

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
==== Increasing the difficulty	11
== Bevy Pong	12
==== Key Points	13
==== Technologies Used	13

= Outline

==== Links	13
- Artsy Projects	14
== Krita Palette Creator	15
==== Key Points	15
==== Technologies Used	15
==== Links	15
== Ray Tracing in Rust	16
==== Key Points	16
==== Technologies Used	16
==== Links	16
== Kait	17
==== Key Points	17
==== Technologies Used	17
==== Links	17
== Pixel Art	18
==== Rust for Lunch	19
==== Fractured Elements	19

= Outline

== Scanlation Work	20
==== Skills Learned	21
= Non-game Development Projects	22
== VRCX Insights	23
==== Key Points	23
==== Technologies Used	23
==== Links	23
== Booth Archiver	24
==== Key Points	24
==== Technologies Used	24
==== Links	24
== Real-Time WebSocket Synchronization with Durable Objects	25
==== Key Points	25
==== Technologies Used	25
==== Links	25
== Brainfuck Compiler with Cranelift	26
==== Key Points	26

= Outline

==== Technologies Used	26
==== Links	26
== NixOS	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
==== Tired	31
==== Wired	31
= Appendix	32
== List of Images	33

= Outline

== List of Tables	36
== Table of Links	36

Portfolio Presentation

Aalto University Game Design and Development Interview

Herschel Pravin Pawar

March 19, 2025

= Introduction





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



Herschel Pravin Pawar
sakurakat.systems

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.

cc GitHub:pawarherschel/UniOfAaltoInterviewSlides

= Game Development Projects

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

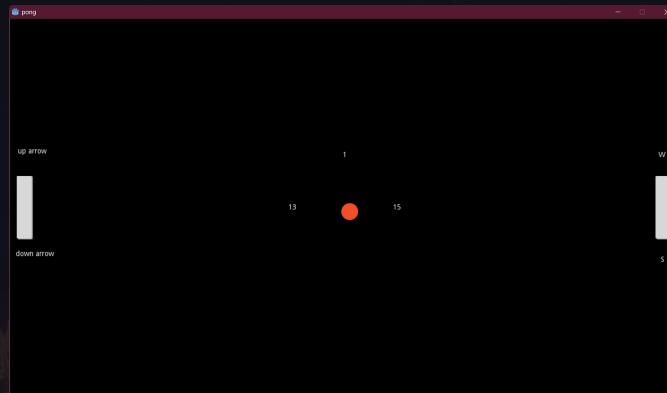


Figure 2: Pong: It starts here
original file @/resources/godot/pong.png



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

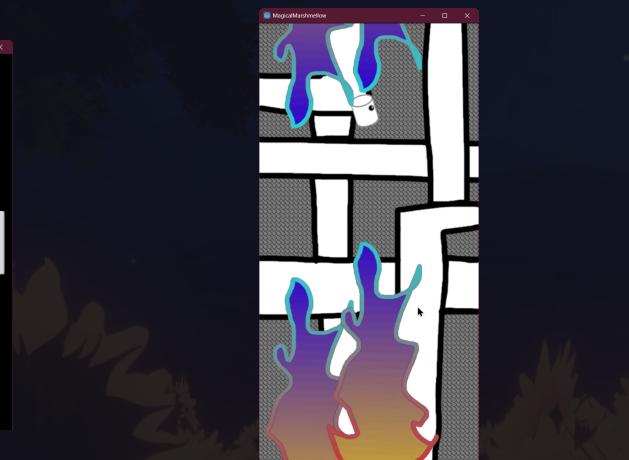


Figure 3: Magical Marshmallow: made my first sfx
original file @/resources/godot/magical_marshmallow.png



Figure 6: Minesweeper: Made the sprites
original file @/resources/godot/minesweeper.png

