

Portfolio Presentation

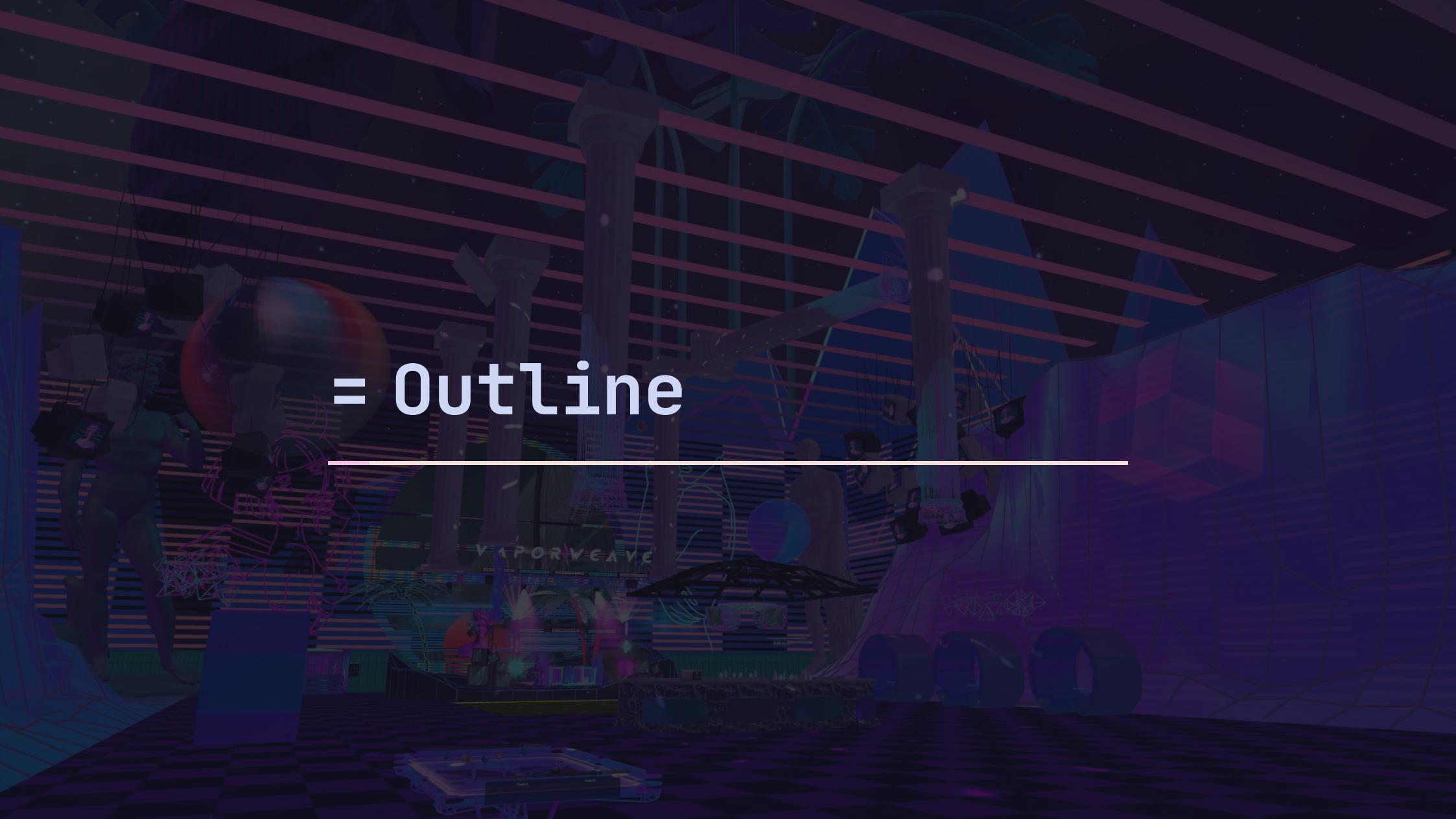
Aalto University Game Design and Development Interview

Herschel Pravin Pawar

March 19, 2025

= Hack so
background images
are preloaded





= Outline

= Outline

= Hack so background images are preloaded	1
= Outline	2
= Introduction	3
= Game Development Projects	5
== Godot Games	6
== Fractured Elements	7
== Key Points	9
== Technologies Used	9
== Links	9
== Cosmos Conquerors	10
== Key Points	10
== Technologies Used	10
== Links	10
== Spicing things up	11
== Bevy Pong	12
== Key Points	13

= Outline

==== Technologies Used	13
==== Links	13
== Learn OpenGL	14
==== Key Points	14
==== Technologies Used	14
==== Links	14
= Artsy Projects	15
== Krita Palette Creator	16
==== Key Points	16
==== Technologies Used	16
==== Links	16
== Ray Tracing in Rust	17
==== Key Points	17
==== Technologies Used	17
==== Links	17
== Kait	18

= Outline

==== Key Points	18
==== Technologies Used	18
==== Links	18
== Pixel Art	19
==== Rust for Lunch	20
==== Fractured Elements	20
== Scanlation Work	21
==== Skills Learned	22
- Non-game Development Projects	23
== VRCX Insights	24
==== Key Points	24
==== Technologies Used	24
==== Links	24
== Booth Archiver	25
==== Key Points	25
==== Technologies Used	25

= Outline

==== Links	25
== Real-Time WebSocket Synchronization with Durable Objects	26
==== Key Points	26
==== Technologies Used	26
==== Links	26
== Brainfuck Compiler with Cranelift	27
==== Key Points	27
==== Technologies Used	27
==== Links	27
= Skills	28
== Natural Languages	29
== Software Tools	29
== Knowledge Areas	29
== Personal Interests	29
= Conclusion	30
== Why Rust?	31

= Outline

==== Tired	31
==== Wired	31
= Appendix	32
== List of Images	33
== List of Tables	37
== List of Links	37

Portfolio Presentation

Aalto University Game Design and Development Interview

Herschel Pravin Pawar

March 19, 2025



= Introduction

= Introduction



Figure 1: Image of me
original file @/resources/me.png



Everything you see in this presentation — scripts, links, and images — are a part of a Typst document available freely on GitHub under a public domain licence.



