOLS

My "set up"

While not professional by any means, this set up is budget-appropriate for someone not ready to make an investment into expensive tools and software like midi keyboards and drum pads and Logic Pro or Ableton Live. Fancy tools don't make artists any better, but only give them more things to play around with. Limitation sparks creation! Therefore, when you're just starting out with music composition the less knobs, sliders, and keys you have, the better.



For my electronic synth instruments and sampling needs I use AudioSauna, which is a comparable free web-based alternative to Garage Band for non-Mac users. If you'd like a standalone suite, there's also LMMS, which is available for Linux users as well.

However, in my experience you get better quality sound from recording off of real instruments. Audacity works perfectly. While I use my phone's microphone to record, a high-quality mic such as a Blue Yeti Pro can yield great results for a fair price.

1. Record in a closed-room and use duct tape to cover the walls with egg cartons and/or blankets for noise cancellation.
2. Make a mic pop filter with a bent wire hanger and pantyhose wrapped around it. If you don't have these materials and you're on the go, you can place one of your fingers or a pencil in front of your mouth to split the sound waves when facing the mic to lessen the pop picked up when speaking with plosives (P's, B's, etc.)
3. If you're using an instrument that has a headphone jack or output, use a headphone splitter and connect it to a pair of headphones (for you to listen) and your computer with an aux cord (to record directly).

CASIO
Simple tone selection! PIANO, ORGAN tone button
Built-in grip for easy portability
Easy-to-read display
Mini keys! Big fun!

"If you can play piano, you can play anything."

Someone much smarter than me once said to me when I was little, "If you can play piano, you can play anything." And that's especially true with electric keyboards with multiple instruments such as the Casio SA76. The Casio SA76 costs less than your average AAA game and has 100 different instruments, a drum pad, and a built-in metronome to keep you on beat. Also, the battery life is amazing; I accidentally left it powered on for an entire month and the batteries did not die. If you're looking for a good versatile keyboard on the cheap, this is a good place to start.

Kazoos, harmonicas, maracas, slide whistles, recorder flutes, and ukuleles, while perhaps a little childish, are also great budget instruments to add to your musical toolbox.

THE BASICS

Every Good Boy Deserves Fudge

While tapping out notes that sound good and recording them works fine for making something quick for a game, it never hurts to learn the basics. Learning how to read and write music, understand time signatures, intervals, scales, chords, and key signatures can only help you. Here's a great site to learn, but also feel free to make a visit to your local library and pick up some literature on music theory there. Some library branches even offer free community workshops and lessons, so be sure to also check their event calendars for listings.

DYNAMIC MUSIC FOR GAMES

Games are interactive, they visually respond to player input, so why shouldn't they respond with changes in the music and sound as well? Mark Brown's Game Maker's Toolkit video essay on "Adaptive Soundtracks" is a great springboard to start thinking about some of the techniques you can adopt and adapt to make your games sound better. I'd also recommend checking out Disasterpeace's *January*, which has downloadable source files to study and learn from. Other than that, just listen closely to your favorite games and analyze how the soundtrack changes and evolves throughout the game.

One of my favorite examples of an adaptive soundtrack is in *Fallout 4* with Diamond City Radio. In *F04* the radio announcements change based on player action and events as well as the DJ's confidence. Throughout the game the radio host is constantly nervous, and it shows, but once you complete the confidence man mission, his entire presentation changes. Be creative in what the soundtrack is tied to and have fun.

DON'T STEAL OTHER PEOPLE'S WORK

JUST DON'T.

“For exposure” kills

If you want to use someone else’s music in your game, you have two options. The first is finding music and recordings of that music within the public domain or creative commons, adhering to the usage guidelines, and giving proper credit. The second is to contact the musician and ask for permission to use their work. If your game is commercial and has a budget, figure out how much it would cost to license their music for it. Work something out, and if they say “no” then find music for your game’s soundtrack elsewhere or create it yourself. Just remember that musicians are artists too and deserve respect.

On finding musicians to collaborate with

Go local or go global. Sometimes the best music is right in your own town, so ask around and attend local basement, bar, and battle of the bands concerts. After attending the concert, if small enough, you’ll probably have a chance to meet a band member and exchange contact info. Musicians can be some of the nicest and most down-to-earth people to work with in games, so be cool and humble when asking if they would like to collab with you.

On the other hand, you can always surf the radios and suggested tracks of SoundCloud or Bandcamp to find music you like and the contact info of the artist(s) who produced it. Once you’ve released a game that gets some attention, you may even have musicians coming to you – which is wonderful! While their music may not fit for whatever you’re working on at that moment, be sure to stay in touch and save their info for the future.

8. DESIGN

A GOOD DESIGNER TAKES INSPIRATION FROM EVERYWHERE

While AAA development favors “T-Shaped” individuals with a depth of knowledge, mastery of one aspect, and general knowledge of other aspects of game development, independent development is different. In design, not only is a generalized development skillset (art, code, sound, music, etc.) helpful for making games, but also knowledge in many other fields from the humanities to competitive duck herding or underwater basket weaving. Everything helps!

A generalized knowledge of gamedev can make you self-sufficient and help you communicate effectively with others in a team setting. Normally artists don’t speak programmer and vice versa, but a designer can help be an intermediary communicator.

If all you know is games though, then your games probably won’t be very interesting. Being inspired by other games is fine, but for innovation to take place it’s important to draw inspiration from things outside of games. If you have a very deep knowledge in something niche like taxidermy, then use it in your games! For example, if you are a taxidermist game developer you may think to capture your in-game art by taking stop-motion photos reminiscent of Victorian taxidermist Walter Potter's work. That suggestion is on the house!

TOILETS

What does a designer actually do? What is design? While Liz England (designer at Ubisoft Toronto) would answer this through "The Door Problem", I prefer using toilets as my reference point in embarking on the questions a designer would ask themselves when approaching toilets in a game project.

- Will there be toilets in game?
- Why or Why not?
- If so, where?
- How many toilets are there in game?
- How long does it take the player on average to reach the first toilet?
- Are the toilets in-game interactable or for show? If so, how?
- How will you telegraph to the player that a toilet is interactable? With non-diegetic user interface, a highlight when hovered over, or is everything in game interactable and by association the player would know this is interactable too?
- Can you pee or poo in the toilet, or is the toilet used for some other purpose? For instance, drinking in *Fallout: NV*, saving in *No Heroes*, or loot in *Borderlands*.
- Is the amount you can defecate limited based on food intake? Time?
- Do you press and hold a key/button to defecate or does it require a series of specific key/button presses or gestures?
- Is defecation limited to just the outside of the toilet?
- Can you interact with your waste post-defecation? E.g., *Duke Nukem Forever* where the player was able to pickup and throw a dukey as well as have their action acknowledged by a fourth-wall-breaking voice-over from Duke.
- If the toilet does more than one thing when interacted with, is it consistent in offering those same options throughout the game or does it vary based off the type of toilet, placement, character progression, or story plot?
- How does the player choose which affordance to execute on if there's more than one option in interacting with the toilet?
- Do enemies spawn inside the toilet and surprise-attack the player?
- Do AI take bathroom breaks? How many AI can fit in one stall? Is the stall door locked whilst the toilet is occupied by the AI? For how long?
- How does the physical interaction on keyboard and gamepad mirror or represent the real-life action of interacting with a toilet?
- Can you lift the toilet seat? How about the lid?

- If so, is this opening and closing done through animation toggling on input or directly driven by physics?
- If animation, if something is in the way of the toilet lid closing or on top of the lid preventing it from opening, will the animation be interrupted to a stop, go through said objects, or push them away?
- Can you also interact with the shut-off valve?
- If so, does it disable the flushing simulation from happening, or is it connected to something else?
- Can you take off the tank lid?
- Is there a Realtime working flush mechanism viewable after taking off the tank lid?
- Is there any surprise or Easter egg inside the tank next to the flushing mechanism?
- Can you flush?
- If so, does flushing just start a particle effect, sound, and animation of urination levels lowering and water turning clear? Is it physically simulated to interact with other objects in the scene should they fall inside?
- What happens when you try flushing when the flush sequence is already in effect?
- Is there any achievement for flushing 100 times? 1000?
- Does flushing a certain amount of times trigger an event? For instance, annoying the AI or causing it to clog or overflow?
- If it overflows, is there physics-driven water simulation that pushes objects in contact?
- Can AI slip and hurt themselves on the overflow? Will there be a janitor AI that comes and puts a “wet floor” sign down near the flooded area? Will other AI use pathfinding to avoid said area?
- What type of pathfinding will the AI use to avoid the fecal-infused overflow spill?
- Can you use the flush handle to communicate in Morse code? Will this be a part of a puzzle?
- Is the toilet a dead end or is there a secret passageway/portal behind it?
- Does the player character have a gender? Are the restrooms gendered with signs on the exterior?
- Is there any difference between a men’s restroom or women’s restroom?
- Are the two gendered restrooms linked together for ease of navigation via a broken wall?
- Is the act of defecation inside the virtual toilet biased? Can you defecate standing up as well as sitting down?
- What values (cultural, political, etc.) do you think toilets present in this game?
- Are these values communicated through interaction with these toilets intentional?
- What does the toilet say in regard to environmental storytelling?

- What's the size of the toilet?
- Is there an accessorily rail modeled next to the toilets? If not, are there no disabled characters in your game world?
- Is the user interface for interacting with the toilet differentiated with color or pattern? Can colorblind players interact with the toilet with as much ease as non-colorblind players?
- Are you excluding a group of players in how you represent toilets? Is it intentional or unintentional?
- When is a good time to stop asking yourself theoretical questions about toilets in-game and move on with the rest of the game?

In summary

A designer's job is to think of literally everything and how everything creates and influences the overall play experience. "The role of the game designer is, first and foremost, to be an advocate for the player." - Tracy Fullerton, Game Design Workshop.

What's the fascination with toilets?

Even something so insignificant as toilets in games must be payed the upmost attention to by the designer. Toilets make the game world feel more believable by relating something so mundane in the real world to the virtual one. Toilets are good game design. For more on toilets in games visit ToiletGameStudies.org.

Q&A: PLAYTESTS WITH PLAYERS

Playing results

As developers we sometimes forget what's it's like to be a player. We're so focused on making games, testing them, and iterating on them that we lose touch of what it's like to play the game without playing results. This is something common to acting too. How does an actor acting as their character fool themselves into being naïve when they know exactly what's going to happen in advance in the script? How can a developer stay objective when they're so close to their work? The answer is playtesting with players who haven't played your game before.

If as designers just do internal playtests, then we'll undoubtedly get tunnel vision and make games only for ourselves. Getting feedback is a vital part of design and revision, and the earlier fresh players are introduced the better! Keep in mind there are several types of focused playtesting for different results.

*TO ADD: IMAGE OF DEV W MAGNIFYING GLASS IGNORING BIG ISSUES OUTSIDE OF VIEW WHILE FOCUSING ON ONE SMALL ISSUE

Focus group testing

Have someone other than you (who doesn't care about your game) introduce the game to a group of strangers with a brief scripted intro (while mentioning their impartiality with a general dispassionate tone), customer persona questions "What games do you play?", "What do you like about those games?", "What was the last game you purchased?", and a guided but open talkback after the playtest where the players give all their honest feedback. This is helpful because as the developer you won't be physically present to defend, explain, or apologize for your choices and influence the sanctity of the playtest. Additionally, if you're able to unobtrusively film the participants (with their consent), you'll be able to see pure reactions to specific moments in-game. Cons: this method takes time and resources to organize.

Usability testing

Not focused on whether the player enjoys the game or not, but rather how usable it is. How easily they're able to learn the mechanics and progress without getting stuck or frustrated and how intuitive the interface is. Record their gameplay, reactions, and analytics on behavior such as heatmaps of the paths they explore, time in defined areas, eye tracking, and mouse tracking. Also, if they're comfortable with it, have them to speak aloud their thought process while playing. Yet, keep in mind that the best feedback is in the player's action, not their words. Their actions can't lie. This test is also best organized by a third party.

*TO ADD: IMAGE OF USABILITY DOUBLESIDED MIRROR OBSERVATION ROOM SIDE VIEW NEXT TO PARTICIPANT

Casual playtests

So, you have a few friends or family members over, and perhaps they aren't even "gamers" to begin with, it's still better than no playtests at all. A casual playtest may not be objective enough, as your confidants may either be too harsh or not harsh enough in the feedback they give. There are two main methods of casual testing: the first is handing them the game and staying absolutely silent (no exceptions) while you watch and take notes on either specifics or general bugs and trouble spots. They'll look lost, break your game, and ask for help but you must not give in to temptation. Just bury yourself into your notepad.

*TO ADD: IMAGE OF DEV WITH ZIPPERMOUTH AND A DISTRESSED POKER FACE

The second method is to talk with them as you normally would and approach the playtest as if you were sharing a finished product and just hanging out playing the game. Buy some food, invite them over, and just be. Maybe ask them some questions like…

- Overall, what do you think of the game?
- What felt right, what felt wrong?
- What did you like? What didn't you like?
- Were you able to learn how to play without any trouble? Was there anything confusing? Anything you wish you knew beforehand?
- How would you pitch this game to someone else?

Q&A: FEEDBACK WITH OTHER DEVELOPERS

Feedback with other developers is fantastic, because devs generally have a better vocabulary to give specific feedback and can back up their ideas with logical reasoning. Potentially unhelpful feedback like "you should add jetpacks and dragons" will be replaced with logical suggestions and the layout of concrete steps needed to get there. However, beware! While you should always keep an open mind and listen to all feedback that others give you, take it all with a grain of salt. Don't immediately dismiss an idea, but rather filter them by the 80/20 rule. If you can implement it with little effort and get a potentially big return, then go for it! Main thing is to keep your artistic intent intact and not allow too many cooks in the kitchen, or else the broth may lose its distinct flavor.

