Software Development Portfolio

Aleksey Panas

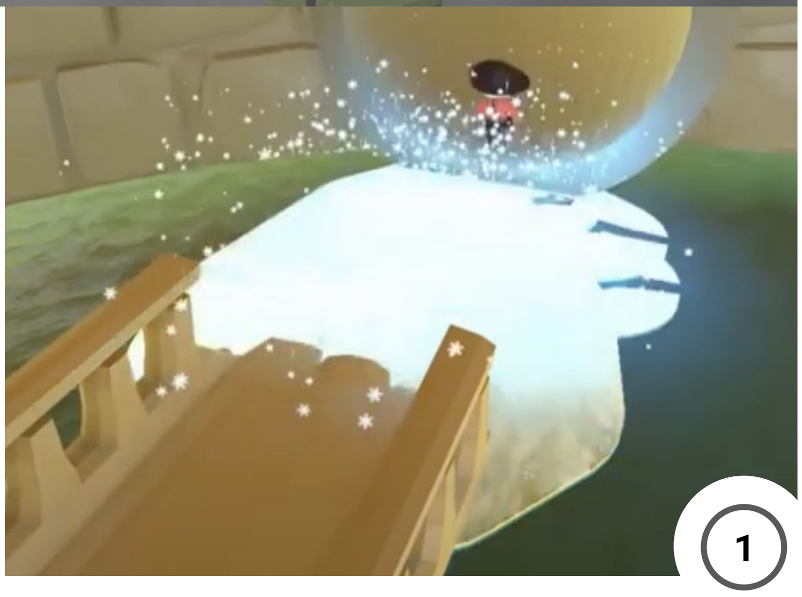
[1/2] Magengaard itch.io link 01/2024 - Present

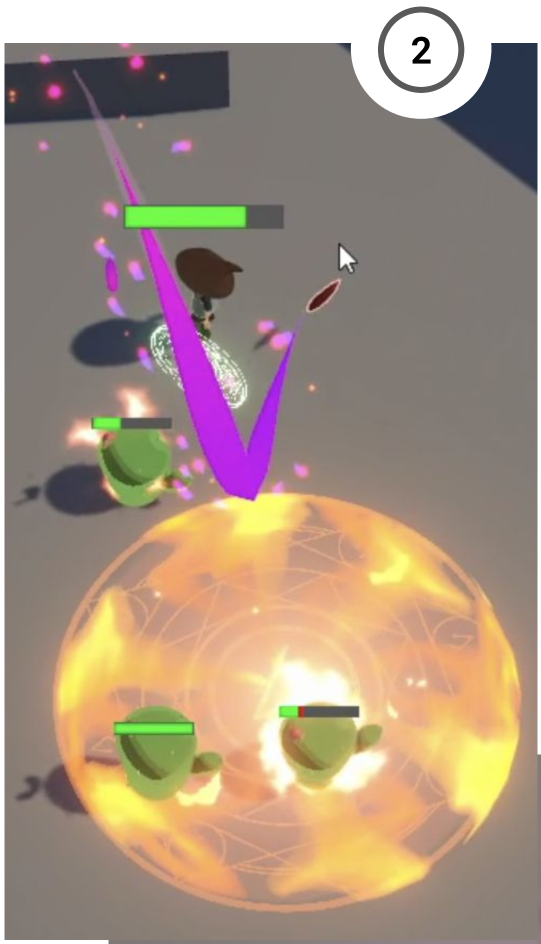
About

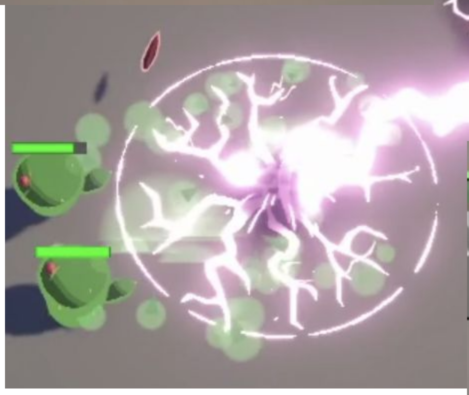
Magengaard is a 3D magic fighting PvP/PvE game where spells are cast by drawing unique symbols on the screen

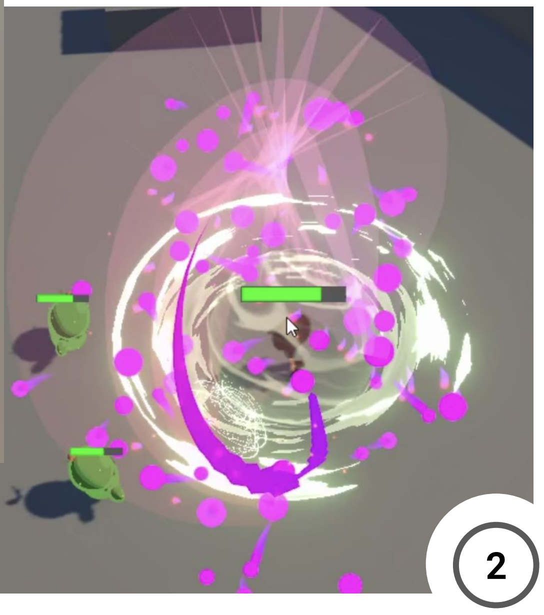
Technologies

- Unity
- C#
- Netcode
- Blender
- Python
- Photoshop
- CSG (Mesh booleans)
- Gesture Recognition
- Git, CI/CD, Agile, Clickup







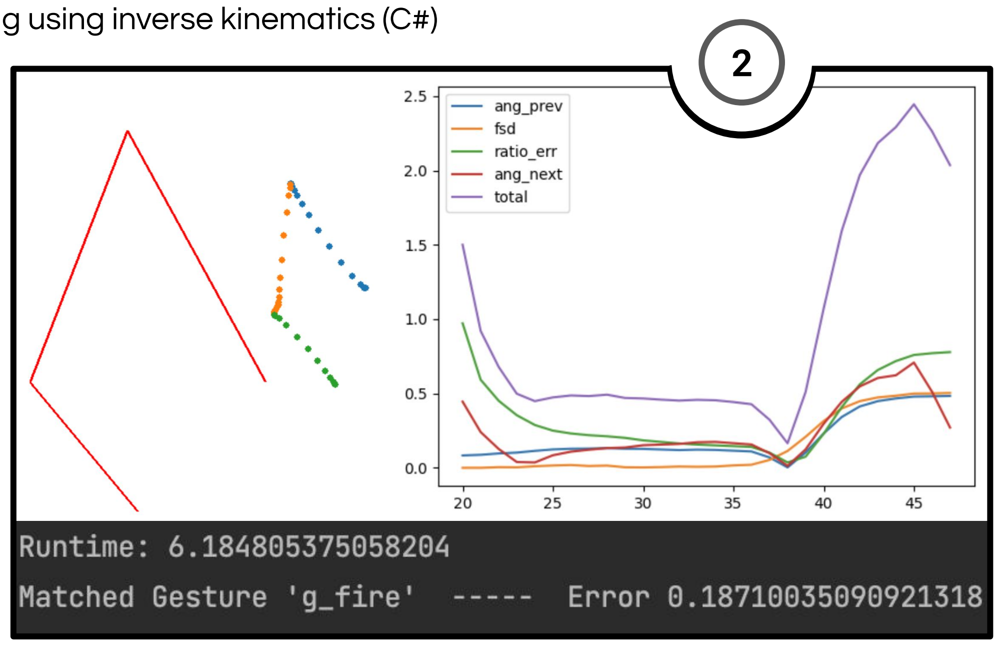


badlek.itch.io/magengaard

Game Demo

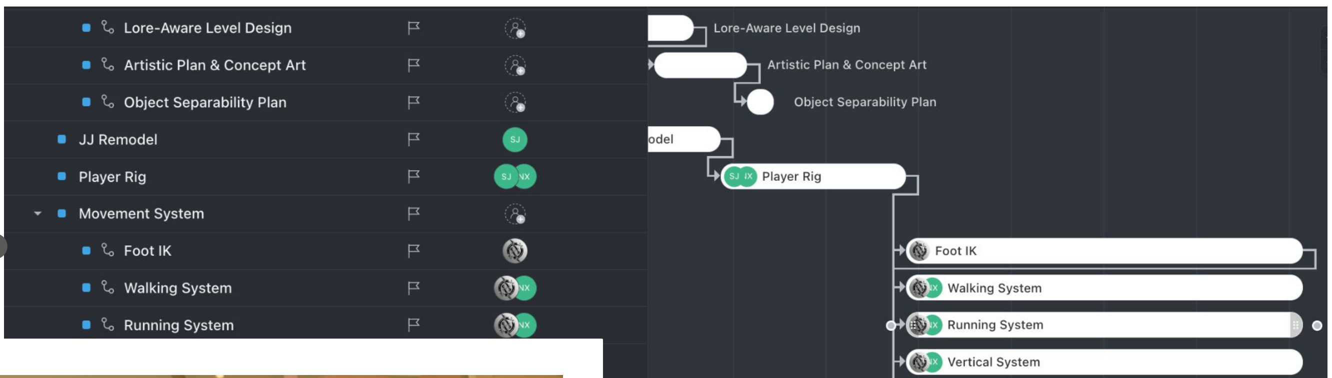
Contribution

- Conceptualized the core game mechanic of gesture-based spellcasting
- 250+ commits including...
  - [1] A custom in-house mesh union algorithm for freezing water (C#)
  - [2] A custom in-house gesture recognition algorithm using function minimization techniques (C#)
  - 3D spellbook UI with pages that turn by mouse dragging using inverse kinematics (C#)
  - System to pick up arbitrary items off the ground (C#)
  - Netcode for multiplayer “Demifox” bossfight where one player controls the boss! (C#)
  - 3D modelling of entire second “Ruins” level in Blender
- Managed our team of 10 artists, programmers, and sound designers, implementing Agile through ClickUp
- Handled team admin tasks, coordinating share distribution and setting development timelines to prepare for upcoming Kickstarter launch and Steam release



```
Runtime: 6.184805375058204
Matched Gesture 'g_fire'  -----  Error 0.18710035090921318
```

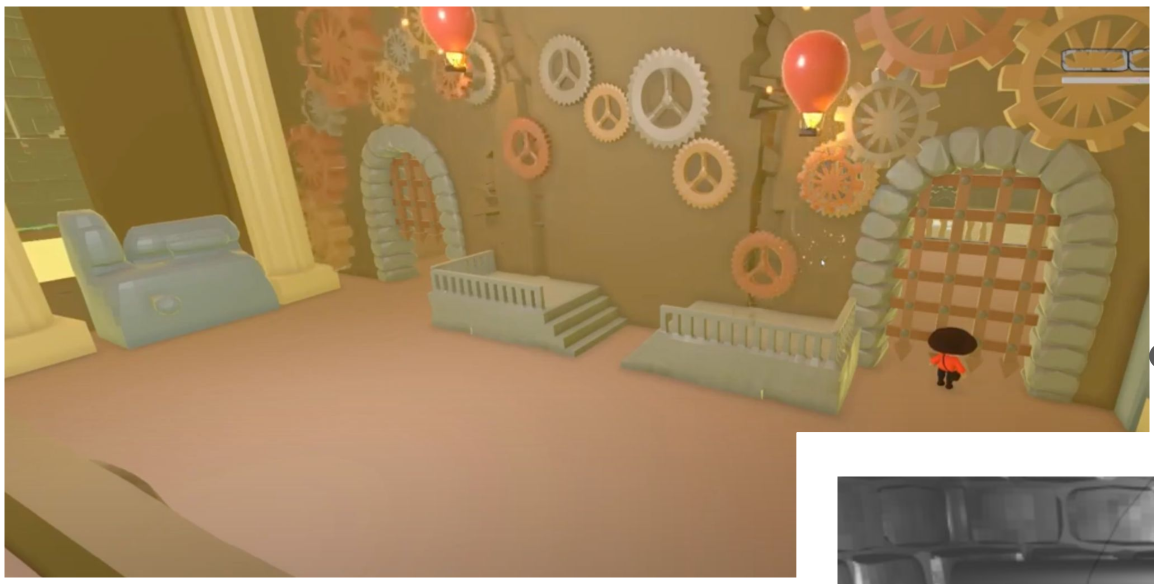
[2/2] Magengaard itch.io link 01/2024 - Present



Lore-Aware Level Design
Artistic Plan & Concept Art
Object Separability Plan
JJ Remodel
Player Rig
Movement System
Foot IK
Walking System
Running System

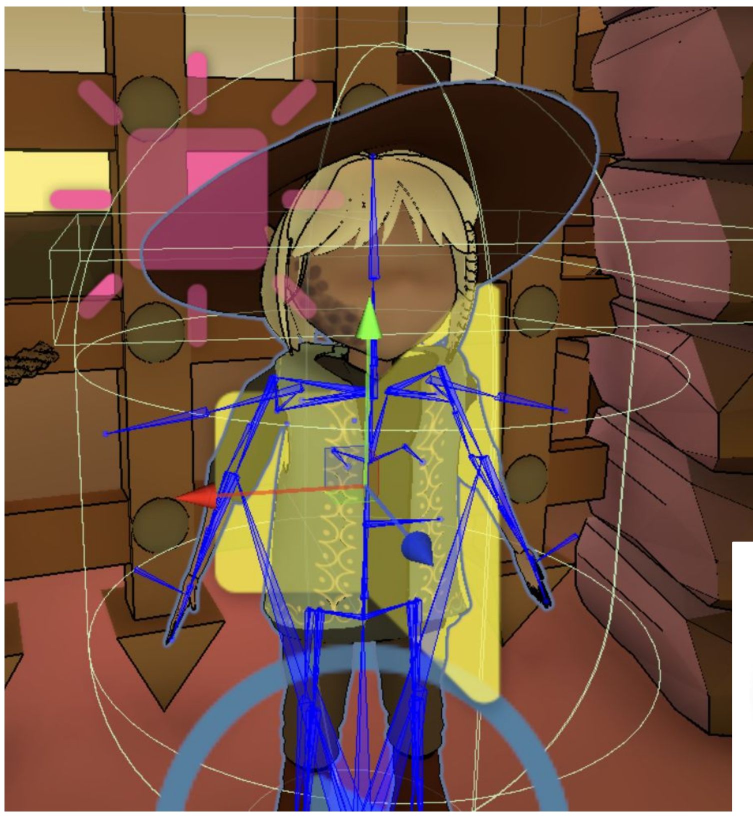
Lore-Aware Level Design
Artistic Plan & Concept Art
Object Separability Plan
odel
Player Rig
Foot IK
Walking System
Running System
Vertical System
Casting Animation System
Spell Hold Effects

Agile project management using Clickup

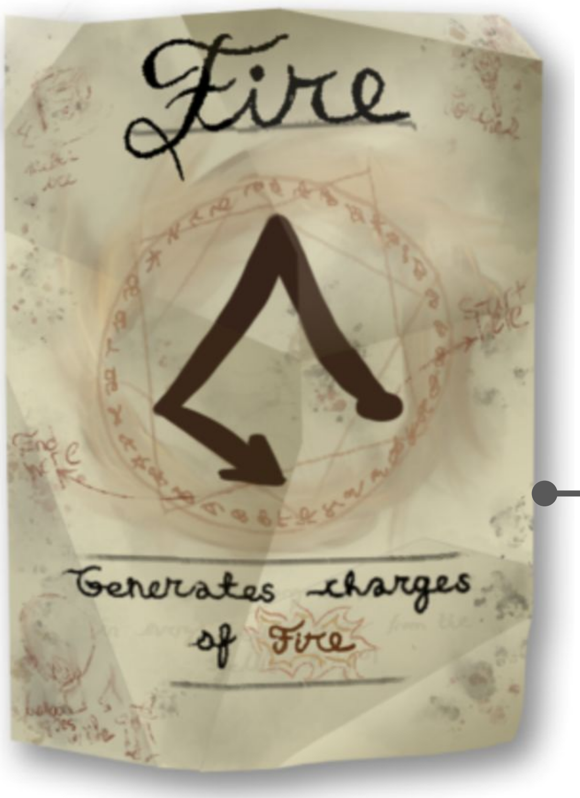


Designed and modelled puzzle in second “Ruins” level

Modelled “boiler room” in second “Ruins” level



Implemented inverse kinematics and animations in code on our rigged main character



Drew the UI page assets that can be picked up throughout the game to learn new spells

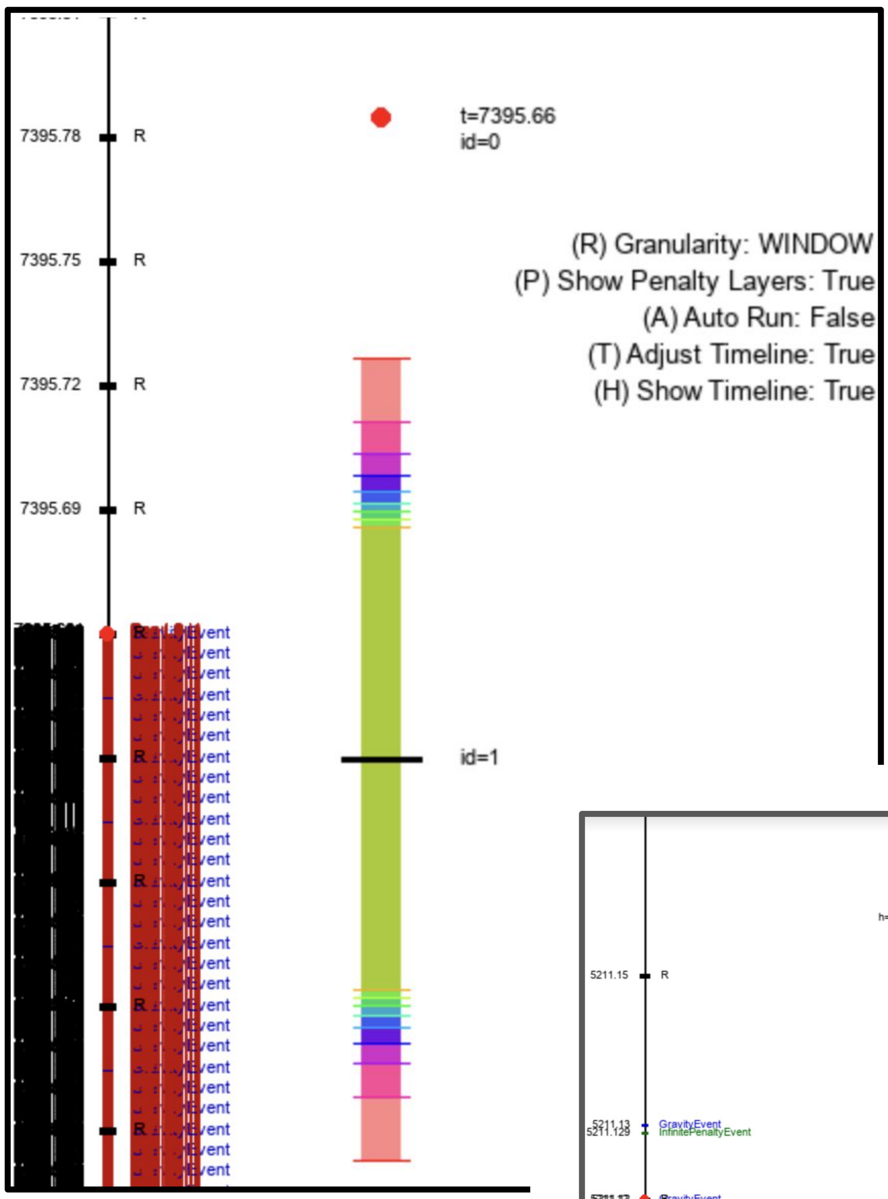
What’s Ahead

While Magengaard’s current PvE experience was great for a proof of concept, we found our PvP multiplayer bossfight very fun!