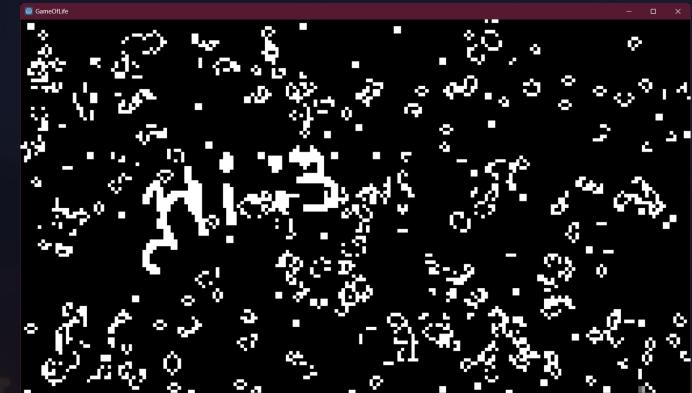


Figure 4: Conway's Game of Life: I ❤️ Maths
original file @/resources/godot/game_of_life.png



Figure 7: Pop The Lock: Used BFXR for SFX
original file @/resources/godot/pop_the_lock.png

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

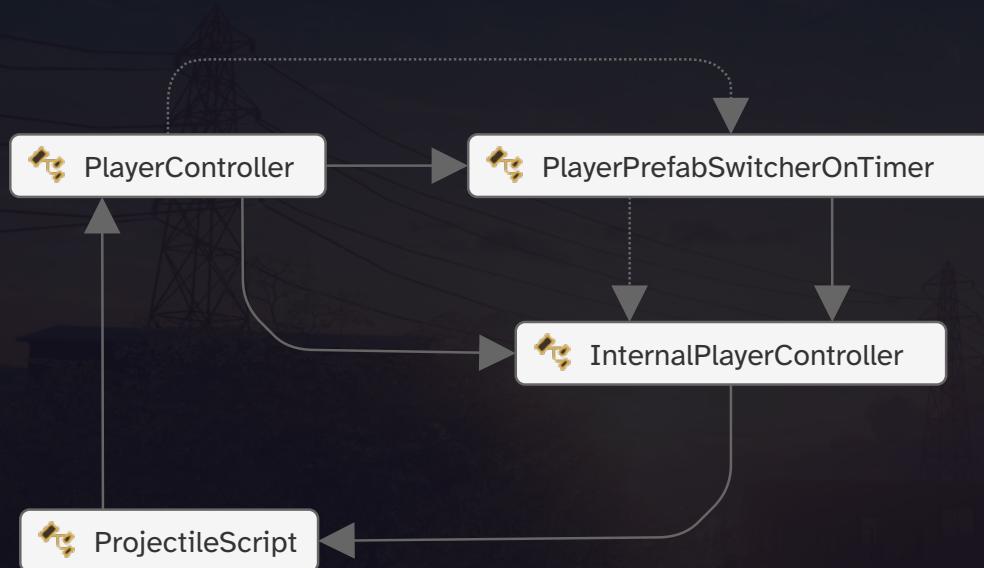


Figure 9: Player Script Graph: Low Coupling
and High Cohesion

original file

@/resources/fractured-elements/player-scripts-hierarchy.svg

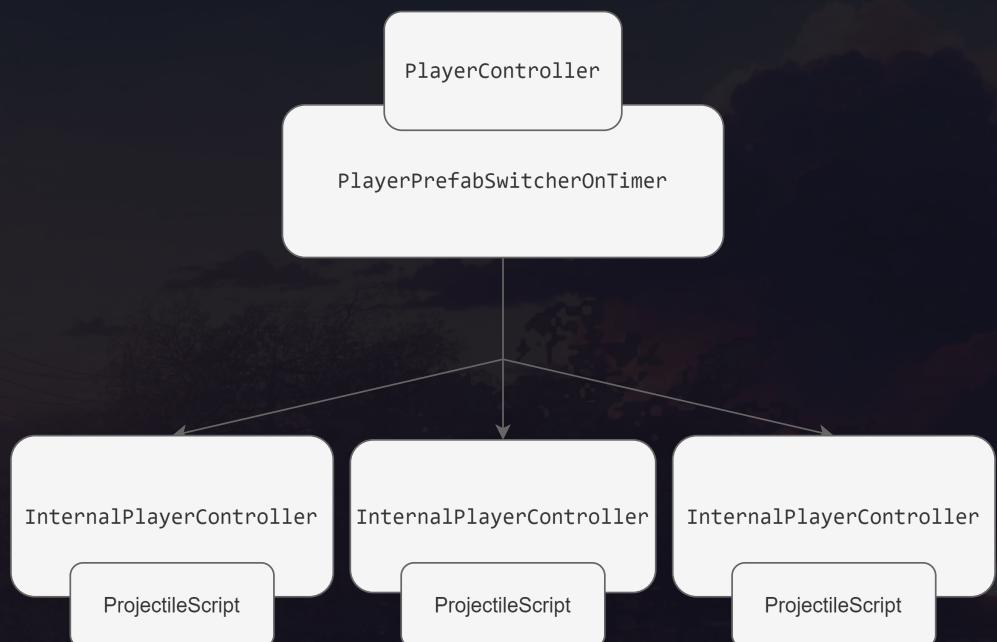


Figure 10: Player Script Hierarchy: Low
Coupling and High Cohesion

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 Drawf
 - Bow and Arrow (ranged) for Elf

==== Technologies Used

- Unity

==== Links

- [GitHub:pawarherschel/CovUniGJ2024](https://github.com/pawarherschel/CovUniGJ2024)
- [itch:pawarherschel:fractured-elements](https://itch.io/game/join/fractured-elements)



Figure 12: Main Mechanic: Roles Reversed

original file

@/resources/cosmos-conquerors/cosmos-conquerors.png

==== Key Points

- Solo Development

==== Technologies Used

- Godot

==== Links

- [GitHub:pawarherschel/GMTK2023](#)
- [itch:pawarherschel:cosmos-conquerors](#)