An unpopular opinion in design

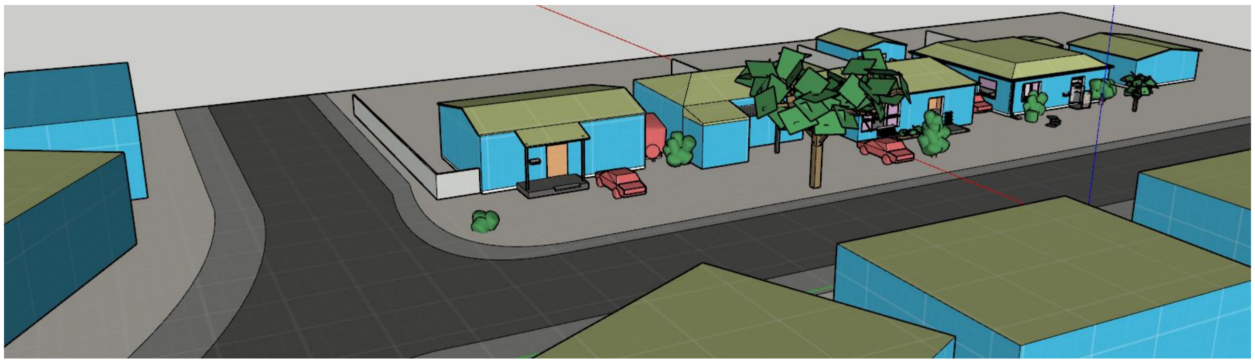
Remember! You know your abilities and your game better than anyone else and like art, sometimes it is too early to demonstrate and communicate what the game is going to be. While many designers advocate for paper and pen playtesting even before you touch art or code, that may not be applicable for every type of game – that may just be how *they* brainstorm.

While it's great to have a platform to bounce your ideas off of, sometimes the most personal things, while perhaps a bit rough around the edges, are developed inside a vacuum. I'm not advocating for not playtesting your game with others -- quite the contrary. Take the time to consider who you have playtest your game and at what stage of development you're comfortable with them doing so, but also be brave.

A sentimental pep talk

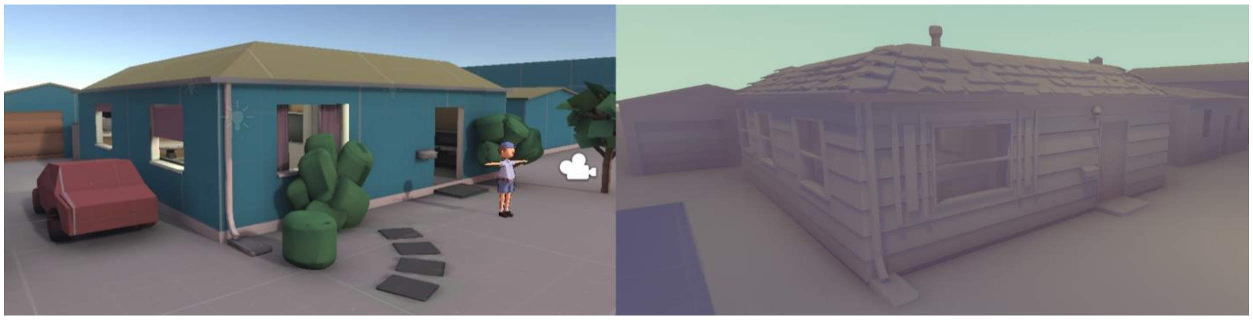
Each game is a little part of yourself, so it can be difficult to push it out into the world. Games are like children. It's scary to put them out into the world because they can get hurt; you can get hurt. And while many say to not attach yourself to your work, it's hard not to. As developers and artists, we pour our entire being into our work, and to detach ourselves from that robs us of our ownership and the work of its lifeblood. If you care so much about your work that you are your work, that's beautiful and should be celebrated. People can still criticize your work, and trust me they will, and that's okay. If it's constructive, use it, if it's not – lose it. Think of who's giving the criticism, their viewpoint, biases, and credibility and contextualize. You can't please everyone, especially online. You must develop thick skin. Everyone has opinions. You might as well make the game you want to see in the world, because odds are if you are happy with it, others will be too.

BLOCKMESHES ARE YOUR BEST FRIEND



Greybox, Whitebox, Whatever You Call It – it's cubes

Environment art takes a lot of time and usually comes after designers create the rough layout for the levels. These blocky layouts are quick to model, playtest, and iterate upon. They're great for establishing the visual language, pacing, and flow of the level. Blockmeshes don't have to be grey either, using different colors to test the lighting of the scene gives the artists (you) a better starting off point!



Nudging the player

- Big landmarks in the distance give the player a destination to move towards
- Openings naturally attract attention, so use visual lines in the environment such as pipes, stones, railings, destroyed cars, and more to pinch and frame the perspective to one focal point.
- "Valves" are commonly used for bossfights areas, but big drops that prevent the player from backtracking are opportunities to stop rendering the previous environment and seamlessly load in the next section of the level.

- Avoid using invisible walls and instead use jagged edges to semantically communicate to players to avoid the area and straight and rounded edges to invite players.
- Players are like trained dogs so if you leave breadcrumbs in the form of collectables, they'll follow them. Extrinsic motivation works just as well as intrinsic motivation in guiding players to points of interest.
- Light & God Rays are great ways to illuminate entrances and exits in low lit environments like caves and ruins.
- Animated moving environmental elements such as animals, characters, and destruction are other ways to direct the player's attention, though should be used when the art is finalized and navigation problems still exist.

Blockmeshing is a vital part of 3D level design and game design as a whole. While some games' art styles are already low poly, blocking out an environment before committing to it will always benefit your workflow.

9. THEORY & FURTHER READINGS

THE FOOTNOTES

What do I start with first?

The three main components of a game are as follows:

Mechanics – The quantifiable formal elements of a game (number of players, objectives, rules, procedures, boundaries, resources). The way data is represented and the algorithms and logic behind it.

Dynamics – The runtime behavior of mechanics acting on inputs over time. Basically, what happens when a player plays the game. The expected and unexpected outcomes ("emergent behavior") of the systems in action.

and ***A E S T H E T I C S*** – The designed emotional response when interacting with a system. The types of aesthetics outlined by Robin Hunicke, Marc LeBlanc, Robert Zubek are as follows: 1. Sensation, 2. Fantasy (make believe), 3. Narrative (drama), 4. Challenge, 5. Fellowship (social framework), 6. Discovery (uncharted territory), 7. Expression (self-discovery and artistry), 8. Submission (games as a pastime, becoming one with the game, i.e., Getting Over It, Euro Truck Simulator, etc...

So, when starting the process of making a game, which aspect of the game do you tackle first? Ideally, both the game mechanics and aesthetics would go hand-in-hand and support one another. However, as designers we typically lean towards one more than the other.

Therefore, is it M -> D -> A or A -> D -> M->?

Do you start with an emotion you'd like to make the player feel? Or a cool mechanic that you'd like to explore? There's no right or wrong way to begin.

Who is the game made for?

There are many schools of thought of how a game should be designed. There's Tracy Fullerton's Playcentric design which focuses on player experience first and an iterative design process. However, you could also just make a game for yourself, make what you think feels good and develop a game with a more auteur stance, as rough around the edges as it may be. Your own design process will come together naturally over time.

Additionally, games can cater to different playstyles. In Richard Bartle's "HEARTS, CLUBS, DIAMONDS, SPADES: PLAYERS WHO SUIT MUDS", Bartle outlines four unique playstyles:

- Achievers – Your speed runners and achievement hunters, completionists, competitive esports players, anyone who plays to master a game.
- Explorers – Walking sim, immersive sim, and open world fans. Players who love gaining knowledge and perhaps enjoy the finer points of a game and its individual parts.
- Socializers – Streamers, MMO players, and casual game players. Players who are proud of the friendships cultivated and contacts made in game as well as their influence and expression.
- Killers – Competitive fighting game players, trolls, sadists. Anyone proud of their K/D ratios, control or domination over other players, getting other players to "ragequit".

Whether you make games targeted towards a specific demographic for profit or a personal project, games in nature are interactive and intended to be played by someone. Or perhaps you're just making it for the sake of making, and that's valid too!

Rules

The best piece of advice I've ever gotten when it came to game rules is that "you don't come in the box". You can't explain the game or demonstrate it in front of someone, so you need the rules to be as legible and clear as possible for the players to understand. In fact, the game should be playable and intuitive enough to play without the rules. The player should be able to glance at the game and automatically have an idea of what to do based on the given affordances. The rules can be learned as the game progresses, because giving the player all the rules at the start may be an overload of information.

Story

The dramatic elements of a game (premise, character, story, arcs, challenge, world building) all provide context to the action on screen. Why are you doing what you are doing? Story can transform a puzzle into a game. Story gives a reason to continue to play. Story can transform a gimmick into a memorable and meaningful experience. There's a story in every game, be it the written ones or the emergent stories players tell themselves. While games may have beautifully written stories such as in Telltale Games' work, emergent narratives from open ended open world games with systems interacting with one another autonomously (as in *S.T.A.L.K.E.R* or *Civilization*)can be just as powerful.

On the other hand, Ralph Koster's "Theory of Fun" states that games aren't stories, but rather are teaching tools that teach underlying patterns of abstract concepts and that stories merely serve as wrapping paper. That flow does not equal fun, but learning, practicing, and mastering a problem with no pressure equals fun.

Meaningful choices

What makes games unique as a medium is that there is a choice, there's player agency, there's options and ways to express themselves in the way they interact and consume art. So, you may as well give the player meaningful choices. Give the player a dilemma.

Here's some aspects to consider when designing meaningful choices.

- Affordances – originally coined by James J. Gibson. Affordances are the many uses of an object and our relationship with those verbs in interacting with objects. The potential of something, i.e, "It could be a... hat!"
- Feedback – or as game designers like to call it "Game Juice".
- Consequences (Real or Fake) – Forces user to weigh pros and cons and make a deliberate decision between the options.
- Context – Making sure the user is invested
- Easy-to-learn systems that are hard to master, moment-to-moment tactical choices become a higher game with long-term strategy

WOT IS A GAME?!?!

A philosophical question indeed that has no definitive answer. Sure, definitions can be tools as they limit us, which helps since design thrives on limitation, but they should never be all-defining.

You know it when you see it

What is considered "a game" can easily vary from one person to the next. Plato may argue for ideal forms to define what something is. However, there is no agreed-upon, perfect game to model all other games from. Therefore, the essence of what is a game is ambiguous. There is no absolute truth. Are games constructs? Jesper Juul, game theorist, in his essay "Game Player World" has a great table of definitions as well as his own model. These are some traditional definitions that are good starting points.

Source	**Definition**
Johan Huizinga 1950, p.13.	[...] a free activity standing quite consciously outside ”ordinary” life as being ”not serious”, but at the same time absorbing the player intensely and utterly. It is an activity connected with no material interest, and no profit can be gained by it. It proceeds within its own proper boundaries of time and space according to fixed rules and in an orderly manner. It promotes the formation of social groupings which tend to surround themselves with secrecy and to stress their difference from the common world by disguise or other means.
Roger Caillois 1961, p.10-11.	[...] an activity which is essentially: Free (voluntary), separate [in time and space], uncertain, unproductive, governed by rules, make-believe.
Bernard Suits 1978, p. 34.	To play a game is to engage in activity directed towards bringing about a specific state of affairs, using only means permitted by rules, where the rules prohibit more efficient in favor of less efficient means, and where such rules are accepted just because they make possible such activity.
Avedon & Sutton Smith 1981, p.7.	At its most elementary level then we can define game as an exercise of voluntary control systems in which there is an opposition between forces, confined by a procedure and rules in order to produce a disequilibrial outcome.
Chris Crawford 1981, chapter 2.	I perceive four common factors: representation ["a closed formal system that subjectively represents a subset of reality"], interaction, conflict, and safety ["the results of a game are always less harsh than the situations the game models"].
David Kelley 1988, p.50.	a game is a form of recreation constituted by a set of rules that specify an object to be attained and the permissible means of attaining it.
Katie Salen & Eric Zimmerman 2003, p.96.	A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.

Table credit: Jesper Juul: "The Game, the Player, the World: Looking for a Heart of Gameness". In *Level Up: Digital Games Research Conference Proceedings*, edited by Marinka Copier and Joost Raessens, 30-45. Utrecht: Utrecht University, 2003.

Why ask the question then?

For posterity? What a game is should be open to interpretation. Regardless, what games are is heavily influenced by the values and agendas of those defining it.

Therefore, it's important to pay attention to definitions because they can change attitudes, ideals, culture, and thought itself. As developers we ask this question for innovation, for change, and most importantly, for a heated talking point in academic and water cooler settings.

HOW DO GAMES MEAN?

Whether intentional or not, games and the worlds, systems, and stories inside them have an embedded message. Games can be metaphors, simulations, and abstractions. Most importantly, games can connect with people; it's extremely important to be conscious of what your game means. Think about what you are trying to say or express as a developer and the disparity to what is perceived by the player. It is your responsibility as a designer to make everything a conscious choice.

