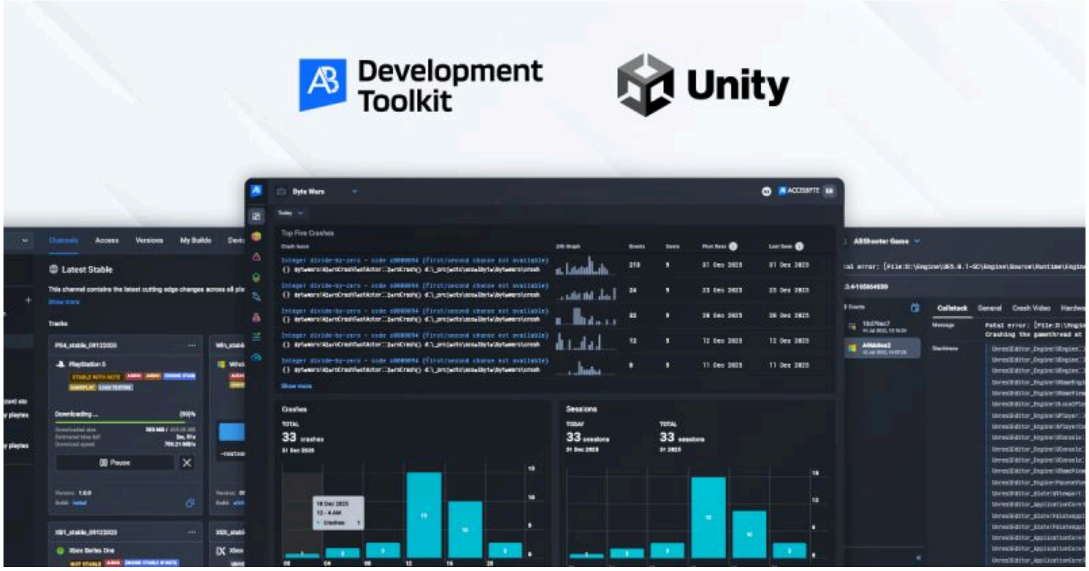
UNITY CRASH REPORTER

Worked on the development and improvement of a **Unity Crash Reporting system** for shipped games, with a strong focus on **Windows platform stability, crash diagnostics, and real-world production issues**.

- Investigated and fixed **crash reproduction and reporting issues on Windows**, ensuring accurate crash capture and video evidence during runtime failures.
- Implemented a **heartbeat monitoring system** within Unity to reliably detect application hangs and crashes, enabling precise identification of crash timing and root causes.
- Improved crash reliability by handling edge cases where crashes occur outside managed (C#) code, including **native** and **engine**-level failures. Made stacktrace working perfectly by utilizing IL2CPP.

Tech: **Unity**, **C++**, C#

Link:
https://www.linkedin.com/posts/accelbyte_if-youve-ever-shipped-a-unity-game-you-activity-7382091995793690624-kqJI?utm_source=share&utm_medium=member_desktop&rcm=ACoAAC9i1DoBDp0grvB1GuiNO5l7T5eeiuUu6Pw



Half Life Mod

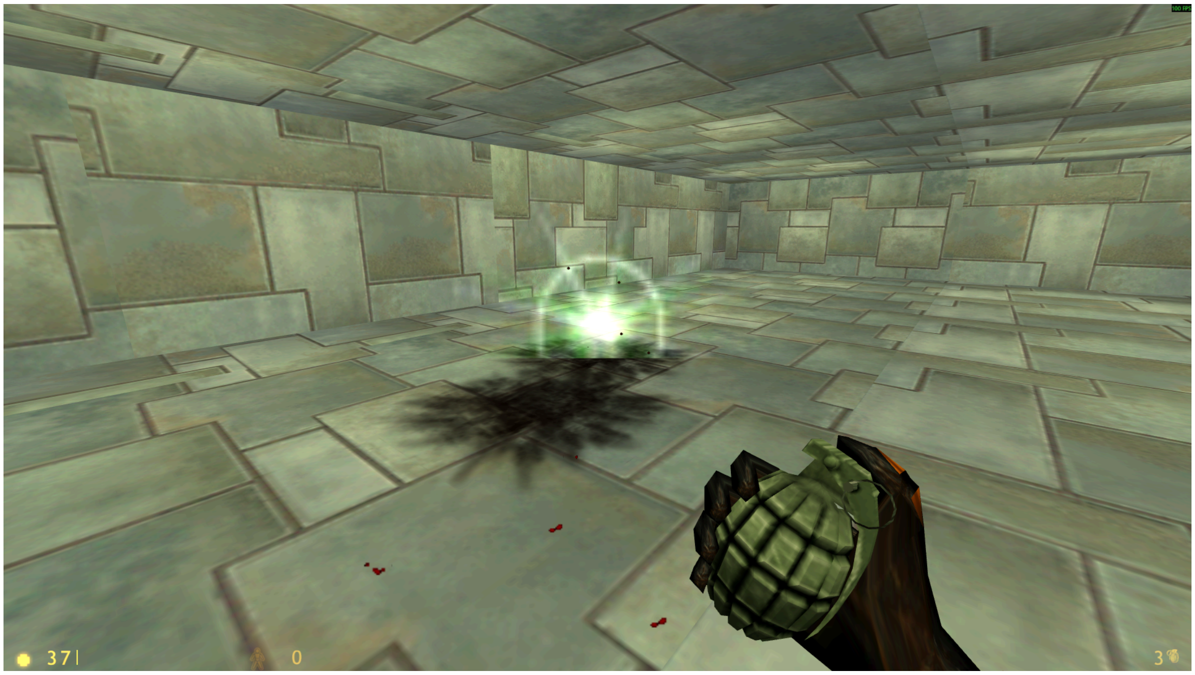
Responsible for developing Half Life mod, named "Project Recon" (LINK), using GoldSrc Valve game engine, specifically **Featureful SDK** (LINK). Work in a team of six people, and I'm the only developer. Collaborating with people in map designer, sound designer, and voice actor roles. Focus on developing **custom grenades** within the mod. Fire grenade, Hallucination grenade, and Corrosive grenade. All new grenades come with a unique custom effect. Mod is planned for release in mid-2026.