We are currently working to cut scope and turn Magengaard into a well-polished competitive PvP experience, aiming for a Kickstarter launch and Steam Release in the first half of 2025

Computer Graphics Research

05/2024 - Present



(R) Granularity: WINDOW
(P) Show Penalty Layers: True
(A) Auto Run: False
(T) Adjust Timeline: True
(H) Show Timeline: True

About

Working with supervision on replicating and innovating on an algorithm in the subfield of contact mechanics in physics simulations.

The goal is to implement this algorithm on the GPU using CUDA and leverage its high parallelizable nature

Contribution

Read, understood, and implemented Speculative Parallel Asynchronous Contact Mechanics (SPACM) in **python**

Wrote an extensive visualization and logging system leveraging **tensorbaord, numpy**, **matplotlib** , and **pygame**

Debugged complex **energy behavior** issues in the simulation

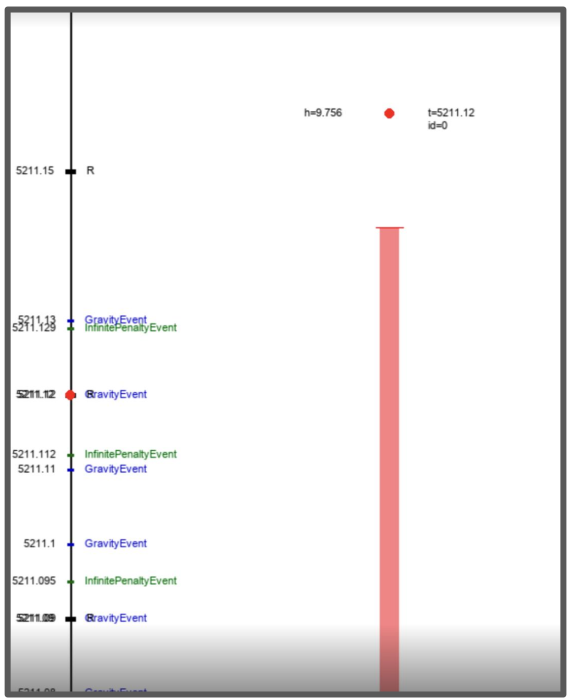
Technologies

- Speculative Parallel Asynchronous Contact Mechanics
- Tensorboard
- Python
- Pygame
- Numpy
- Matplotlib

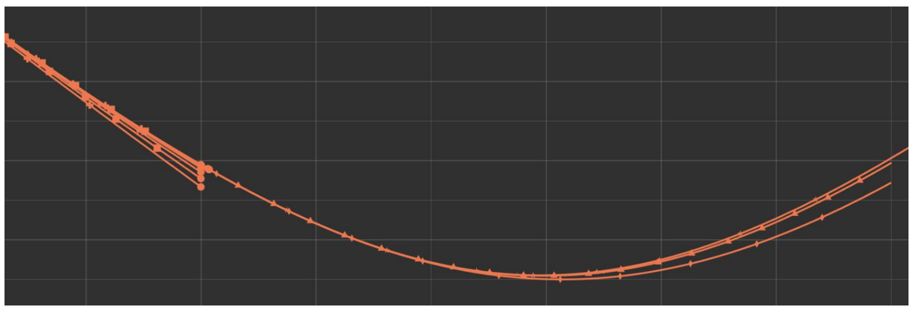
AlekseyPanas/GPUSPACM

What's Ahead

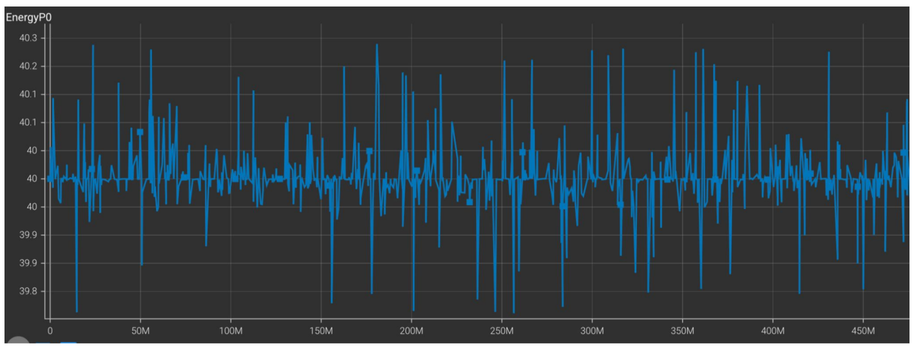
Currently working on implementing SPACM on the GPU using **CUDA** through **warp**



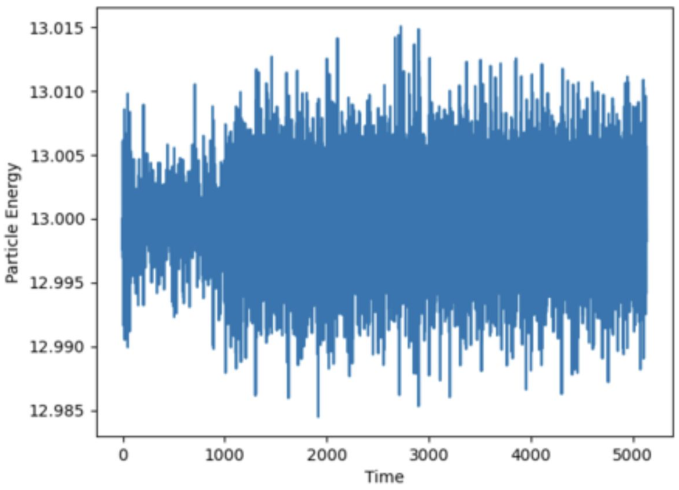
Pygame visualizer for 1D sim: colorful layers are penalty regions which apply collision-preventing repelling forces via repeating events as shown in the timeline view on the left



Tensorboard particle position plot with rollback visualization

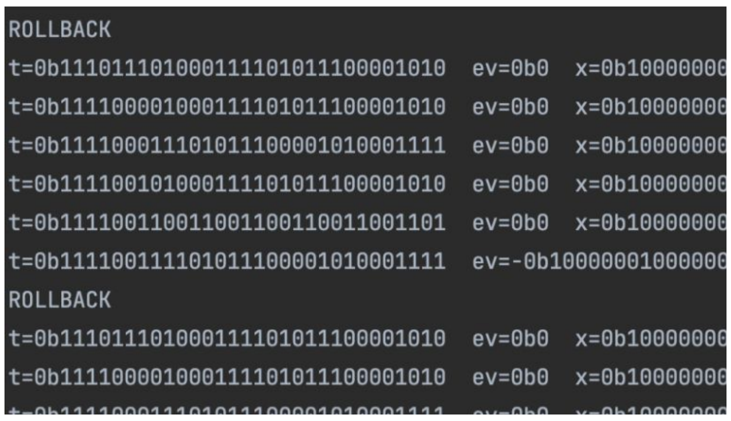


Tensorboard particle energy plot

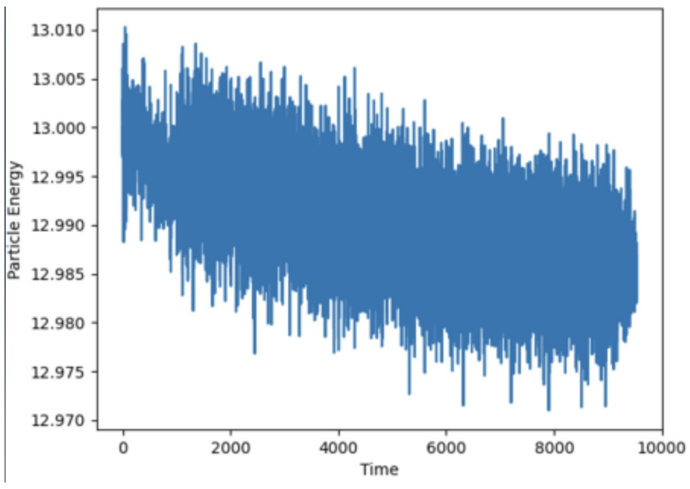


Good energy behavior (after fix)

Simulation logs in binary used to debug bad energy behavior



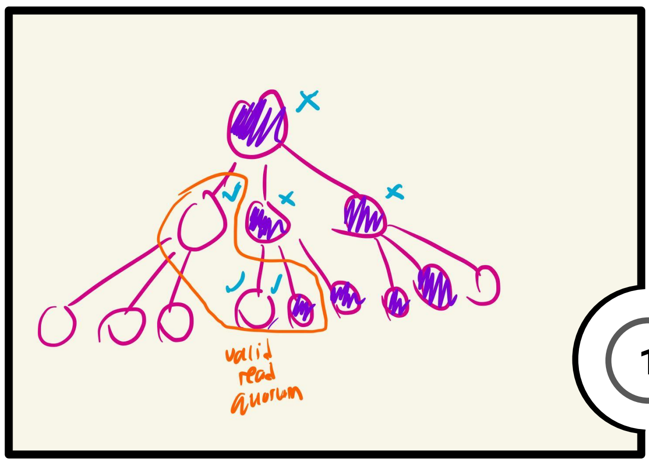
```
ROLLBACK
t=0b11101110100011110101110000101O  ev=0b0  x=0b10000000
```



Bad energy behavior (before fix)

Strong Consistency Research Work

01/2023-Present



1

About

Primary work as Research Assistant involves developing novel algorithms in the strong consistency field of distributed networks. Latest work in particular involves optimization of local reads and attempted generalization from Paxos to other classes of algorithms such as Leaderless.

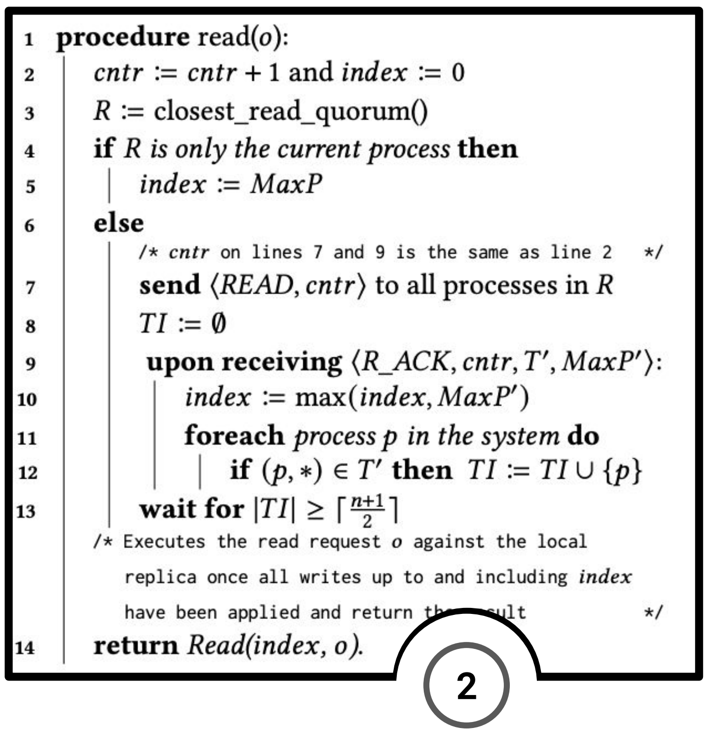
Contact my supervising Professor Eyal de Lara at **delara@cs.toronto.edu** if a reference letter is needed

Contribution

With mentorship from a PhD student...

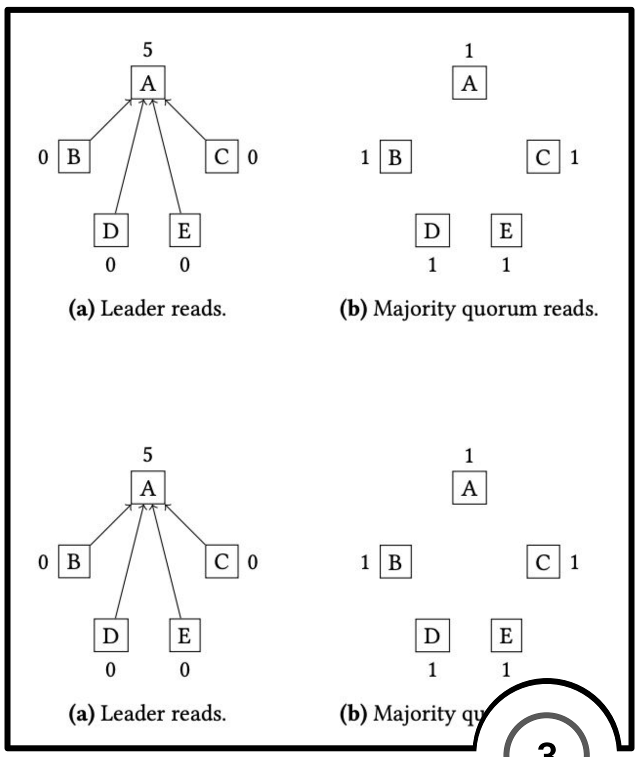
- Contributed over **50%** of the content to a completed research paper .
- Worked on brainstorming and discovering over 10 minor and major innovations in the strong consistency field, particularly for local reads
- Implemented the algorithms to test them experimentally
- Formally analyzed the innovations to ensure their correctness and efficiency compared to other algorithms
- Presented my discoveries to colleagues and Professors in the lab
- Read over **50** papers in the area

Research Paper Link



```
1  procedure read($o$):
2      $cntr := cntr + 1$ and $index := 0$
3      $R :=$ closest_read_quorum()
4      if $R$ is only the current process then
5          $index := MaxP$
6      else
           /* cntr on lines 7 and 9 is the same as line 2 */
7          send $\langle READ, cntr \rangle$ to all processes in $R$
8          $TI := \emptyset$
9          upon receiving $\langle R\_ACK, cntr, T', MaxP' \rangle$:
10             $index := \max(index, MaxP')$
11             foreach process $p$ in the system do
12                 if $(p, *) \in T'$ then $TI := TI \cup \{p\}$
13         wait for $|TI| \geq \lceil \frac{n+1}{2} \rceil$
       /* Executes the read request o against the local
          replica once all writes up to and including index
          have been applied and return the result */
14     return $Read(index, o)$.
```

2



(a) Leader reads. (b) Majority quorum reads.

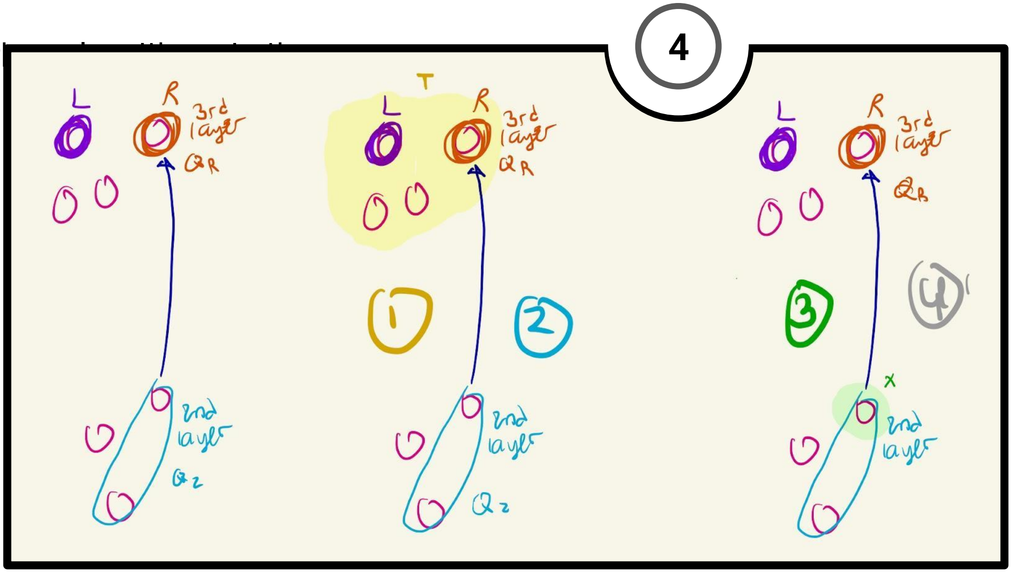
(a) Leader reads. (b) Majority qu

3

Technologies

- Strong Consistency Algorithms (**MultiPaxos** , **Raft** , **Leaderless**)
- Local Reads Algorithms (**Paxos Quorum Leases** , **Paxos Quorum Reads**)
- **Clock Sync Algorithms** , **Lease Algorithms**
- Various properties/metrics of distributed systems: **Fault Tolerance** , **Read/Write latency** , **Throughput** , **Recovery Time** , etc
- Reconfiguration Algorithms: **Matchmaker Paxos** , **Vertical Paxos** , **SMART**
- **Linearizability**
- Quorum Intersection Algorithms: **Crumbling Walls** , **Tree Quorums**
- Many ot

4



Captions

1. Tree Quorums sketch for self-verification
2. Small lemma from algorithm proofs
3. Lease diagram from local reads work
4. Hierarchy scenario comparisons from local reads work

xxFS: Distributed Filesystem Paper

06/2023-08/2023

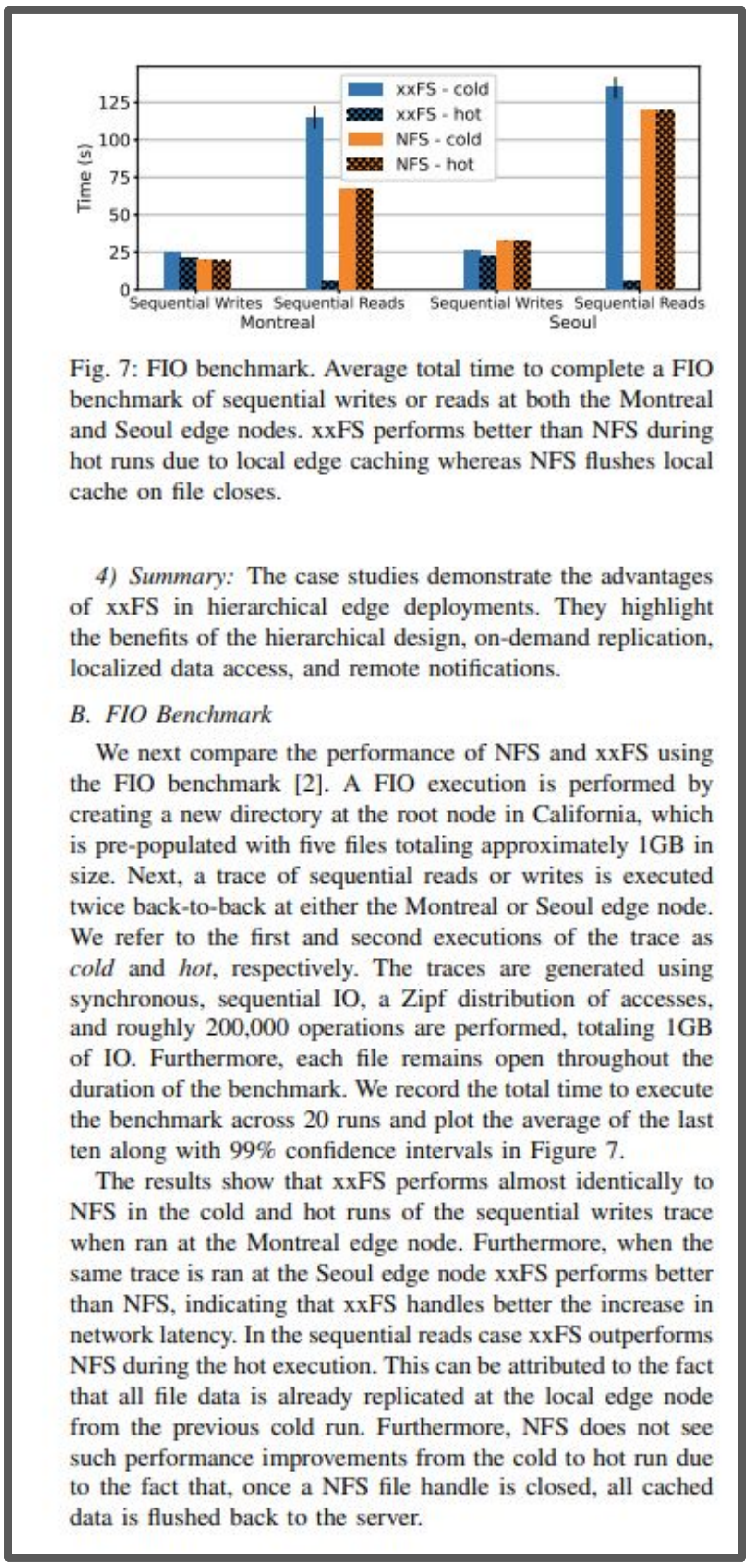

xxFS - cold
xxFS - hot
NFS - cold
NFS - hot
Time (s)
125
100
75
50
25
0
Sequential Writes
Sequential Reads
Montreal
Sequential Writes
Sequential Reads
Seoul

Fig. 7: FIO benchmark. Average total time to complete a FIO benchmark of sequential writes or reads at both the Montreal and Seoul edge nodes. xxFS performs better than NFS during hot runs due to local edge caching whereas NFS flushes local cache on file closes.