==== Increasing the difficulty



Figure 13: All invaders shooting
original file

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

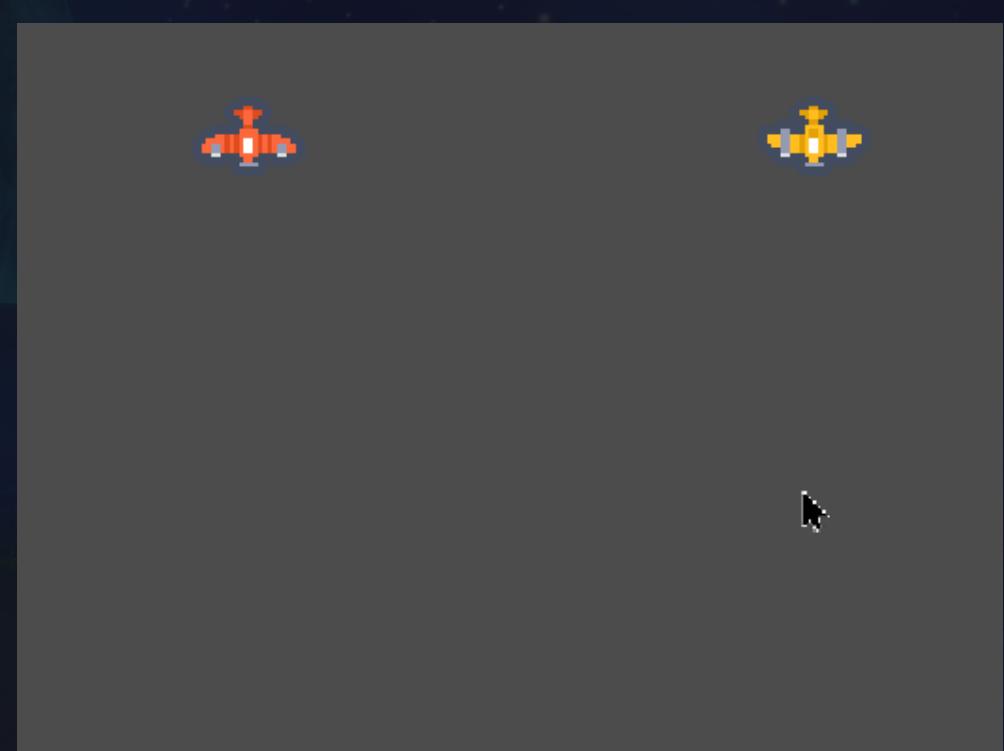


Figure 14: Two invaders shooting
original file

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

== Bevy Pong

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



```
20 fn respawn_handler(
21     mut respawn: EventReader<Respawn>,
22     mut query: Query<
23         (&mut Transform, &mut Velocity), With<Ball>
24     >,
25 ) { /* ... */ }
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>)>, Without<Ball>)
35     >,
36     mut flip: EventWriter<Flip>,
37 ) { /* ... */ }
```



== Bevy Pong

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



==== Key Points

- Learn ECS and Bevy

==== Technologies Used

- Rust
- Bevy (Game Engine)

==== Links

[GitHub:pawarherschel/bevy-pong](https://github.com/pawarherschel/bevy-pong)

= Artsy Projects

- Krita Palette Creator September 2024
- Ray Tracing in Rust November 2023
- Kait 2021
- Pixel Art { March 2024, 2024, 2023 }
- Scanlation Work 2020



Figure 15: Original Image

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



Figure 16: Quantized Image

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



Figure 17: Colors Used

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

==== Key Points

- Implemented color quantization by using median cut algorithm

==== Technologies Used

- Rust

==== Links

- [Rosetta Code Page](#)
- [Wikipedia:Median_cut => Median Cut](#)
- [Wikipedia:Color_quantization => Color Quantization](#)
- [GitHub:pawarherschel/krita-palette-creator](#)

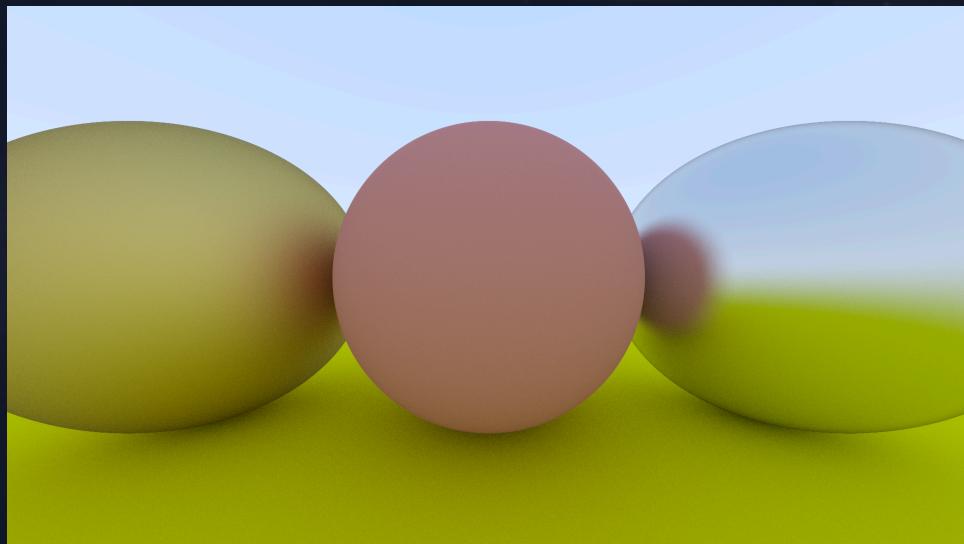


Figure 18: Ray Tracer Output
[original file @/resources/raytracing.png](#)

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 ... 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