Tech: **GoldSrc**, **C++**, J.A.C.K Editor

Link:

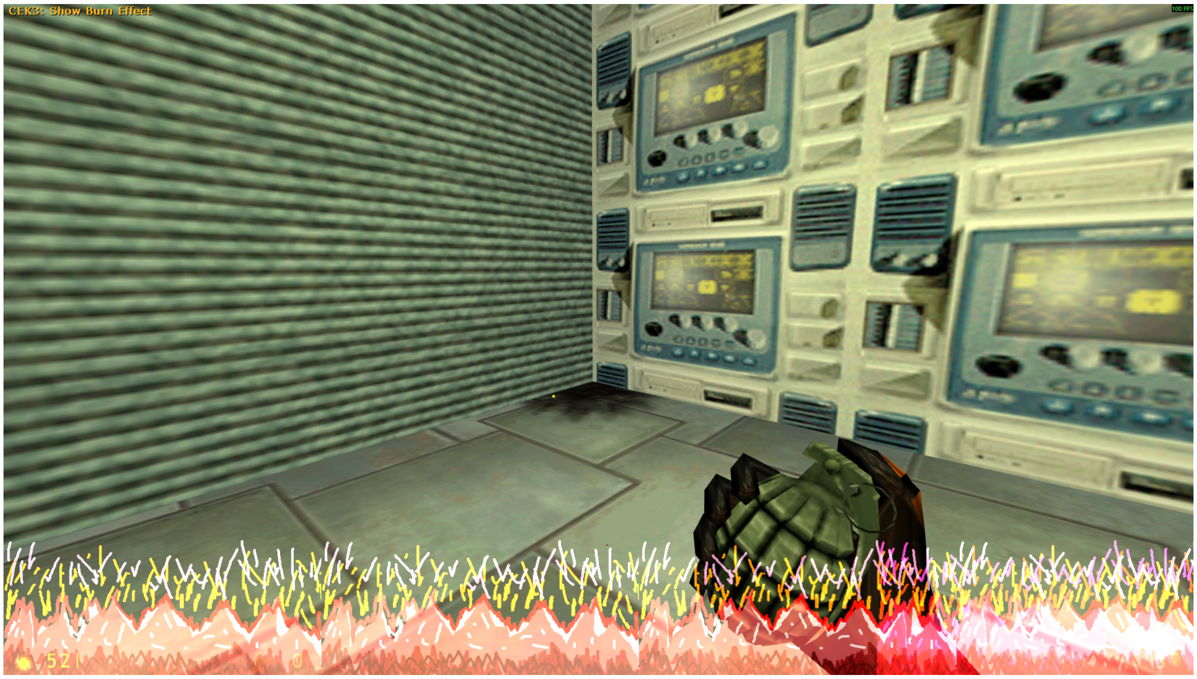
- https://www.moddb.com/mods/half-life-recon
- https://github.com/mhnaufal/halflife-featureful

Custom Smoke.
We develop a custom smoke sprite and animation to engage more players with beautiful graphics.