FURTHER READINGS

Brian Sutton-Smith: Play & Ambiguity

Roger Caillois: Man, Play and Games

Alexander R. Galloway: Gaming Essays on Algorithmic Culture "Gaming action, four moments"

Henry Jenkins: Game Design as Narrative Architecture

Guy Debord: Theory of the Dérive

Yi-Fu Twan: Space and Place: The Perspective of Experience

Ian Bogost: The Rhetoric of Video Games

Henry Jenkins: Complete Freedom of Movement: Video Games as Gendered Play Spaces

Jenova Chen: Flow in Games (and Everything Else)

Daniel Vella: No Mastery Without Mystery: Dark Souls and the Ludic Sublime

Bonnie Ruberg and Adrienne Shaw: Queer Game Studies

Rob Zacny: When a Game Becomes a Troubling Psychological Portrait

Nicky Case: How I Make Explorable Explanations

Bonnie Ruberg and Adrienne Shaw: Queer Game Studies

10. FINISHING POLISHES

THE LAST 10% IS ALWAYS THE HARDEST

While prototyping is a skill in its own, finishing is a more difficult skill to master. Making something performant, “bug free”, and ready for release isn’t easy. More importantly, finishing and releasing a full game requires social media and marketing action, pitching your game, and presentation which is not development and therefore difficult for some. However, what’s important is to take breaks, remind yourself why you started the project in the first place, and measure how far you’ve come and how little there is left to go in comparison. If you ever hit a wall, don’t be afraid to reach out to peers for advice and guidance.

HELPFUL QUESTIONS TO ASK YOURSELF

What is your purpose as a designer/design team?

This will serve as your mission statement (what you’re doing now to achieve your goals), but also think about your vision (long term).

What is lacking or broken in games?

What is “the problem”?

What is the solution?

You and your game.

What platforms will your game be released on?

Desktop, mobile, console, etc...

Who is your ideal player?

Targeted demographics (regions and populations/customer personas); niche.

Market Dynamics

What is your total addressable market (all the players who can buy/play your game)? What are the stats (ratings & reviews, total amount of owners, drop-off rates) of games comparable to yours? IT’S NOT A COMPETITION, but what would your “competition” be?

How is your game the best experience for the player?

What makes your game unique and special opposed to other titles? The “X factor” of your game? What is your ownable position that hooks people into about your game? What part of your game do people enjoy the most?

What should players believe about the value of the game?

Why it’s worth paying for/spending time with.

What are your key revenue drivers?
How are you going to make money from this?

- Free to Play with microtransactions (cosmetic, energy, timers, downloadable content, etc.)
- Premium Price Tag $60
- Merchandising (plush toys, phone cases, clothing, etc.)

What's the development roadmap?
- When does pre-production (planning) begin?
- When does production begin?
- When is the alpha complete (core gameplay finished, features in-game)?
- When is the beta (game is finished, maybe a few bugs and balancing issues to iron out) ready?
- When will the game be finished? (gold master)
- Will you be doing a soft launch (releasing in one territory as a test, then releasing everywhere else later)?

Where do you promote the game, what platform(s)?
Social media platforms, selected games writers to send press kits to (ones who have written about similar things in the past who may write favorably about your game), and conventions to showcase, promote, and gather feedback from.

Audience Acquisition?
Paid features on storefronts? Earned festivals and reputation? Viral guerrilla marketing campaign? Grassroots development vlogs/streams? Highly-active social media presence?

Audience Retention?
How do you keep people playing or interested in your work? Is your game episodic? Is your game a service? Do you post content updates or maintain an active social media presence? How do you maintain the momentum from this game to the next?

How do you talk about the game?
Personal? Professional? Memey? What language do you use when talking about the game? How are you going to convey the experience through as few words as possible to sell the fantasy?

SCOPE: KILL YOUR DARLINGS OR SUFFER

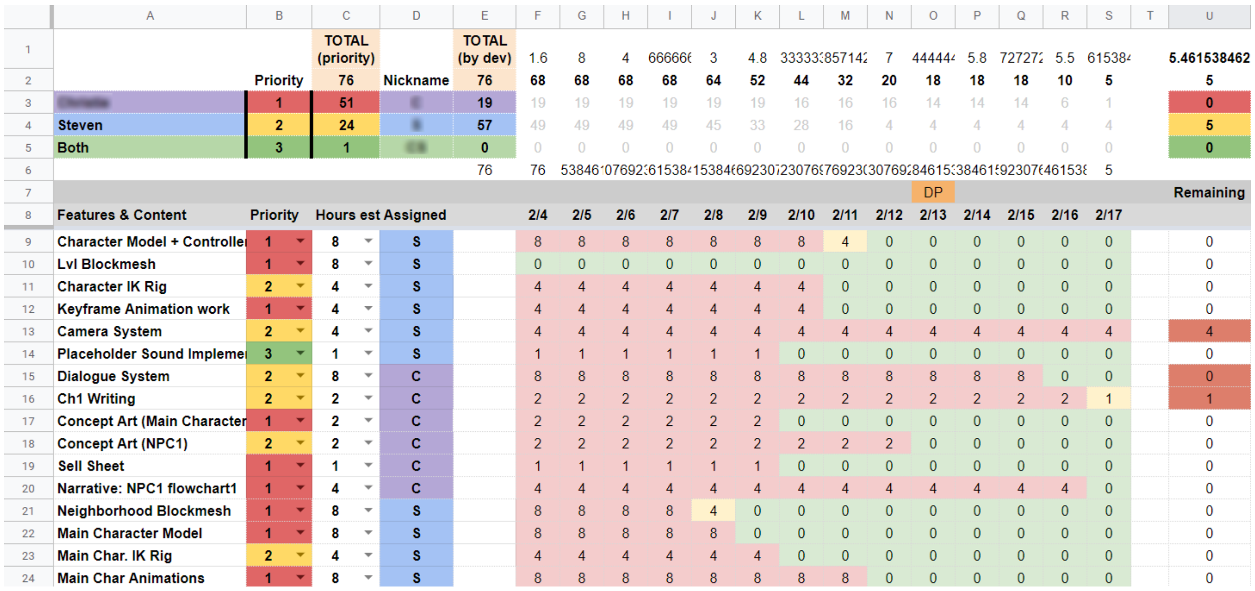
What is scope?
Scope is the work required and features that make a product what it is. The bigger the scope, the more ambitious it is – the smaller the scope, the more doable and less risky the project becomes. Referring back to Chapter 2 section 4 "feature creep" is also known as "scope creep". A teacher once told me that "perfection isn't

in adding more, but chipping away at something until you've got a gem". Sometimes features need to be cut for games to be shipped. There's always an opportunity for those ideas to become realized in a future project! Don't get stuck adding features to your game forever. No game ever wants to be finished, so it's your responsibility as the dev to finish it.

BE YOUR BEST PRODUCER

Time management is key!

It's extremely difficult to estimate how long something will take you to complete before you've ever done it. It'll be especially hard to estimate the time it takes to complete a task if nobody in the world has ever even attempted it. Because games are such a new medium and the tech used to make them is always changing, everyone is always learning how long things take. This is why you should create a "burn down chart" or at the very least keep a running "to-do" list to keep track of your time so you know when to call it a day.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1			TOTAL (priority)		TOTAL (by dev)	1.6	8	4	666666	3	4.8	33333	857142	7	44444	5.8	727272	5.5	615384		5.461538462
2		Priority	76	Nickname	76	68	68	68	68	64	52	44	32	20	18	18	18	10	5		5
3	[redacted]	1	51	[redacted]	19	19	19	19	19	19	19	16	16	16	14	14	14	6	1		0
4	Steven	2	24	[redacted]	57	49	49	49	49	45	33	28	16	4	4	4	4	4	4		5
5	Both	3	1	[redacted]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
6					76	76	538461076923615384153846692307230769769230307692846153384615923076461538												5		
7															DP						Remaining
8	Features & Content	Priority	Hours est	Assigned		2/4	2/5	2/6	2/7	2/8	2/9	2/10	2/11	2/12	2/13	2/14	2/15	2/16	2/17		
9	Character Model + Controlle	1	8	S		8	8	8	8	8	8	8	4	0	0	0	0	0	0		0
10	Lvl Blockmesh	1	8	S		0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
11	Character IK Rig	2	4	S		4	4	4	4	4	4	4	0	0	0	0	0	0	0		0
12	Keyframe Animation work	1	4	S		4	4	4	4	4	4	4	0	0	0	0	0	0	0		0
13	Camera System	2	4	S		4	4	4	4	4	4	4	4	4	4	4	4	4	4		4
14	Placeholder Sound Impleme	3	1	S		1	1	1	1	1	1	0	0	0	0	0	0	0	0		0
15	Dialogue System	2	8	C		8	8	8	8	8	8	8	8	8	8	8	8	0	0		0
16	Ch1 Writing	2	2	C		2	2	2	2	2	2	2	2	2	2	2	2	2	1		1
17	Concept Art (Main Characte	1	2	C		2	2	2	2	2	2	0	0	0	0	0	0	0	0		0
18	Concept Art (NPC1)	2	2	C		2	2	2	2	2	2	2	2	2	0	0	0	0	0		0
19	Sell Sheet	1	1	C		1	1	1	1	1	1	0	0	0	0	0	0	0	0		0
20	Narrative: NPC1 flowchart1	1	4	C		4	4	4	4	4	4	4	4	4	4	4	4	4	0		0
21	Neighborhood Blockmesh	1	8	S		8	8	8	8	4	0	0	0	0	0	0	0	0	0		0
22	Main Character Model	1	8	S		8	8	8	8	8	0	0	0	0	0	0	0	0	0		0
23	Main Char. IK Rig	2	4	S		4	4	4	4	4	4	0	0	0	0	0	0	0	0		0
24	Main Char Animations	1	8	S		8	8	8	8	8	8	8	8	0	0	0	0	0	0		0

Becoming a hermit

By not tracking your own development time, you lose out on a lot of helpful information on how long it took you to do X,Y, and Z. Therefore you won't be able to make educated estimations and strategic (long-term) decisions that could save you precious time that could be spent elsewhere – like sleeping! By working 12+ hour days and getting lost in your craft without a care in the world, you could also forget to shower, eat, stay hydrated, and sleep (computer monitors restrain melatonin production). Not only are you risking your physical and emotional health, but you lose sight of your own self-worth, commercial rate, and interpersonal relationships.

Be more kind to yourself
Don’t overwork yourself. It’s a marathon, not a sprint. Others’ success aren’t your failures. Hard work isn’t bad, but make sure it’s sustainable. No game is worth a dev, so ask yourself if you’d ask this of someone else – and if the answer is “no” then you shouldn’t be asking it of yourself.

ELEVATOR PITCH: TALKING ABOUT YOUR GAME

Always assume they’re skeptical
Unless the target audience for your game is just your family and friends, they probably don’t care about you or your game. Attention is the most scarce commodity in today’s economy and you’ve only got one chance to grab and hold it. People, generally speaking, don’t care. Don’t be disheartened though! People want to care about something! You’ve just got to communicate your passion and paint a picture, and give them a story to hold onto.

Setting yourself apart
What are you values?

> How are those values aligned with what you are currently doing? Are you making aspirational work? Don’t just wait for the opportunity to present itself, make the opportunity.

What’s your niche?

> What do you offer that nobody else does? What is your artistic voice? Don’t worry, it’ll come naturally with time (lots of time).

Don’t assume they’re stupid
Yes, generally in game development designers have to make things very “idiot proof” in regard to exploits and boundaries to make a smooth frictionless experience. While your design can be simplified to become more accessible, your pitch shouldn’t be come off as condescending, off-putting, or insulting to the audience to whom it’s addressed.

Assume your game isn’t a masterpiece
You can be proud of your game, but attention is going to drop real fast if all you keep saying about the game is how “awesome” it is. Players generally don’t care about the developer’s opinion on their work, they only value their own opinions and peer judgements. If you can quote positive reviews from well-known publications – that’s great! However, use your time wisely to actually describe your game, NOT hype it up. If the game is good and the stars are aligned, the hype will come naturally.

Save artistic intent for the artist statement

While describing your game by means of theme is good when talking about it to other developers, players just want to hear about the gameplay and context. Save the artistic intent and inspirations for the dev vlogs or postmortem blog post.

NO LORE

If your game is lore-heavy, save the lore for the game itself. This is the most frequent mistake many newcomers make. The elder gods, a longtime feud between factions, mystical creatures, and prophecies can wait. There is no quicker way to lose your audience, than to recite them a novel. Pitches above all else should be short, to the point, and respectful of everyone's time.

Your primary duty when pitching a game is to get an idea across as quickly and efficiently as possible and leave your audience wanting to hear more about it. Trying to paint a world in someone's head by summarizing story lore is asking a lot from them, and therefore they'll likely get confused, bored, or will just entirely zone out.

Actionable next step

Your pitch could be amazing, but if your audience has nothing to follow up on, you'll have wasted the opportunity. If your audience seems interested, don't forget to mention your…

- Kickstarter/Fig/Patreon crowdsourcing campaign
- Game page (wishlist/pre-order/buy)
- Social media presence (streams/podcasts/blog/etc.)
- Mailing list (the most important)

Some Pitch Examples

"You play as a recently-divorced tour guide at Gettysburg National Military Park just trying to keep it together" (paints clear picture, conversational & casual)

"Couch co-op twinstick glitch em' up for your home computer" (nostalgic, play on words in genre, know what you're getting)

"Meditative open-world vaporwave skateboarding game about being on the brink of life and death collecting *aesthetics* to reach enlightenment in skate-purgatory whilst trying to expiate your sins, help lost souls, find meaning in life and closure in death." (bit long, but demands a second take)

There are no rules

A good "elevator pitch" comes with time and practice. There is no formula to a perfect pitch, but a little enthusiasm goes a long way. The most important thing is that you believe in your game. If you are passionate when talking about your game, others will feel and understand that. Good pitches can be ruined with bad presentation and vice versa, so the sooner you start practicing pitch the better!

THE ART OF MAKING GOOD GAME TRAILERS

What a trailer needs to do

1. Capture the viewer’s attention
2. Hold onto the viewer’s attention
3. Communicate the experience in as short a time as possible
4. Leave them wanting more
5. Call to action

Sounds familiar? A good trailer is there to give your pitch while you’re not present to give it yourself, and when done well, can elevate it.