== Kait

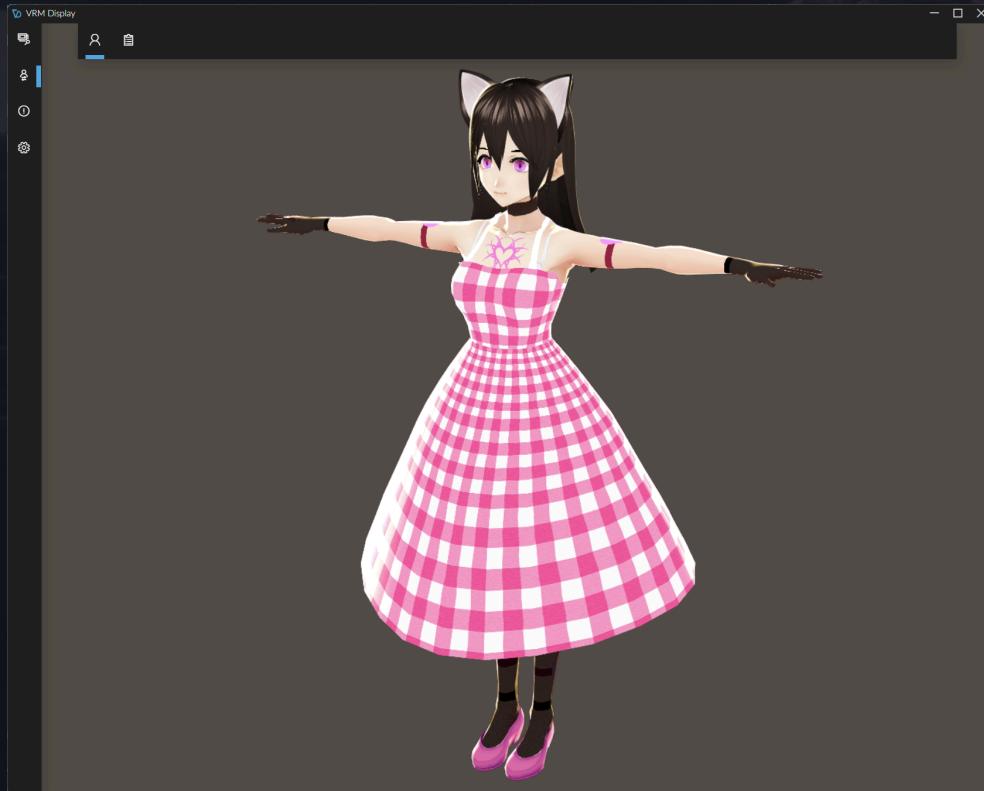


Figure 19: Original Character: Kait
[original file @/resources/kait.png](#)

==== Key Points

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

==== Technologies Used

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

==== Links

- [VRM File](#)



Figure 20: Pixel Art of Charmander waving it's arms looking stupid (affectionate)
[original file @/resources/pixel-art/banner.gif](#)

== Pixel Art

==== Rust for Lunch



Figure 21: Rust for Lunch Fan Art
[original file](#) @/resources/pixel-art/rfl.png

==== Fractured Elements



Figure 22: Fractured Elements Logo
[original file](#)
@/resources/pixel-art/fractured-elements.png



Figure 23: Original Ver.
original file

@/resources/scanlation/original.jpg



Figure 24: 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
- Brainfuck Compiler with Cranelift November 2024
- NixOS October 2024