4) Summary: The case studies demonstrate the advantages of xxFS in hierarchical edge deployments. They highlight the benefits of the hierarchical design, on-demand replication, localized data access, and remote notifications.

B. FIO Benchmark

We next compare the performance of NFS and xxFS using the FIO benchmark [2]. A FIO execution is performed by creating a new directory at the root node in California, which is pre-populated with five files totaling approximately 1GB in size. Next, a trace of sequential reads or writes is executed twice back-to-back at either the Montreal or Seoul edge node. We refer to the first and second executions of the trace as *cold* and *hot*, respectively. The traces are generated using synchronous, sequential IO, a Zipf distribution of accesses, and roughly 200,000 operations are performed, totaling 1GB of IO. Furthermore, each file remains open throughout the duration of the benchmark. We record the total time to execute the benchmark across 20 runs and plot the average of the last ten along with 99% confidence intervals in Figure 7.

The results show that xxFS performs almost identically to NFS in the cold and hot runs of the sequential writes trace when ran at the Montreal edge node. Furthermore, when the same trace is ran at the Seoul edge node xxFS performs better than NFS, indicating that xxFS handles better the increase in network latency. In the sequential reads case xxFS outperforms NFS during the hot execution. This can be attributed to the fact that all file data is already replicated at the local edge node from the previous cold run. Furthermore, NFS does not see such performance improvements from the cold to hot run due to the fact that, once a NFS file handle is closed, all cached data is flushed back to the server.

About

xxFS is a **distributed file system** that optimizes for **edge** applications developed in the **University of Toronto Systems & Networks Group** . The system prototype is built in **Java** on top of **Cassandra** using a custom **FUSE** implementation

Contribution

As **Research Assistant** , conducted an experiment using a standard file system benchmark, generated a plot, and wrote the corresponding section in the paper *(final result shown on the left)*

In the process, discovered several bugs in the system, **3** of which were **mission critical** , deducing their root cause, and **fixing 2** of them in code.

e.g one of the bugs involved an obscure case of a file being shown in full from the `cat` command but not when indexed sequentially via `vim`. The cause was an indexing error when file seeking.

Technologies

- Used **Python Fabric library** to automate experiment execution allowing us to perform over **100** repetitions
- **AWS** used to create an edge-like setup with over **30** machines
- **FIO Benchmark Tool** used to perform the standard benchmark
- **Traffic Control** on **Ubuntu Linux** used to manipulate **network tables** and simulate latencies for various experiment setups
- **Docker** used to deploy xxFS across the machines
- Deployed **NFS** across the machines to compare against xxFS

Research Paper Link

Contact my supervising Professor Eyal de Lara at
delara@cs.toronto.edu if a reference letter is needed

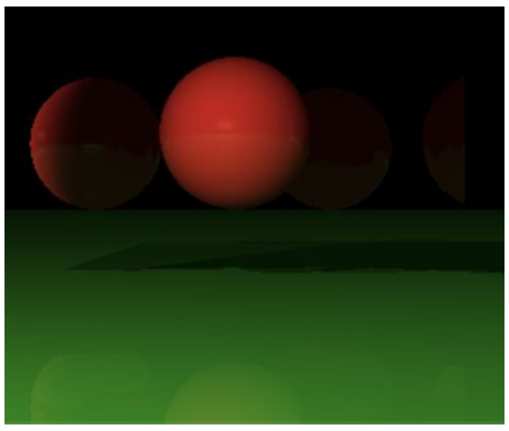
Computer Graphics Course Projects [1/2]

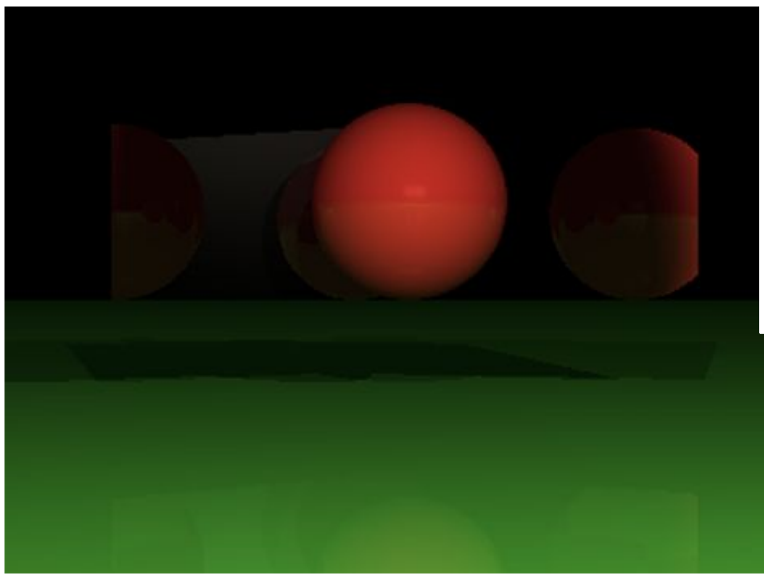
09/2023-12/2023

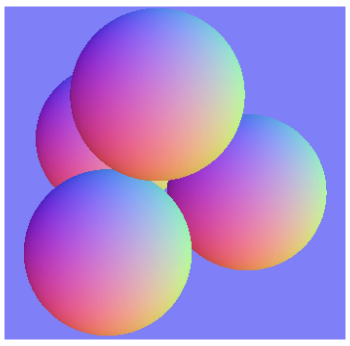
About

The computer graphics course involved 8 major independent C++ projects. Here are some highlights.

AlekseyPanas/ CG-Projects







Technologies

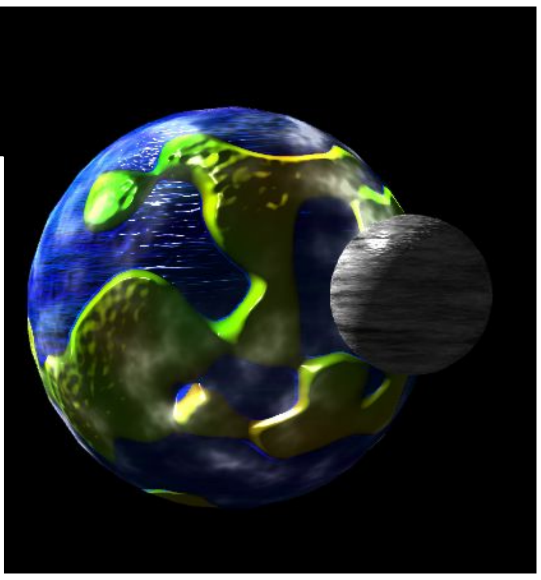
- **C++**
- **GLSL**
- Ray Tracing
- Physics-based animation
- Rigid Skinning
- Linear Algebra
- Calculus (Gradient descent)

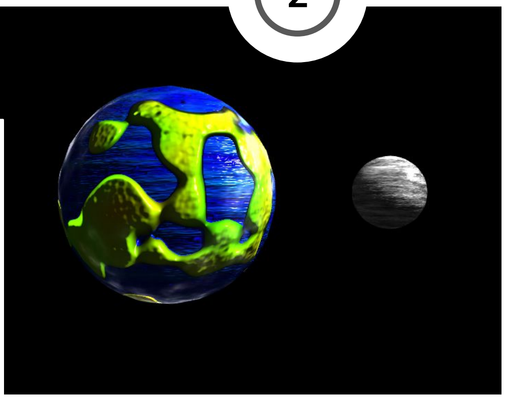
1

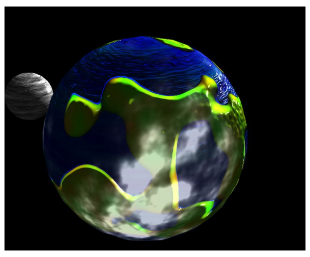
1: Ray Tracing

A ray tracer built in C++ capable of rendering scenes containing planes, spheres, and triangle meshes implementing shadows, reflections, and specular lighting

2







2: Shader Scene

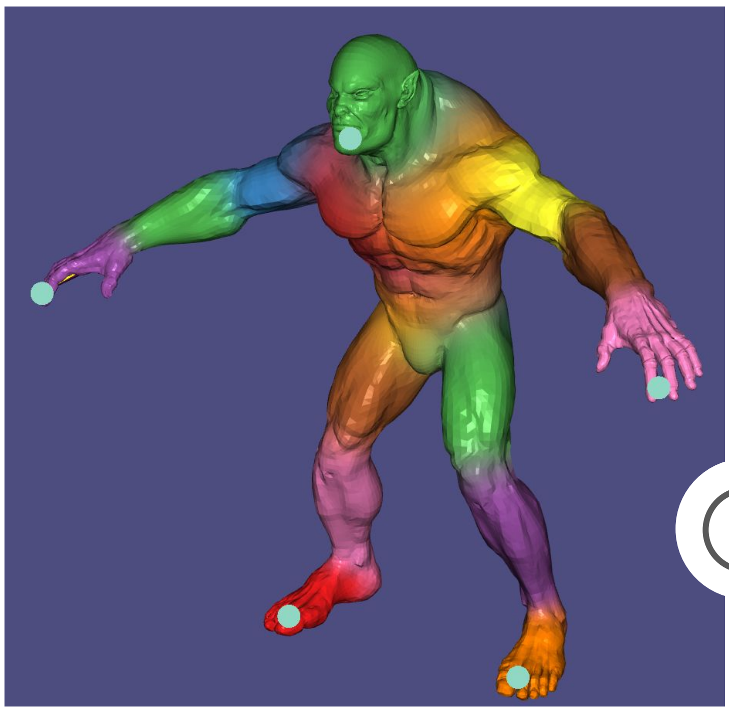
An animated planet scene generated entirely with shader code.

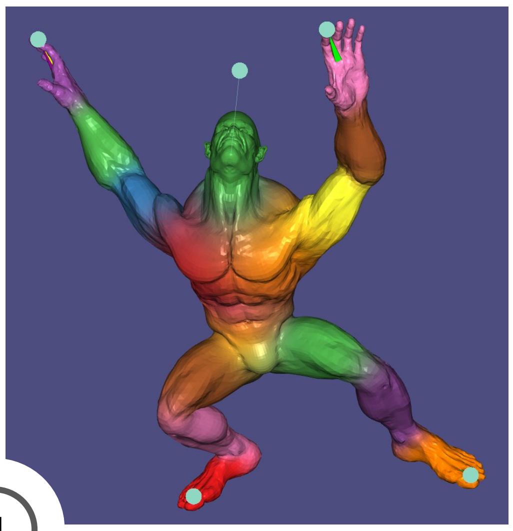
Involves vertex shaders, tessellation, and fragment shaders

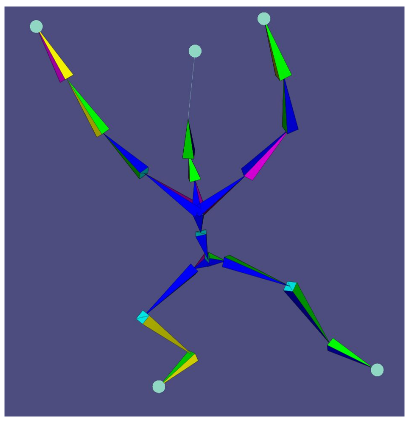
3: Surface Normals

Implementation of per-face normals, per-vertex normals, and per-corner normals for different shading effects

3



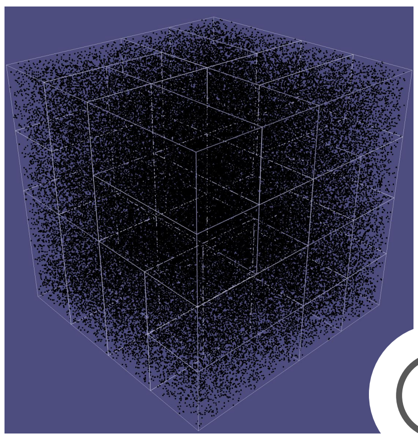


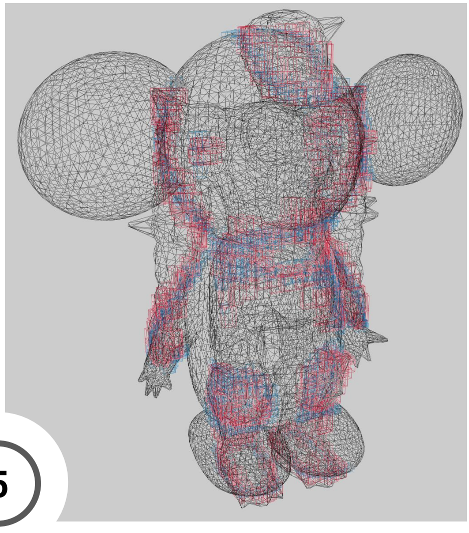


4: Rigid Skinning

C++ implementation of the rigid skinning method which involves bone matrix transformations and mesh deformation to the bones.

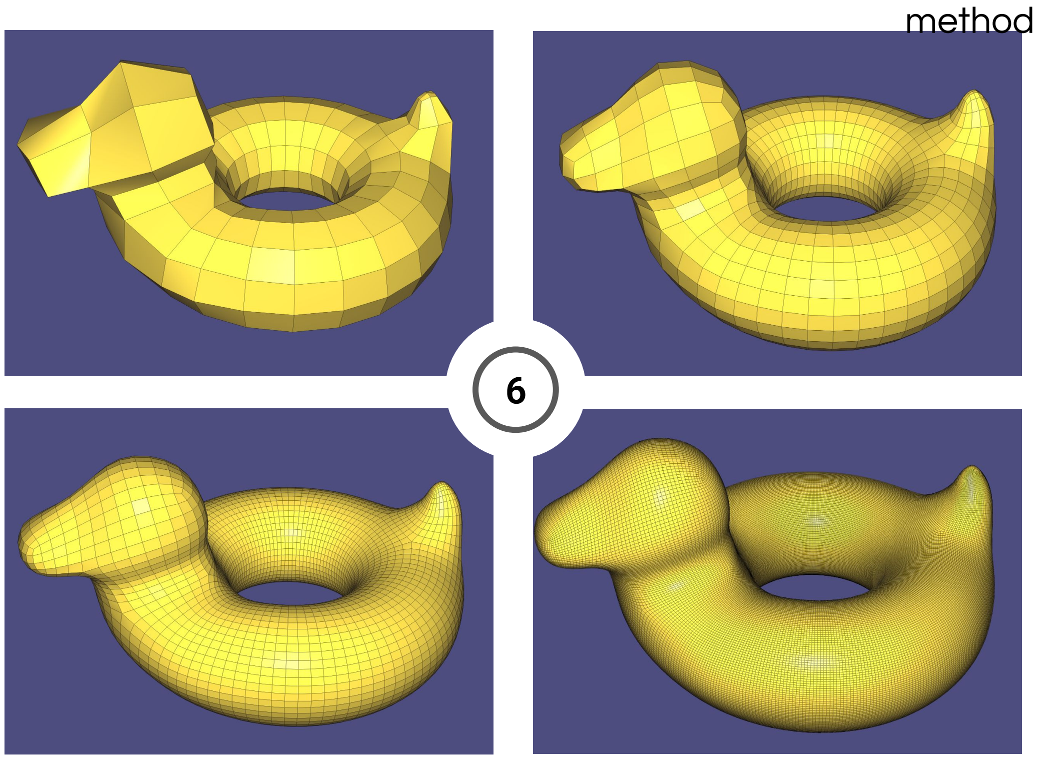
The images show inverse kinematics achieved via gradient descent to deform the mesh via target endpoint positions.