Steps of making said trailer

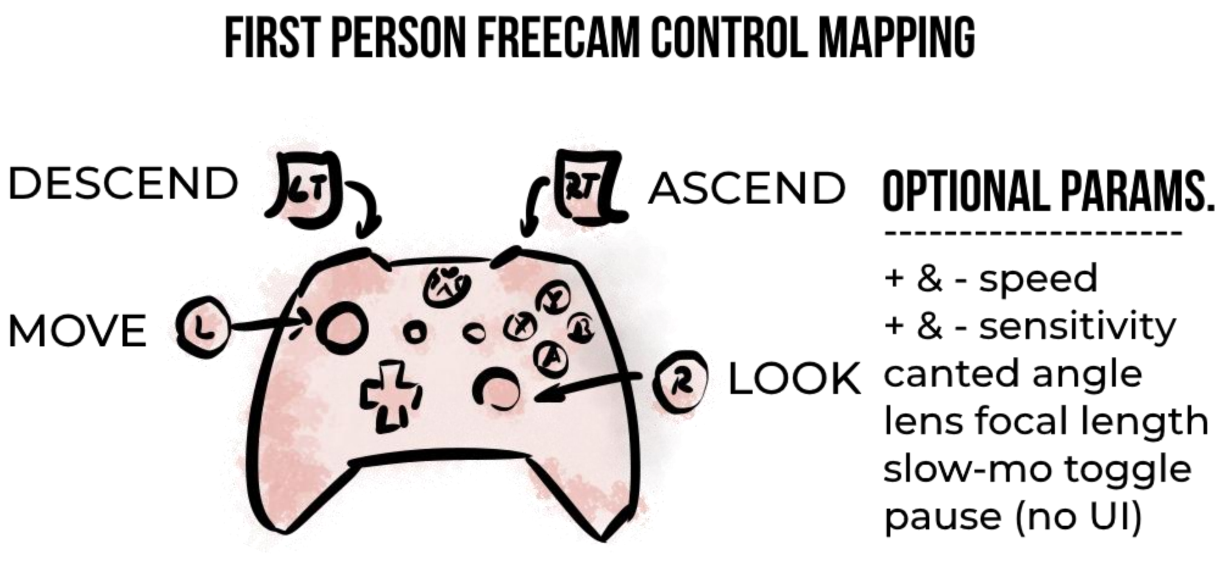
1. Think about what you want to select

   You can’t capture everything in your game within less than a minute, and if you can then your game doesn’t need a trailer. Storyboards work for some people, but given enough source material, a trailer will naturally come together in the editing process.

2. Give yourself the tools for good game capture

   Gamepads (controllers) are good for slow, smooth camera movement.

   Mouse and keyboards are good for fast and precise camera movement.



A toggle between gameplay and free cam spectator flythrough mode or two player co-op with one controller being a virtual camera operator for best results!

You'll want to implement a disable UI option to remove crosshairs and other unwanted user interface that takes up precious screen space. Because at the end of the day, most people will be viewing your trailer on their phone.

Metronomes when set to the tempo of the trailer music can help sync action to the beat. Your capture should only include in game sound effects, but not music. The music will come in the editing later. If you don't do this and just mute the in-game sound, the trailer will feel lifeless and dull with only music.

A level skip debug code can help save time when recording gameplay for a challenge-based game. Using NPC bots to play with, or against, make it easy to record multiplayer gameplay when you don't have friends readily available to record with you.

3. Capture *cinematic* gameplay
   All Windows 10 computers have the XBOX app which can be opened up in game by holding [WIN] + [G] and provides decent game capture. Nvidia has shadowplay, but there's also OBS. There are plenty of free tools to record your screen

   To ensure good composition while you're recording, try taping lines of string across your monitor to make the rule of thirds overlay your screen.
4. Editing

Editing is an artform unto itself. Learning all the different types of cuts and transitions can be overwhelming, but as long as you follow the 180-degree rule — try to match cut and cut on the action or beat – the action should be fairly easy to follow. If your game relies heavily on past conventions, go light on the text and let the gameplay speak for itself. If the game is a bit harder to follow from images alone, perhaps text slides (with legible consistent font) are necessary.

I recommend watching rocket jump film school's video essays and tutorials.

Additionally, Mark Brown's "Game Makers Toolkit" has a wonderful video on the subject of independent successful and not so successful game trailers.

https://youtu.be/OAH0MoAv2CI https://youtu.be/4CSYA9R70R8

Additional readings if you want to better understand filmic language:

Film Art: An Introduction by David Bordwell and Kristin Thompson

Introduction to Film Reader by Drew Casper

PUBLISHING, LANDING PAGE, AND PRESS KIT

To self-publish or not self-publish

You've got a game, now it's time to finish and release it…

Are you comfortable doing all these things on your own?

- Funding: self-funded, crowdfunded, venture capital?
- Marketing: public relations, advertising, event showcases, award circuits
- Production: QA Testing, Localization (translation into different languages), voice over, music (obtaining rights/finding composers)
- Distribution: digital, retail, merch (swag & plushies), etc.

If you are comfortable doing those alone, then you don't need a publisher.

If you aren't, you still don't need a publisher. You can still get one though, if you feel you don't want to go it alone and have someone in your corner. However, with time and effort, it is possible to do everything by yourself. Even with the "indiepocalypse", a good game, a little bit of a guerrilla marketing, exhibition, and luck can make your game stand out and get into the hands of those who will enjoy it.

What a publisher can offer you

- Storefront featured spots on launch day
- An ambassador for your game, someone who can take your game places and showcase it for you, or even pay your way to attend these game festivals.
- A personal relationship with an unbiased third party who can be brutally honest with you and give good feedback.
- $$$, usually in exchange for a cut of your game's profit.

What to look out for (red flags)

- You should always own your own IP (intellectual property)
- If they try to rush a contract on you. Deadlines aren't real. NEVER AGREE TO SOMETHING YOU DON'T UNDERSTAND (ESPECIALLY LEGAL JARGON). Ask questions and wait patiently for answers. You are in control.
- Know how the money flows. Never lowball them in needed funding for development salaries, rent, and overheads. Always leave buffer room in the budget to buy yourself extra time for development in case your production schedule doesn't go to plan. The less you have to revise "the deal", the better.

Just be honest and upfront about how much money you need to make the game.

- Do your research! Know the publisher's history, how they're funded, and BEWARE OF SCAM PUBLISHERS.

Other things to consider

- Registration of dev kits, and set up
- Live connectivity with console SDKs, how to save user data using their networks
- Entering catalog and marketing Info
- Pricing & release date availabilities (more complicated if your game is multiplatform)
- Certification (art *correct UI graphics, trailer and promotional materials branding*, bug free & other miscellaneous. tests)
- Final Certification & Post Certification (sending off your game to be rigorously tested and approved for shipping)
- Maintaining your title with updates

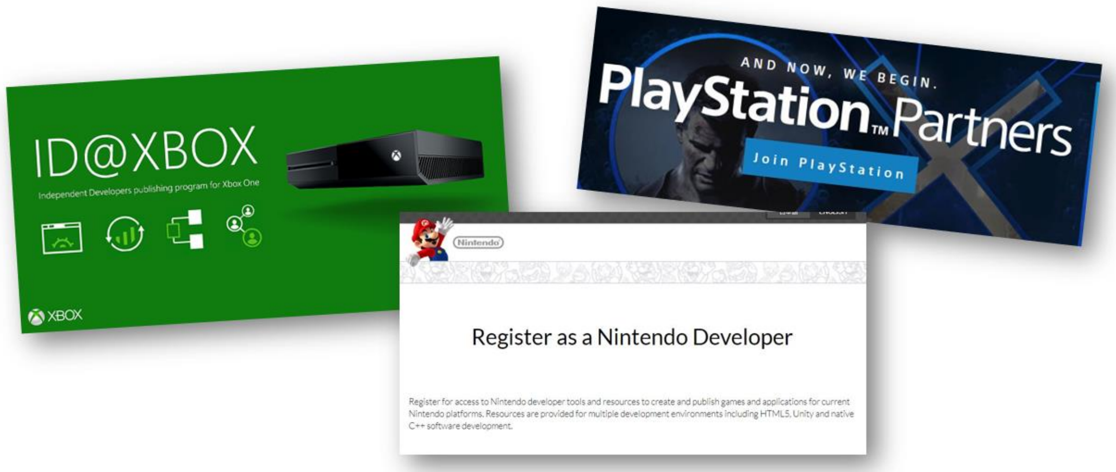
Interfacing directly with the distributor

If you want to get serious about releasing a game on a big platform like Steam or a modern gaming console, you'll need a company to interface with them. You should create an LLC, a "limited liability company". The company acts like a person once created. Instead of you being on the hook if your game gets sued, it's the company that gets sued.

Creating a company is different process depending on where you live, but if you live in the states it's usually goes like this.

1) Visit the Secretary of State website for your state
2) Determine your legal structure (sole proprietorship, partnership, corportation, or LLC)
3) Search their database and check to see if your company name is available
4) File the document to register your company name and have it notarized
5) After that, you pay a one-time filing fee and will be charged an annual fee to keep the company in good standing. Some states like New Mexico don't even have an annual required fee where others like California have an $800 fee.
6) Secure the business record, so nobody but you is able to edit the details.
7) Open a business bank account. You'll need a copy of the registration documents and certificate of good standing from the Secretary of State website.
8) Obtain a federal tax ID number (F.E.I.N) from the IRS.
9) Obtain a state tax number and wage withholding account from your state's Department of Revenue

10) Congratulations, you have a business now! Google and many other companies will now inundate you with a lot of mail advertisements for their services, including but not limited to free samples and many pamphlets.
11) You're now ready to apply to indie publisher programs



Register as a Nintendo Developer

Register for access to Nintendo developer tools and resources to create and publish games and applications for current Nintendo platforms. Resources are provided for multiple development environments including HTML5, Unity and native C++ software development.

Programs like ID@Xbox, Playstation Partners, and Nintendo Developers open up console development to anyone as long as they have a company of their own and a game that they believe will sell. There's a lot of paperwork involved, but the end of the day you get dev kits and are able to port your game to these platforms.

Console publishing aside, mobile app stores and digital marketplaces like Steam are now more accessible than ever, for better or for worse. If you don't want to shell out $100 USD on a publishing license, nor want to make the effort in creating a company, there's always free alternatives like Gamejolt, Itch.io, and Indie DB.

What's the cost?

The percentage platform holders take varies, but generally it's 15-30% of all revenue, and sometimes you only start receiving your revenue only after you've made ~$200 in profit. However, that's variable depending on the platform and contract you get. More indie focused platforms like Gamejolt and Itch.io let you decide what cut you want to give to the platform for hosting your game which is fantastic.

Why not launch to as many platforms as possible?

Each storefront has their own banner and thumbnail dimensions, as well as requirements for screenshots and trailers. Creating and managing multiple game pages can become tedious, especially when patching the game and uploading new builds to each of them for every different OS platform. There is a benefit for trying several storefronts, you may have better luck with some over others regarding exposure through features and a different community. A good number I've found has been three storefronts. Any more to manage becomes difficult for a singular indie developer.

What is a landing page?

With all these storefronts, trailers, screenshots, game descriptions and mailing lists, having a central hub for everything streamlines everything and allows you to present your game on your terms. If you have some knowledge in HTML, JS, and CSS then you can already go ahead and create a landing page website.

You'll need:

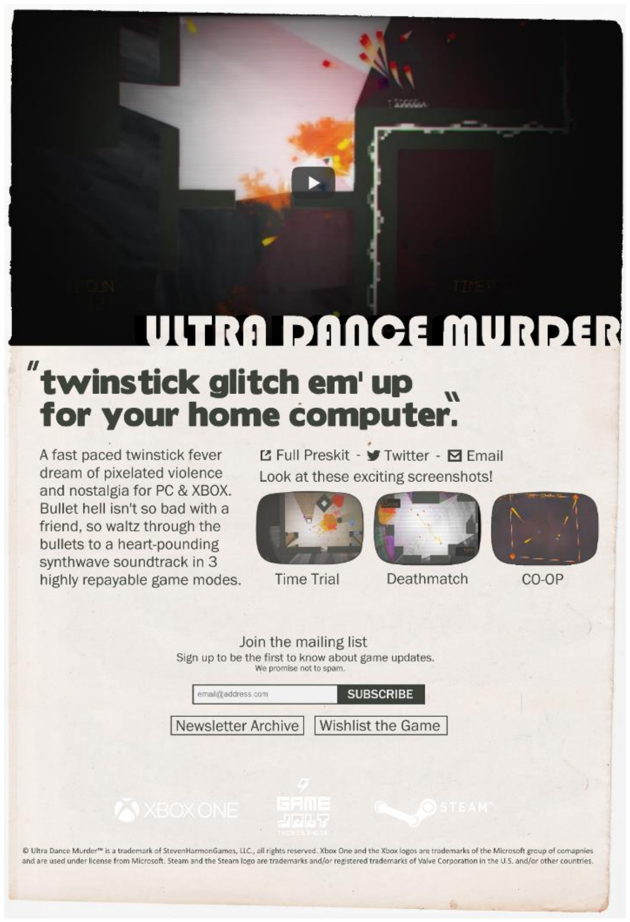
- A banner image for the title of your game
- A trailer of your game
- An embedded widget to buy the game/buttons to various storefront pages
- At least four nice high-resolution screenshots that show variety
- Links to social media for your game/studio
- Link to your press kit for your game (we'll visit this later)
- A mailing list widget for future updates on the game and future releases

What if I don't know HTML, JS, and CSS?

No website coding experience? No problem.