[GitHub:pawarherschel/UniOfAaltoInterviewSlides](https://github.com/pawarherschel/UniOfAaltoInterviewSlides) |
First Commit: 2025-03-19

= Game Development Projects

- Godot Games 2021
- Fractured Elements July 2024
- Cosmos Conquerors July 2023
- Bevy Pong October 2024
- Learn OpenGL November 2024

= Godot Games

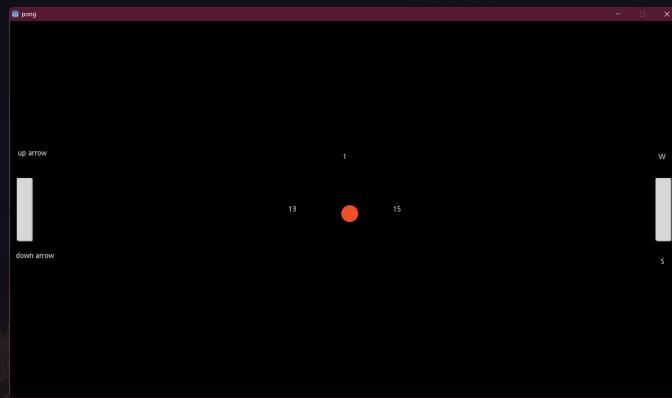


Figure 2: Pong: It starts here
original file @/resources/godot/pong.png

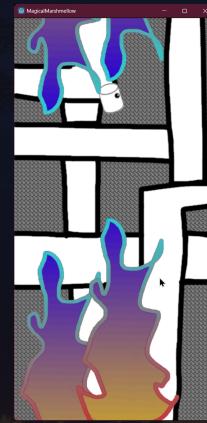


Figure 3: Magical Marshmellow: Drew all the sprites, and made the “boing” SFX
original file @/resources/godot/magical_marshallmellow.png

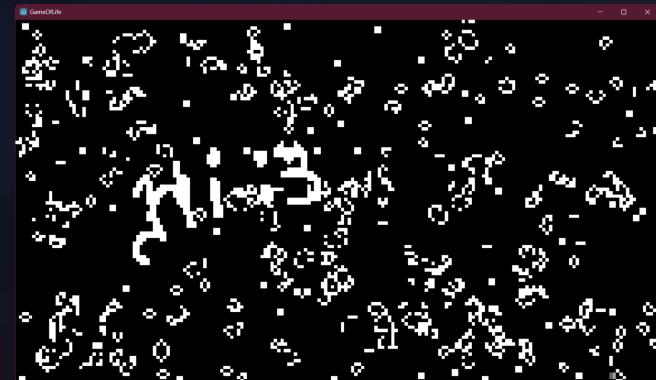


Figure 4: Conway's Game of Life: Implemented
Wikipedia:Conway%27s_Game_of_Life ==> Conway's Game of Life
original file @/resources/godot/game_of_life.png



Figure 5: Meme Invaders: First from scratch game
original file @/resources/godot/meme_invaders.png

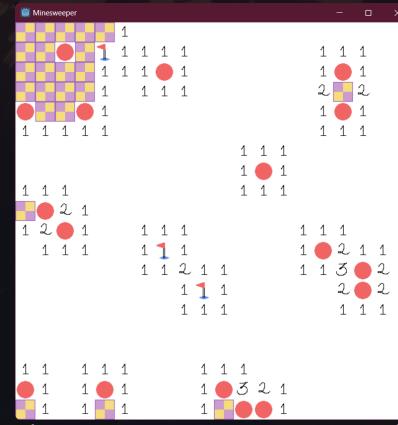


Figure 6: Minesweeper: Wrote all the logic by myself
(Wikipedia:Flood_fill ==> Flood_fill)
original file @/resources/godot/minesweeper.png



Figure 7: Pop The Lock: Made all the assets using Krita's Vector Layers
original file @/resources/godot/pop_the_lock.png

= Fractured Elements

Timer (seconds): 1.3 of 3.0
Sprite Index: 1 of 2
Player Speed: (0.00, 0.00)
(max horizontal speed: 1.5)



Figure 8: Main Mechanic: Continuous Change
[original file](#)

@/resources/fractured-elements/hero-switching.gif

“ Visually nice platformer game with a pixel art approach and switching characters that allow either melee or ranged attacks. Multiple levels make it interesting and force the player to master their jumping and attacking skills. There is a final boss for the finale. One improvement I would suggest is that I was unsure what caused me to change between the characters - I was not sure if it was timed, depending on where I was in the level or a button press. Well done to all involved in the game and the hard work you put into it! ”

— Anonymous judge feedback

Criteria	Rank	Score*	Raw Score
Gameplay	#2	3.286	3.600
Presentation	#2	3.469	3.800
Creativity	#3	2.739	3.000
Enjoyment	#3	2.921	3.200

= Fractured Elements

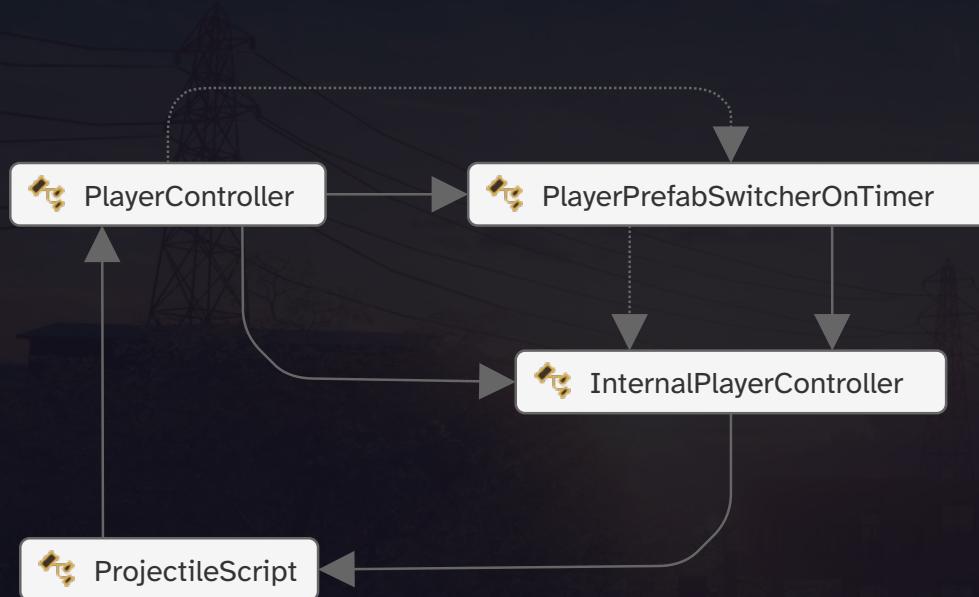


Figure 9: Player Script Graph
original file

[@/resources/fractured-elements/player-scripts-hierarchy.svg](#)