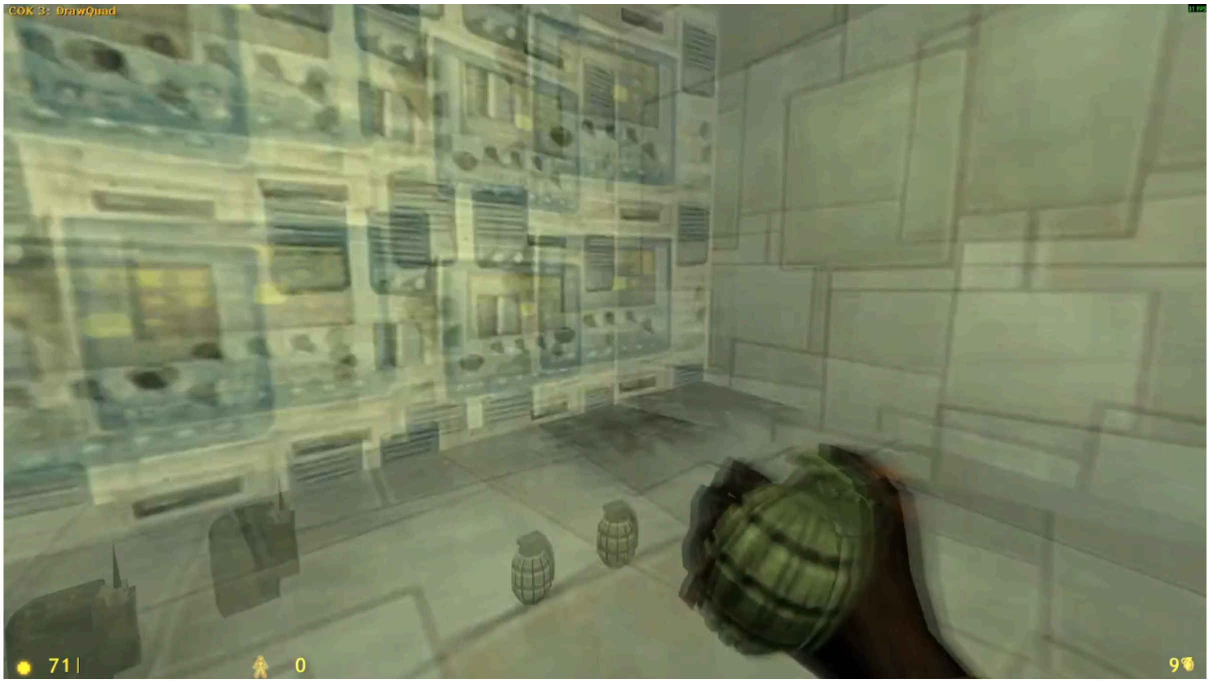
Player Camera Burned Effect.

The burn grenade creates a fire-and-smoke effect around the player's camera, mimicking the feeling of being burned, like in any modern game.



Player Camera Blur Effect.

The hallucination grenade made the player's camera blur, mimicking the effect of the smoke.

