

Demystifying Game Development

How to focus on Game Design under time
pressure - Game Jamming

Who_am_I?

And what do I do?

- Software Engineer @ Sword
- Graduated RGU in July 2025 - MSci Computing Science
- Developer of Ender IO (Minecraft Mod)
- Game Jam Enthusiast
- Writer of Crap Code & Hoarder of Side Projects



What's_a_Jam?

**“If this year works out, we'll do
a new Jam every year, with a new
wacky technology each time.”**

Chris Hecker, 23/12/2001 - “A Small Group of Game Developers” via Email

The_First_Jam*

Indie Game Jam 0 - 15/03/2002

- The organisers would build a custom game engine with some **experimental technologies**
- They would invite indie developers to **build games** with the engine over the course of 4 days
- The goal? **to encourage experimentation and innovation in the game industry**

Modern_Jams

Getting with the times

- Generally replace the “new engine” with a theme or constraint
- Shorter timeframe
- Bring your own engine/tools
- Way more of them

Ludum Dare





Next weekend!

6th - 8th of March at UHI in Elgin



What_About_Hackathons?

What's the difference?

- Nothing really... Just a change in the subject matter and *sometimes* the method of presenting your work.
- First* Hackathon run by OpenBSD in June, 1999
- Wider focus on software as a whole, rather than just games
- **RGUHack**: Pitch your solution, rather than play-test/hands-on assessment - This is the important difference today



* again, probably...

Pre_Requsites

Pre_Requisites

Make sure you know what you're doing before you spend time you don't have

- Think of a handful of **unique gameplay aspects** - **pick one**, or even $\frac{1}{2}$ of one
- Define your core gameplay loop, what makes it **satisfying** to play and how does it help your **narrative**
- **Optimise** for 5 or so mins of gameplay at most, **demonstrate all of your features** in that time
- **Keep it simple!** - Players wouldn't be reading your code!

Game_Design

**“The best ideas come as jokes.
Make your thinking as funny as
possible.”**

David M. Ogilvy

How_to_Design_a_Game

Making it *fun*

- Think outside the box - don't knock ideas that seem 'too big', you can distill these down
- Focus on quality > quantity. One really good level will destroy a game with 10 mid levels
- Try and keep things interesting, in 5 mins of playtime you should avoid repeating a substantial puzzle
- Avoid dull patterns like "Find a key, unlock a door for, find another key, open the next door..."

Building_Quickly

“Write Games, Not Engines”

Josh Petrie, 30/08/2007, and almost everyone else who has written an Engine.

Starting_on_the_Right_Path

Making good use of the early hours

- Pick technology early - it's easier to plan when you know how it's built
- Assets aren't important to build features - use placeholder shapes in the meantime!
- Get your Git repo ready early and branch as soon as you can
- It may be hard to delegate in the early stages, try to collaborate in other ways like further planning while someone sets up the basics

Keeping_Pace

Once you're in the thick of it

- Commit early and PR/merge often!
- Set aside 15-20 minutes for a final merge before the end of the hack - you don't want a bad merge to ruin the project at this stage either!
- Avoid making sweeping refactors - clean code is good, but working code is better*
- Play-test early and often.
- Build features first, polish after

Choosing_Tech



“The Right Tool for the Job”

Scott Ambler, 01/12/2003

**“The Most Appropriate Tool for, the
“The Right Tool, for the Job”**

Kasoltin Attorney., 017/107/2002

“The Most Appropriate Tool for the Job”

Scott Ambler, 01/12/2003

What_is_Appropriate?

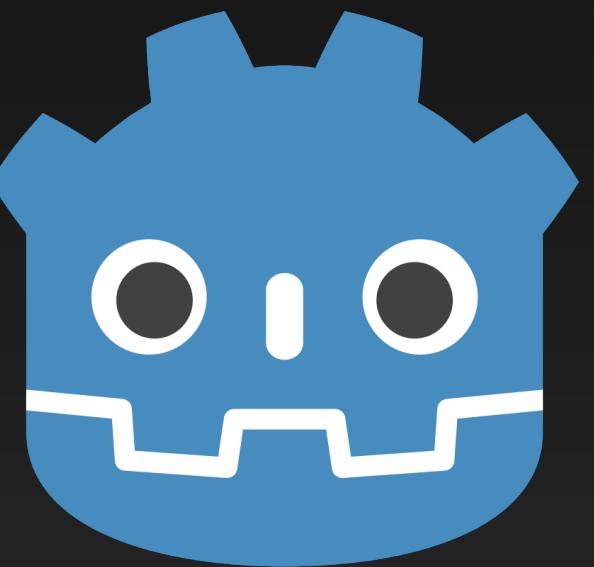
How do you decide what is right?

- If you have **already chosen** a technology - avoid changing course unless you encounter a **technical reason** to!
- When under time pressure, aim for **familiarity**
- Ensure that the tech you select provides the **building blocks** you need
- Make sure you're suitably **comfortable** before using a brand new tool - taking some time to **learn a tool** may save many hours of pain

My_Suggestions

1 - Godot Engine

- Open Source!
- Write code in GDScript or C#
- All-in-one editor
- Lightweight, runs on anything
- Lots of built-in tools which will speed up development



GODOT
Game engine

My_Suggestions

2 - Unity

- Large community - lots of tutorials and forum posts
- Write code in C#
- Extensive feature set, with a large number of packages at your disposal
- **CAVEAT:** Can be somewhat complex, especially when dealing with older tools - ask me how I know!



My_Suggestions

3 - Web Technologies

- If you're more comfy with web technology, why leave!
- Runs in the browser with tooling you already know
- Can use simple HTML markup and links to build a simple game
- Render to the Canvas, either by hand or using a 3rd party library!



Standing_Out

Standing_Out

Give your game some *Juice*

- A **polished game** with one feature feels nicer to play than six features in a trench coat
- Set aside time for SFX, Particles and other effects to provide satisfying feedback
- You should reward your player for doing well, only punish mistakes if it's thematically relevant - consider your audience
- Remember, you're presenting your game, optimise for a good story and appearance!

Juice Example 1

Skyrim Lock Picking

- Audible feedback while lock picking
- Visual indication when you are close to succeeding
- Physically based movement of lock pick and cylinder, emulates real life



Credit: [Brandon Dávila Sánchez via YouTube](#)

Juice Example 2

Bejeweled 2 Explosion

- Shining spotlights to show gems of interest
- Explosion of particles when producing a combo
- Congratulatory text overlay to reward the player



Credit: ChimaTronX via YouTube

Juice Example 3

Gore Splatters

- Visual and audible impact upon failure
- Playtesters at RGU Game Jam 2024 loved it
- We won 1st place in the end :)



Credit: Cool Catz (me!) @ RGU Game Jam 2024

Game_Assets

Game_Assets

Need graphics or sounds? I know a place...

- <https://kenney.nl/>
- <https://freemusicarchive.org/>
- <https://freesound.org/>
- <https://opengameart.org/>
- <https://spritedatabase.net/>
- **Remember to credit artists when using their assets!**



“I know all of this already! ”

Some of you, probably...

Making_it_Interesting

You've been here before...

- Try out **new things** - just don't change more than you can handle!
- Test out a new way to build your games, be it **architecturally**, with a **different engine or language**!
- Consider designing a game outside of your usual genre and see how it feels
- A **hackathon** is a perfect time to **learn and experiment**



Go Forth & Make Games

I'm happy to take any questions now, or later in the Bear Pit -
I'll be here all night if you need any help!

// TODO: Build game lol