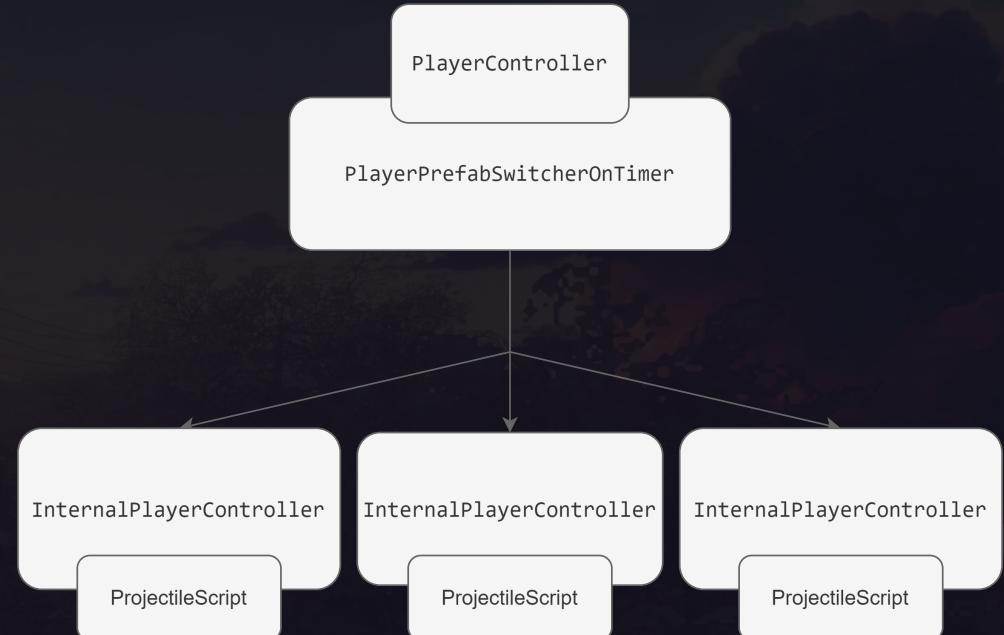


Figure 10: Player Script Hierarchy

original file [@/resources/fractured-elements/player.svg](#)

= Fractured Elements



Figure 11: Shasank and Me
original file

@/resources/fractured-elements/shashank-and-me.jpg

☰ Key Points

- Core Game Mechanics
- Character Transition System
- Attack Mechanic (thanks Terraria)
 - Hammer (melee) for Dwarf
 - Bow and Arrow (ranged) for Elf

☰ Technologies Used

- Unity

☰ Links

- [GitHub:pawarherschel/CovUniGJ2024](#) | First Commit: 2024-07-22
- [itch:pawarherschel:fractured-elements](#) ⇒ [Fractured Elements](#)

= Cosmos Conquerors



Figure 12: Main Mechanic: Roles Reversed

[original file @/resources/cosmos-conquerors/screenshot.png](#)



Gabriel Toschi 1 year ago (+1)

Cool concept and you made it work! More enemies and a better control of the ships and what ship would shoot would be nice

[Like](#) [Reply](#) [Delete](#) [Report](#) [Ban...](#)

Figure 13: Comment by “Gabriel Toschi”

[original file @/resources/cosmos-conquerors/comment 1.png](#) (has alt text)



Zimizeh Submitted 1 year ago (+1)

Interesting concept! More variations in the future would be great!

Submitted
EB-87

[Like](#) [Reply](#) [Delete](#) [Report](#) [Ban...](#)

Figure 14: Comment by “Zimizeh”

[original file @/resources/cosmos-conquerors/comment 2.png](#) (has alt text)

☰ Key Points

- Solo Development

☰ Technologies Used

- Godot

☰ Links

- [GitHub:pawarherschel/GMTK2023](#) | First Commit: 2023-07-08
- [itch:pawarherschel:cosmos-conquerors](#) ⇒ [Cosmos Conquerors](#)

= Cosmos Conquerors

☰ Spicing things up



Figure 15: All invaders shooting
original file

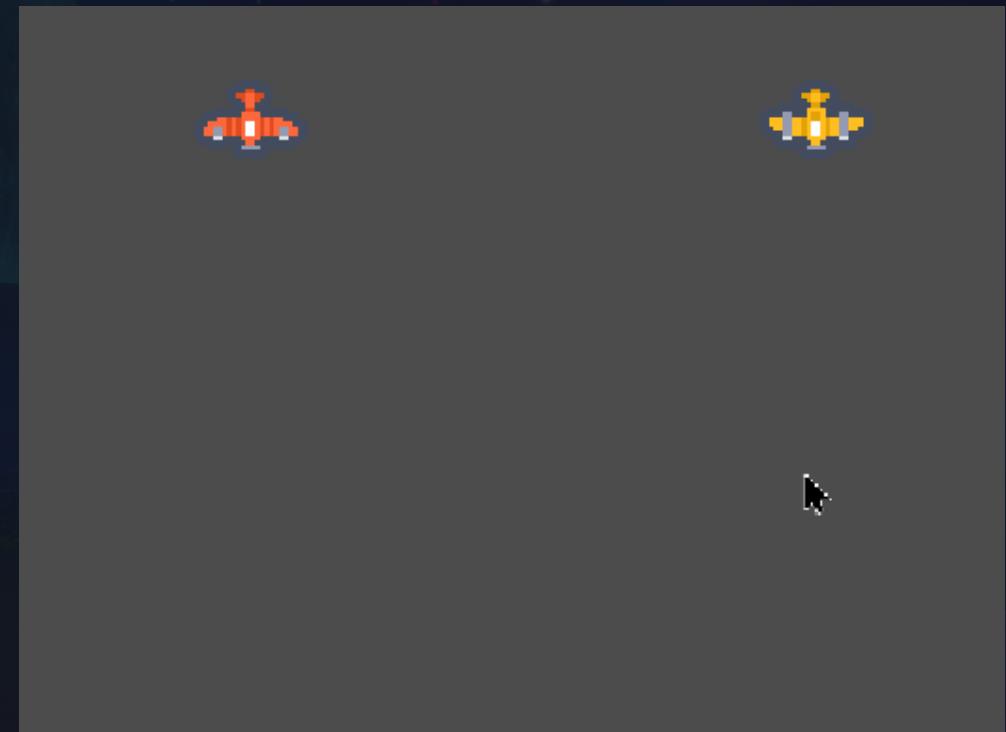


Figure 16: Two invaders shooting
original file

[@/resources/cosmos-conquerors/invaders-shooting.gif](#)