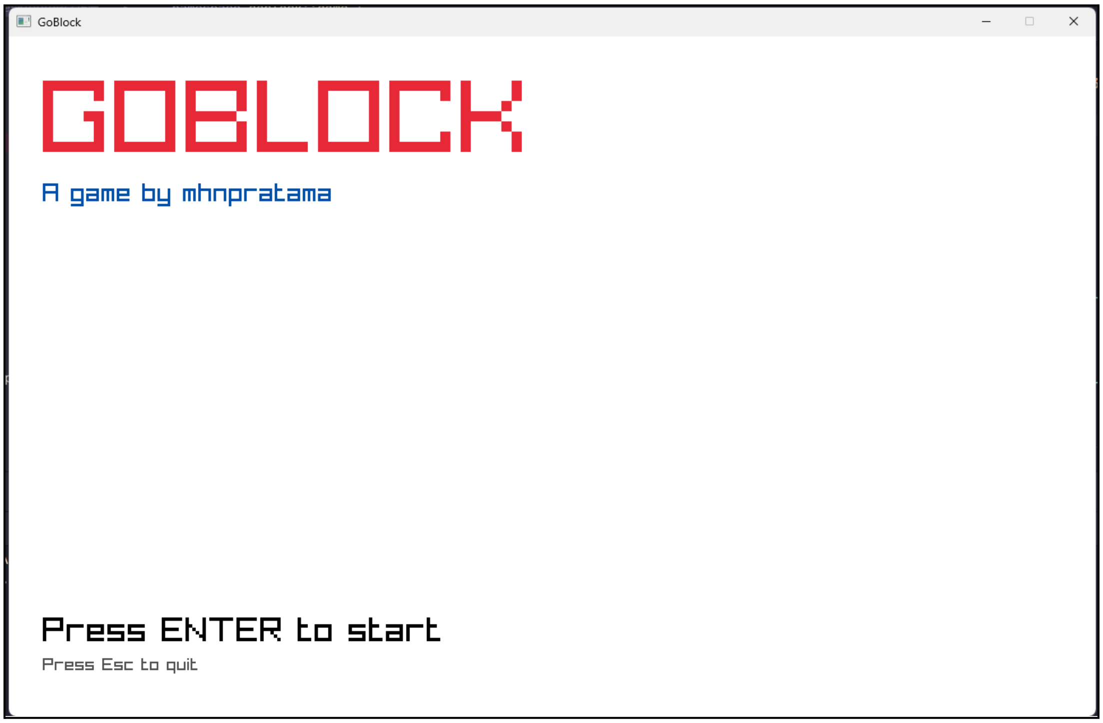


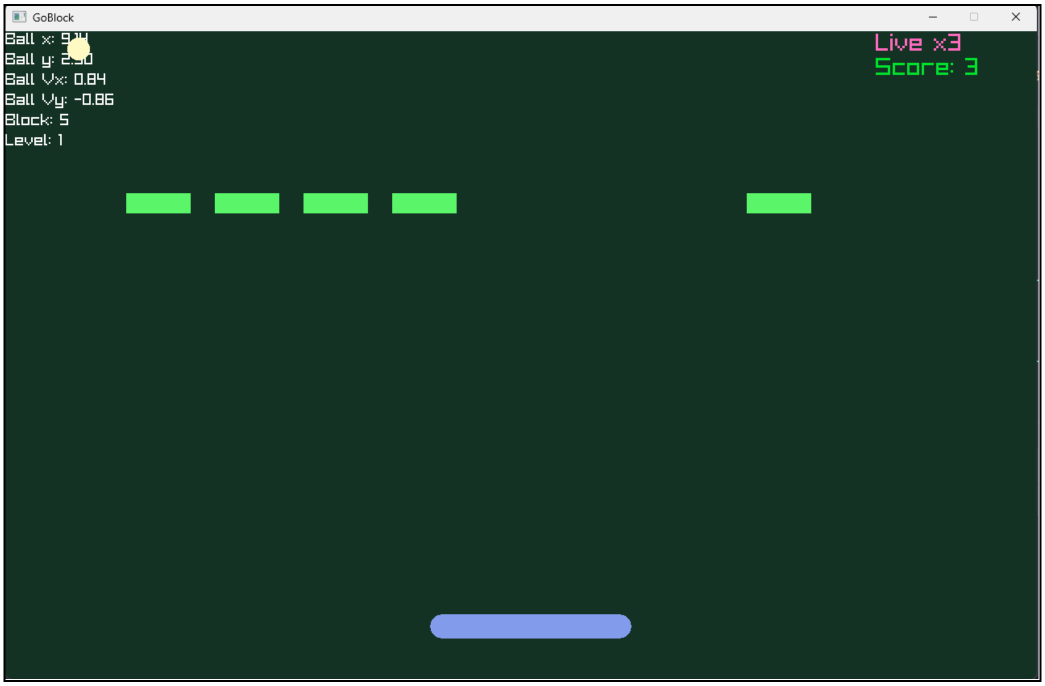
GoBlock

Develop a ***block destroyer*** game in pure **C++** without a game engine. Featuring C++ as the programming language with additional game libraries such as **Raylib** for most of the graphics, sound, and gameplay, **Flecs** for entity-component-system (ECS), and **Conan** for managing dependencies.

Tech: C++, Raylib, Flecs, Conan.

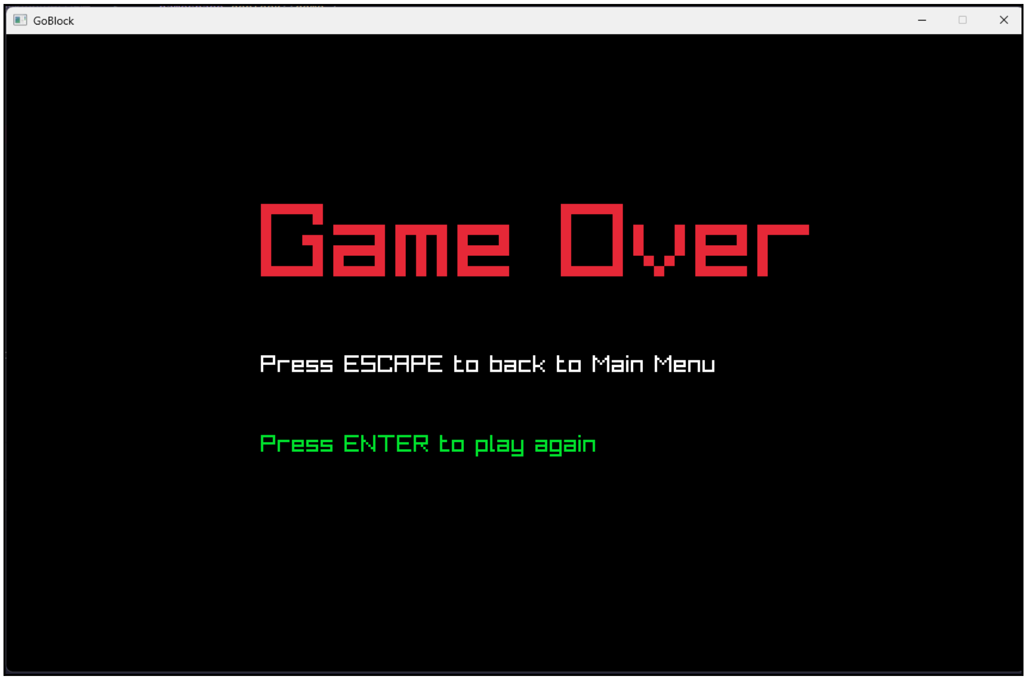
- **GoBlock**: https://github.com/mhnaufal/goblock





```
Ball x: 9.14
Ball y: 2.50
Ball Vx: 0.84
Ball Vy: -0.86
Block: 5
Level: 1
```

```
Live x3
Score: 3
```



