

Contents

Godot Games	1
Fractured Elements	1
Fractured Elements	2
Fractured Elements	2
Cosmos Conquerors	2
Cosmos Conquerors	2
Bevy Pong	3
Bevy Pong	3
Learn OpenGL	3
Krita Palette Creator	3
Ray Tracing in Rust	4
Kait	4
Pixel Art	4
Pixel Art	4
Scanlation Work	4
Scanlation Work	5
VRCX Insights	5
Booth Archiver	5
Real-Time WebSocket Synchronization with...	6
Brainfuck Compiler with Cranelift	6
Conclusion	6
Why Rust?	6

Godot Games

Page Number: 11

- The first proper game engine I used was godot
- I tried to do something different everytime I made a game
- Pong: learned the game engine
- Magical Marshmellow: Used audacity to record and edit a sound effect, drew all the sprites
- Conway's Game of Life: Implemented the algorithm by copying from wikipedia
- Meme Invaders: Made the whole game by myself without watching tutorial
- Minesweeper: Implemented the flood fill algorithm without knowing the flood fill algorithm
- Pop The Lock: Used Krita's Vector layers for first time

Word Count: 86

Fractured Elements

Page Number: 12

- Originally, the game was supposed to have 4 characters but due to time constrains, we weren't able to do it
- The judge feedback is pretty positive, they mentioned that they were unsure about what caused them to change between the characters
- My partner and I had decided that the character changing mechanic was going to be explained during the presentation, but we weren't given the chance to show our presentation

Word Count: 72

Fractured Elements

Page Number: 13

- My biggest contribution to the game was the changing character mechanic
- My aim was to have low coupling and high cohesion
- Lets call the outermost shell, the character and the switching character, that is, the dwarf, and the elf, sub-characters
- Coming from Godot I thought it would be easy, just create signals and connect them, and run function at a certain animation frame to spawn projectile
- But, Unity was less friendly than I expected
- The PlayerController script handled the movement, switching between the sub-characters, and sending "signals" for the animation, and attack to the sub-character
- I wanted to use Unity events, however, I decided that it was more important to complete the game, than to do it the proper way

Word Count: 122

Fractured Elements

Page Number: 14

- Years ago I heard somewhere about how Terraria handles weapons
- They spawn a projectile with an initial position, and an initial speed
- Everything else was handled by the projectile script
- So, thats exactly how I did it
- Another problem I encountered was spawning the attack projectiles at the correct frame during animation
- I ended up using a janky way to detect if the animation time has reached a certain percentage and then spawning the projectile

Word Count: 77

Cosmos Conquerors

Page Number: 15

- Cosmos Conquerors was my first game jam game
- In the game, you are the swarm of enemies, and your objective is to survive against ever more powerful heros
- I had quite a few ideas but I decided to first make the core game and then focus on the aesthetics and tweaks
- I wanted to have a procedurally generated background but due to health problems I was unable to complete it

Word Count: 72

Cosmos Conquerors

Page Number: 16

- The twist was that you couldn't control which invader shoots when you press spacebar
- The bullet was shot from a random invader
- This made it so more invaders equals more health, but you're less likely to shoot from the invader you want
- The hero gains more and more speed everytime it dies, eventually, it will catch up to the invaders and kill them all

Word Count: 66

Bevy Pong

Page Number: 17

- Bevy is a game engine in rust which uses ECS architecture
- I am interested in high performance compute, so, I wanted to learn how to use ECS
- I had also heard good things about bevy, like, how its great at progressive disclosure of complexity
- Like you can replace the renderer (check out: Tiny Glade)
- There's also some render graph thing going on but I don't know what it is
- Bevy uses WGPU, an implementation of WebGPU standard in rust
- I find WGSL easier to read

Word Count: 87

Bevy Pong

Page Number: 18

- I originally wanted to add online multiplayer via websockets but while building proof of concept for the websocket, I encountered problems with the websocket library and I don't have the skills to troubleshoot it
- I will revisit and continue it later
- Also, it took me 3 tries to learn how to use ECS
- I also want to use bevy to write compute shaders in WGSL
- Godot's signals instantly clicked with me because of how they decouple scripts
- Bevy's `EventWriter<T>` and `EventReader<T>` do the same for me

Word Count: 88

Learn OpenGL

Page Number: 19

- Coming from rust, where I rely a lot on tools like Clippy to reduce cognitive load, catch errors, use idiomatic code, dodge potential issues, etc, the tooling for C++ was not helpful at all
- I tried to use CLang's lsp and CLion IDE, but they weren't anywhere as helpful as Clippy
- I also couldn't explore and learn the way I do in Rust, by reading the documentation by hovering on the functions

Word Count: 74

Krita Palette Creator

Page Number: 21

- I made this project as I often wanted to pick colors from an image, but due to the shading and jpeg compression artifacts, it was hard to choose a representative color
- I checked if Rosetta Code had the algorithm, and it didn't
- I checked other languages, and Java was the most readable one, so I ported the java code almost line to line, and then contributed the code
- I wanted a program which helped me pick distinct colors, while reducing the number of actual colors which are similar
- Turns out, the algorithm isn't suitable for my usecase, but I still manage to use it