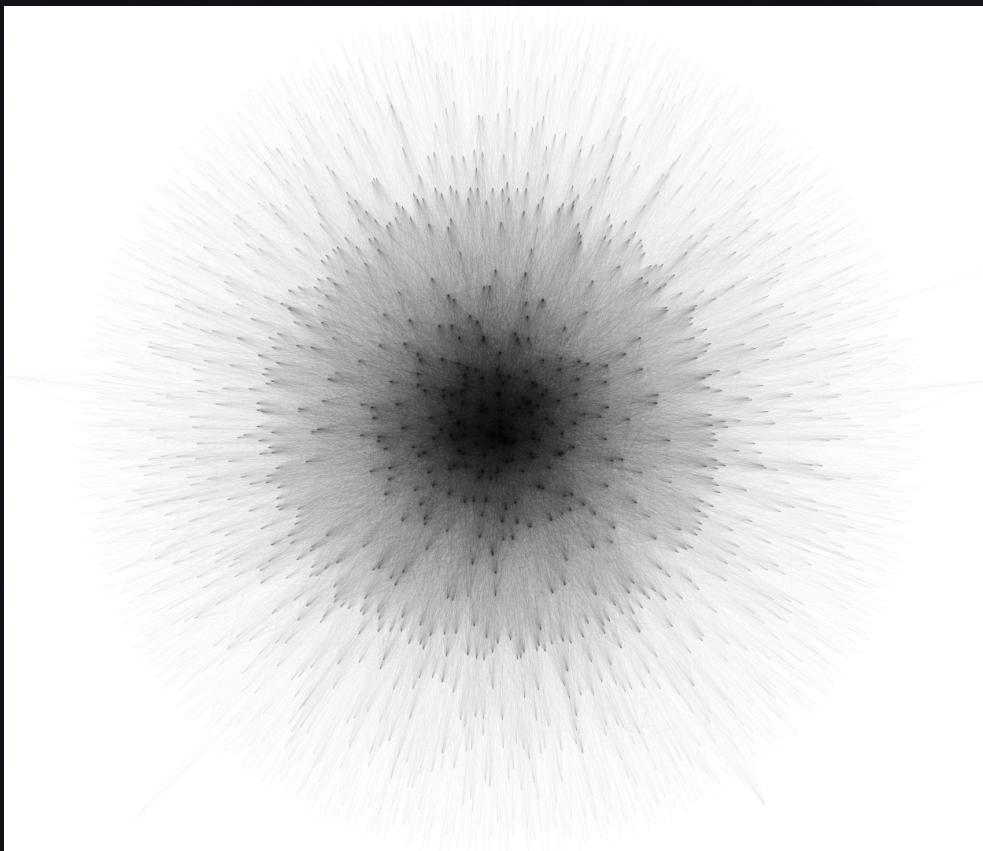


Figure 25: Preview of the image generated by the program

original file @/resources/vrcx.png

==== Key Points

- Implemented a simple (but surprisingly effective) data mining algorithm based on thresholds
- Uses an external data source (not controlled by me)

==== Technologies Used

- Rust
- Tokio (Async Framework)
- SQLite

==== Links

- [GitHub:pawarherschel/vrcx-insights](https://github.com/pawarherschel/vrcx-insights)

==== Key Points

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

==== Technologies Used

- Rust

==== Links

[GitHub:pawarherschel/booth_archiver](#)

“ 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
- Durable Objects
- WebSocket
- TypeScript

==== Links

- GitHub:[pawarherschel/workers...](https://github.com/pawarherschel/workers...)

==== 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](#)

==== Key Points

- NixOS is a linux distro known for its declarative and reproducible system configurations
- “Flakes” help remove inconsistencies

==== Technologies Used

- Linux
- Nix

==== Links

- [GitHub:pawarherschel/nixos-config](https://github.com/pawarherschel/nixos-config)