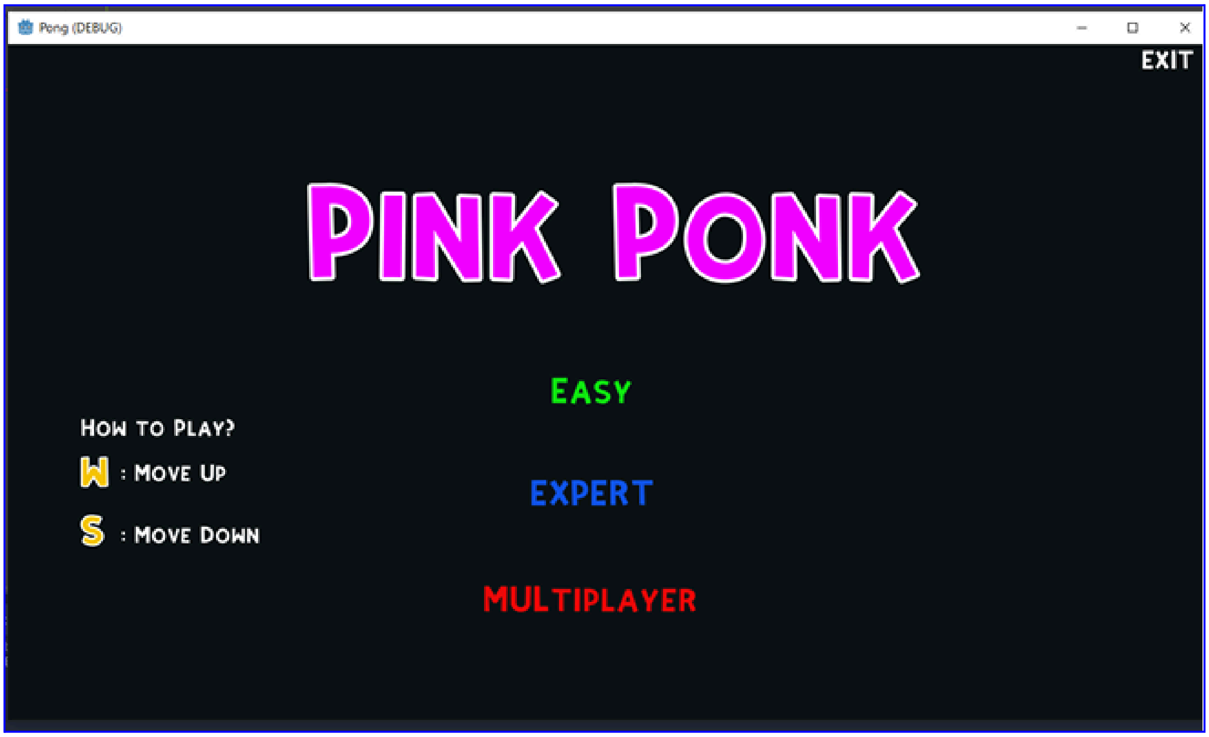
```
Game Over

Press ESCAPE to back to Main Menu

Press ENTER to play again
```

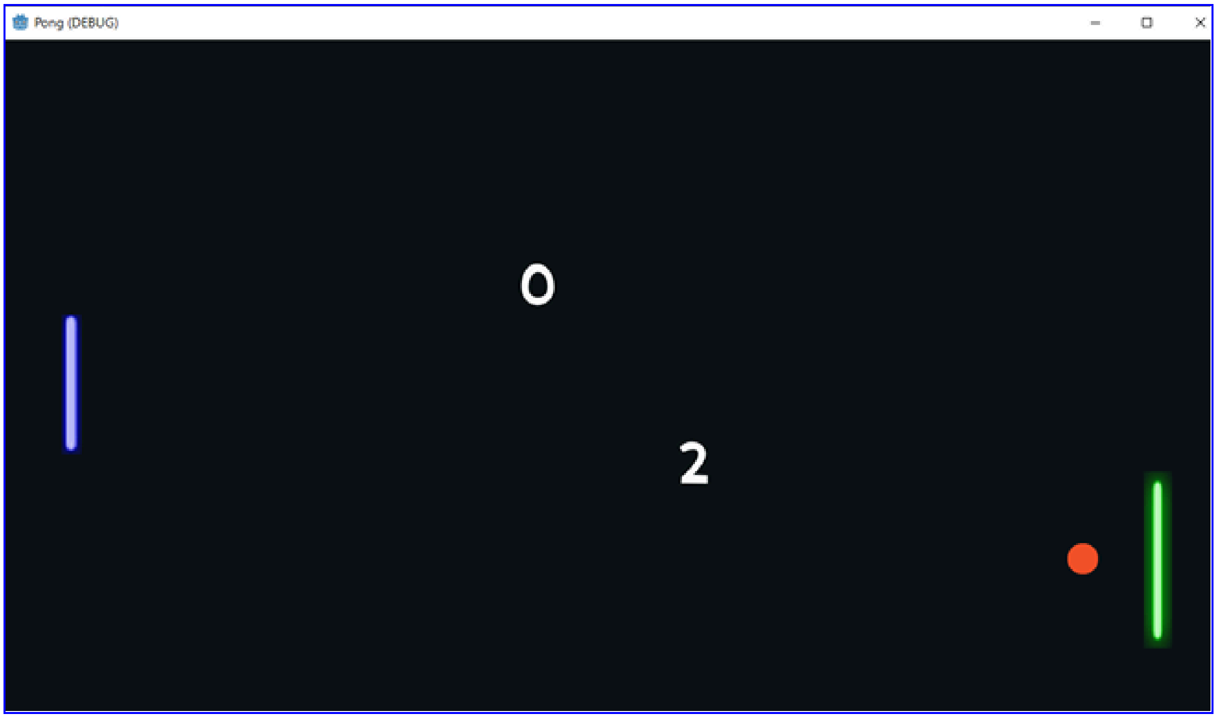
Pink Ponk

A ***ping-pong*** two local multiplayer game developed using **Godot 3**. The AI computer is too smart to predict where the ball gonna fall.