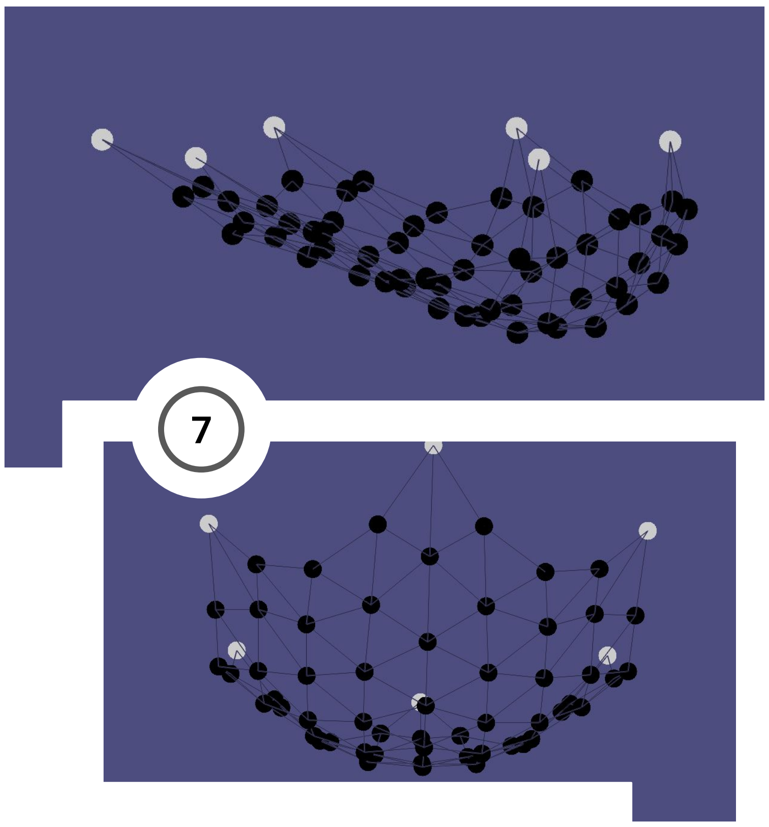




5: Bounding Volume Hierarchies

Hierarchical structure to organize scene for optimizing intersection and distance queries to **O(log(n))** time. Additionally implemented Devillier's triangle intersection method





Mass
springs

6: Catmull Clark Subdivision

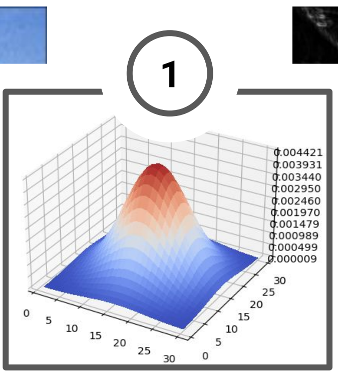
Implementation of the Catmull Clark subdivision scheme for increasing the number of polygons in a mesh

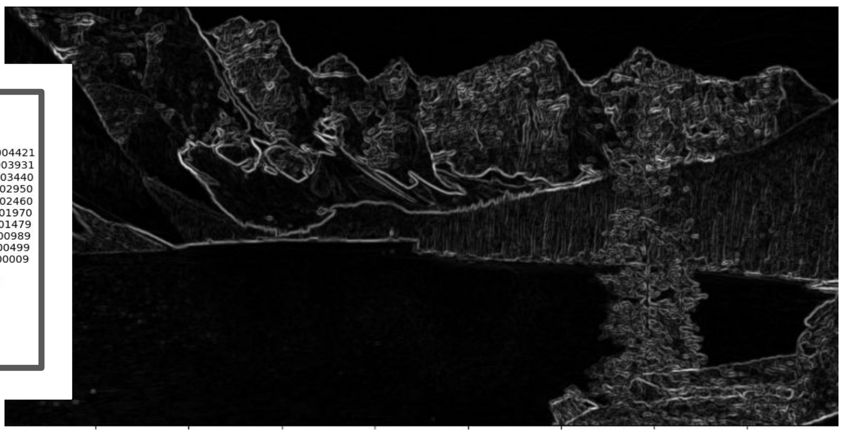
7: Physics-based Animation

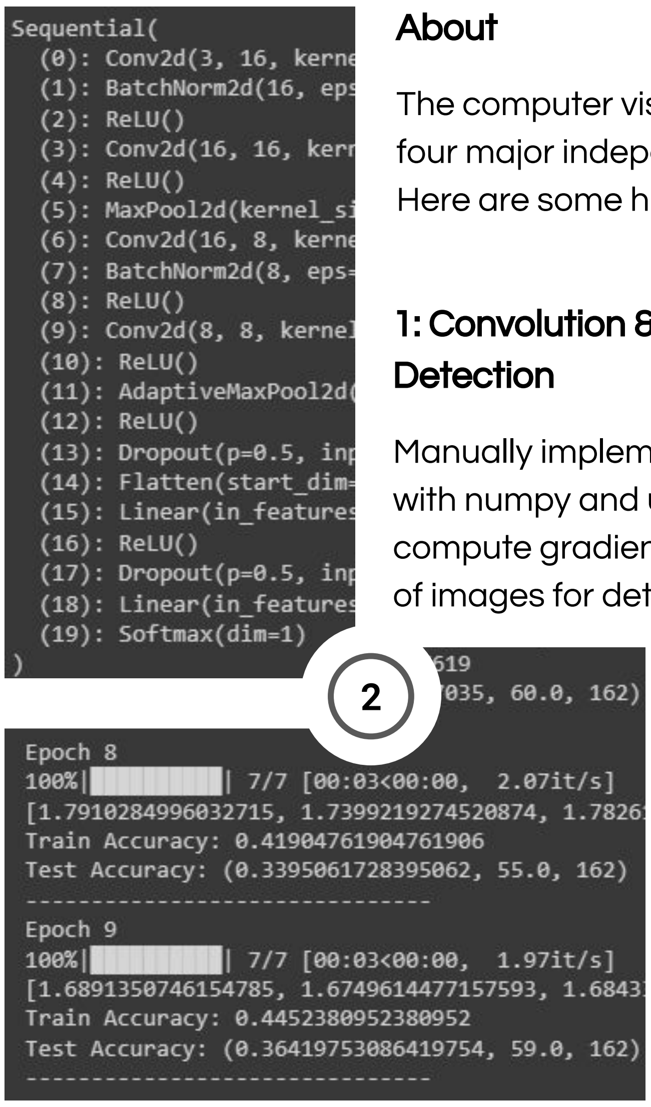
Implemented Fast Simulation of Mass-Spring Systems paper to simulate cloth-like objects

Computer Vision Course Projects

09/2023-12/2023







```
Sequential(
  (0): Conv2d(3, 16, kerne
  (1): BatchNorm2d(16, eps
  (2): ReLU()
  (3): Conv2d(16, 16, kern
  (4): ReLU()
  (5): MaxPool2d(kernel_si
  (6): Conv2d(16, 8, kerne
  (7): BatchNorm2d(8, eps=
  (8): ReLU()
  (9): Conv2d(8, 8, kernel
  (10): ReLU()
  (11): AdaptiveMaxPool2d(
  (12): ReLU()
  (13): Dropout(p=0.5, inp
  (14): Flatten(start_dim=
  (15): Linear(in_features
  (16): ReLU()
  (17): Dropout(p=0.5, inp
  (18): Linear(in_features
  (19): Softmax(dim=1)
)
```

```
619
035, 60.0, 162)

Epoch 8
100%|██████████| 7/7 [00:03<00:00,  2.07it/s]
[1.7910284996032715, 1.7399219274520874, 1.7826
Train Accuracy: 0.41904761904761906
Test Accuracy: (0.3395061728395062, 55.0, 162)
------------------------------
Epoch 9
100%|██████████| 7/7 [00:03<00:00,  1.97it/s]
[1.6891350746154785, 1.6749614477157593, 1.6843
Train Accuracy: 0.4452380952380952
Test Accuracy: (0.36419753086419754, 59.0, 162)
------------------------------
```

About

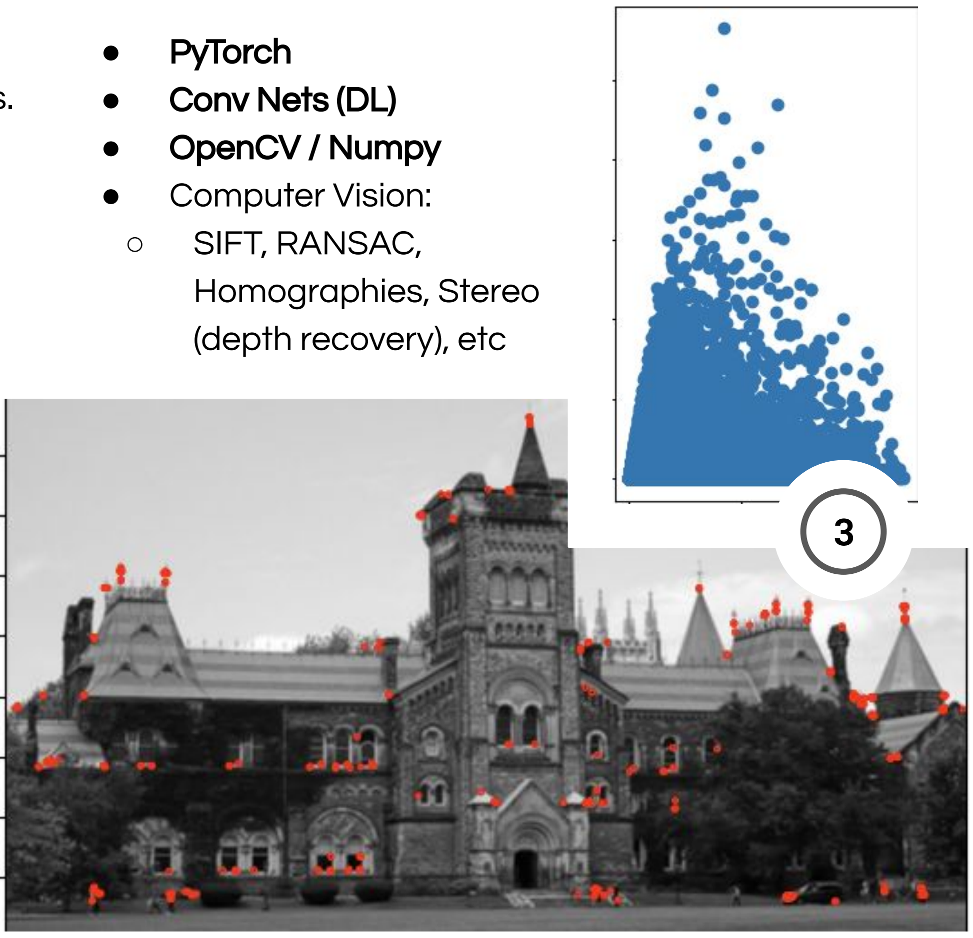
The computer vision course involved four major independent assignments. Here are some highlights.

1: Convolution & Edge Detection

Manually implemented convolution with numpy and used it to compute gradient magnitude of images for detecting edges

Technologies

- **PyTorch**
- **Conv Nets (DL)**
- **OpenCV / Numpy**
- Computer Vision:
  - SIFT, RANSAC, Homographies, Stereo (depth recovery), etc



3: Histogram of Gradients

Implemented an algorithm to compute the histogram of gradients for an image: a process commonly used in old-school object detection

2: Deep Learning

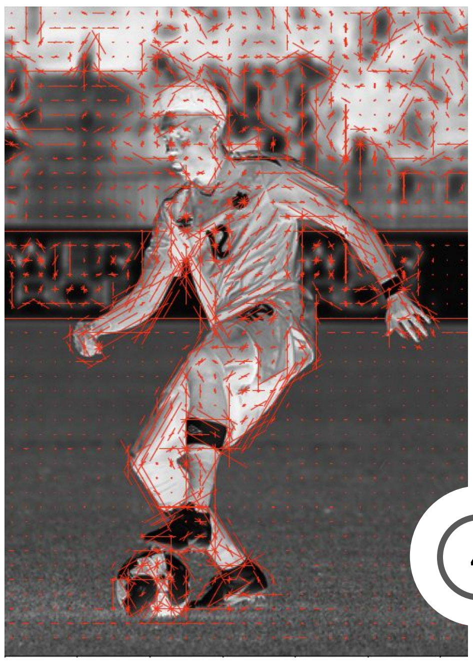
Designed and trained a conv net in Pytorch to distinguish dog breeds. Also applied transfer learning on ResNet

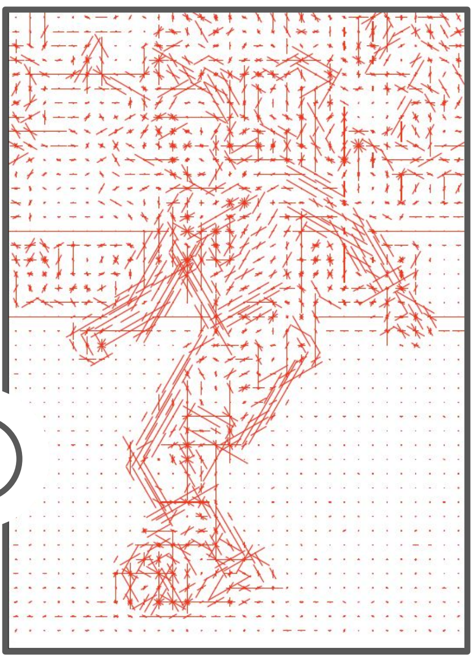
3: Corner Detection

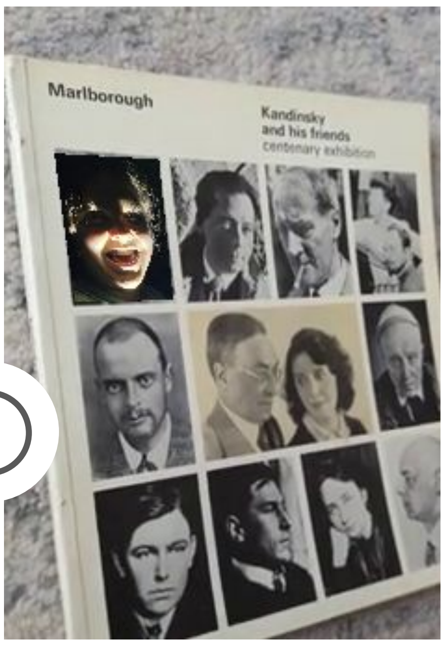
Detecting corners via the eigenvalues of the second moment matrix (scatterplot for threshold-setting)

4: Homographies, RANSAC, SIFT

Tasked with replacing an arbitrary planar object (selected via 4 points) throughout a video, applied SIFT and RANSAC to determine a homography transformation within each frame

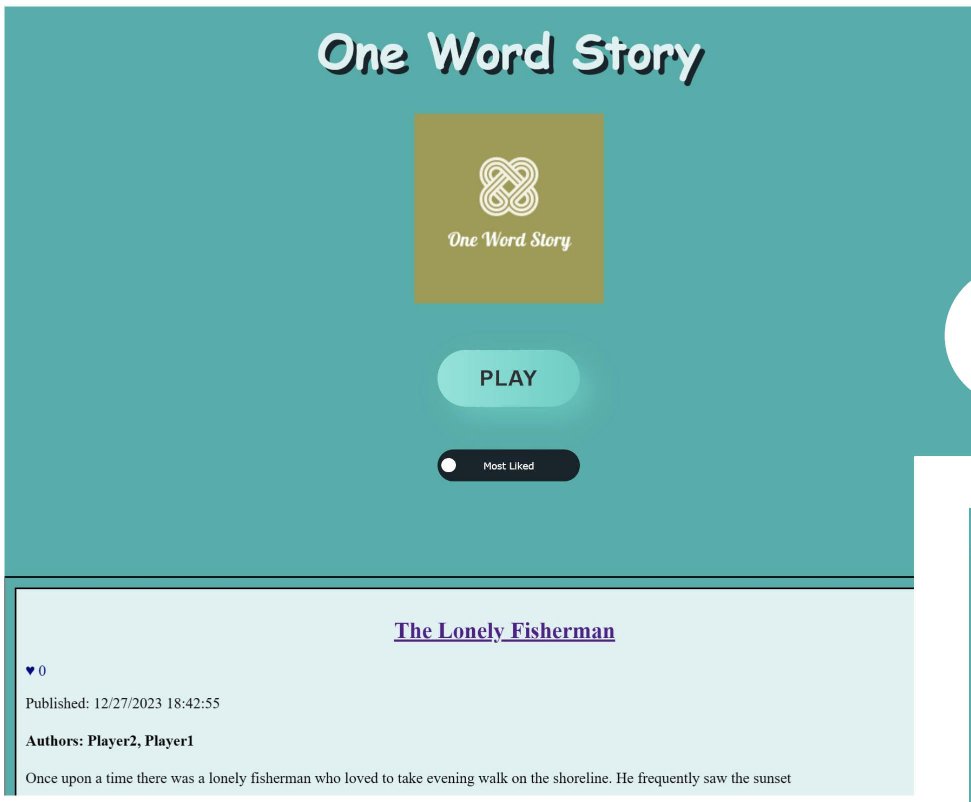






[1/2] One Word Story

www.onewordstory.net 09/2022 - 05/2023



The Lonely Fisherman

♥ 0

Published: 12/27/2023 18:42:55

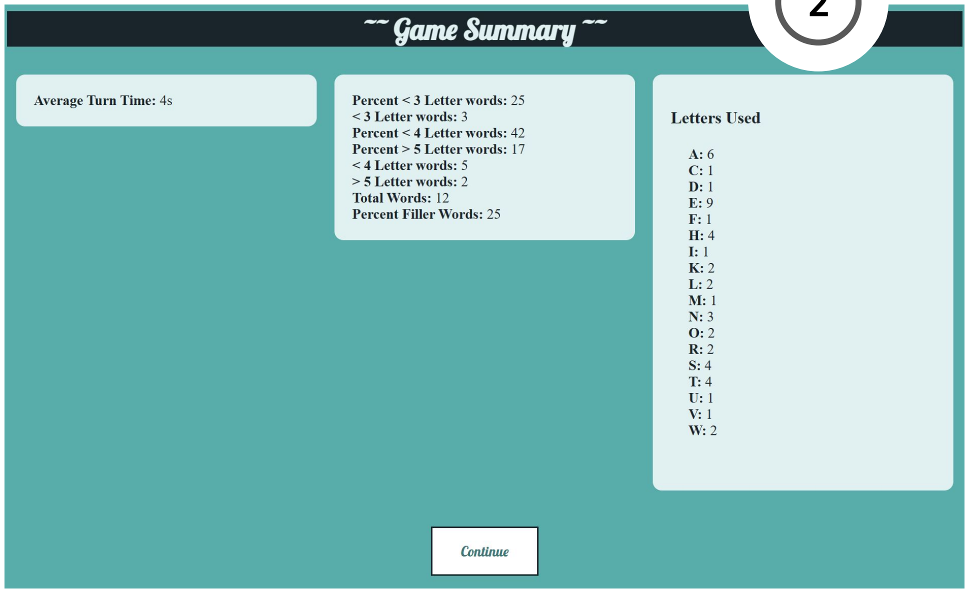
Authors: Player2, Player1

Once upon a time there was a lonely fisherman who loved to take evening walk on the shoreline. He frequently saw the sunset

Purpose

"One Word Story" is a web-based game inspired by the classic word game where players build a story one word at a time.

Developed by an eight-member team for a **Software Design** course, the platform allows players to join a match in an io-game fashion and automatically displays completed stories on its homepage after each game session.