There are many WYSIWYG (what you see is what you get) website editors on the market. Adobe has Dreamweaver (coding intensive but flexible) and Muse (no code). However, there are many great free alternatives. Mobirise4 is a simple templated approach to web design, and Gamejolt has a feature called Gamejolt Sites that includes a great editor but also supports hosting .zip game landing pages exported from other third party offline editors. Additionally, if you host through Gamejolt you can use your own domain name that you've purchased through Google Domains, Go Daddy, or any of the other providers to show up easily in search engines.

First step I'd recommend is to mockup what you'd like your landing page to look like in an image editing package. Try to relate the visual style of the landing page to the aesthetic of your game as best as possible for consistency. For example, I made my game's home page look like an ad inside an old gaming magazine from the 90s.



"twinstick glitch em' up for your home computer."

A fast paced twinstick fever dream of pixelated violence and nostalgia for PC & XBOX. Bullet hell isn't so bad with a friend, so waltz through the bullets to a heart-pounding synthwave soundtrack in 3 highly repayable game modes.

Full Preskit - Twitter - Email

Look at these exciting screenshots!

Time Trial | Deathmatch | CO-OP

Join the mailing list

Sign up to be the first to know about game updates.

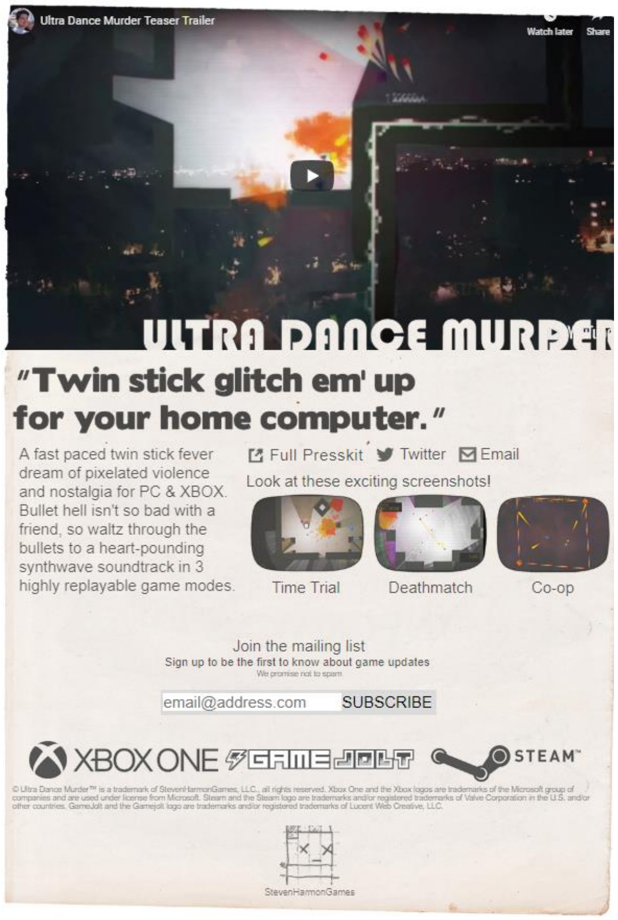
We promise not to spam.

email@address.com SUBSCRIBE

Newsletter Archive | Wishlist the Game

© Ultra Dance Murder™ is a trademark of StevenHarmonGames, LLC., all rights reserved. Xbox One and the Xbox logos are trademarks of the Microsoft group of comapnies and are used under license from Microsoft. Steam and the Steam logo are trademarks and/or registered trademarks of Valve Corporation in the U.S. and/or other countries.

Landing Page Image Mockup



"Twin stick glitch em' up for your home computer."

A fast paced twin stick fever dream of pixelated violence and nostalgia for PC & XBOX. Bullet hell isn't so bad with a friend, so waltz through the bullets to a heart-pounding synthwave soundtrack in 3 highly replayable game modes.

Full Presskit Twitter Email

Look at these exciting screenshots!

Time Trial | Deathmatch | Co-op

Join the mailing list

Sign up to be the first to know about game updates

We promise not to spam

email@address.com SUBSCRIBE

© Ultra Dance Murder™ is a trademark of StevenHarmonGames, LLC., all rights reserved. Xbox One and the Xbox logos are trademarks of the Microsoft group of companies and are used under license from Microsoft. Steam and the Steam logo are trademarks and/or registered trademarks of Valve Corporation in the U.S. and/or other countries. GameJolt and the Gamejolt logo are trademarks and/or registered trademarks of Lucent Web Creative, LLC.

StevenHarmonGames

Final Online Landing Page

The Press kit

Getting your game noticed today is harder than ever before. With so many games all vying for attention, there are simply not enough eyes nor time to go around to all of them. It's vital to your game's success to make it as easy as possible for press, game journalists, and let's players to cover your game, and a press kit is the best way to go about doing that. There's plenty of free tools online to help in the process of making a press kit. Rami Ismail from Vlambeer has a free template called Presskit(), Indie DB automatically generates and hosts a press kit when you upload your game onto their site, and if neither of those suit you, you can format your own however you'd like when you create your game landing page.

The format of your press kit doesn't matter as long as it includes the following information:

- Name of the developer (studio name) and publisher (if self-publishing, leave blank)
- Credits of everyone involved in the development hyperlinked to their sites
- Release date
- Platforms released on (desktop/mobile/etc.)

- Link to game page
- Short pitch description
- Short features list
- Short history of the game's development (Why was the game made? Any challenges? Interesting tid bits that would make for a good story?
- List of the tools used and their respective hyperlinks
- Monetization permission (Do you allow let's players to make monetize and profit from videos of them playing your game?)
- Media (A link to somewhere where all the thumbnail images, logos, hero images, screenshots, gifs, and trailers that anyone would need to write a review on the game are available to download)

GETTING NOTICED AS AN INDIE: SOCIAL MEDIA & MARKETING

Post Quality Content & Post Often

Social media is a full-time job when done well, but for independent developers it's a challenge to strike a balance between generating an audience for your game while actually developing the game. Reminding people you and your game exist while also bringing something interesting and new to the table in terms of posts is a hefty order, especially if you're not working full time on gamedev. So here are some tools to lighten the burden. Just remember! Plan your posts ahead of time in a separate word document with spellcheck!

TWEETDECK

https://tweetdeck.twitter.com/

Price:
Free

Available for:
Web

Notes:
Schedule tweets ahead of time, extra step before posting to avoid mistakes, powerful tools for monitoring and engaging with social media trends

GIF CAM

http://blog.bahraniapps.com/gifcam/

Price:
Free

Available for:
Windows

Notes:
Gif capture & editing suite

SCREEN TO GIF

https://www.screentogif.com/

Price:
Free

Available for:
Windows

Notes:
An alternative Gif capture & editing suite

EZGIF.COM

https://ezgif.com/

Price:
Free

Available for:
Web

Notes:
Convert between video and gifs, edit and compress gifs

Make Your Social Media Work for You

Social media isn't one-sided. Good social media evokes a response and in today's algorithmic stream of content, engagement is the way to be seen, but that isn't to say every post should be sensational. Furthermore, if you're just trying to catch up to the algorithm through yellow journalism tactics, there's no way to tell if what you're doing is actually working on not… because the algorithm always moves. What get's attention one day gets ignored the next. Social media is a complex enigma, and instead of wasting brain power trying to wrestle with that – just use it to make your game better and I assure you that eventually you will find your audience… with a little help from the use of the appropriate hashtags.

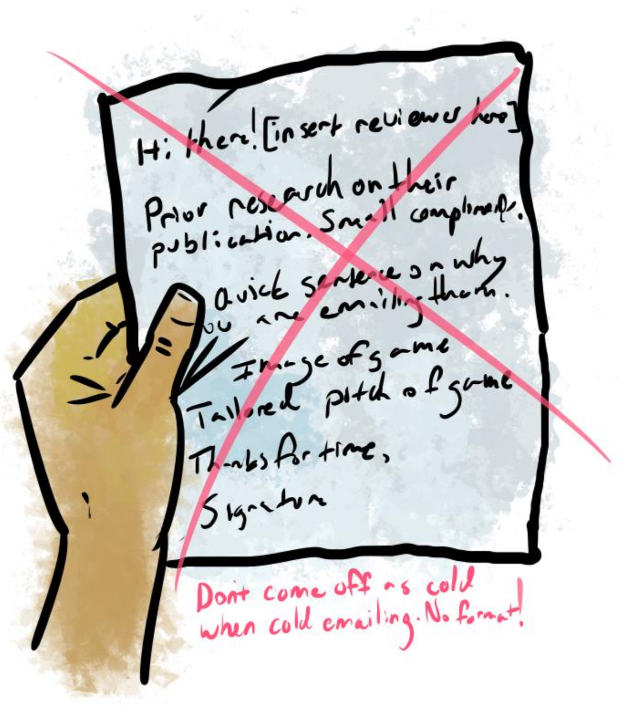
#gamedev #screenshotsaturday #madewithunity
#indiedev #indiedevhour #indiegames
#AR #VR #pixelart #blocktober #lowPoly #VFX
#WIP #artistsontwitter #art #UE4 #gamemaker

Early in development, you can share early prototype footage in the form of gifs or short videos and solicit feedback. Not only does posting early on in development let you stress test the marketability of your idea, but also helps you iterate upon it. Kill two birds with one stone and share your development! Think of your Twitter as an informal developer diary. Just be sure to stay transparent that the game is a work in progress.

On Brand

Social media is more than pitching your game 1000 times to a choir of crickets. Social media is more than catchy taglines and behind-the-scenes development. Social media first and foremost should play to your strengths.

I'm not saying everyone should start a separate podcast, blog, pervasive ARG game, or YouTube channel to funnel more people to their game. However, finding something of collateral that you can offer publicly that may bring more eyes to you and your main work is the quickest way to build an audience. If you can be yourself online and have your audience get to know you, they'll want to support you and your work. Do not reach out only to popular influencers; target smaller publications and let's players/streamers who have covered games similar in scope to yours in the past, because they'll be more likely to give you the time of the day when you send a cold email. Be personable!


Hi there! [insert reviewer here]
Prior research on their publication. Small compliment.
Quick sentence on why you are emailing them.
Image of game
Tailored pitch of game
Thanks for time,
Signature
Don't come off as cold when cold emailing. No format!

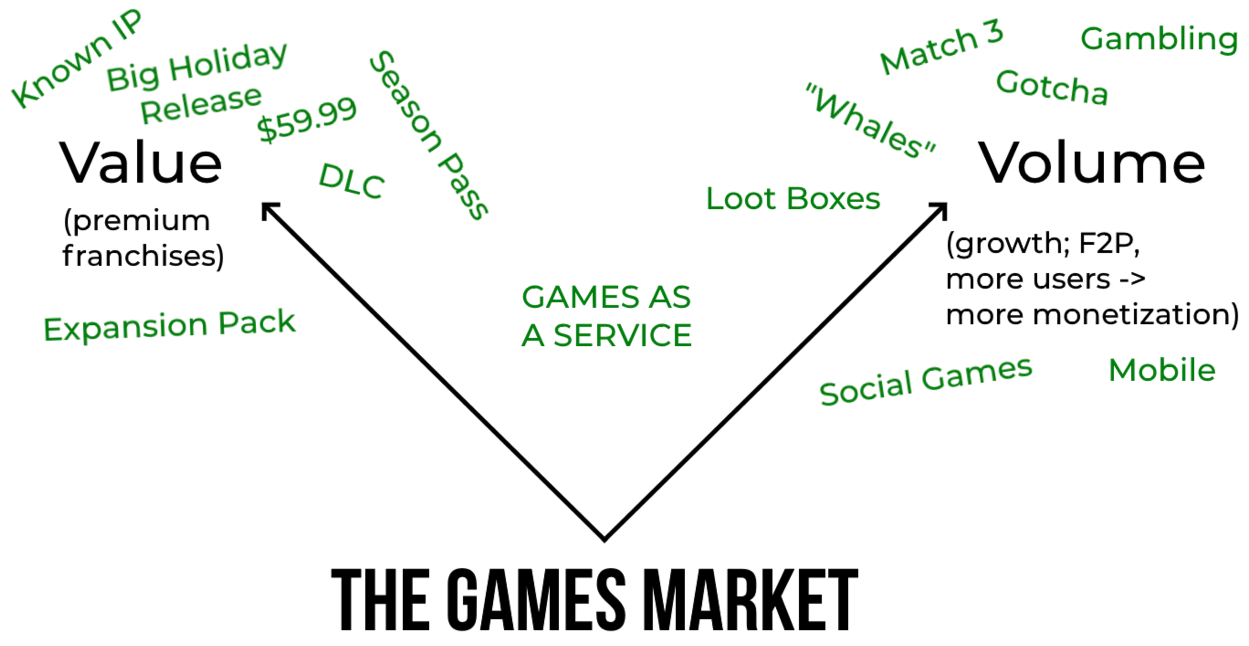
Just be yourself. Don't be afraid to self-promote, but don't spam. Avoid showing the same thing twice, and don't be afraid to mix things up once in a while. Post in the morning, post midday, post at night. Be professional. Be personable. Be whoever you want to be online, but most importantly, don't be a jerk.

For some, it happens overnight, and for others it takes years. Regardless, you don't need likes to like yourself. You don't need hearts to love yourself. You don't need approval nor validation to follow your dreams. Running social media for a game is a lot like creating a dating profile – don't worry if there's a lull. As long as you stay true to yourself and stay on brand, your audience will find you.

For every game, there is a player.

BUSINESS: UNDERSTANDING THE MARKET & MAKING MONEY

The Big V



Currently in 2019, the games market is split between value and volume. Making a living in games means either working on a premium franchise with a high price point and a publisher or the free to play mobile social market. While existing well known IPs dominate the top grossing charts, freemium titles mean more users which equates to more eyes, more ad impressions, and more monetization opportunities. According to a report by Swrve "0.15% of Mobile Gamers Contribute 50% of All In-Game Revenue" and these "big spenders" are called "Whales"; these "whales" are being exploited via their addictive tendencies, impatience, and short attention spans. There is space for premium single player experience-based games, but they are a bigger financial risk than the two dominant avenues of "the big V".