Tech: Godot

- **PinkPonk**: https://github.com/mhnaufal/PinkPonk





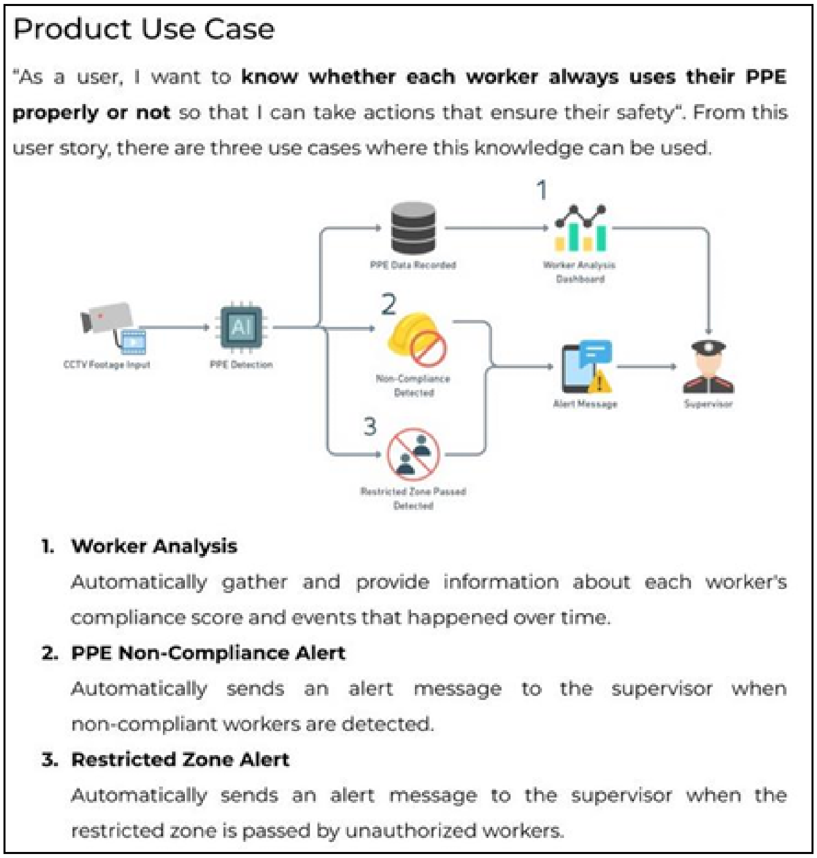
Pong (DEBUG)
0
2

PPE Detection

PPE Detection is one of the three final projects during my internship at PT Nodeflux Teknologi Indonesia. I worked with the other three members in this project to develop a PPE detection system. The PPE detection runs on the web platform and checks whether workers are not wearing safety equipment, such as helmets and vests. Detection is performed using deep learning models. My role on this project is as a Software Engineer, managing the software architecture and codebase.

Tech: Python, Streamlit, Tensorflow, Git.

Link:
https://www.linkedin.com/posts/christianale_personal-protective-equipment-prd-visionaire-activity-7015619743797694464-b901?utm_source=share&utm_medium=member_desktop



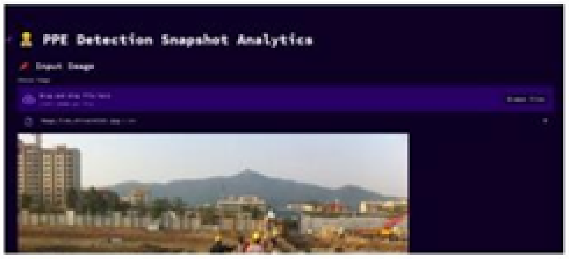
Product Use Case

“As a user, I want to **know whether each worker always uses their PPE properly or not** so that I can take actions that ensure their safety”. From this user story, there are three use cases where this knowledge can be used.

CCTV Footage Input
AI
PPE Detection
1
PPE Data Recorded
Worker Analysis Dashboard
2
Non-Compliance Detected
3
Restricted Zone Passed Detected
Alert Message
Supervisor

1. **Worker Analysis**
   Automatically gather and provide information about each worker's compliance score and events that happened over time.
2. **PPE Non-Compliance Alert**
   Automatically sends an alert message to the supervisor when non-compliant workers are detected.
3. **Restricted Zone Alert**
   Automatically sends an alert message to the supervisor when the restricted zone is passed by unauthorized workers.