Average Turn Time: 4s

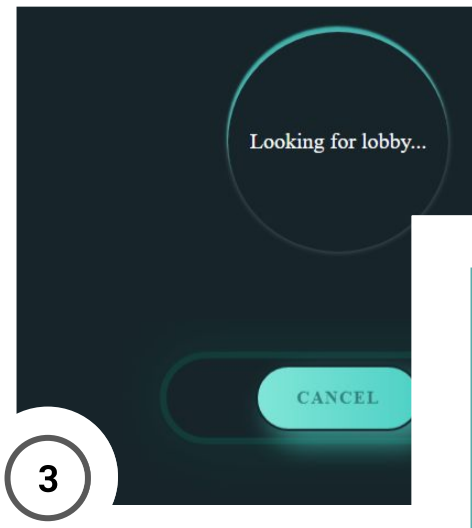
Percent < 3 Letter words: 25
< 3 Letter words: 3
Percent < 4 Letter words: 42
Percent > 5 Letter words: 17
< 4 Letter words: 5
> 5 Letter words: 2
Total Words: 12
Percent Filler Words: 25

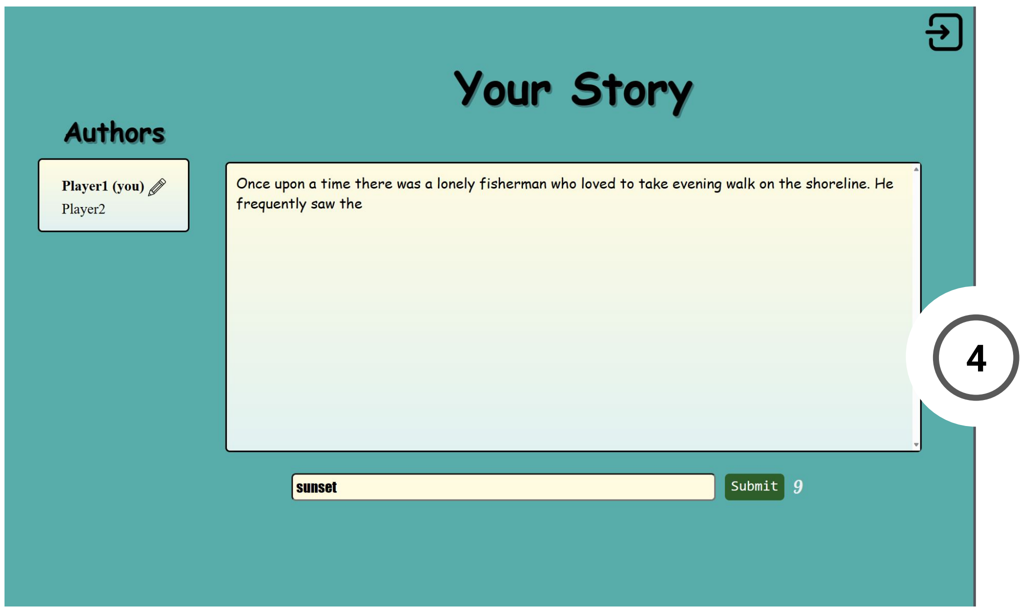
Letters Used

A:	6
C:	1
D:	1
E:	9
F:	1
H:	4
I:	1
K:	2
L:	2
M:	1
N:	3
O:	2
R:	2
S:	4
T:	4
U:	1
V:	1
W:	2

Technologies

- Java Spring Boot
- AWS
- PostgreSQL
- Websockets
- Multi-threading
- ReactJS
- HTML/CSS/JS
- JQuery
- Clean Architecture
- SOLID Principles
- Git / Github
- CI/CD
- Regex





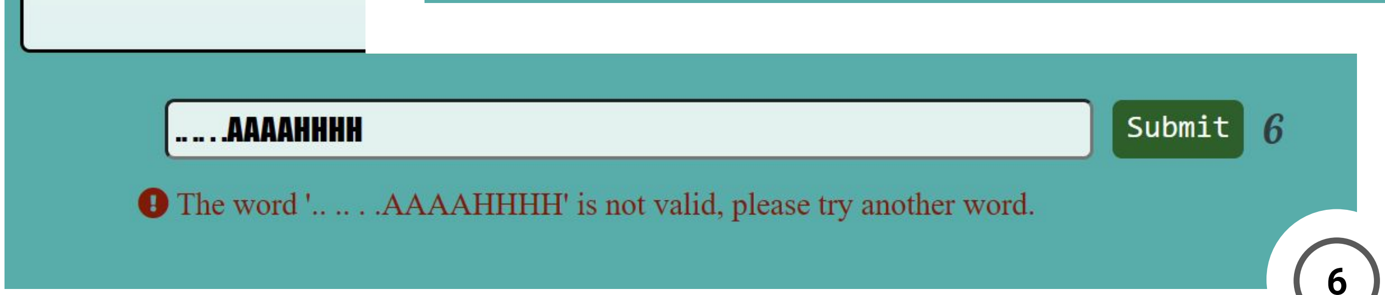
Player1 (you)
Player2

Once upon a time there was a lonely fisherman who loved to take evening walk on the shoreline. He frequently saw the

sunset 9

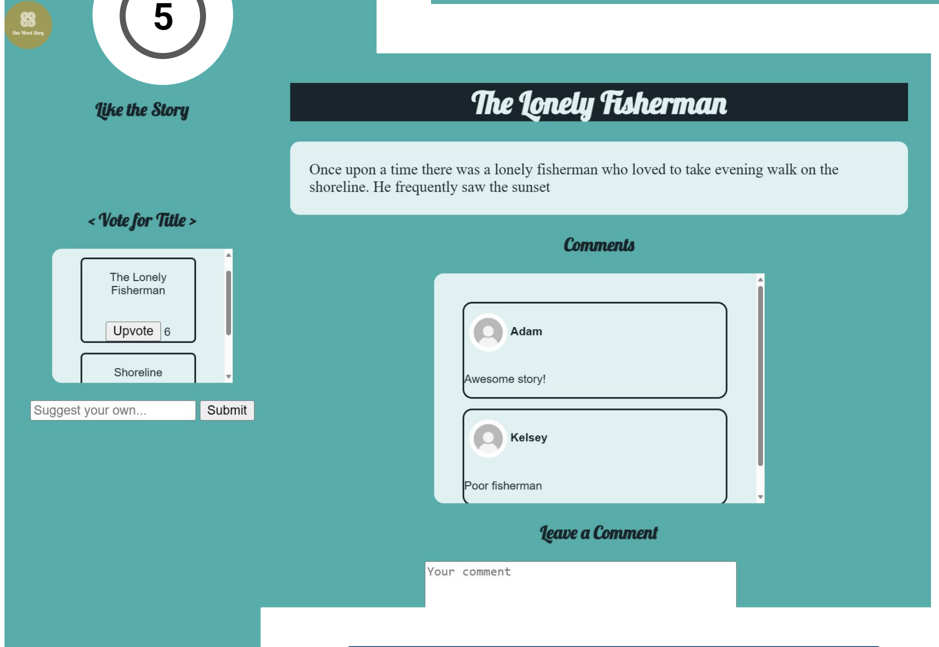
Contribution

Created the vast majority of the backend, including the **websocket** code for the core game, created the **frontend** for the **[4]**, **[6]** game page, and took care of domain setup and deployment via **AWS EC2, RDS,** and **Route53**



.. .. . .AAAAHHHH 6

The word '.. .. . .AAAAHHHH' is not valid, please try another word.



The Lonely Fisherman

Once upon a time there was a lonely fisherman who loved to take evening walk on the shoreline. He frequently saw the sunset

The Lonely Fisherman 6

Shoreline

Suggest your own...

Adam
Awesome story!

Kelsey
Poor fisherman

Your comment

Team-Tree-Dog/One-Word-Story

Captions

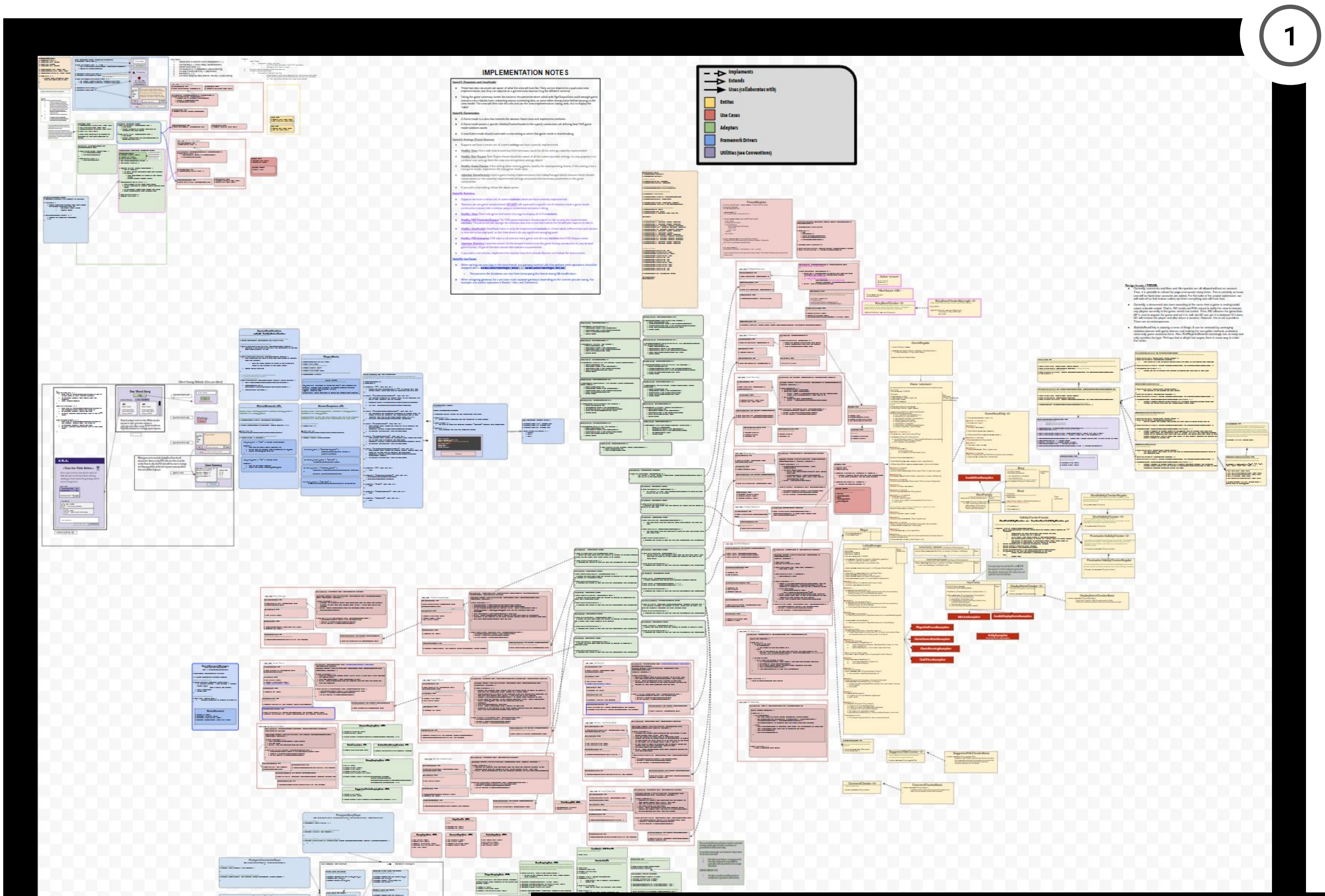
1. Homepage with completed stories on showcase
2. Post-game summary shows fun statistics for each player from the finished game
3. Loading screen
4. In-game UI
5. Page for a specific story, allowing comments and title voting
6. Regex-based input parsing

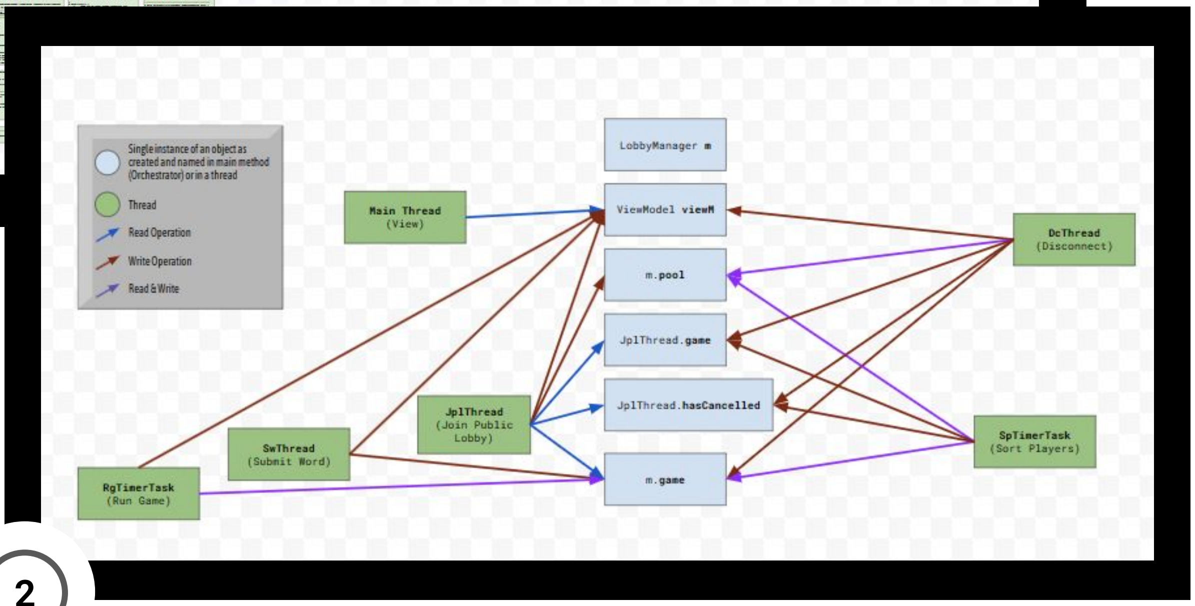
What's Ahead

Ongoing work to port the game to **ReactJS**.
Implementing an account system for private lobbies.
Adding new game modes for variety

[2/2]

1





2

Captions

1. **Design diagram** used by team software architects to design new features in high-level pseudocode and organize classes into **clean architecture layers**

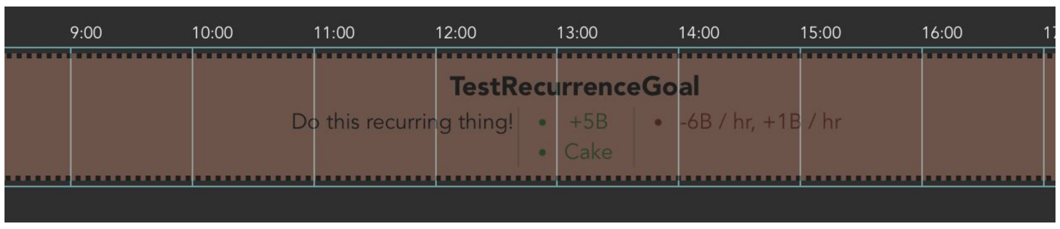
2. **Thread** chart mapping Use Cases threads to core objects they access for reads and writes. This was used extensively to resolve or prevent **deadlocks**

Goal-Setting App

05/2024 - Present

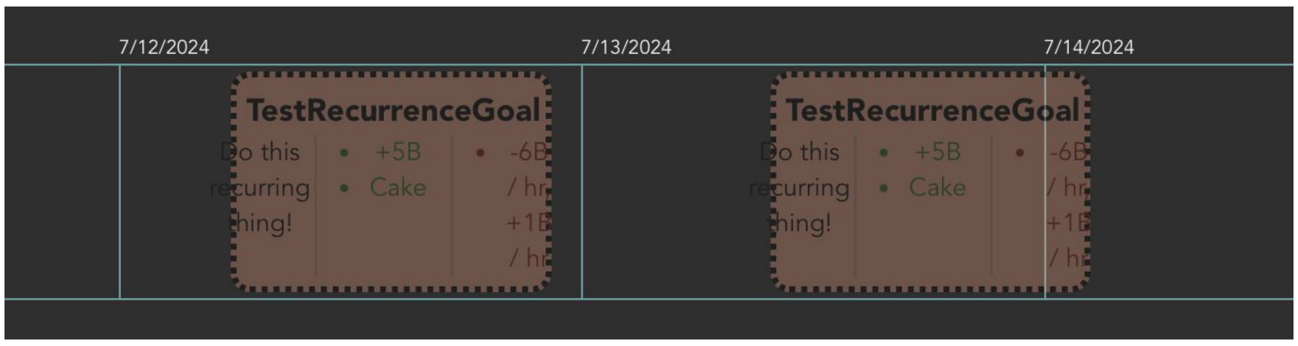
About

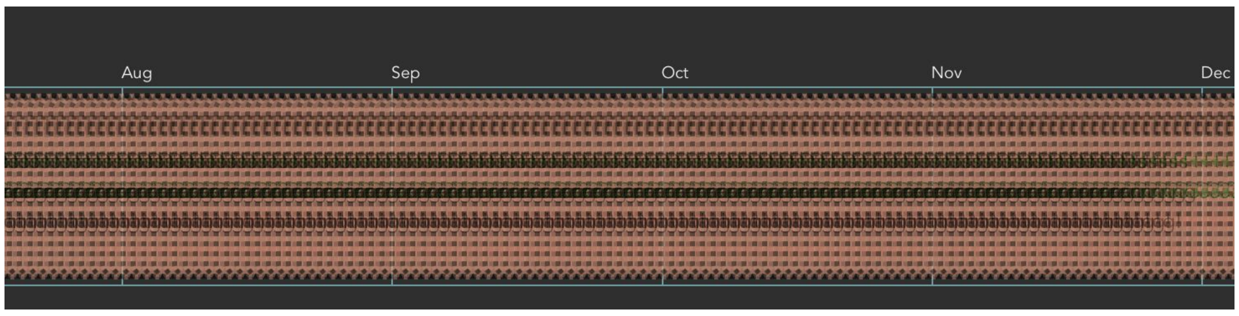
No matter how many goal-setting apps I tried over the years, nothing seemed to give me the exact features I needed, so finally I decided I'd write my own solution.



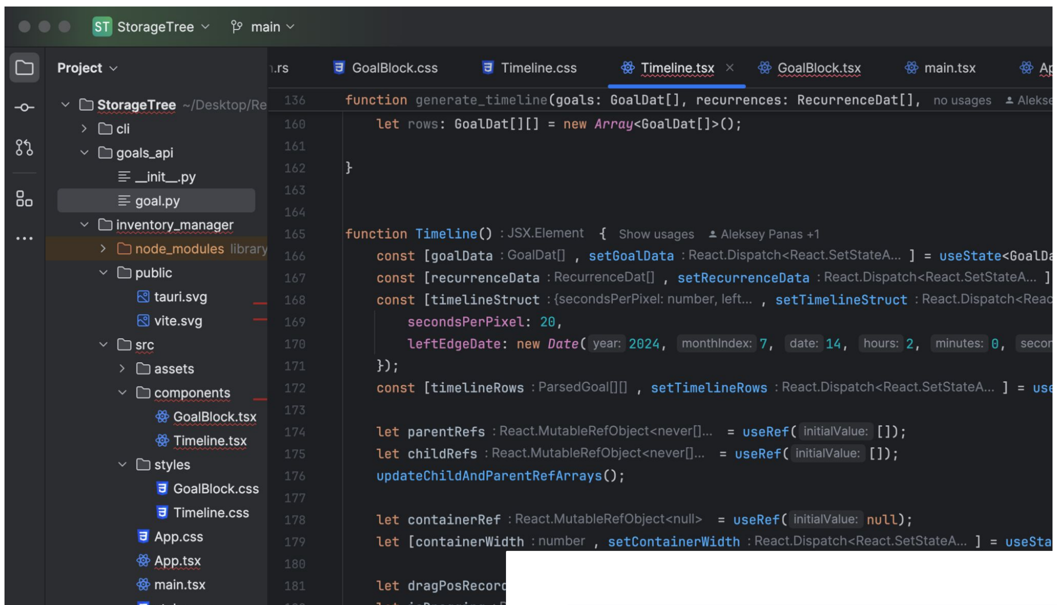
Technologies

- Tauri
- ReactJS
- Vite
- TS/TSX/CSS
- Rust
- Postgres





[AleksеyPanas/StorageTree]