If the game doesn't get featured, you're screwed.

A temporary game feature (product placement) on the front page of a storefront can be the difference between selling 2k copies to 200k copies on launch day. 10k copies sold on launch day for an indie is considered "a success". Some publishers have deals and promotions that can highlight your game. Getting your game featured when you're self-publishing can be due to your game's merit, a promotion, or just luck. There are some things you can do to increase your luck from networking opportunities at the game developers conference to launching your game at a good time (in a lull when no big-name games will overshadow yours).

Other places to understand the market

- Look at stock exchange earning reports of publicly-traded companies

- Press releases also are a great resource for current trends and stats
- Trade organizations like the ESA Foundation have an annual report
- Market research firms have the best data: See New Zoo & Digi Capital
- Storefront analytic tracker pages App Annie, Steam Spy, etc..
- The GDC Vault has plenty of great presentations from every year of GDC

Pitching your game to investors & publishers

Pitching to investors is less about selling the fantasy or experience, but rather about selling yourself as a competent person and your game as a sure thing!

When convincing an investor, prove these things.

- Need in market for this
- Want from consumers for this
- I can build it. Proof of concept (selling yourself & team)
- Consumers will be there for it when it launches (data to support)
- I can monetize the audience, return on investment, and make a profit

You must showcase the expansiveness of your idea, future thinking possibilities such as trans media (movie & television spin offs, merchandise, comic books, etc.) while also keeping the presentation crisp (showing expertise, nothing over obvious nor hard to digest, a presentation that doesn't overstay its welcome). Be deadly serious but be authentic in your personal relationship with and investment in the idea; showcase your passion, back it up with research. Know who you're pitching to and tailor your pitch deck (see below) to them. If they're smart, don't spend too much time on the market dynamics slides and abbreviate - they already know the market and lingo.

Pitch deck template

- Title page (company name, website, professional email [No Gmail])
- Agenda (table of context)
- Product snapshot (sets tone for product, release platforms and territories, target customer gender and age range)
- Company Snapshot (Team composition - CEO, CTO, Creative Director, etc.)

  Pro tip: prove your credibility, name drop when possible, look smart

- Vision (opportunity in the market, studio's mission statement aka "goal", big ambition pipe dream for future)
- Market dynamics (write a narrative backed by recent data and trends that explains exactly how big your total addressable market is and how there's money to be reaped there.)

  Pro tip: set up the perfect environment, instill a sense of urgency, "get in on the ground floor now or miss the boat!" Show numbers, cite sources.

- Product overview (explain your game's selling points in depth but only show them the tip of the metaphorical iceberg)

- Key revenue drivers (how do you plan to continually bring in money)

  Pro tip: Get creative with ideas on how to deliver the game. Come up with a game plan and demographic statistical information based on comparable games to make broad estimates

- Key competition (using a venn diagram or spreadsheet compare and contrast similar games in the market sharing a set of characteristics and explain why your game is unique and superior)

  Pro tip: make the genre market your game is in an attractive one

- Development plan (roadmap/timeline of development beats – preproduction, production, vertical slice “beautiful corner of game, e3 presentation build never to be seen in final game”, alpha “feature complete”, beta “content complete with bugs”,”, soft launch “launching in one territory as a test”, gold master “finished and ready to ship everywhere”)

  Pro tip: include the conference circuit you plan to tour to show off the game

- Business development timeline (partnerships, IP, legal – optional)

  Pro tip: Prove to them that you aren’t dependent on a platform. Cut out “the middleman” and show that you are in control.

- Audience acquisition strategy (paid vs. earned media, how do you plan to get the word out about your game? How do you intend to keep your audience coming back to the game / franchise and keep paying?

This is a joke, don't actually use this. Do actually think how you'll leverage tools to monetize your game.

- Planned use of funds (come up with a realistic budget and do the math for all expenses – development salaries of team, rent, marketing costs, travel costs to festivals & conventions, overheads, legal costs, buffer room *be generous*, and then a total of everything above)

  Pro tip: showcase your budget via a labeled pie chart

- Title page reiterated with thanks, contact info, and time for questions

Extra Extra Read All About It!

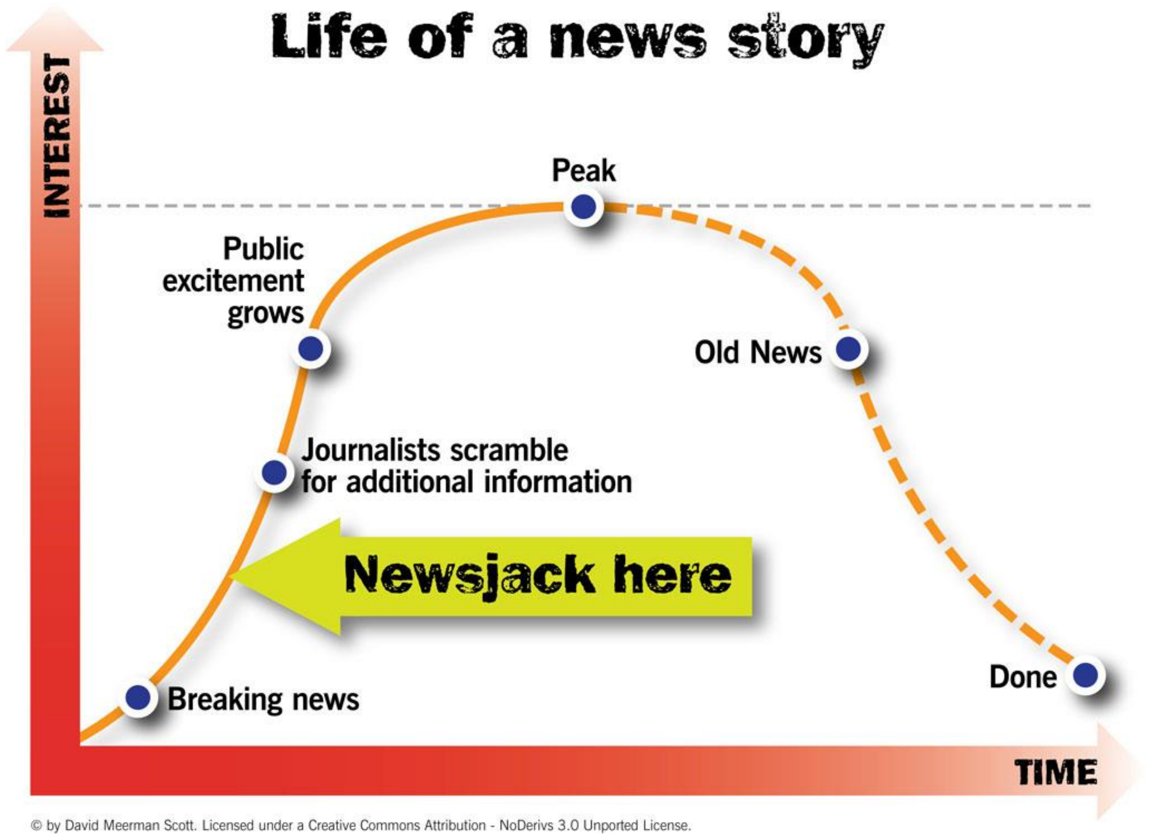
Staying relevant is staying alive in the fast-moving world of games journalism.

Here are some tactics to keep your game in the limelight and its story on track.

- Develop thought leadership (be an expert in your field with opinions) Become Rami Ismail, be someone who press want a comment from.
- React real-time to events and news in the world (tweet regularly)
- Comment on regulatory change in your industry (blog regularly)
- News releases: have a new take on an old problem: "we've solved the VR locomotion problem!", serve a unique market "A game with revolutionary accessibility features!", announcing awards and conferences, anything interesting...

  Pro tip: News releases should be on a Sunday night, Monday morning, or Tuesdays. Never on a Friday afternoon. Leave time for games journos to write their own articles.

- Before big releases, offer exclusive coverage to the story you're pitching to journalists. Offer them something that will impress their editor.
- News Jacking: Take control of a news story through speed and agility by inserting yourself into a breaking news story. Don't capitalize on a tragedy or try to force it; not all stories can nor should be news jacked.



© by David Meerman Scott. Licensed under a Creative Commons Attribution - NoDerivs 3.0 Unported License.

11. WHEN GAMES BECOME WORK: TAKING CARE OF YOURSELF

IMPOSTOR SYNDROME

What is impostor syndrome?

- Self-doubt and anxiety about your place in the industry
- Perfectionism (setting high goals and falling short, feeling like you need to prove yourself to defend and affirm your position)
- Fear of failure/fear of rejection
- The feeling of not being “good enough” and the fear of being discovered, outed, and exiled

Usually impostor syndrome comes after you get any recognition that you believe you don't deserve, that it was due to luck, or any other forces outside of your control. Impostor syndrome is the lurking pit in your stomach that reminds you that you have no idea what you're doing, and you'll be found out a fraud and your life will soon be over. A feeling of impending doom. It happens quite often in fields of constant critique (like gamedev and art). We never fully feel like we've "made it" and belong here until we've contributed something worthwhile to the field. That's super subjective and the cult of awards ceremonies and the “indie darlings” inner social circles and cliques don’t help.

How do I know if I have impostor syndrome?

In every field there’s impostor syndrome. It’s quite common, especially for those coming from marginalized backgrounds and not a part of the industry culture.

If you’ve commonly

- Rejected compliments (either externally or internally)
- Felt like you were presenting a “false self” to others when making first impressions
- Felt pressured to not lose face in front of peers
- Refused help because you didn’t want to appear incapable, vulnerable, weak. Worn too many hats or took on too much work just so you could prove you didn’t need anyone.
- Deliberately hidden your achievements when in conversation, because you didn’t feel you achieved them.
- Became addicted to validation and praise by means of achievement

Above are the symptoms, but the real effect impostor syndrome can have on someone can affect if they’ll go out of their way into dangerous territory or hold back. This can be harmful and lead to stalled projects, unfinished masterpieces, and general unhappiness.

What's to be done?

I'm not a psychologist or counselor. How can I possibly help someone redefine their personal worth, method of internalizing accomplishments, and way they view themselves in the world? How can I do that when I can't even do it for myself most of the time?

The answer is I can't.

Yet, just being aware of it is half the battle. Since impostor syndrome is a cycle, by becoming self-aware of it you can stop yourself in the act of the symptoms. Take those negative feelings and attribute them to just being an outsider and realize you aren't alone in feeling that way.

Pro tip: Treat yourself for all the small victories, daily if possible, but at a surprise time. "You've gotta' accentuate the positive."

Fake it till ya' make it!

None of the advice above helping you?

You're too far deep in "faking it"?

Okay… Accept that.

"It" may never end. You may never arrive to "it" and will always continue "faking it" forever.

Revel in that truth.

You are the trickster extraordinaire! You've fooled everyone into thinking you are a competent developer/artist/etc. Now rest on your laurels of deception, your secret of being a fraud is safe. Now go on with business as usual!

What if I don't have impostor syndrome?

That's wonderful! Be aware that there may be more people like you working in games which may make you feel more comfortable and entitled to be there. Just do what you can to support those who may not have those same support structures in place.

STAYING HEALHTY AT CONVENTIONS

Showcasing your game at a convention is more than just lugging monitors, computers, cords, décor, peripherals and printouts to a table and calling it a day. Demos need to be reset, you'll always be talking to someone, and whatever can go wrong will go wrong. For those who may not be good with crowds, lots of noise, and the occasional rude con-goer – conventions will be a challenge. Being enthusiastic 24/7 for 12-hour shifts multiple days in a row while standing sedentary and not losing your voice is a skill that comes only with practice.

Your support structures at the expo

Your fellow developers and their neighboring booths are arguably the most valuable part of attending these conventions. Network, make friends, and follow up afterward with an email to keep in touch. During the convention, sharing is caring, so be sure to bring some extension cords with extra slots, snacks, and offer to watch each other's booths for bathroom and food breaks.

Stay hydrated!

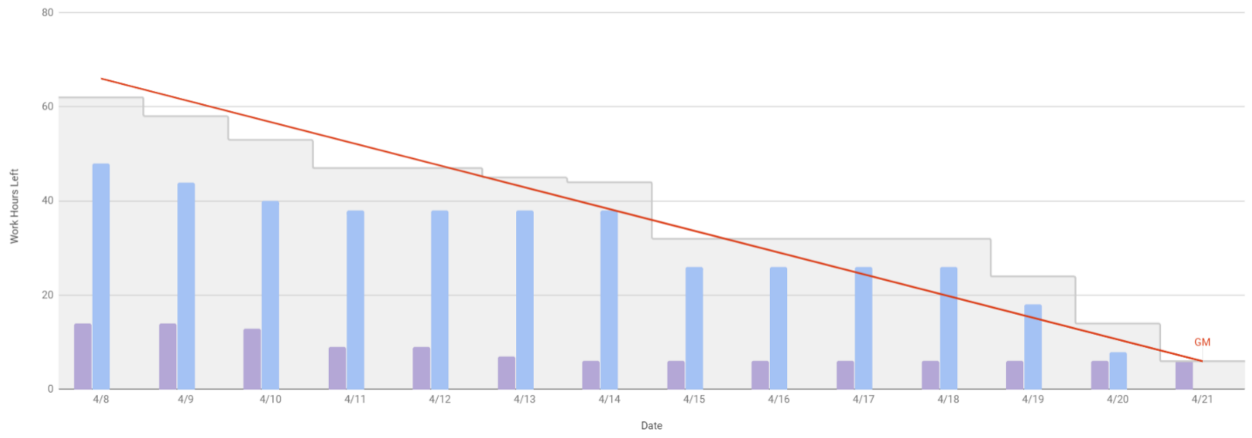
I cannot stress enough how much drinking water non-stop can help keep you healthy and talking throughout the conference. Your pee should be clear and your trips to the restroom frequent. Use them as your body's natural excuse to take breaks to walk around and temporarily enjoy the convention as a con-goer, not an exhibitor.