[@/resources/cosmos-conquerors/two-invaders-shooting.gif](#)

= Bevy Pong

```
1 // taken from: @/src/ball.rs          Rust
2
3 fn startup(
4     mut commands: Commands,
5     mut meshes: ResMut<Assets<Mesh>>,
6     mut materials: ResMut<Assets<ColorMaterial>>,
7 ) { /* .. omit .. */ }
8
9 fn update(
10    mut query: Query<(Entity, &mut Transform),
11    With<Ball>,
12    mut flip: EventWriter<Flip>,
13    mut create_score: EventWriter<CreateScore>,
14 ) { /* .. omit .. */ }
15
16 fn flip_handler(
17     mut flips: EventReader<Flip>,
18     mut query: Query<&mut Velocity, With<Ball>>,
19 ) { /* .. omit .. */ }
```

```
20 fn respawn_handler(          Rust
21     mut respawn: EventReader<Respawn>,
22     mut query: Query<
23         (&mut Transform, &mut Velocity), With<Ball>
24     >,
25 ) { /* .. omit .. */ }
26
27 fn paddle_collision(
28     ball: Query<
29         (Entity, &Transform),
30         (With<Ball>, (Without<Player>, Without<Enemy>))
31     >,
32     paddles: Query<
33         &Transform,
34         (Or<(With<Player>, With<Enemy>)>,
35         Without<Ball>)
36     >,
37 ) { /* .. omit .. */ }
```

= Bevy Pong

```
1 // taken from: @/src/lib.rs          Rust
2
3 impl bevy::prelude::Plugin for Pong {
4     fn build(&self, app: &mut App) {
5         app.add_plugins(
6             DefaultPlugins.set(
7                 WindowPlugin { /* .. omit .. */ }
8             ),
9         )
10        .add_systems(Startup, setup_camera)
11        .add_plugins(velocity::Plugin)
12        .add_plugins(player::Plugin)
13        .add_plugins(resolution::Plugin)
14        .add_plugins(last_mouse_position::Plugin)
15        .add_plugins(ball::Plugin)
16        .add_plugins(score::Plugin)
17        .add_plugins(enemy::Plugin);
18    }
19 }
```

☰ Key Points

- Learn ECS and Bevy

☰ Technologies Used

- Rust
- Bevy (Game Engine)

☰ Links

- [GitHub:pawarherschel/bevy-pong](#) | First Commit: 2024-10-26
- [YouTube:CnoDOc6ML0Y](#) ⇒ Chris Biscardi
 - Bevy: A case study in ergonomic Rust ←
 - Rust Nation UK
- [Steam:2198150](#) ⇒ Tiny Glade

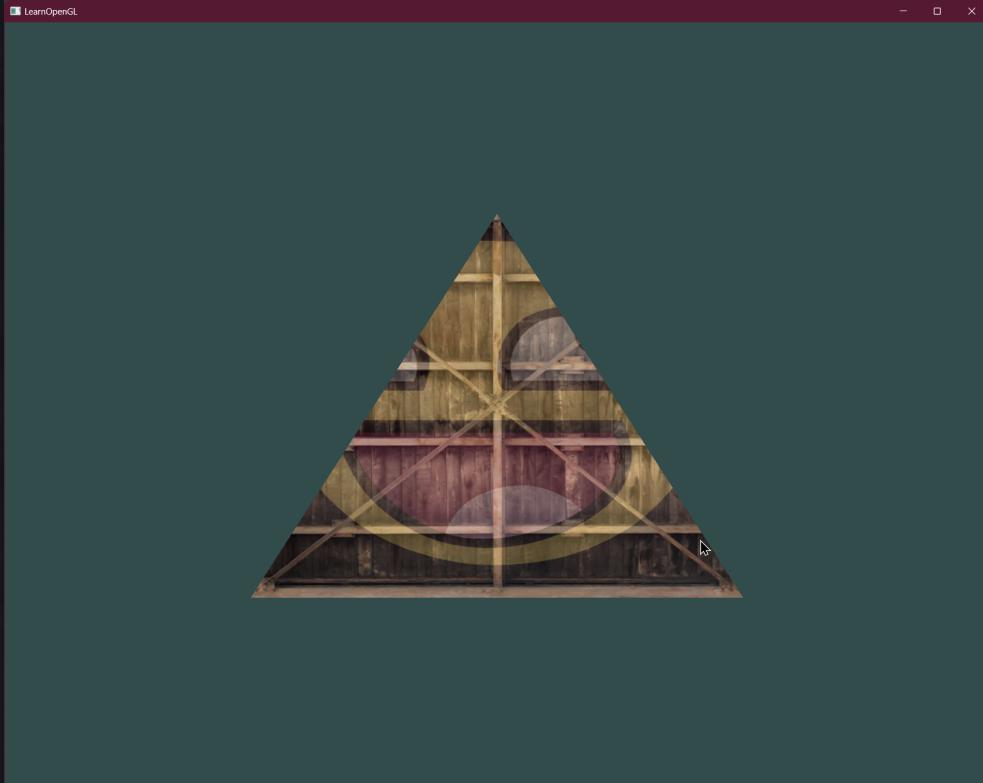


Figure 17: OpenGL Renderer Progress
[original file @/resources/learn-opengl.png](#)

≡ Key Points

- Tried to use modern C++
 - Failed because I'm on Windows
 - Modern C++ didn't feel cohesive
- Reached the [Transformations Chapter](#)

≡ Technologies Used

- C++
- OpenGL

≡ Links

- [GitHub:pawarherschel/learn-opengl](#) | First Commit: 2024-11-01
- [Learn OpenGL Book](#)

= Artsy Projects

- [Krita Palette Creator](#) September 2024
- [Ray Tracing in Rust](#) November 2023
- [Kait](#) 2021
- [Pixel Art](#) { 2023, 2024 }
- [Scanlation Work](#) 2020

= Krita Palette Creator



Figure 18: Original Image

original file @/resources/krita-palette-creator/original.png

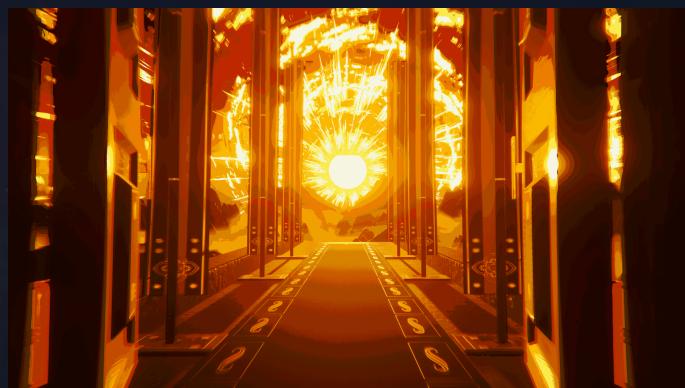


Figure 19: Quantized Image

original file @/resources/krita-palette-creator/processed.png



Figure 20: Colors Used

original file @/resources/krita-palette-creator/palette.png

☰ Key Points

- Implemented color quantization by using median cut algorithm (Ported the Java code)