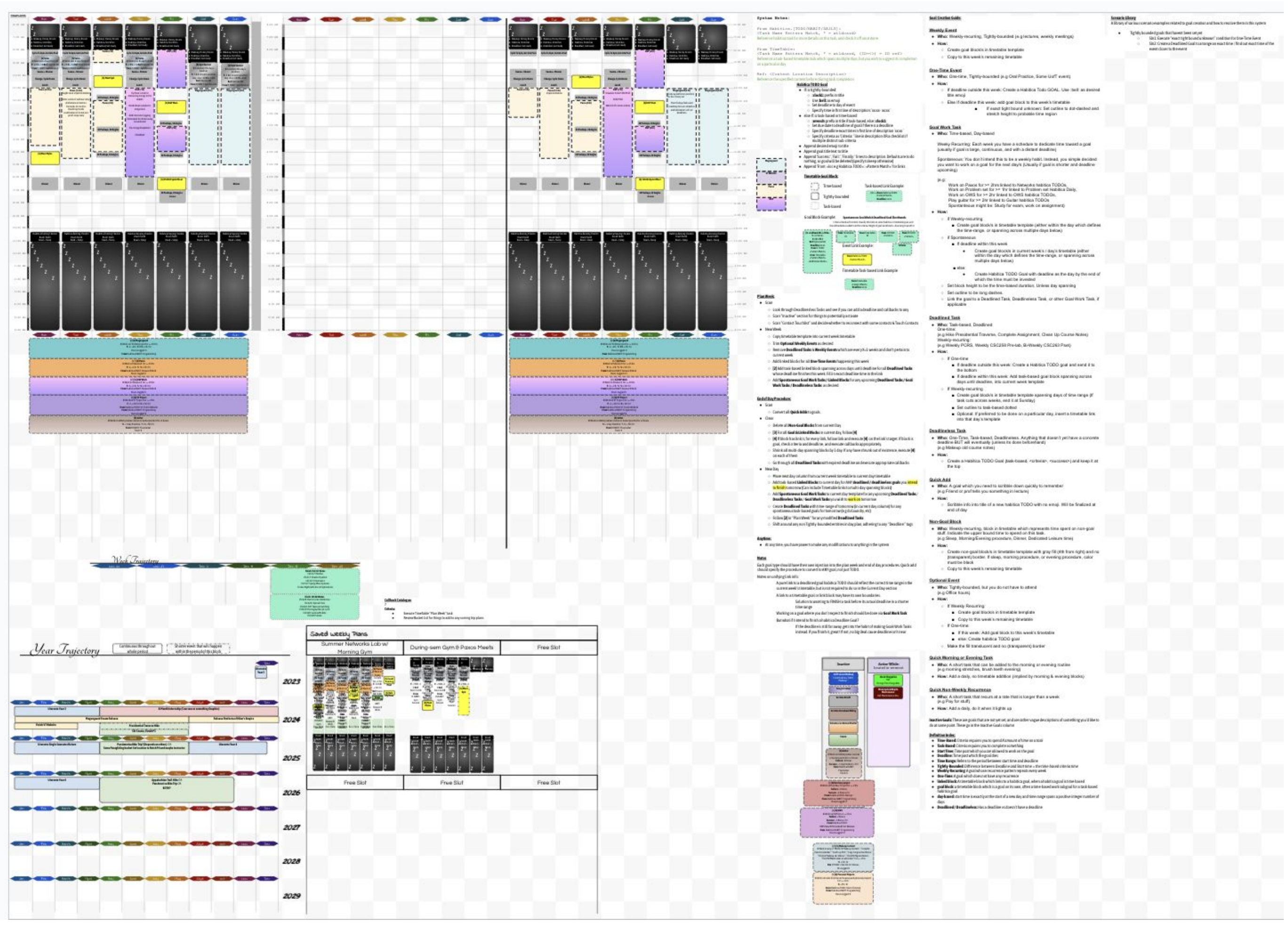


```
function generate_timeline(goals: GoalDat[], recurrences: RecurrenceDat[],   no usages   Alekse
    let rows: GoalDat[][] = new Array<GoalDat[]>();

}


function Timeline() : JSX.Element  {   Show usages   Aleksey Panas +1
    const [goalData : GoalDat[] , setGoalData : React.Dispatch<React.SetStateA... ] = useState<GoalDa
    const [recurrenceData : RecurrenceDat[] , setRecurrenceData : React.Dispatch<React.SetStateA... ]
    const [timelineStruct : {secondsPerPixel: number, left... , setTimelineStruct : React.Dispatch<Reac
        secondsPerPixel: 20,
        leftEdgeDate: new Date( year: 2024,  monthIndex: 7,  date: 14,  hours: 2,  minutes: 0,  secor
    });
    const [timelineRows : ParsedGoal[][] , setTimelineRows : React.Dispatch<React.SetStateA... ] = use

    let parentRefs : React.MutableRefObject<never[]...  = useRef( initialValue: []);
    let childRefs : React.MutableRefObject<never[]...  = useRef( initialValue: []);
    updateChildAndParentRefArrays();

    let containerRef : React.MutableRefObject<null>  = useRef( initialValue: null);
    let [containerWidth : number , setContainerWidth : React.Dispatch<React.SetStateA... ] = useSta

    let dragPosRecord
    let isDragging : R

    function updateCh
```

Contribution

As part of this solo project developed a full-stack web & desktop application using **Tauri**:

- Wrote an extensive **API backend** in **Rust** allowing the creation, modification, deletion, and resolution of goals
- Designed an in-house zoomable timeline view in the frontend using **ReactJS**
- Deployed a **PostreSQL** database using **AWS RDS** connecting it to the backend in **Rust**

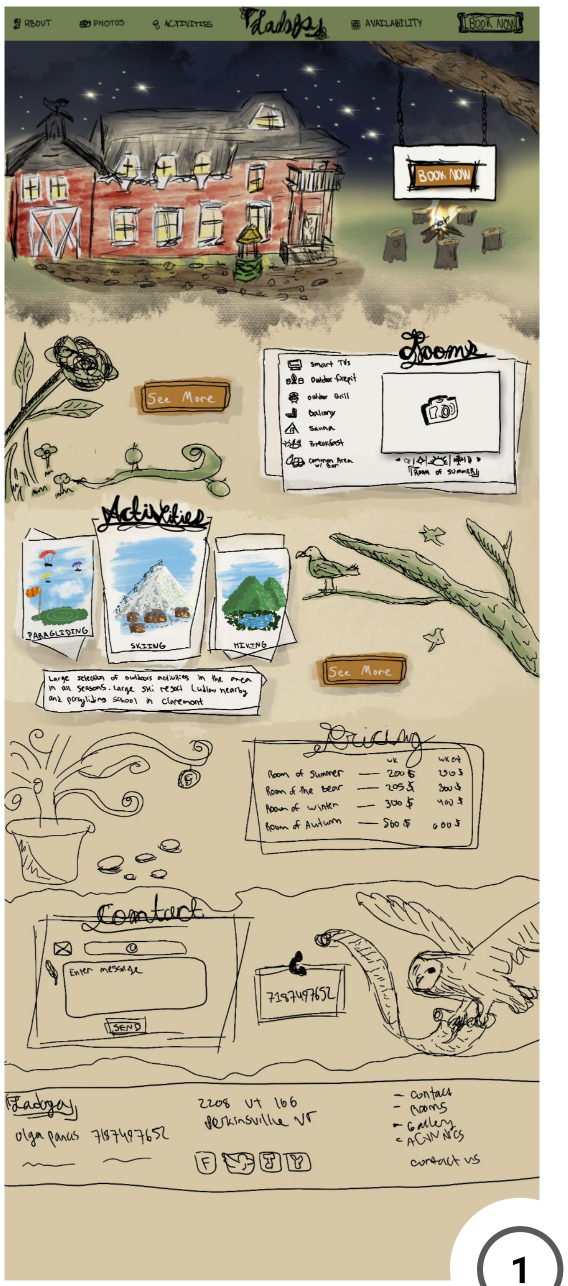


This document is my current goal-setting solution. It involves manually performing many automatable-tasks such as executing callbacks, recursively checking goal dependencies, and more.

After years of adding to this doc, it now serves as "pseudocode" for the app

New England BnB Hotel System

06/2023 - Present

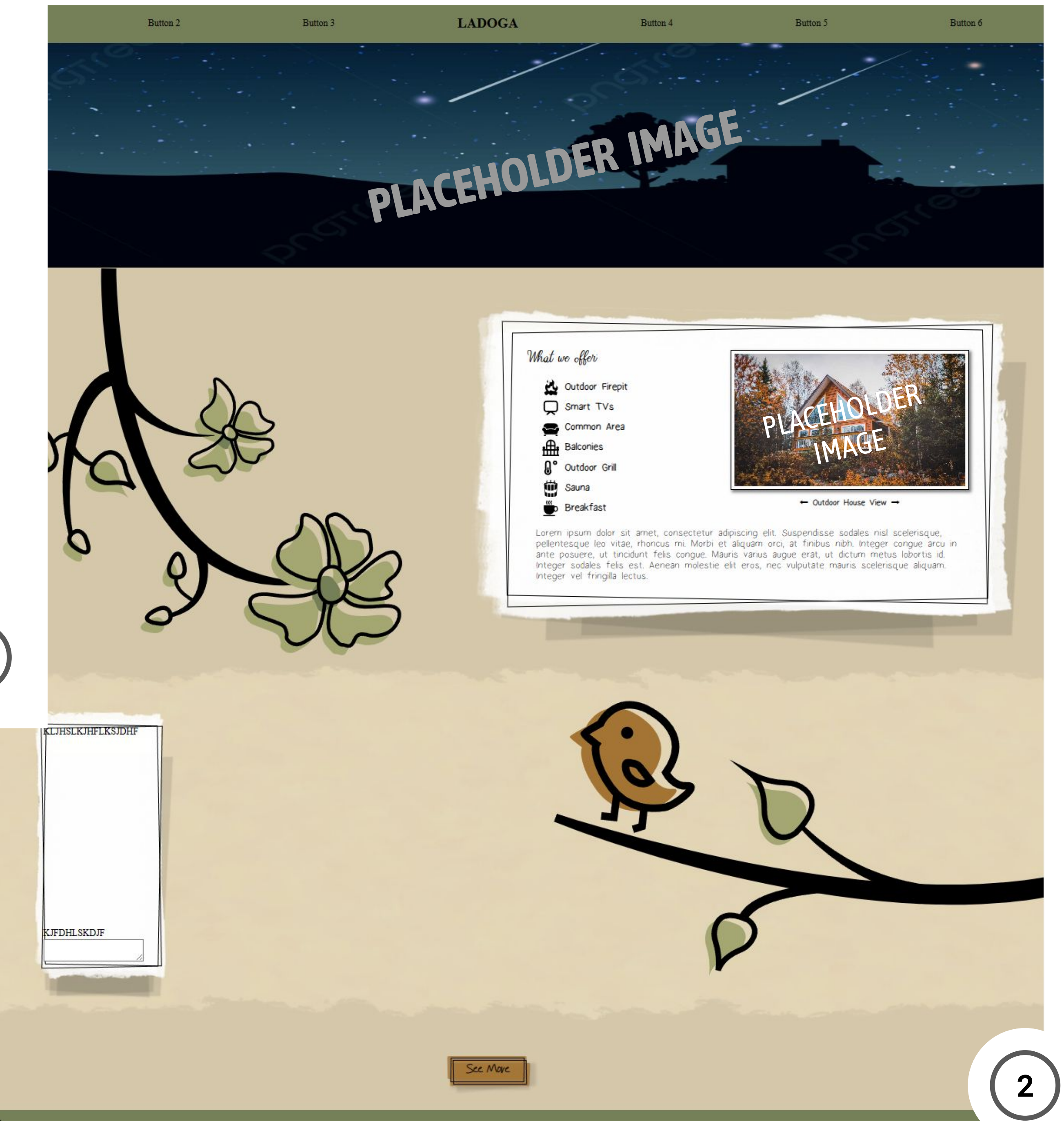


1

Purpose

This is a two-person **ongoing client project** for a **bed & breakfast business** in New England, USA. The project's first priority is a customer-facing visually-appealing frontend to view information about the B&B, book rooms, see local activities, or contact the owners.

An additional **React-based admin dashboard** will allow management of bookings, cancellations, payments, and customer correspondences. To move off of AirBnB, the platform also includes **Stripe** payments



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse sodales nisl scelerisque, pellentesque leo vitae, rhoncus mi. Morbi et aliquam orci, at finibus nibh. Integer congue arcu in ante posuere, ut tincidunt felis congue. Mauris varius augue erat, ut dictum metus lobortis id. Integer sodales felis est. Aenean molestie elit eros, nec vulputate mauris scelerisque aliquam. Integer vel fringilla lectus.

2

Captions

1. Design mockup made in Procreate
2. Custom component showcase

Technologies

- **Java Spring Boot** backend
- **Gradle** with customized **Kotlin** script to compile the multi-project build
- **HTML / Typescript / SASS** for customer-facing website
- **ReactJS** for admin dashboard
- **Stripe** for payment processing
- **AWS EC2, RDS, Route53** for hosting
- **Git/Github** for version control and project management
- **ViteJS** for frontend tooling
- **PostgreSQL** as database choice
- **Procreate** and **Photoshop** for mockups and frontend design
- **SVG** vector art for animating frontend visuals

Contribution

Configured the **Gradle** **Kotlin** script to build multiple **React** and non-React frontend projects with a **Spring Boot** backend stack, as well as to execute **unit tests** and automatically pull production changes on the server

Designed the website frontend mockup and working towards implementing it in **HTML/SASS/TS** and **React** components. Additionally working on the backend for the customer facing website, admin dashboard, and **Stripe** integration.

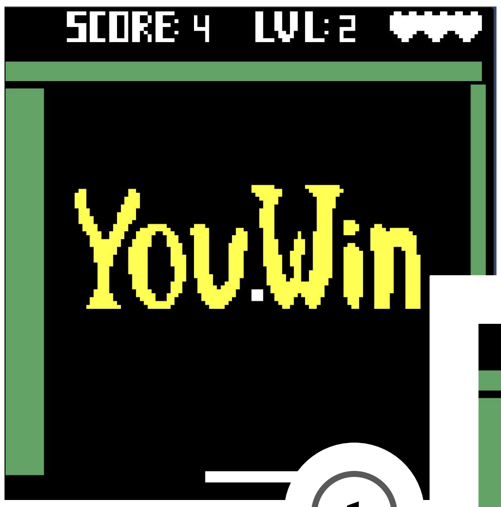
Managing project tasks using Agile with a partner, delegating SVG work and 30% of the frontend

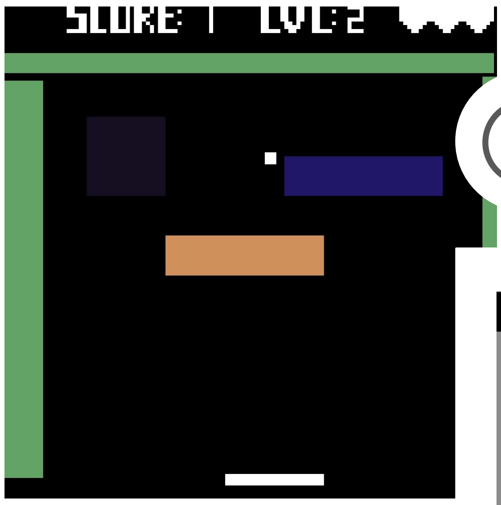
MIPS Assembly Breakout

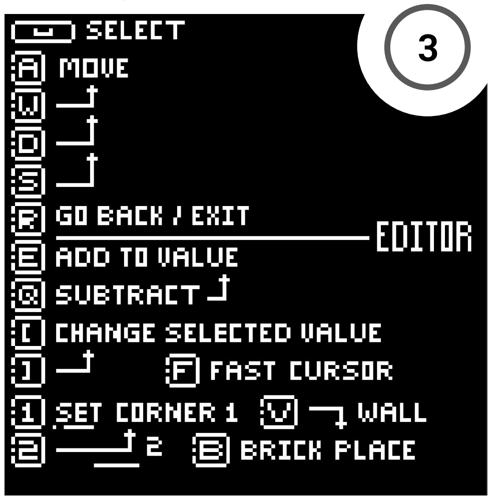
01/2023 - 05/2023

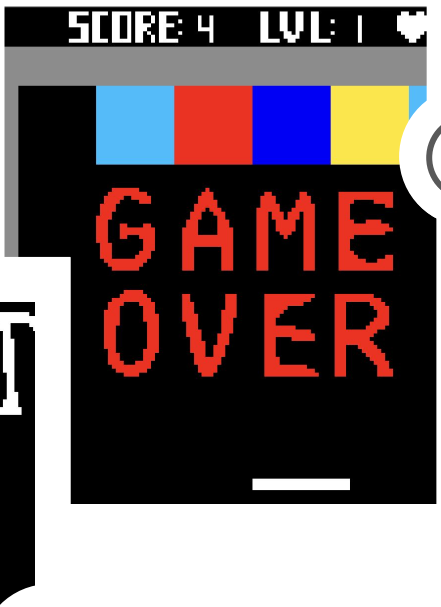
Purpose

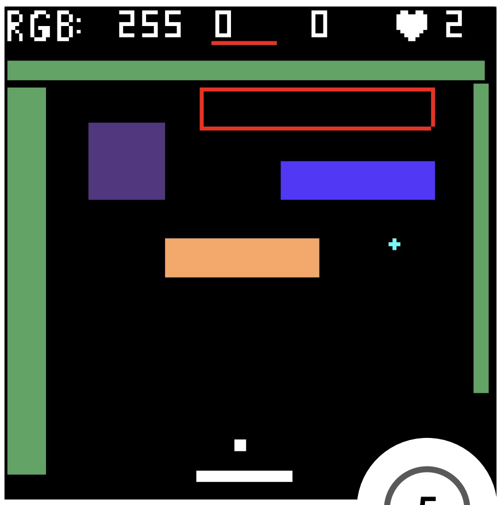
Solo project for a **Computer Organization Course** written entirely in **MIPS Assembly**. This submission goes **beyond course expectations** implementing an entire custom level editor, a menu, saving to disk, and detailed bitmaps designed with the help of a supplementary **Python** script

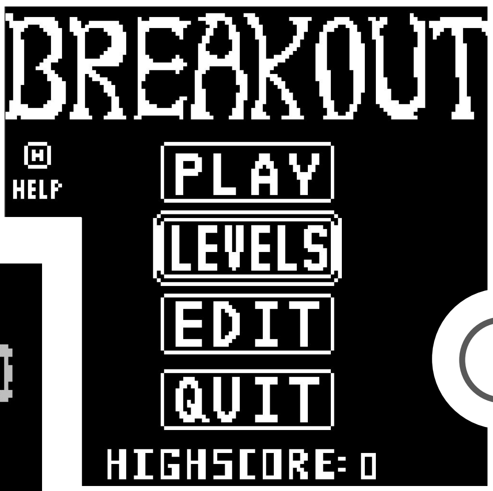


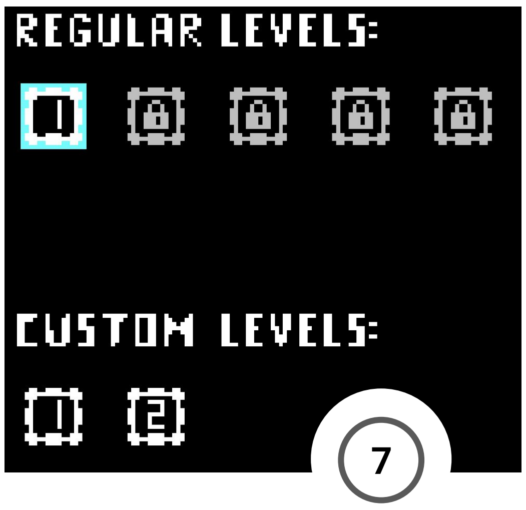












Technologies

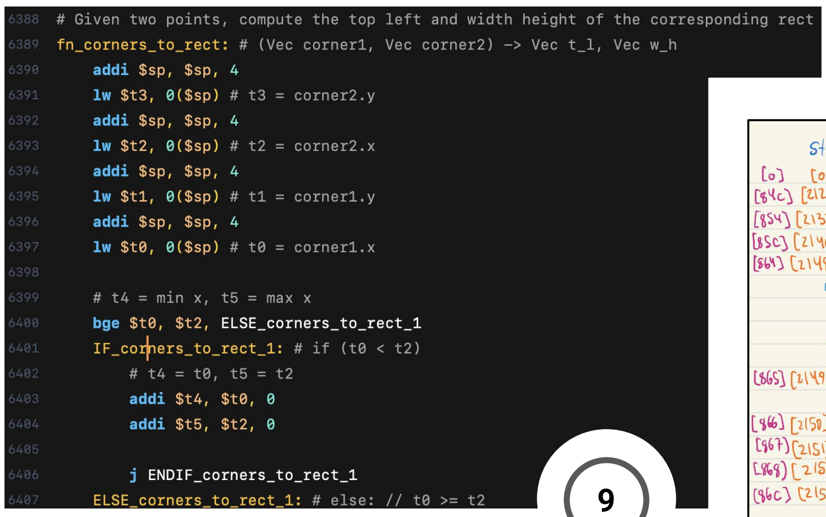
- MIPS Assembly
- Python / Pygame
- Bitmaps