Always be prepared!

Never rely on anything that isn't promised in writing from the site. Bring your own tablecloth, signage (height wins the day), pen and paper, duct tape, string, speakers and headphones, controllers, computers & monitors, etc. Have a flexible set-up that can be erected within 15 minutes or less anywhere. Don't give yourself a hernia trying to lift your materials, there's no shame in wrapping all your electronics in blankets then dragging them in luggage cases or atop dollies. Set up early, give yourself time to get acquainted with the space and its various bathrooms, exits, and if you're lucky, private exhibitor lounges.

DEADLINES, CRUNCH, AND BURNOUT – OH MY!

Even in agile workflows where all work is broke down into short two-week periods called "sprints", the entire project development cycle shouldn't feel like a never-ending sprint. While game jams, short 24 – 48 hr./1 week game projects, are fun ways to break outside of your comfort zone and learn new skills and tools in a low pressure environment, working non-stop with little-to-no sleep nor breaks shouldn't be your norm when it comes to working.



Crunch Culture

"Crunch" isn't the occasional overtime that is asked of employees, it's something more sinister. Crunch is when that unrewarded overtime becomes the normal and expected, when that time is unpaid and even rewarded, when your dream job becomes a waking nightmare. Crunch is usually attributed to the final weeks of a game's production cycle, but that's not always the case. It can happen because of poor production management, unreasonable publisher expectations, trying to reach tight deadlines and big milestones. However, the most insidious thing about crunch is that even if many of those external pressures are gone, the culture surrounding it continues to perpetuate the practice – even in independent work and academia. Crunch is addictive.

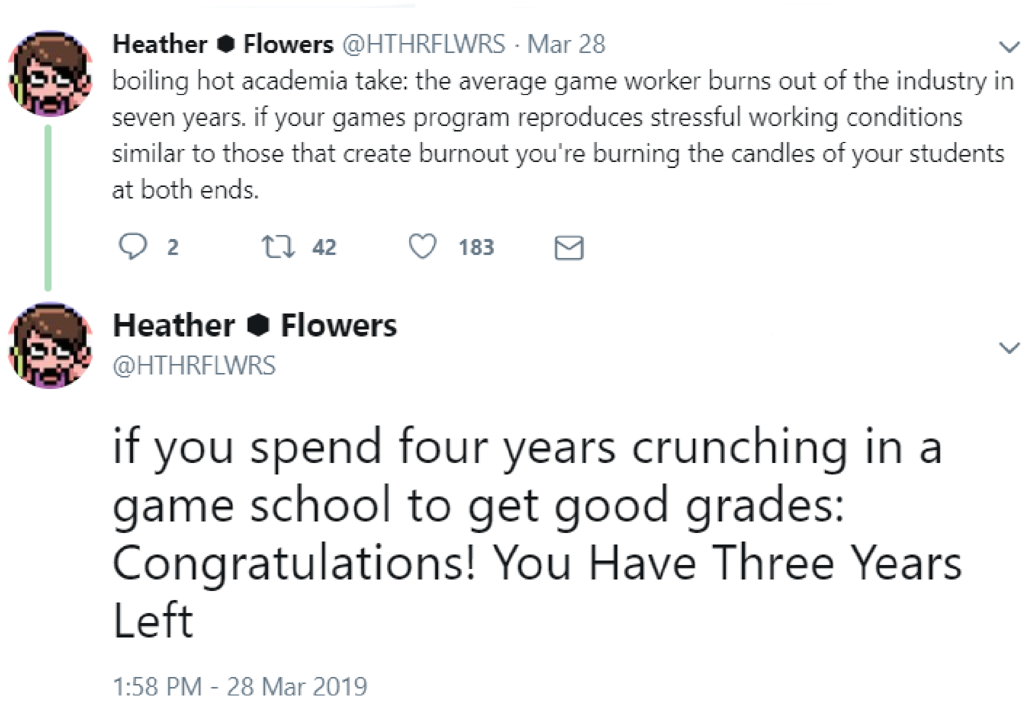
There's nothing wrong with hard work and ambition, but when work takes over all of your life and the other aspects of your life fall by the wayside – that's crunch. Crunch when used in moderation can be a short-term solution to a fundamental problem, but after being used once it becomes easier and easier to use again and harder to avoid. Crunch produces visible results quickly and part of the joy of gamedev is seeing the fruit of your labor in the expressions and validations from those playing your work. Crunch is like performance-enhancing drugs, but instead of getting disqualified from the sports competition, nobody is stopping you nor preventing you from getting more. Consecutive all-nighters are seen as a right of passage and suffering and sacrificing for one's art is seen as dedication. The abysmal amount of sleep becomes a watercooler point of bragging and is met with either praise or sympathy, neither of which do anything to stop the behavior.

Crunch is seen as natural and expected, because of survivorship bias with those whom have crunched in the past. However, it doesn't have to be that way – there are alternative methods in balancing quality of life and self-care with working in games.

A Hot Take

Institutions such as colleges and universities are present to prepare students for the workplace of tomorrow, but when that workplace has high burnout and turnover rates with massive seasonal layoffs and exploitative practices preying on passion and competition, it's obvious why mental breakdowns and health problems are common and widespread in some of the top performing games, animation, art, and CS programs in the country. When academia becomes a microcosm of the industry at large without enough theory classes to think critically about the problems plaguing the industry, it becomes a part of the pipeline supplying disposable workers.

One of the most talented and prolific developers I know, Heather Flowers once told me "if you spend four years crunching in a game school to get good grades: Congratulations! You have three years left" [three years left of the current game worker's seven-year burnout rate]. Seeing as I personally am in my seventh year of game development and witnessed my relationship with my work become more toxic ever since I started college… this nugget of wisdom has stayed with me.


Heather ⬢ Flowers @HTHRFLWRS · Mar 28

boiling hot academia take: the average game worker burns out of the industry in seven years. if your games program reproduces stressful working conditions similar to those that create burnout you're burning the candles of your students at both ends.

2 42 183

Heather ⬢ Flowers
@HTHRFLWRS

if you spend four years crunching in a game school to get good grades: Congratulations! You Have Three Years Left

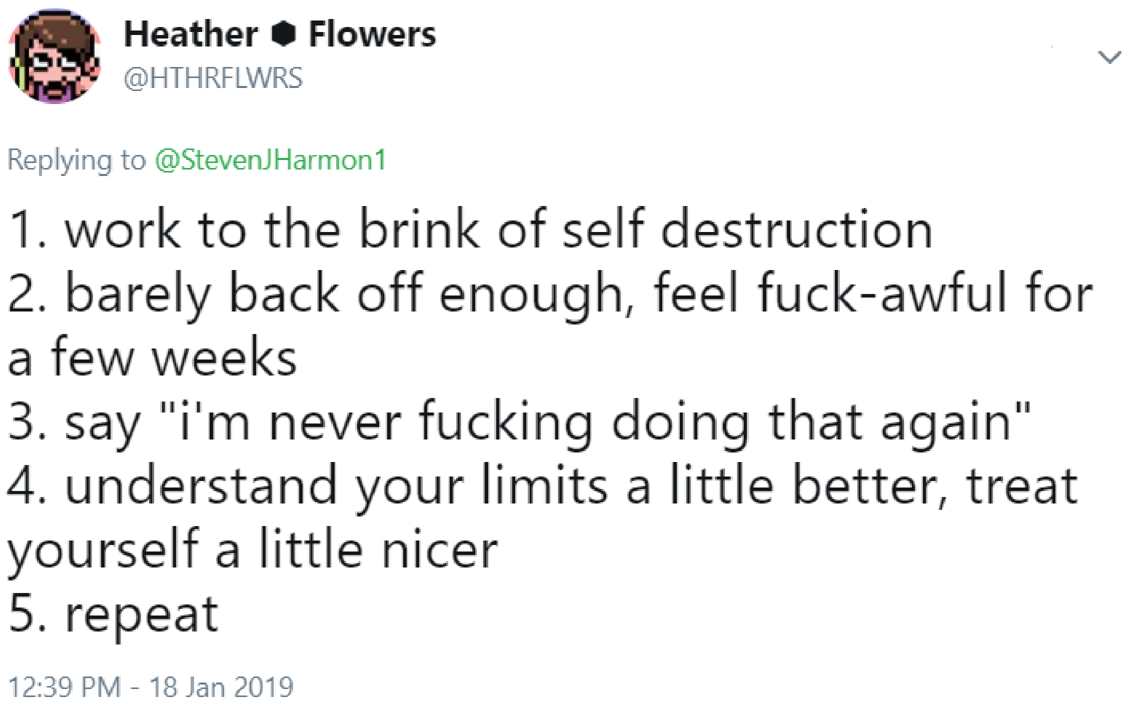
1:58 PM - 28 Mar 2019

One solution to crunch that some universities (publicly criticized of enabling crunch culture in the classroom) say they employ is changing their language, quite similar to how the !Kung, also known as the Ju'Hoansi, insult their hunting game to keep male ego at bay and their society civil. However, students who willingly crunch don't enjoy being chastised for creating polished portfolio ready work when they're just trying to get hired. And on the other hand, actions speak louder than words when high-performing students that crunch are shown favoritism and are offered opportunities that their non-crunching counterparts are not.

My advice for improving game academia: full transparency in grading, setting (realistic) expectations, and above all compassion for students. Courses that scale to what you put in is what you get out, with no shame or difference in grade are the most beneficial. Everyone starts from a different place and as long as students are showing some progression — that's all that matters. More focus should be placed on the process rather than the final polished product. While schools should keep current with industry trends, there should be an equal number of classes outside of what's commercially lucrative. Classes with corporate sponsors under the guise of "real world experience" are free labor that may not actually benefit the students.

You're damned if you crunch, you're damned if you don't – it's a catch 22!

The best piece of advice I've gotten about how to work in this ambiguity was also from Heather. She outlined her process of working herself to the brink of destruction, reflecting on her limits and through iteration, then treating herself better each time.



Heather ⬢ Flowers
@HTHRFLWRS

Replying to @StevenJHarmon1

1. work to the brink of self destruction
2. barely back off enough, feel fuck-awful for a few weeks
3. say "i'm never fucking doing that again"
4. understand your limits a little better, treat yourself a little nicer
5. repeat

12:39 PM - 18 Jan 2019

I believe this process is the most forgiving and most practical for anyone stuck on steps 1,2,3, and 5. And while it's nice to think knowing your limits better will stop crunch outright, the honest truth is – sometimes crunch may be outside of your control. Sometimes crunching, while not explicitly asked for, is the only way to make enough to make ends meet and survive. So if 5 step plans aren't your thing, perhaps a story about two wood cutters will be (also shared to me by Heather).

Burnout

While difficult to pinpoint in a creative industry that you're passionate about because it's something that happens gradually, there are symptoms to over-exhaustion from your work.

- Being easily overwhelmed by small tasks
- Dissociation with one's work "I made that, but it's not mine"
- Difficulty concentrating and getting visible work done, procrastination
- Insomnia, lack of appetite and self-grooming, self-isolation "hermitage"
- Anger, frustration, apathy, denial of burnout

In a society that values a person based on their productivity, it's hard not to be busy all the time. However, down time is absolutely necessary for your health and creative output. Down time isn't looking at social media or reading articles as "research"; DO NOT feel guilty for not constantly working and producing content. Whether that's meditation or exercise, anything where you can turn off your brain for a little while is helpful.

Why is taking in no stimuli good? Isn't the brain a "use it or lose it" system?

The brain is a "use it or lose it" system, so keeping in constant practice is important. However, did you ever notice how your best ideas come to you in the middle of the night or in the shower? Have you ever had an epiphany when zoning out, staring into nothing blankly? Do you daydream?

- i at komme i en særlig stemning
... 'en måde at arbejde på' -

John Cleese has this talk on "Creativity In Management" where he outlines an "open mode" and "closed mode". The open mode being the relaxed and playful and the closed being the more rigid and focused state where you typically get the most work done. However, when it comes to generating ideas, collecting inspiration, processing information, and receiving feedback, the open mode excels.

My personal definition for burnout is being so mentally exhausted from work that you are unable to return to the open mode and have become resentful towards the project, the team, or yourself.

Time heals all wounds

Taking steps to reduce crunch and prevent burnout is advised, but all the advice in the world can't stop people from learning it for themselves the hard way. Just know that taking a step back for a while to focus on yourself is okay, there's no shame in it.

YOUR MENTAL HEALTH & WELL BEING ISN'T A GAME

Game development can and likely will wreck you, but always remember why you got into it in the first place. It's those reasons that will guide you to making the best decisions for your happiness and creative fulfilment.

12. MARKETING YOURSELF

Resume Advice

Your resume should be able to be scanned within six to ten seconds

When you're applying for a job, you won't be the only one applying. There will be dozens of other applications in the same stack for review, and your job is to make your resume stand out from the others and get you an interview. This shouldn't be done through flashy art and formatting, because applicant tracking systems (ATS) may not be able to parse them into plain text well. No fancy formatting. Keep it simple, clean, and readable. Make the hiring team's job easy and focus on the employer's needs, not yours. Some employers spend less than 10 seconds to make a decision after an initial pass on a resume.

Your resume is there to make a good first impression and get your foot in the door, not get you a job

Your resume's sole purpose is to hook the employer in and land you an interview, nothing else. Don't overshare; leave them wanting more. Think of your resume as "the audition" and your interview as "the callback".