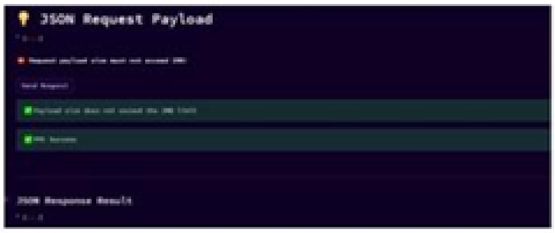
User Interface



PPE Detection Snapshot Analytics

Input Image

[illegible]

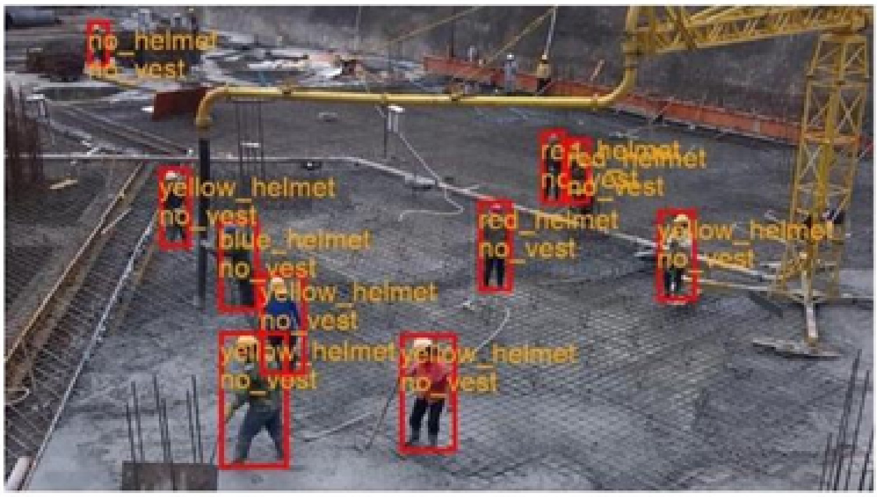


JSON Request Payload

[illegible]

JSON Response Result

Example of Output Image

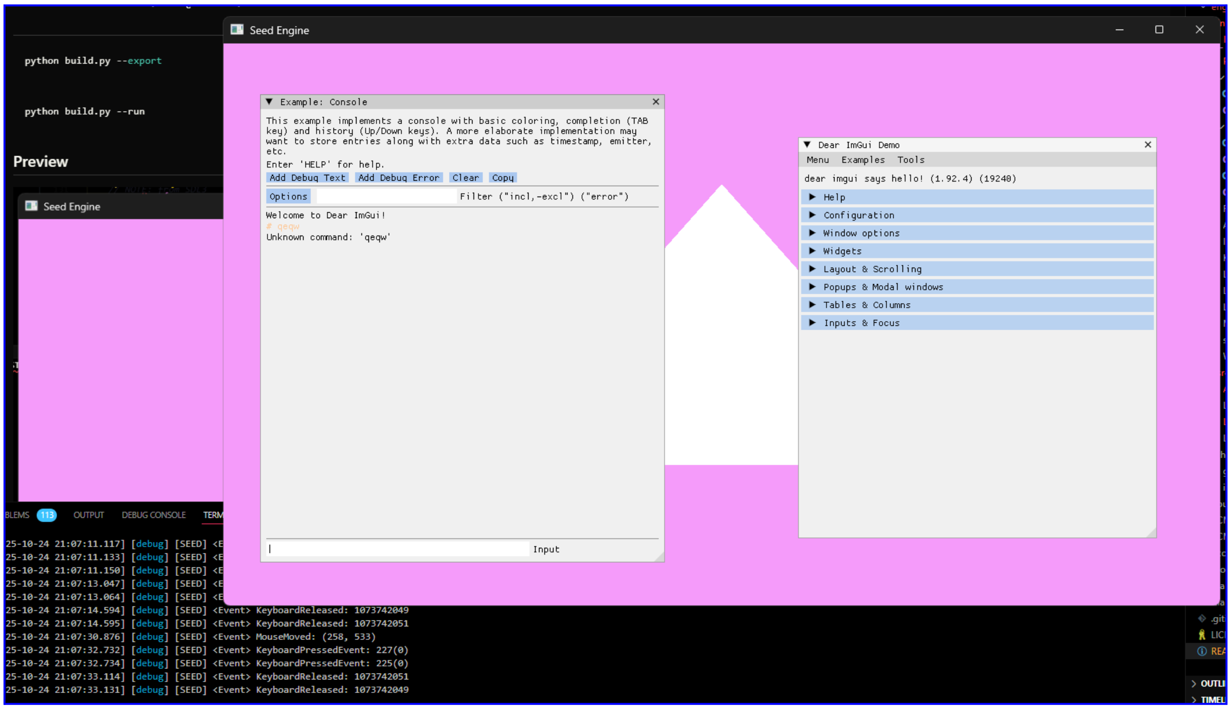


Seed Engine

Seed, a custom game engine developed using ***C++***, ***SDL GPU***, ***ImGUI***, and ***miniaudio***. Focusing on the most straightforward and minimal approach for developing a game. Powered by an easy custom shader that mainly targeted to render a "*toon style*" graphic.

Tech: C++, SDL GPU, OpenGL, Vulkan, spdlog, Conan, ImGUI.