☰ Technologies Used

- Rust

☰ Links

- [Rosetta Code Page](#)
- [Wikipedia:Median_cut](#) ⇒ Median Cut
- [Wikipedia:Color_quantization](#) ⇒ Color Quantization
- [GitHub:pawarherschel/krita-palette-creator](#) | First Commit: 2024-12-16

= Ray Tracing in Rust

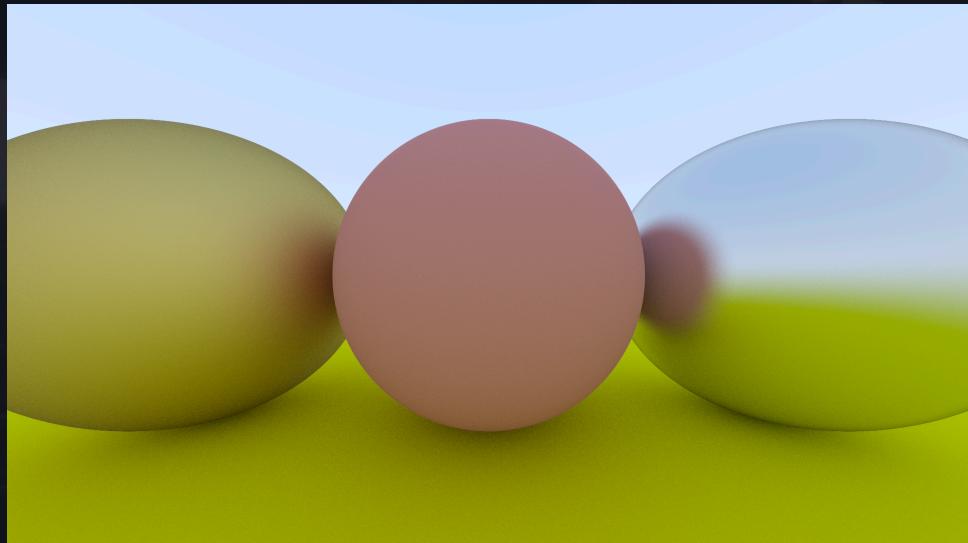


Figure 21: Ray Tracer Output
[original file @/resources/raytracing.png](#)

Resolution : 1920 * 1080

Laptop CPU : i5-8350U (8 Cores)

Number of Runs : 10

Time (mean $\pm \sigma$) : $76.861s \pm 2.590s$

Range (min ... max) : $70.564s \dots 79.588s$

≡ Key Points

- Learned how to make declarative macros
- Made a custom Vec3 struct with auto-vectorization in mind
- Used traits (interfaces) for modularity
- Parallel Rendering

≡ Technologies Used

- Rust

≡ Links

- <https://raytracing.github.io/>
- GitHub:[pawarherschel/raytracinginrust](https://github.com/pawarherschel/raytracinginrust) | First Commit: 2023-11-23

= Kait

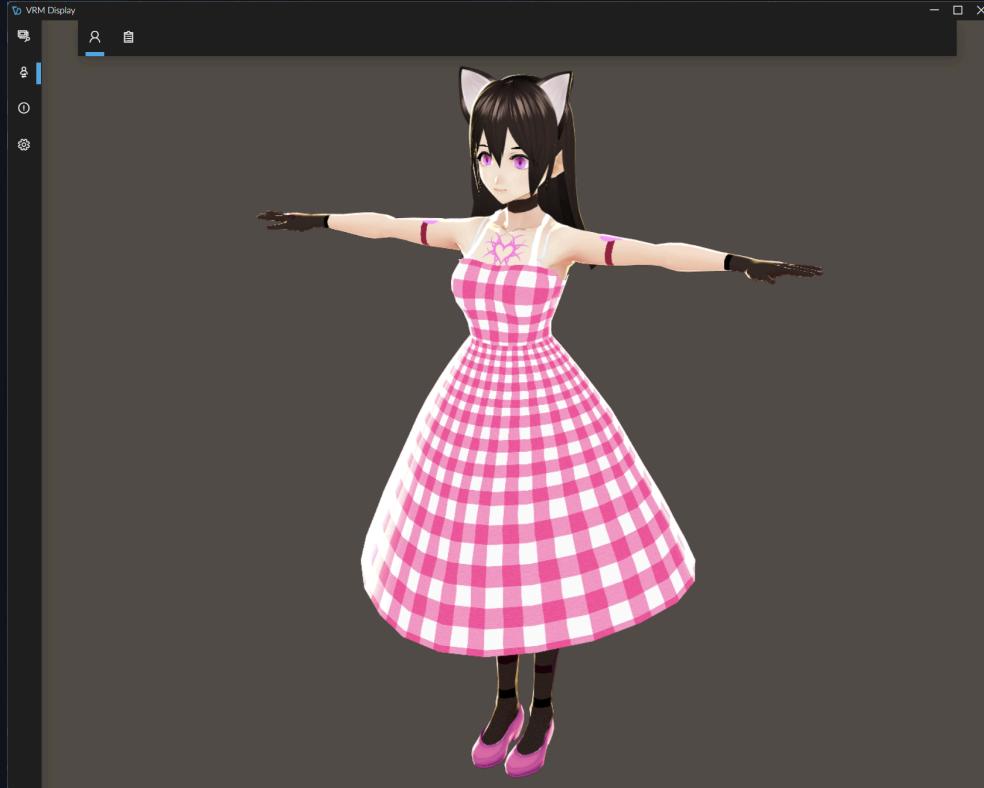


Figure 22: Original Character: Kait

[original file @/resources/kait.png](#)

≡ Key Points

- First foray into 3d model development
- Learned character design
- Consistently used Heart motifs

≡ Technologies Used

- [VRoid Studio](#)
- [Adobe Photoshop](#)

≡ Links

- [VRM File](#)

= Pixel Art



Figure 23: Pixel Art of Charmander waving it's
arms looking stupid (affectionate)
[original file](#) [@/resources/pixel-art/banner.gif](#)