Word Count: 107

Ray Tracing in Rust

Page Number: 22

- One of my favorite type of content on youtube is some person optimizing a single aspect of a program until it's limit
- I used this project as a testbed to write efficient code
- I was following a tutorial written in some old version of rust
- I used my knowledge of rust to write it in modern rust while trying my best to use good and idiomatic rust practices
- I was a fool back then and didn't write down the link to the book I was following, so, I can only put the legendary book by Peter Shirley

Word Count: 101

Kait

Page Number: 23

- I made Kait to use as my vtuber persona
- My aim was to make a simple countryside girl inspired by cottagecore who has a lot of heart motifs
- I made or edited the textures in photoshop
- While I ultimately didn't stream much, I'm still proud of the character I made

Word Count: 51

Pixel Art

Page Number: 24

- I made this adorable idiot while learning pixel art from a course I bought

Word Count: 16

Pixel Art

Page Number: 25

- The pixel art on left is for rust's mascot ferris the crab obtaining a sandwich
- It was made for a monthly online rust meetup
- I was planning speak in the meetup but the last meetup was december of last year
- The pixel art on right is the logo I made for the Coventry University Summer School Game Jam
- It's the elemental powers for the characters
- If I wanted to continue the game, I would add the element for the other powers when I add them to the game

Word Count: 90

Scanlation Work

Page Number: 26

- Scanlation is the act of fan-made scanning, translation, and editing of comics from a language into another language
I have read a lot of manga, and I was always curious about how hard it would be to clean and typeset manga pages
- I did it to practice my editing and typesetting skills
- Here you see an example of my work

Word Count: 62

Scanlation Work

Page Number: 27

- I cannot do the whole scanlation process
- I would need to build up good will so artists don't mind me processing their art, learn and practice Japanese, learn about Japanese culture so I can translate more accurately, and then finally I can do what I originally wanted to do, that is, practice my editing and typesetting skills
- So, I joined the "Danke fürs Lesen" scanlation group, and was able to work with people across timezones and be part of an inclusive community

Word Count: 84

VRCX Insights

Page Number: 29

- There's a concept called six degrees of separation which says that you are connected to any other person on earth by at most six people
- One of my friends kept saying that "I know everyone" since I kept bumping into her in unexpected friend circles
- So, I wanted a tool to see how people are connected directly and indirectly when I'm not part of the link
- For example, if I know three people: X, Y, and Z, I wanted to see if X and Z are connected through Y
- One of the other goals was to identify social circles within the community
- It uses frequency of joining and leaving a room to guess if someone is connected to another person
- If people are often in the same room then it's highly likely they're connected
- For a real life example, I had classmates Manish, and Aryan, and underclassmates Divyesh, and Tejas
- Divyesh is the same person who is in the credits for most of my slides
- Aryan, Manish, Tejas, and I were part of the student association
- Manish knew Aryan, Aryan knew Tejas, and Tejas knew Divyesh
- So, the indirect chain is Manish -> Aryan -> Tejas -> Divyesh

Word Count: 200

Booth Archiver

Page Number: 30

- I made this project to learn rust by creating a practical project
- It uses all the available threads
- I learnt how to use AMD uPerf and flamegraph to find out the bottleneck
- I also had to learn how to use a debugger for this project
- I eventually found out that I keep running into deadlock situations while consolidating data from the threads

Word Count: 64

Real-Time WebSocket Synchronization with... — Page Number: 31

- Very dumb name, I know
- I just used this project to learn how to use websockets in their quote-on-quote native environment, that is, the web browser
- I chose websockets instead of UDP as it's easier to use in browser and I'm not even sure if you can use UDP sockets directly
- Also, I just wanted to use cloudflare's durable objects
- My plan was to make the pong multiplayer, and then create a 2 player chess game and put it up as one of my portfolio piece

Word Count: 92

Brainfuck Compiler with Cranelift — Page Number: 32

- The only reason this exists is because I wanted to use cranelift and compile code to native
- Aside from creating for the sake of creating, I wanted to have some experience with compilers as I was going to attend a meetup in Bangalore hosted by Bosch
- One of their lead engineers is creating a the RISC-V compilation target for Cranelift
- I shared my experience of C++ with him and It was validating to hear that he shared a similar experience

Word Count: 84

Conclusion — Page Number: 35

- I wanted to share this quote from Steve Jobs, which beautifully captures my philosophy towards creativity
- He talks about how creativity isn't about inventing something entirely new out of thin air, but rather about connecting existing experiences and ideas in new ways
- My project topics are very diverse, but I believe that when the time comes, they will all connect in a beautiful way
- It's very exciting when I can apply knowledge I learned in one project helps me in another project

Word Count: 83

Why Rust? — Page Number: 36

- People are tired of hearing rust is better due to memory safety, while yes, that's good, it's not the only reason rust is good
- For me, rust was better for the reasons on right

Word Count: 36