= Skills

Just a quick overview :3

== Personal Interests

== 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

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

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

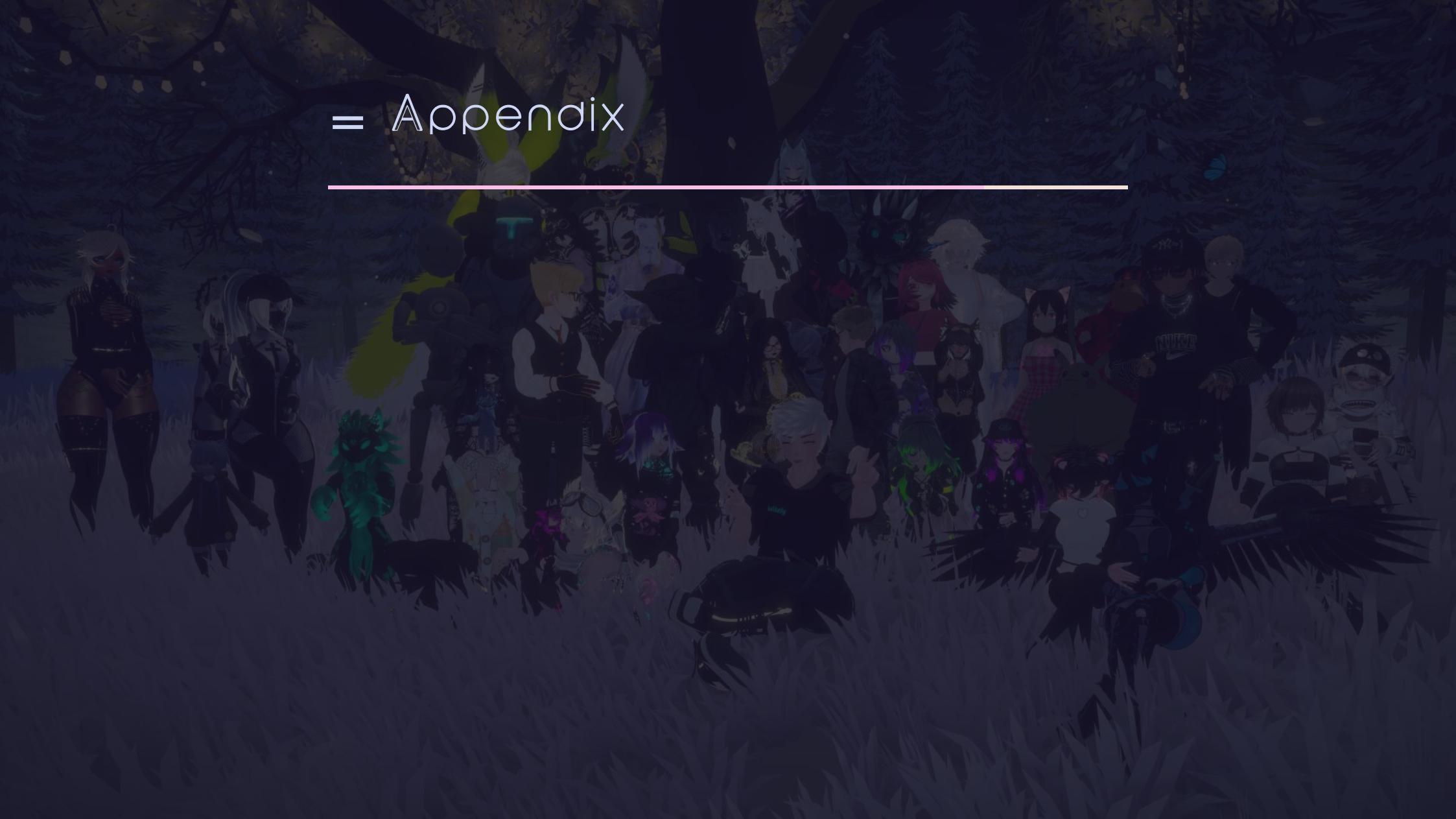
==== Tired

- Memory safety without garbage collection
- Fast code, comparable to C
- Increasing industry adoption
- One language which truly fits all
 - Game Development: Bevy ECS
 - Rendering: WGPU
 - Embedded: Embedded 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 Marshmellow: made my first sfx
[original file @/resources/godot/magical_marshmellow.png](#) 13
- Figure 4: Conway's Game of Life: I ❤ Maths
[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: Made the sprites
[original file @/resources/godot/minesweeper.png](#) 13
- Figure 7: Pop The Lock: Used BFXR for SFX
[original file @/resources/godot/pop_the_lock.png](#) 13
- Figure 8: Main Mechanic: Continuous Change
[original file @/resources/fractured-elements/hero-switching.gif](#) 14
- Figure 9: Player Script Graph: Low Coupling and High Cohesion
[original file @/resources/fractured-elements/player-scripts-hierarchy.svg](#) 15
- Figure 10: Player Script Hierarchy: Low Coupling and High Cohesion
[original file @/resources/fractured-elements/player.svg](#) 15