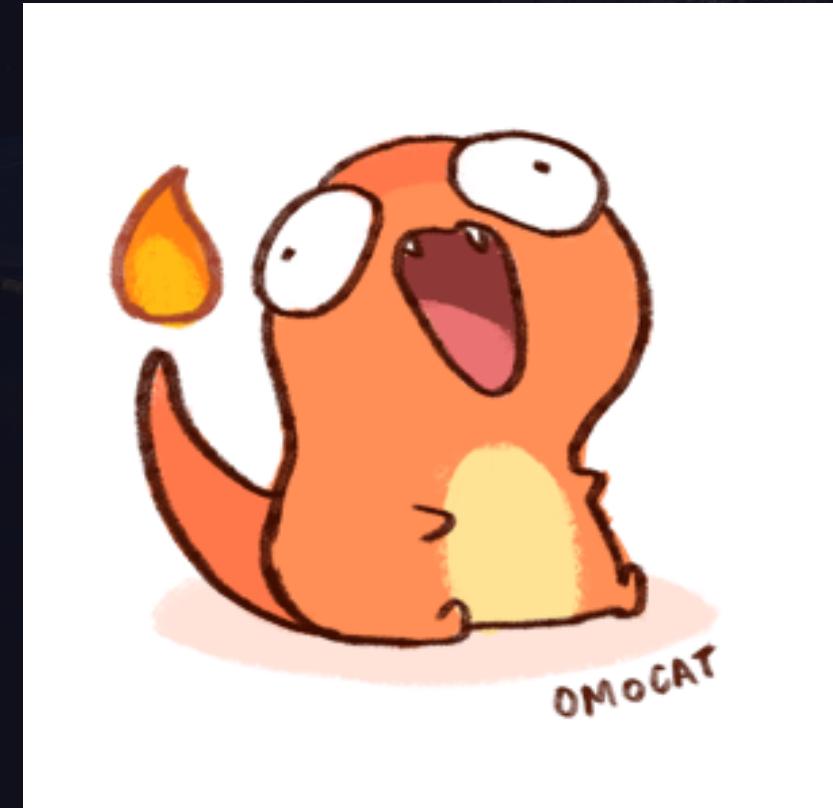


Figure 24: [original source: omocat's tumblr blog: pikachu needed friends original file](#)
[@/resources/pixel-art/charmander-original.gif](#)

≡ Rust for Lunch



Figure 25: Rust for Lunch Fan Art
original file [@/resources/pixel-art/rfl.png](#)

≡ Fractured Elements

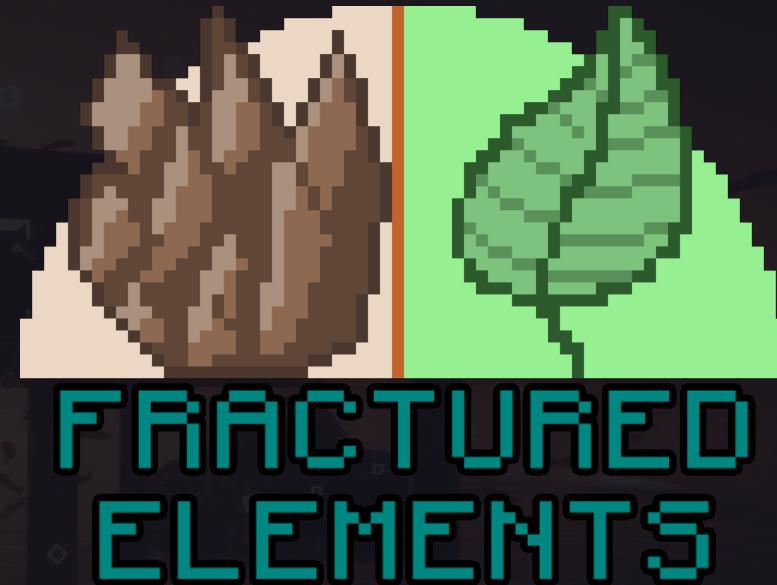


Figure 26: Fractured Elements Logo
original file
[@/resources/pixel-art/fractured-elements.png](#)

= Scanlation Work



Figure 27: Original Ver.

[original file @/resources/scanlation/original.jpg](#)



Figure 28: Edited Ver.

[original file @/resources/scanlation/product.jpg](#)

Danbooru:pools:17121

Original Title: 助けた少女が吸血鬼だった医者の話。

Translated Title:
The Story of a
Doctor Who Saved
a Girl Who Turned
Out to Be a
Vampire

Original By:
Tsukinami
Kousuke
Original at:
Twitter:tuki_nami

= Scanlation Work

☰ Skills Learned

- Image Editing and Cleaning (Adobe Photoshop)
- Typesetting and Lettering
- Teamwork and Collaboration across timezones

= Non-game Development Projects

- VRCX Insights 2023
- Booth Archiver 2023
- Real-Time WebSocket Synchronization with Durable Objects December 2024