AlekseyPanas/assembly-breakout

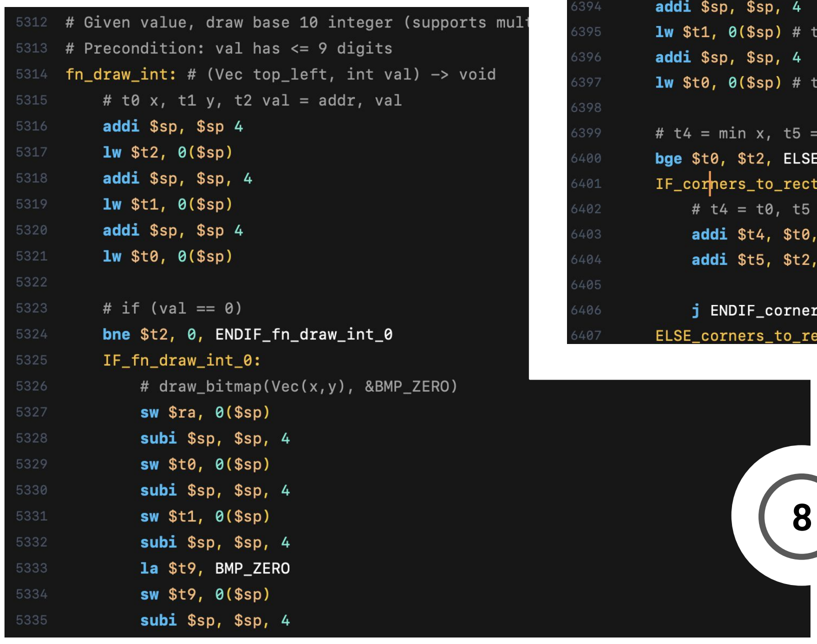
Game Demo

Captions

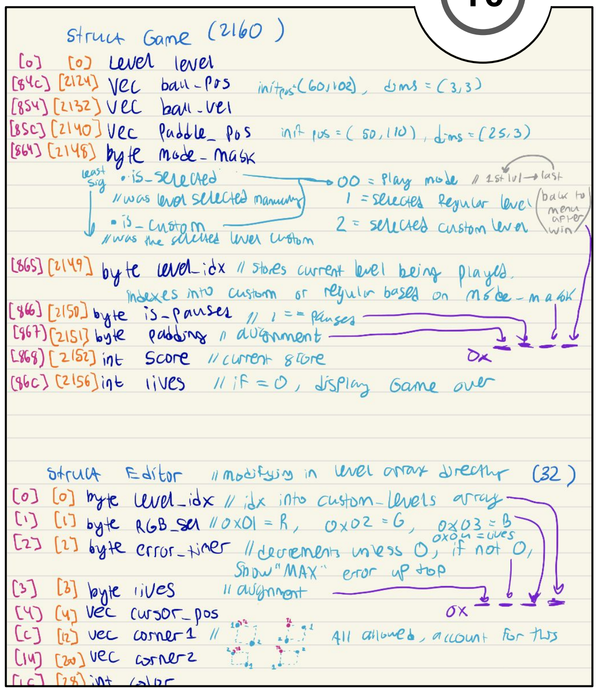
1. Win Screen
2. Gameplay
3. Help Screen
4. Game Over Screen
5. Level Editor
6. Main Menu
7. Level Selection Screen
8. Function to draw an arbitrary multi-digit base 10 number on screen (e.g for score or lives display)
9. Function to convert two points to a rectangle
10. Struct planning for game objects



```
# Given two points, compute the top left and width height of the corresponding rect
fn_corners_to_rect: # (Vec corner1, Vec corner2) -> Vec t_l, Vec w_h
    addi $sp, $sp, 4
    lw $t3, 0($sp) # t3 = corner2.y
    addi $sp, $sp, 4
    lw $t2, 0($sp) # t2 = corner2.x
    addi $sp, $sp, 4
    lw $t1, 0($sp) # t1 = corner1.y
    addi $sp, $sp, 4
    lw $t0, 0($sp) # t0 = corner1.x

    # t4 = min x, t5 = max x
    bge $t0, $t2, ELSE_corners_to_rect_1
    IF_corners_to_rect_1: # if (t0 < t2)
        # t4 = t0, t5 = t2
        addi $t4, $t0, 0
        addi $t5, $t2, 0

        j ENDIF_corners_to_rect_1
    ELSE_corners_to_rect_1: # else: // t0 >= t2
```



```
# Given value, draw base 10 integer (supports mult
# Precondition: val has <= 9 digits
fn_draw_int: # (Vec top_left, int val) -> void
    # t0 x, t1 y, t2 val = addr, val
    addi $sp, $sp 4
    lw $t2, 0($sp)
    addi $sp, $sp, 4
    lw $t1, 0($sp)
    addi $sp, $sp 4
    lw $t0, 0($sp)

    # if (val == 0)
    bne $t2, 0, ENDIF_fn_draw_int_0
    IF_fn_draw_int_0:
        # draw_bitmap(Vec(x,y), &BMP_ZERO)
        sw $ra, 0($sp)
        subi $sp, $sp, 4
        sw $t0, 0($sp)
        subi $sp, $sp, 4
        sw $t1, 0($sp)
        subi $sp, $sp, 4
        la $t9, BMP_ZERO
        sw $t9, 0($sp)
        subi $sp, $sp, 4
```



Struct Game (2160)

[0] [0] Level level
[84c] [2124] Vec ball_pos init pos: (60,102), dims = (3,3)
[854] [2132] Vec ball_vel
[85c] [2140] Vec Paddle_Pos init pos = (50,110), dims = (25,3)
[864] [2148] byte mode_mask
least sig
• is_selected → 00 = Play mode // 1st lvl → last
// was level selected manually 1 = selected Regular level (back to menu after win)
• is_custom 2 = selected custom level
// was the selected level custom
[865] [2149] byte level_idx // stores current level being played. indexes into custom or regular based on mode_mask
[866] [2150] byte is_paused // 1 == paused
[867] [2151] byte padding // alignment
0x
[868] [2152] int score // current score
[86c] [2156] int lives // if = 0, display Game over

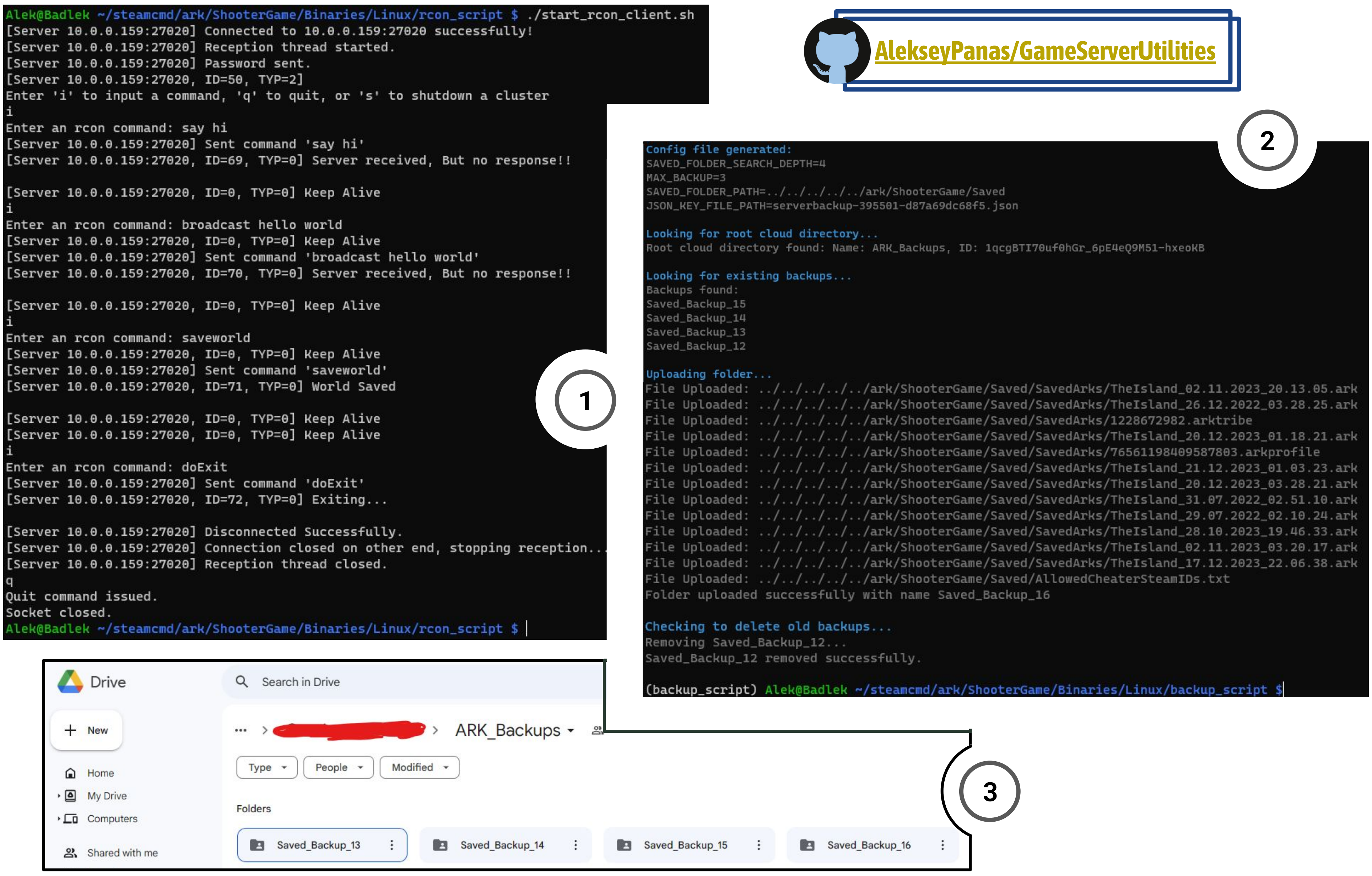
struct Editor // modifying in level array directly (32)

[0] [0] byte level_idx // idx into custom_levels array
[1] [1] byte RGB_sel // 0x01 = R, 0x02 = G, 0x03 = B 0x04 = lives
[2] [2] byte error_timer // decrements unless 0, if not 0, Show "MAX" error up top
[3] [3] byte lives // alignment
0x
[4] [4] vec cursor_pos
[c] [12] vec corner1 // All allowed, account for this
[14] [20] vec corner2
[1c] [28] int color

Game Server Utilities (ARK)

10/2023 - 11/2023

AlekseyPanas/GameServerUtilities



```
Alek@Badlek ~/steamcmd/ark/ShooterGame/Binaries/Linux/rcon_script $ ./start_rcon_client.sh
[Server 10.0.0.159:27020] Connected to 10.0.0.159:27020 successfully!
[Server 10.0.0.159:27020] Reception thread started.
[Server 10.0.0.159:27020] Password sent.
[Server 10.0.0.159:27020, ID=50, TYP=2]
Enter 'i' to input a command, 'q' to quit, or 's' to shutdown a cluster
i
Enter an rcon command: say hi
[Server 10.0.0.159:27020] Sent command 'say hi'
[Server 10.0.0.159:27020, ID=69, TYP=0] Server received, But no response!!

[Server 10.0.0.159:27020, ID=0, TYP=0] Keep Alive
i
Enter an rcon command: broadcast hello world
[Server 10.0.0.159:27020, ID=0, TYP=0] Keep Alive
[Server 10.0.0.159:27020] Sent command 'broadcast hello world'
[Server 10.0.0.159:27020, ID=70, TYP=0] Server received, But no response!!

[Server 10.0.0.159:27020, ID=0, TYP=0] Keep Alive
i
Enter an rcon command: saveworld
[Server 10.0.0.159:27020, ID=0, TYP=0] Keep Alive
[Server 10.0.0.159:27020] Sent command 'saveworld'
[Server 10.0.0.159:27020, ID=71, TYP=0] World Saved

[Server 10.0.0.159:27020, ID=0, TYP=0] Keep Alive
[Server 10.0.0.159:27020, ID=0, TYP=0] Keep Alive
i
Enter an rcon command: doExit
[Server 10.0.0.159:27020] Sent command 'doExit'
[Server 10.0.0.159:27020, ID=72, TYP=0] Exiting...

[Server 10.0.0.159:27020] Disconnected Successfully.
[Server 10.0.0.159:27020] Connection closed on other end, stopping reception...
[Server 10.0.0.159:27020] Reception thread closed.
q
Quit command issued.
Socket closed.
Alek@Badlek ~/steamcmd/ark/ShooterGame/Binaries/Linux/rcon_script $ |
```

```
Config file generated:
SAVED_FOLDER_SEARCH_DEPTH=4
MAX_BACKUP=3
SAVED_FOLDER_PATH=../../../../../ark/ShooterGame/Saved
JSON_KEY_FILE_PATH=serverbackup-395501-d87a69dc68f5.json

Looking for root cloud directory...
Root cloud directory found: Name: ARK_Backups, ID: 1qcgBTI70uf0hGr_6pE4eQ9M51-hxeoKB

Looking for existing backups...
Backups found:
Saved_Backup_15
Saved_Backup_14
Saved_Backup_13
Saved_Backup_12

Uploading folder...
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_02.11.2023_20.13.05.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_26.12.2022_03.28.25.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/1228672982.arktribe
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_20.12.2023_01.18.21.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/76561198409587803.arkprofile
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_21.12.2023_01.03.23.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_20.12.2023_03.28.21.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_31.07.2022_02.51.10.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_29.07.2022_02.10.24.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_28.10.2023_19.46.33.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_02.11.2023_03.20.17.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/SavedArks/TheIsland_17.12.2023_22.06.38.ark
File Uploaded: ../../../../../ark/ShooterGame/Saved/AllowedCheaterSteamIDs.txt
Folder uploaded successfully with name Saved_Backup_16

Checking to delete old backups...
Removing Saved_Backup_12...
Saved_Backup_12 removed successfully.

(backup_script) Alek@Badlek ~/steamcmd/ark/ShooterGame/Binaries/Linux/backup_script $
```

Purpose

Two scripts were developed to streamline the management of an ARK: Survival Evolved server on a local machine:

- **RCON Script:** Using Python with TCP sockets, this script implements Valve's RCON protocol from scratch, enabling remote execution of server commands outside of the game
- **Google Drive Backup Script:** This Python script automatically detects the game save folder within the server directory and subsequently uploads it to a Google Drive folder for backup

The ARK server was configured on a custom **Gentoo Linux** installation where the two scripts are now used on a regular basis

Captions

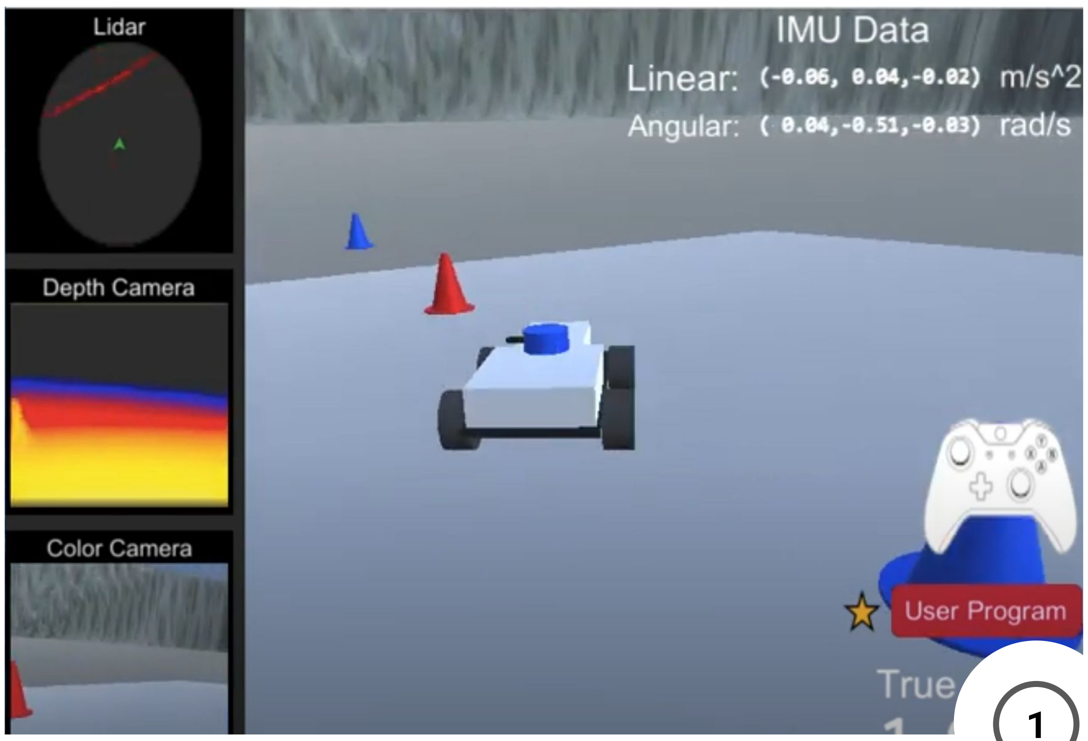
1. **RCON Script** in action
2. Execution of the **Google Drive backup script**
3. Shows latest backup appearing in Drive

Technologies

- Python
- Gentoo Linux
- TCP Socket Programming
- RCON Protocol
- Google API

MIT Beaverworks: RACECAR Course

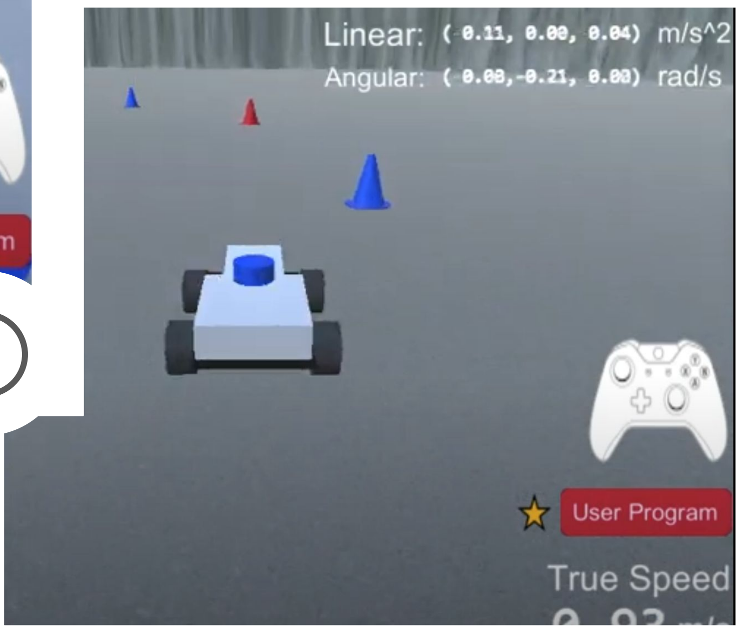
07/2020 - 08/2020


Lidar
Depth Camera
Color Camera
IMU Data
Linear: (-0.06, 0.04,-0.02) m/s^2
Angular: (0.04,-0.51,-0.03) rad/s
User Program
True

1

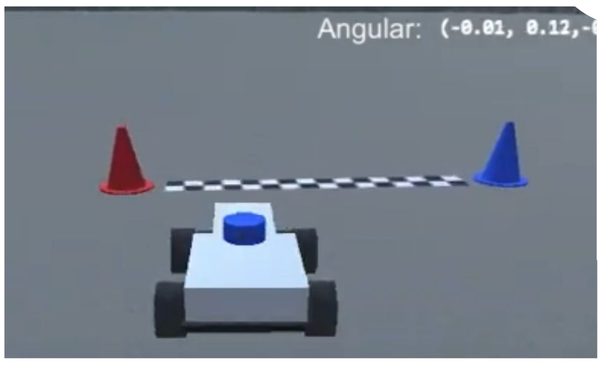
Purpose

Completed a month-long course at MIT Beaverworks progressing from the foundations of OpenCV to writing autonomous code for a car to navigate an obstacle course consisting of lane following, cone slaloms, ledge avoidance, and more