== List of Images

- 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/cosmos-conquerors.png	17
- Figure 13: All invaders shooting original file @/resources/cosmos-conquerors/invaders-shooting.gif	18
- Figure 14: Two invaders shooting original file @/resources/cosmos-conquerors/two-invaders-shooting.gif	18
- Figure 15: Original Image original file @/resources/krita-palette-creator/original.png	22
- Figure 16: Quantized Image original file @/resources/krita-palette-creator/processed.png	22
- Figure 17: Colors Used original file @/resources/krita-palette-creator/palette.png	22
- Figure 18: Ray Tracer Output original file @/resources/raytracing.png	23
- Figure 19: Original Character: Kait original file @/resources/kait.png	24
- Figure 20: Pixel Art of Charmander waving it's arms looking stupid (affectionate) original file @/resources/pixel-art/banner.gif	25

== List of Images

- Figure 21: Rust for Lunch Fan Art original file @/resources/pixel-art/rfl.png	26
- Figure 22: Fractured Elements Logo original file @/resources/pixel-art/fractured-elements.png	26
- Figure 23: Original Ver. original file @/resources/scanlation/original.jpg	27
- Figure 24: Edited Ver. original file @/resources/scanlation/product.jpg	27
- Figure 25: Preview of the image generated by the program original file @/resources/vrcx.png	30

== Table of Links

pg no	Accompanying Text	Link Destination
11	sakurakat.systems	https://sakurakat.systems
11	GitHub:pawarherschel/ UniOfAaltoInterviewSlides	https://github.com/pawarherschel/UniOfAaltoInterviewSlides
16	Terraria	https://www.terraria.org
16	GitHub:pawarherschel/CovUniGJ2024	https://github.com/pawarherschel/CovUniGJ2024
16	itch:pawarherschel:fractured-elements	https://pawarherschel.itch.io/fractured-elements
17	GitHub:pawarherschel/GMTK2023	https://github.com/pawarherschel/GMTK2023

== Table of Links

pg no	Accompanying Text	Link Destination
17	itch:pawarherschel:cosmos-conquerors	https://pawarherschel.itch.io/cosmos-conquerors
20	GitHub:pawarherschel/bevy-pong	https://github.com/pawarherschel/bevy-pong
22	Rosetta Code Page	https://rosettacode.org/wiki/Color_quantization#Rust
22	Wikipedia:Median_cut => Median Cut	Median_cut
22	Wikipedia:Color_quantization => Color Quantization	Color_quantization
22	GitHub:pawarherschel/krita-palette-creator	https://github.com/pawarherschel/krita-palette-creator

== Table of Links

pg no	Accompanying Text	Link Destination
23	GitHub:pawarherschel/raytracinginrust	https://github.com/pawarherschel/raytracinginrust
24	VRoid Studio	https://vroid.com/en
24	VRM File	https://r2.sakurakat.systems/kait.vrm
27	Danbooru:pools:17121	https://danbooru.donmai.us/pools/17121
27	Twitter:tuki_nami	https://x.com/tuki_nami/status/1224267670037483520
30	GitHub:pawarherschel/vrcx-insights	https://github.com/pawarherschel/vrcx-insights
31	Booth	https://booth.pm/en

== Table of Links

pg no	Accompanying Text	Link Destination
31	GitHub:pawarherschel/boot_her	https://github.com/pawarherschel/boot_her
32	Cloudflare Durable Objects Documentation	https://developers.cloudflare.com/durable-objects/
32	GitHub:pawarherschel/worker	https://github.com/pawarherschel/worker-durable-object-websocket-synced
33	GitHub:pawarherschel/BFCranelift	https://github.com/pawarherschel/BFCranelift
34	GitHub:pawarherschel/nixos-con	https://github.com/pawarherschel/nixos-con