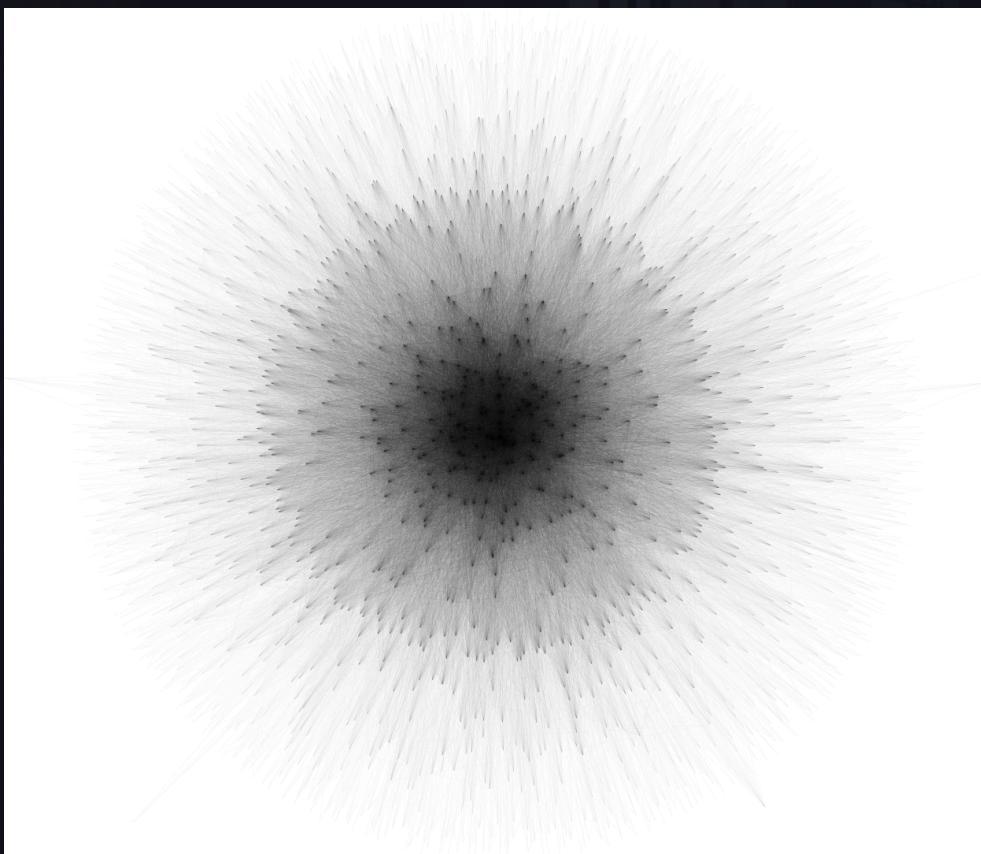


Figure 29: Preview of the image generated by the program

[original file @/resources/vrcx.png](#)

☰ Key Points

- Implemented a simple (but surprisingly effective) data mining algorithm based on thresholds
- Uses an external data source

☰ Technologies Used

- Rust
- SQLite
- Tokio (Async Framework)

☰ Links

- [GitHub:pawarherschel/vrcx-insights](#) | First Commit: 2023-11-19
- [Wikipedia:Six_degrees_of_separation](#) ⇒ Six degrees of separation

☰ Key Points

- First Rust project
- Archives wishlist from Booth
- Compiles the data into an Excel Spreadsheet

☰ Technologies Used

- Rust

☰ Links

[GitHub:pawarherschel/booth_archiver](https://github.com/pawarherschel/booth_archiver) | First Commit: 2023-07-28

= Real-Time WebSocket Synchronization with Durable Objects

“ Durable Objects provide a building block for stateful applications and distributed systems.

Use Durable Objects to build applications that need coordination among multiple clients, like collaborative editing tools, interactive chat, multiplayer games, and deep distributed systems, without requiring you to build serialization and coordination primitives on your own.

...omitted...

Therefore, Durable Objects enable stateful serverless applications.”

— Cloudflare Durable Objects Documentation

☰ Key Points

- Learned networking code
- Precursor to making online multiplayer games
- Ease of use in browser

☰ Technologies Used

- | | |
|----------------------|--------------|
| - Cloudflare Workers | - WebSocket |
| - Durable Objects | - TypeScript |

☰ Links

- [GitHub:pawarherschel/workers...](#) | First Commit: 2024-12-01

= Brainfuck Compiler with Cranelift

☰ Key Points

- Wanted to learn how to make frontend for a language
- Used a language known for it's small compiler frontend footprint
- Used Cranelift, an alternative to LLVM written in pure rust

☰ Technologies Used

- Rust
- Cranelift

☰ Links

- [Compiling Brainfuck code - Part 3: A Cranelift JIT Compiler](#)
- [GitHub:pawarherschel/BFCranelift | First Commit: 2024-11-17](#)
- [YouTube:jlhSdmv6bAY ⇒ Compilers in Rust: Cranelift, the All-Rust Codegen Alternative to LLVM \(No C/C++, Part 1\) ← regionaltantrums](#)



= Skills

Just a quick overview :3

Note: I only have a surface level understanding of these things as I learnt just enough for my work requirements

= Natural Languages

1. English
2. Hindi
3. Marathi
4. Finnish (learning)

= Software Tools

1. Photoshop
2. Krita
3. Godot
4. Typst
5. VRoid Studio

= Knowledge Areas

1. Cloudflare
2. Linux
3. NixOS
4. Docker
5. Ansible

= Personal Interests

1. Game Development
2. GPU Acceleration
3. VRChat
4. NixOS
5. Manga & Scanlation

= Conclusion

“ Creativity is just connecting things. When you ask creative people how they did something, they feel a little guilty because they didn't really do it, they just saw something. It seemed obvious to them after a while. That's because they were able to connect experiences they've had and synthesize new things. And the reason they were able to do that was that they've had more experiences or they have thought more about their experiences than other people. Unfortunately, that's too rare a commodity. A lot of people in our industry haven't had very diverse experiences. So they don't have enough dots to connect, and they end up with very linear solutions without a broad perspective on the problem. The broader one's understanding of the human experience, the better design we will have. ”

– Steve Jobs

= Why Rust?

≡ Tired

- Memory safety without garbage collection
- Fast code, comparable to C
- Increasing industry adoption
- One language which truely fits all
 - Game Development: Bevy ECS
 - Rendering: WGPU
 - Embeded: Embeded Rust Ecosystem
 - Data Science
 - Systems Programming
 - Web Dev (if required)

≡ Wired

- Reduced cognitive overhead
- Awesome tooling
- Welcoming and inclusive community
- Ownership and Borrowing system works perfectly with my brain
- Functional language, but people ship with it
- No `throw` exception or `nullptr` or ambiguous null objects!
 - `Result<T, E>` and `Option<T>` FTW
- No implicit conversion

= Appendix

= List of Images

- Figure 1: Image of me
[original file @/resources/me.png](#) 11
- Figure 2: Pong: It starts here
[original file @/resources/godot/pong.png](#) 13
- Figure 3: Magical Marshmallow: Drew all the sprites, and made the “boing” SFX
[original file @/resources/godot/magical_marshmallow.png](#) 13
- Figure 4: Conway’s Game of Life: Implemented [Wikipedia:Conway%27s_Game_of_Life](#) ⇒ Conway’s Game of Life
[original file @/resources/godot/game_of_life.png](#) 13
- Figure 5: Meme Invaders: First from scratch game
[original file @/resources/godot/meme_invaders.png](#) 13
- Figure 6: Minesweeper: Wrote all the logic by myself ([Wikipedia:Flood_fill](#) ⇒ Flood fill)
[original file @/resources/godot/minesweeper.png](#) 13
- Figure 7: Pop The Lock: Made all the assets using Krita’s Vector Layers
[original file @/resources/godot/pop_the_lock.png](#) 13
- Figure 8: Main Mechanic: Continous Change
[original file @/resources/fractured-elements/hero-switching.gif](#) 14