Relevant work experience only

This is a touchy subject, but generally I'd recommend not putting where you went to high school, your past retail/service industry job, honors society. Nobody cares about those, and unless you can make an argument onto how those positions have made you a perfect candidate for this job, then leave those details out.

Work Experience > School Experience

If you're a student and don't have any work experience yet, that's okay. However, employers want to see the passion and dedication to your craft that can only be seen through finished personal and professional projects. A college degree only shows employers that you were able to finish your degree, nothing more.

Quantify, Quantify, Quantify

The format of a resume should go as follows:

Title of Position + Employer, your responsibilities (what you've done)

Be sure to quantify your previous responsibilities and make everything sound impressive. Never lie, because they will independently fact-check everything, and if something is off then your application will become null and void.

That being said, do not quantify everything. Don't quantify your skills. Progress bars, percentages, loops, and rings make you look incompetent. You either have the skills at the level the employer is looking for or you don't. The only skills that could be quantified on a resume are languages that have distinct levels to show your progress in learning. For example, Mandarin has HSK proficiency levels.

They don't know lingo

The rule of thumb for acronyms and abbreviations is you can use them to display your knowledge of your field's lingo (if it's common knowledge) but if it's a proper noun that's not well known, then you should include the full title.

Give context, and spell it out if you must. Otherwise, the employer won't know what to make of it and pass over it.

Spelling and grammatical errors will ruin your resume

It doesn't matter how good the content is on your resume, if the viewer sees any errors in your resume it will come off as lack of thought and effort, and that assumption will carry over to the rest of your work. Get as many eyes in front of your resume as possible, listen to feedback, and make many edits because a resume is an ever-changing document.

Know the difference between a resume and a portfolio

A resume is just a plain text overview of your relevant work, accolades, skills (hard and soft), and education (education overview should be no more than one sentence; it's not very important). No pictures belong here.

A portfolio is a portable showcase of your talent/experience. Pictures are okay here! Showcase your work with screenshots, concepts, design schematics, etc.

Your selfie does not belong on your resume. Period.

Unless you're an actor or model, your headshot should not appear on your resume. It is illegal for employers to discriminate based on race/ethnicity, so they'll try to avoid paying attention to the photo or even skip it altogether.

Privacy is important!

Do not put your address on your resume. If the company is located in the same city as you, you can include that you are "LA-based" so they know relocating won't be a barrier. Otherwise, leave out where you live completely. This resume will end up online and you don't want a stranger to know where you live.

What to include:

- Name
- Website (personal portfolio website)
- Email (yourname@yourwebsitedomain is most professional)
- Linked In
- GitHub/Art Station (whichever is applicable to your field of practice)
- Phone number is iffy, if you do include it – get a Google number that reroutes to your real number so in case you get spammed you can just shut off that number.

“References upon request”

Is a given. <u>If it goes without saying, then don’t say it.</u> Employers already know if they want to contact references, they’ll ask you for a reference or contact someone involved with one of the projects you’ve listed on your portfolio independently.

Portfolio & Cover Letter Advice

Your portfolio is only as strong as the weakest work on it

This isn’t “the weakest link” from *Doctor Who*, but it’s vital you put your best work forward. Quality over quantity, and if you do have quantity, you can still mention it in a footnote.

A note on past work

You can definitely cringe at past work to the point of not wanting to be affiliated with it. It’s perfectly reasonable to have a pseudonym where you release your less-polished work. However, know that your games will always be a part of you. Games are like children. Deep down you’ll love each of them no matter what, because with each game you make you give a part of yourself away. Not to mean that if you make enough, you’ll eventually be empty – you can always recharge by living life and experiencing new things. Mainly, each game shows the progression and growth of you as an artist and as a person. Past work no matter how objectively bad it is, is good work. As long as you’ve learned something while creating it, you should be proud of it.

Creative freedom with portfolios

A portfolio can be as long as you need (just don’t cover all your work). One-page portfolios are great for iteration, polish, and talent scouts to easily share around. Just be sure to make it a pdf and set the file name to something like “lastname, firstname resume.pdf” for easy finding for online recruiters to share around. Other than that, there aren’t any standards or rules for portfolios, so get creative, make it pretty, and showcase those graphic design skills.

Cover letters are optional

If you do choose to write one, showcase your familiarity with the company and how their workplace is a perfect fit for you as a person. Don’t be disingenuous about it, just be personable and to the point. Speak to you as a person (motivated, loyal, etc.) and make sure the letter works for you and not against you as it’s another additional piece of information for the employer to make an informed decision. However, the benefit of a personally tailored cover letter is it shows an extra ounce of effort towards your application and your personality which may not come through in your resume. Cover letters aren’t required for most applications, but if you have a connection to the company you’re applying to – go for it!

Sell yourself

You need an angle. You need hooks (on your resume) to catch employers' attention when scanning. You are a product and this resume is the advertisement – make sure it's a gleaming one. Make yourself easy to say "yes" to and hard to say "no" to. Make sure all links and projects listed on your resume are easily accessible for further inquiry. Prepare to talk about them in detail in interviews.

And remember… it's a numbers game. Apply to 10 studios and you may only get one or two interviews, so keep applying and giving yourself a chance. The worst thing they can do is say "No, try again later."

The pilgrimage to GDC

The cost

GDC is prohibitively expensive. It's one of my biggest issues with the conference. There are online alternatives to GDC such as #notGDC and gamedev.world, but they'll never fully replace meeting your fellow developers face to face. San Francisco is one of the most expensive places on earth the convention could be held, and for students and novices, a $1000 USD badge is out of the question. Good thing is if you're really out of funds, the early-bird expo pass $149 USD is all you need. The best talks are posted online to the official GDC Youtube channel after the conference, and someone you know will probably let you borrow their password to get vault access (if you ask nicely). Most roundtables, sponsored talks, and general conference events are open to those with expo passes, so you won't miss the exclusive (not recorded) parts of GDC.

Additionally, there are plenty of scholarships, volunteer, and work opportunities out there to get into GDC. While competitive, I've heard nothing but good things from the conference associates position. Organizations like GaymerX, I Need Diverse Games, Able Gamers, Pixelles, IGDA, ESA Foundation and many more have scholarships open every year. If you make it happen, getting a pass to GDC isn't impossible. However, finding an affordable place to sleep for the week is another challenge. There's "the indie hostel" and perhaps some cheaper accommodations the closer you get to the Tenderloin. My recommendation is to do your homework ahead of time to get the best deals for Airbnbs a few BART stations away from Powell. If you're lucky and have friends or family who live in or near San Francisco, you could couch surf.

Pro tip: conceal your lanyard in public, walk with purpose like you have some place to be, and be aware of your surroundings. High-risk areas that people say to avoid aren't all that bad in comparison with other cities. The homeless are more at risk than you. Don't make yourself a target, travel in groups, and you should be fine.

What to bring

- Really comfortable walking shoes (you're going to be on your feet the entire time – minus the talks and panels you'll be sitting on)

- Toothbrush, toothpaste, floss, deodorant, fingernail clippers, nail file, etc. (nothing ends conversations faster than lack of personal hygiene)
- Business-casual clothes (T-shirts are fine, just make sure whatever you're wearing doesn't have holes or massive wrinkles – look like you put some effort into your look. Dress clothes aren't necessary outside of pitch meetings)
- Some cash for meals near Moscone
- 50 to 250 personal business cards "The Currency of GDC"

  Pro tip: the biggest newbie mistake is to force "networking" on others and give your card to everyone. This diminishes its value and makes you look desperate. Not every conversation you have has to end with exchanging cards. Focus more on making friends instead of connections. Don't expect anything from anyone, just introduce yourself and get to know others. If someone is busy talking to someone else, do more listening than talking and be polite, or just wait to approach them later when they're open. Don't only talk to developers with notable games, talk to everyone regardless of what they can potentially offer you.

- Notebook & pen for miscellaneous notes

  Pro tip: after you finish an initial conversation, jot down a brief summary with some key points of what you talked about and whom (along with their contact info) to make follow-up emails post GDC easy. Writing everything down also helps you remember names better, which always helps!

- The official GDC mobile app (you can schedule the talks you want to attend and get push notifications with an embedded map to help you arrive on time)
- A backpack for your water bottle, snacks, notebook, cards, phone charger, personal health products, and whatever swag/leaflets you manage to accumulate on the expo floor.

  Pro tip: you don't need to bring your laptop, there'll be no time nor place to do any actual work or showcase your game(s) guerilla style, so why hurt your back carrying a heavy laptop and battery?

What to do?

- GDC Lost Levels isn't technically part of GDC, it's usually held in Yerba Buena Park for free and is counter programming radical soapbox short talks from any developers. Signups at the door, err… tree.
- Touristy San Francisco attractions:
  - Golden Gate Bridge
  - Fisherman's Warf/Pier 39
  - Lombard Street
  - Coit Tower

    - Twin Peaks
    - The Painted Ladies
    - Fort Point National Historic Site
    - The Castro

  Pro tip: The biggest regret many devs have is not taking time to see San Francisco outside of the convention center. Give yourself an extra day to be a tourist! It'll be well worth it!

- The after parties (2019's unofficial list – for up-to-date information check the facebook group "The Fellowship of Game Developer parties" or do a Twitter search for "GDC parties")

  Pro tip: don't make a fool of yourself by drinking too much, stay safe by never leaving your drink unattended and staying within arm's length of a trusted friend. Let friends always know your whereabouts and keep your head on a swivel. While I'd love to say that these parties are professional, there have been numerous instances of inappropriate behavior and advances. This goes for everyone, stay safe, try to get home at a reasonable hour, and get some sleep for the following day!

- Meals with friends!

  Pro tip: a meal eaten alone at GDC is a wasted opportunity to meet and get to know other devs. Even if you aren't actively making new contacts, catching up with old friends is a great use of time too! Just try to schedule your meals in advance!

The walls have ears

Your industry is smaller than you think it is, don't down talk products/teams/companies in public, because odds are whoever is hiring you personally knows and or has worked with them before. Put petty consumer rivalries behind you when you enter the industry. If you have something nasty to say, hold it for your pet when you're back home, far far away from any other folx in the game industry. Don't burn bridges unintentionally.

Where are all the opportunities?

GAMEDEVMAP

https://gamedevmap.com/index.php

Notes:

Gamedevmap is an interactive catalogue of all game studios with five or more employees and previously-published games. Convenient for finding work without having to relocate yourself.

AKUPARAGAMES FESTIVALS

http://www.akuparagames.com/festivals/

Notes:

Gathers deadlines for different game competitions and showcases from around the world. A must-use tool for doing the festival circuit with your game.

MEETUP.COM

https://www.meetup.com/

Notes:

Find developer casual meetups and local gamedev events in your area. If nothing exists here nor with the IGDA, create your own group and start hosting events!

ARTSTATION JOBS

https://www.artstation.com/jobs

Notes:

A great resource for live job openings for artists. If you create an Art Station account, you'll get access to a mailing list that updates you with openings every week, which can always be unsubscribed from.

GAMASUTRA & ITCH.IO JOBS

https://jobs.gamasutra.com/

https://itch.io/jobs

Notes:

Two resources for job listings for all positions in studios worldwide. However, both listings are quite limited at the moment.

Pro tip: You'll likely have better luck looking up game studios' careers pages and looking for work manually. A few Twitter searches can yield better results than any website tracker can. Meanwhile, attending local events in person and maintaining relationships yields far better opportunities than applying online.

GAME SCHOOL MAILING LISTS

Notes:

If you have a good friend who attends or has attended a notable college game or art program. Odds are they're in a mail list for updates on job offers and showcase opportunities. If they can create a filter to forward all those emails to you, these departmental mailing lists are invaluable!

12. CLOSING NOTES

There is No One Straight Path to A Career in Games

You don't need a technical or artistic background in game development to work in games. There are plenty of production, QA, press, PR, Esports, and business-related positions in games that want outside expertise. Your first job will likely not be your last, and lateral position moves are just as valid as working towards a promotion. Even if you're working in a games adjacent field or something entirely different, it may free you up to have enough creative energy at the end of the day to pursue your passion for game development untainted by commercial pressures to monetize a healthy hobby. You don't need to have a career in games to be a happy game developer.

Make the games you wish to see in the world

Make games with representation for underrepresented groups, because representation is affirmation. It's the culture at large letting you know that you were considered in the making of this and that you are valid. Being able to see themselves represented with positive role models can allow people to feel more comfortable being their true selves and living their life to their full potential. Life imitates art just as much as art imitates life, so it's incredibly important what message is being broadcasted, intentionally or unintentionally, in the media we consume. As far as developers go, knowing someone like you made the game you enjoyed playing can inspire that person to make games of their own. It's a positive cycle that makes the industry a more diverse place, and the games better too.

Make games that don't fit into any one single genre or play convention. How many times have we seen the big appeal of a game be "you can be sneaky or loud" or "the actions you do can be bad, good, or just crazy". Try to find more nuanced verbs and ways of interacting with the world than what's already out there. Draw inspiration from all things outside of games. There's nothing wrong with improving and working within familiar constraints, but making games that aren't considered games by a typical consumer audience opens up infinite possibilities.

Most importantly, make the games you wish to see in the world. Because odds are if you want it to exist, others do too.

Be good to one another

One of the best pieces of advice I've ever gotten was from a teacher of mine Adam Sulzdorf-Liszkiewicz who simply stated after every lecture "Be good to one another". Raise one another up, not view your friends' accomplishments as personal losses, but instead as an opportunity for them to drop the ladder for others to rise up as well. Make introductions for friends and connect others whenever possible. The game industry doesn't have to be a cutthroat one if we can show kindness and basic human decency to each other.