- Seed: https://github.com/mhnaufal/seed-engine

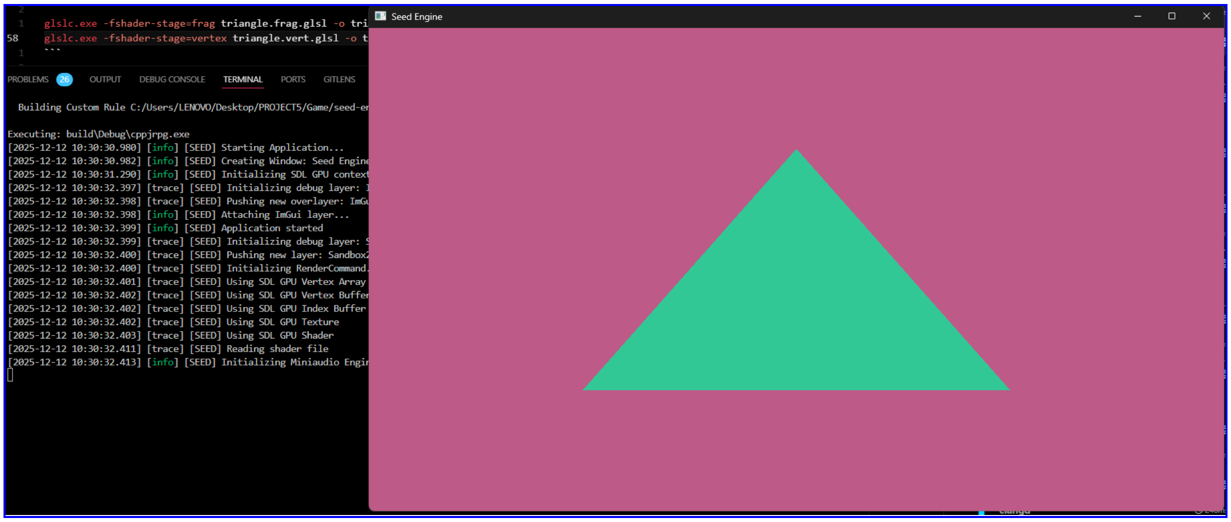


```
python build.py --export

python build.py --run
```

```
This example implements a console with basic coloring, completion (TAB
key) and history (Up/Down keys). A more elaborate implementation may
want to store entries along with extra data such as timestamp, emitter,
etc.
Enter 'HELP' for help.
Filter ("incl,-excl") ("error")
Welcome to Dear ImGui!
# qeqw
Unknown command: 'qeqw'
```

```
dear imgui says hello! (1.92.4) (19240)
```

```
25-10-24 21:07:11.117] [debug] [SEED] <E
25-10-24 21:07:11.133] [debug] [SEED] <E
25-10-24 21:07:11.150] [debug] [SEED] <E
25-10-24 21:07:13.047] [debug] [SEED] <E
25-10-24 21:07:13.064] [debug] [SEED] <E
25-10-24 21:07:14.594] [debug] [SEED] <Event> KeyboardReleased: 1073742049
25-10-24 21:07:14.595] [debug] [SEED] <Event> KeyboardReleased: 1073742051
25-10-24 21:07:30.876] [debug] [SEED] <Event> MouseMoved: (258, 533)
25-10-24 21:07:32.732] [debug] [SEED] <Event> KeyboardPressedEvent: 227(0)
25-10-24 21:07:32.734] [debug] [SEED] <Event> KeyboardPressedEvent: 225(0)
25-10-24 21:07:33.114] [debug] [SEED] <Event> KeyboardReleased: 1073742051
25-10-24 21:07:33.131] [debug] [SEED] <Event> KeyboardReleased: 1073742049
```



```
glslc.exe -fshader-stage=frag triangle.frag.glsl -o tri
glslc.exe -fshader-stage=vertex triangle.vert.glsl -o t
```

```
 Building Custom Rule C:/Users/LENOVO/Desktop/PROJECT5/Game/seed-en

Executing: build\Debug\cppjrpg.exe
[2025-12-12 10:30:30.980] [info] [SEED] Starting Application...
[2025-12-12 10:30:30.982] [info] [SEED] Creating Window: Seed Engine
[2025-12-12 10:30:31.290] [info] [SEED] Initializing SDL GPU context
[2025-12-12 10:30:32.397] [trace] [SEED] Initializing debug layer: I
[2025-12-12 10:30:32.398] [trace] [SEED] Pushing new overlayer: ImGu
[2025-12-12 10:30:32.398] [info] [SEED] Attaching ImGui layer...
[2025-12-12 10:30:32.399] [info] [SEED] Application started
[2025-12-12 10:30:32.399] [trace] [SEED] Initializing debug layer: S
[2025-12-12 10:30:32.400] [trace] [SEED] Pushing new layer: Sandbox2
[2025-12-12 10:30:32.400] [trace] [SEED] Initializing RenderCommand.
[2025-12-12 10:30:32.401] [trace] [SEED] Using SDL GPU Vertex Array
[2025-12-12 10:30:32.402] [trace] [SEED] Using SDL GPU Vertex Buffer
[2025-12-12 10:30:32.402] [trace] [SEED] Using SDL GPU Index Buffer
[2025-12-12 10:30:32.402] [trace] [SEED] Using SDL GPU Texture
[2025-12-12 10:30:32.403] [trace] [SEED] Using SDL GPU Shader
[2025-12-12 10:30:32.411] [trace] [SEED] Reading shader file
[2025-12-12 10:30:32.413] [info] [SEED] Initializing Miniaudio Engin
```

Muhammad Naufal Pratama
mhnpratama@gmail.com
https://mhnaufal.vercel.app/
https://www.linkedin.com/in/mnpratamaa/