= List of Images

- Figure 9: Player Script Graph
[original file @/resources/fractured-elements/player-scripts-hierarchy.svg](#) 15
- Figure 10: Player Script Hierarchy
[original file @/resources/fractured-elements/player.svg](#) 15
- Figure 11: Shasank and Me
[original file @/resources/fractured-elements/shashank-and-me.jpg](#) 16
- Figure 12: Main Mechanic: Roles Reversed
[original file @/resources/cosmos-conquerors/screenshot.png](#) 17
- Figure 13: Comment by “Gabriel Toschi”
[original file @/resources/cosmos-conquerors/comment 1.png](#) (has alt text) 17
- Figure 14: Comment by “Zimizeh”
[original file @/resources/cosmos-conquerors/comment 2.png](#) (has alt text) 17
- Figure 15: All invaders shooting
[original file @/resources/cosmos-conquerors/invaders-shooting.gif](#) 18
- Figure 16: Two invaders shooting
[original file @/resources/cosmos-conquerors/two-invaders-shooting.gif](#) 18
- Figure 17: OpenGL Renderer Progress
[original file @/resources/learn-opengl.png](#) 21

= List of Images

- Figure 18: Original Image original file @/resources/krita-palette-creator/original.png	23
- Figure 19: Quantized Image original file @/resources/krita-palette-creator/processed.png	23
- Figure 20: Colors Used original file @/resources/krita-palette-creator/palette.png	23
- Figure 21: Ray Tracer Output original file @/resources/raytracing.png	24
- Figure 22: Original Character: Kait original file @/resources/kait.png	25
- Figure 23: Pixel Art of Charmander waving it's arms looking stupid (affectionate) original file @/resources/pixel-art/banner.gif	26
- Figure 24: original source: omocat's tumblr blog: pikachu needed friends original file @/resources/pixel-art/charmander-original.gif	26
- Figure 25: Rust for Lunch Fan Art original file @/resources/pixel-art/rfl.png	27
- Figure 26: Fractured Elements Logo original file @/resources/pixel-art/fractured-elements.png	27
- Figure 27: Original Ver. original file @/resources/scanlation/original.jpg	28

= List of Images

- Figure 28: Edited Ver.
[original file @/resources/scanlation/product.jpg](#) 28
- Figure 29: Preview of the image generated by the program
[original file @/resources/vrcx.png](#) 31

= List of Links

- sakurakat.systems (<https://sakurakat.systems>) 11
- GitHub:pawarherschel/UniOfAaltoInterviewSlides (<https://github.com/pawarherschel/UniOfAaltoInterviewSlides>) 11
- Wikipedia:Conway%27s_Game_of_Life == Conway's Game of Life (https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life) 13
- Wikipedia:Flood_fill == Flood fill (https://en.wikipedia.org/wiki/Flood_fill) 13
- Terraria (<https://www.terraria.org>) 16
- GitHub:pawarherschel/CovUniGJ2024 (<https://github.com/pawarherschel/CovUniGJ2024>) 16
- itch:pawarherschel:fractured-elements == Fractured Elements (<https://pawarherschel.itch.io/fractured-elements>) 16
- GitHub:pawarherschel/GMTK2023 (<https://github.com/pawarherschel/GMTK2023>) 17
- itch:pawarherschel:cosmos-conquerors == Cosmos Conquerors (<https://pawarherschel.itch.io/cosmos-conquerors>) 17
- GitHub:pawarherschel/bevy-pong (<https://github.com/pawarherschel/bevy-pong>) 20
- YouTube:CnoDOc6ML0Y == Chris Biscardi - Bevy: A case study in ergonomic Rust == Rust Nation UK (<https://www.youtube.com/watch?v=CnoDOc6ML0Y>) 20
- Steam:2198150 == Tiny Glade (<https://store.steampowered.com/app/2198150>) 20
- Transformations Chapter (<https://learnopengl.com/Getting-started/Transformations>) 21

= List of Links

- GitHub:pawarherschel/learn-opengl (<https://github.com/pawarherschel/learn-opengl>) 21
- Learn OpenGL Book (<https://learnopengl.com>) 21
- Rosetta Code Page (https://rosettacode.org/wiki/Color_quantization#Rust) 23
- Wikipedia:Median_cut == Median Cut (https://en.wikipedia.org/wiki/Median_cut) 23
- Wikipedia:Color_quantization == Color Quantization (https://en.wikipedia.org/wiki/Color_quantization) 23
- GitHub:pawarherschel/krita-palette-creator (<https://github.com/pawarherschel/krita-palette-creator>) .. 23
- GitHub:pawarherschel/raytracinginrust (<https://github.com/pawarherschel/raytracinginrust>) 24
- VRoid Studio (<https://vroid.com/en>) 25
- VRM File (<https://r2.sakurakat.systems/kait.vrm>) 25
- original source: omocat's tumblr blog: pikachu needed friends (<https://www.omocat-blog.com/post/24438605155/pikachu-needed-friends>) 26
- Danbooru:pools:17121 (<https://danbooru.donmai.us/pools/17121>) 28
- Twitter:tuki_nami (https://x.com/tuki_nami/status/1224267670037483520) 28
- GitHub:pawarherschel/vrcx-insights (<https://github.com/pawarherschel/vrcx-insights>) 31
- Wikipedia:Six_degrees_of_separation == Six degrees of separation (https://en.wikipedia.org/wiki/Six_degrees_of_separation) 31

= List of Links

- Booth (<https://booth.pm/en>) 32
- GitHub:pawarherschel/booth_archiver (https://github.com/pawarherschel/booth_archiver) 32
- Cloudflare Durable Objects Documentation (<https://developers.cloudflare.com/durable-objects/>) ... 33
- GitHub:pawarherschel/workers... (<https://github.com/pawarherschel/workers-durable-object-websocket-synced>) 33
- GitHub:pawarherschel/BFCranelift (<https://github.com/pawarherschel/BFCranelift>) 34
- YouTube:ilhSdmv6bAY == Compilers in Rust: Cranelift, the All-Rust Codegen Alternative to LLVM (No C/C++, Part 1) == regionaltantrums (<https://www.youtube.com/watch?v=ilhSdmv6bAY>) 34
- Wikipedia:Conway%27s_Game_of_Life == Conway's Game of Life (https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life) 40
- Wikipedia:Flood_fill == Flood fill (https://en.wikipedia.org/wiki/Flood_fill) 40
- original source: omocat's tumblr blog: pikachu needed friends (<https://www.omocat-blog.com/post/24438605155/pikachu-needed-friends>) 42