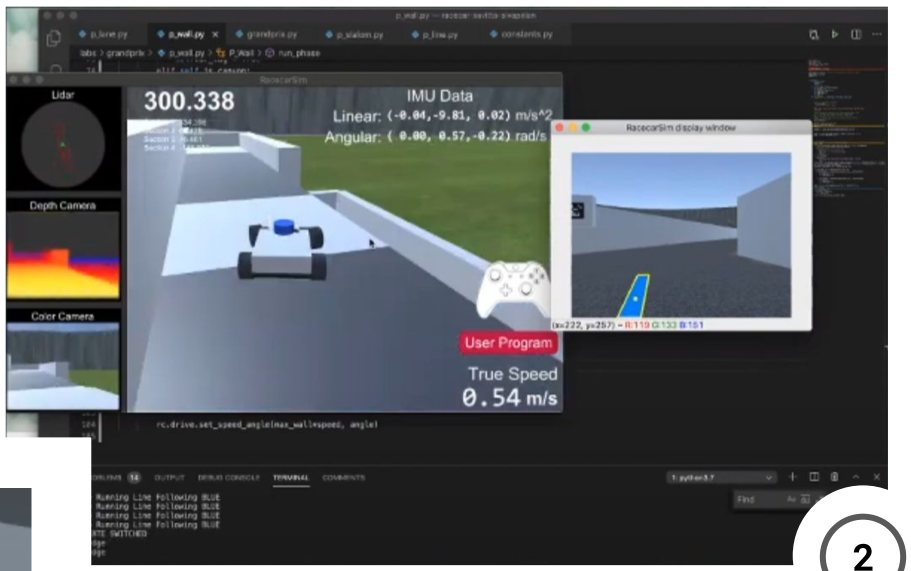

Linear: (0.11, 0.00, 0.04) m/s^2
Angular: (-0.00,-0.21, 0.00) rad/s
User Program
True Speed

Technologies

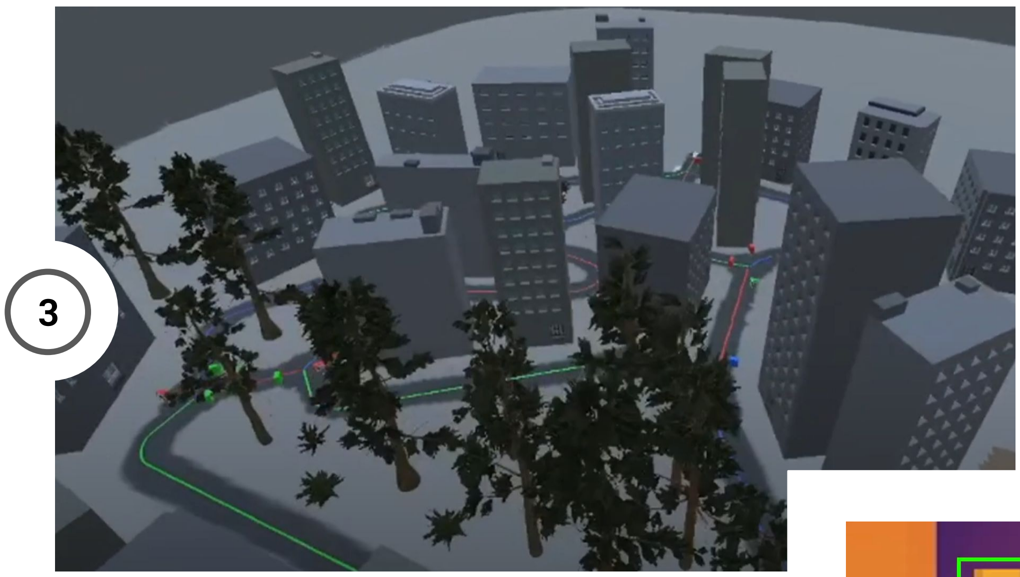
- OpenCV
- Numpy
- Depth Camera
- LiDAR
- SLAM
- Ubuntu
- Jetson NANO
- Sensor Fusion
- Unity
- Python
- AR Codes



- Cone Slalom
- Group Presentation Video
- Time Trial 1st Place
- Final Grand Prix Course Run



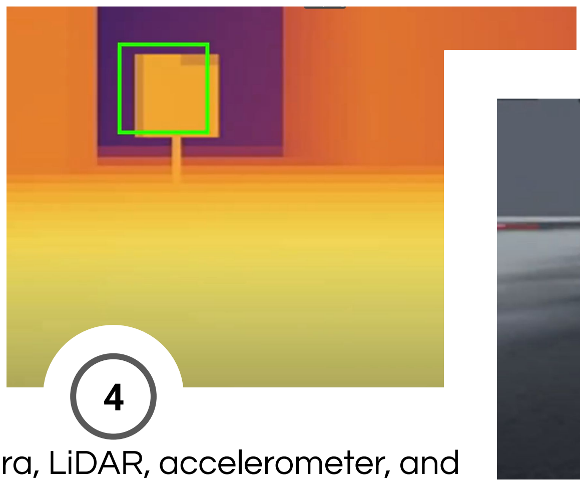
2



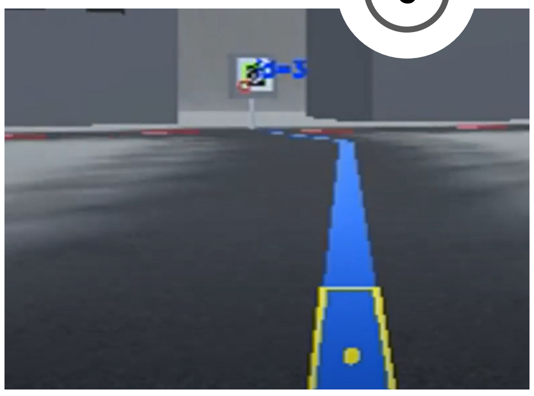
3

Captions

1. Cone Slalom
2. Ledge avoidance and wall following
3. City designed in Unity for navigation testing
4. Depth camera closest AR tag detection
5. Line follower and AR tag detection



4



5

Course Assignments Involved...

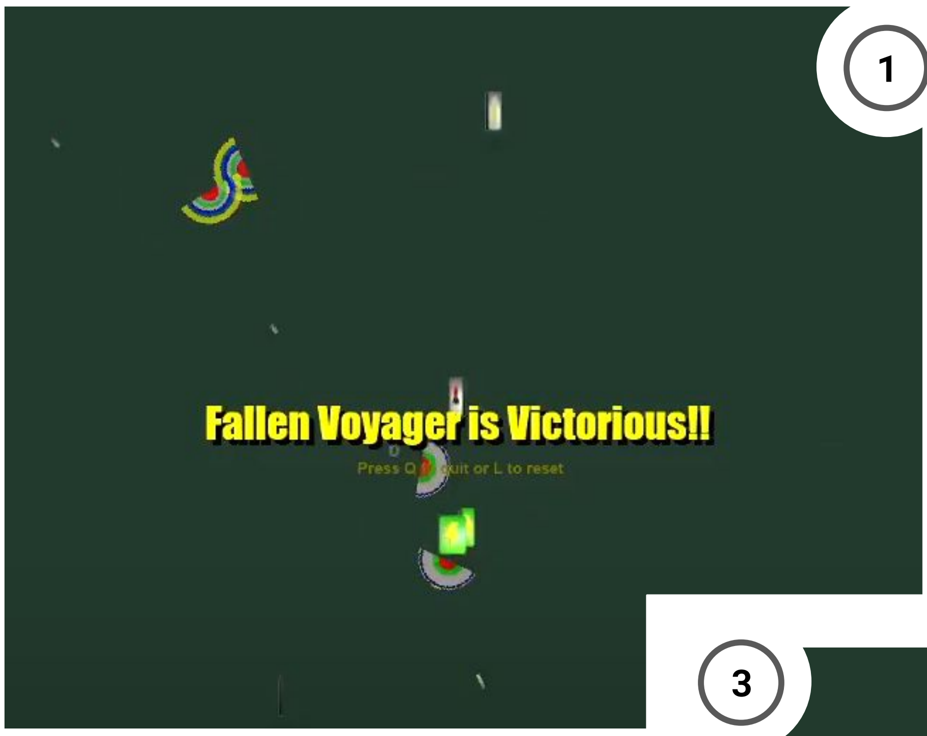
- Wrote a line and lane follower with color switching based on AR tags
- Wrote a cone slaloming algorithm
- Wrote SLAM code to navigate a wall obstacle course and keep track of location
- Incorporated data from a color camera, depth camera, LiDAR, accelerometer, and gyro in the autonomous navigation code applying sensor fusion techniques
- Deployed code to real robotic race car via Ubuntu on Jetson NANO
- Created the video for and contributed to our group's final presentation
- Wrote code for an autonomous time trial course winning 1st place among all students

AlekseyPanas/BWSI-Labs

apanas9246/GRANDPRIX

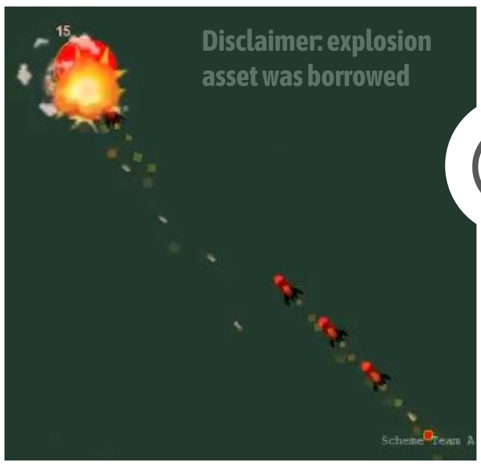
CodeBrawl

01/2021 - 04/2021

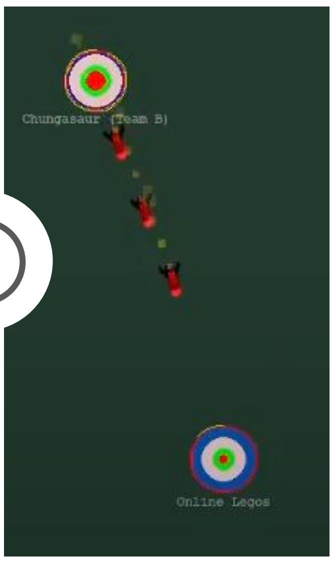


1

AlekseyPanas/CodeBrawl



2



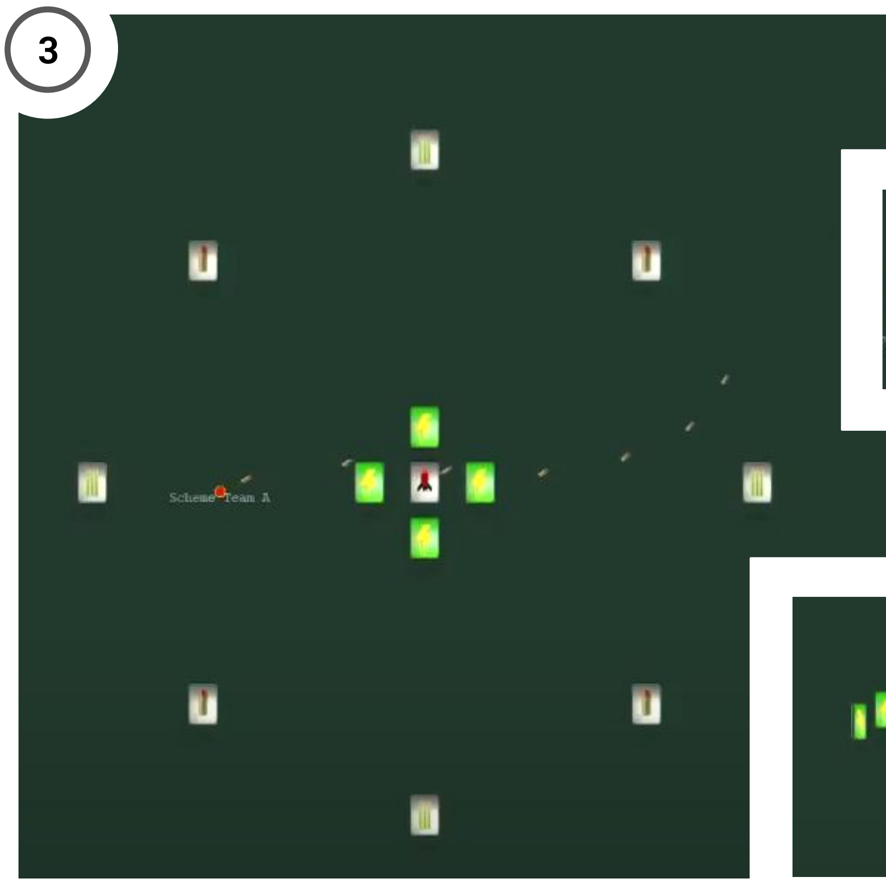
Purpose

Codebrawl was written to host a final coding competition in the **Brooklyn Tech PyUnityd Game Dev club**

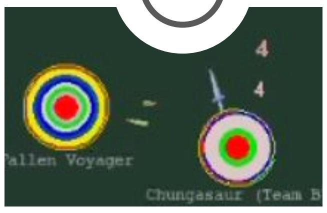
The game is multiplayer over **TCP** and is played by implementing an AI client script. The server then calls the AI scripts of each participating player and displays the visuals of the ongoing battle.

The **gameplay** involves customizing player stats pre-game, moving to acquire energy, shooting, close-range swordplay, and missiles.

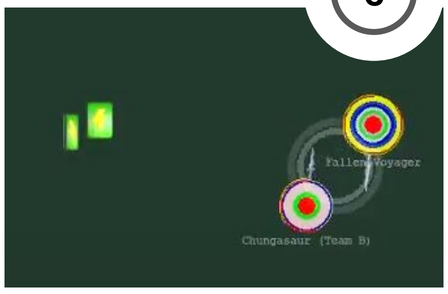
3



4



5

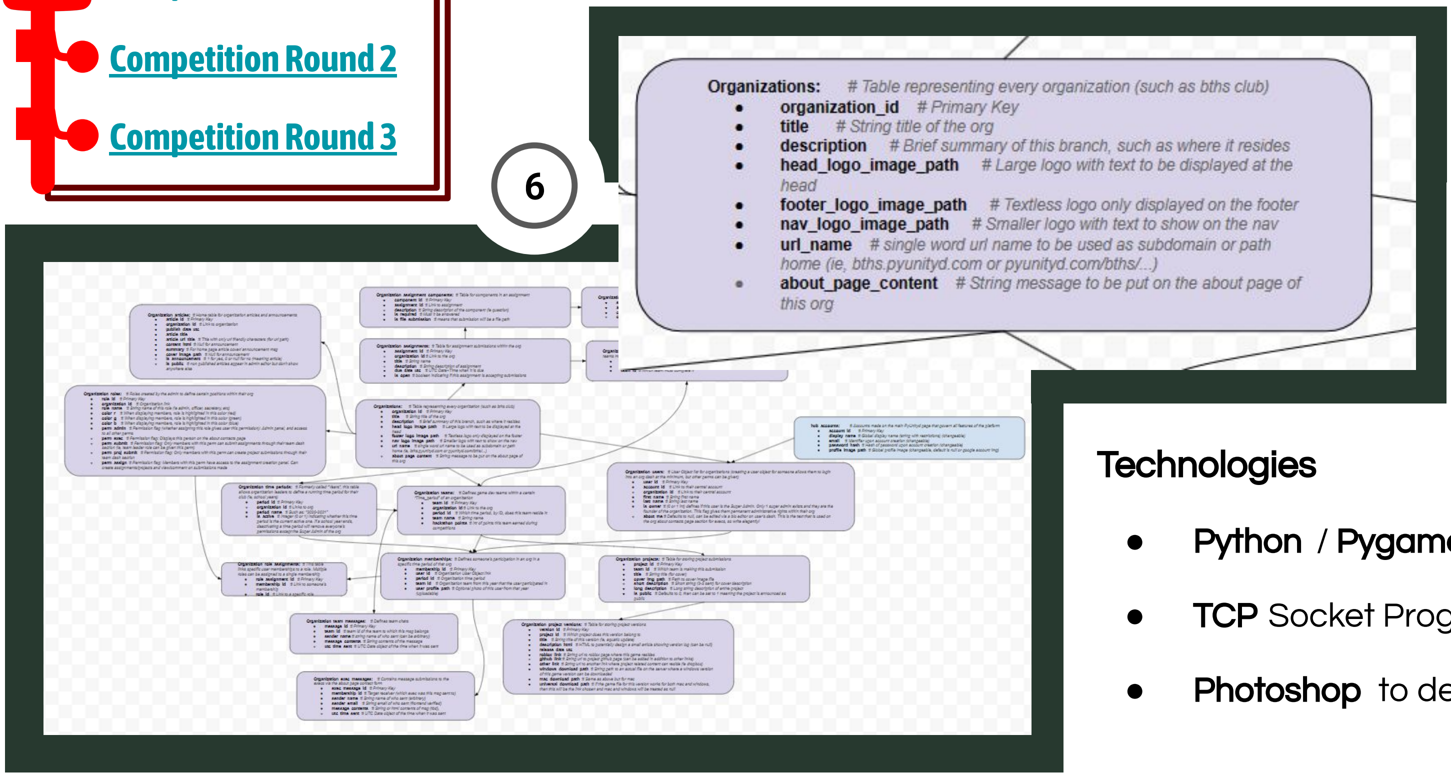


Competition Round 1

Competition Round 2

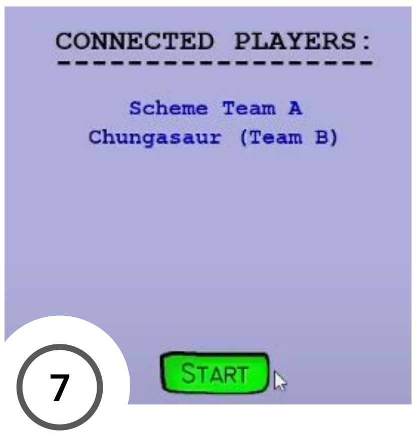
Competition Round 3

6



Organizations: *# Table representing every organization (such as bths club)*

- **organization_id** *# Primary Key*
- **title** *# String title of the org*
- **description** *# Brief summary of this branch, such as where it resides*
- **head_logo_image_path** *# Large logo with text to be displayed at the head*
- **footer_logo_image_path** *# Textless logo only displayed on the footer*
- **nav_logo_image_path** *# Smaller logo with text to show on the nav*
- **url_name** *# single word url name to be used as subdomain or path home (ie, bths.pyunityd.com or pyunityd.com/bths/...)*
- **about_page_content** *# String message to be put on the about page of this org*



CONNECTED PLAYERS:

Scheme Team A
Chungasaur (Team B)

START

7

Technologies

- **Python** / **Pygame**
- **TCP** Socket Programming
- **Photoshop** to design assets

Captions

1. Victory screen with custom splitting and fading animations on defeat
2. Missiles being fired
3. Starting arena appearance and bullet firing
4. Damage number indicators, sword, and bullets
5. Sword blocking mechanic
6. Database map for ongoing work to port project to web
7. Lobby screen

Contribution

Developed the entire game start to finish in Python. Brought on two others to assist in writing additional clients in Scala, C#, and Java

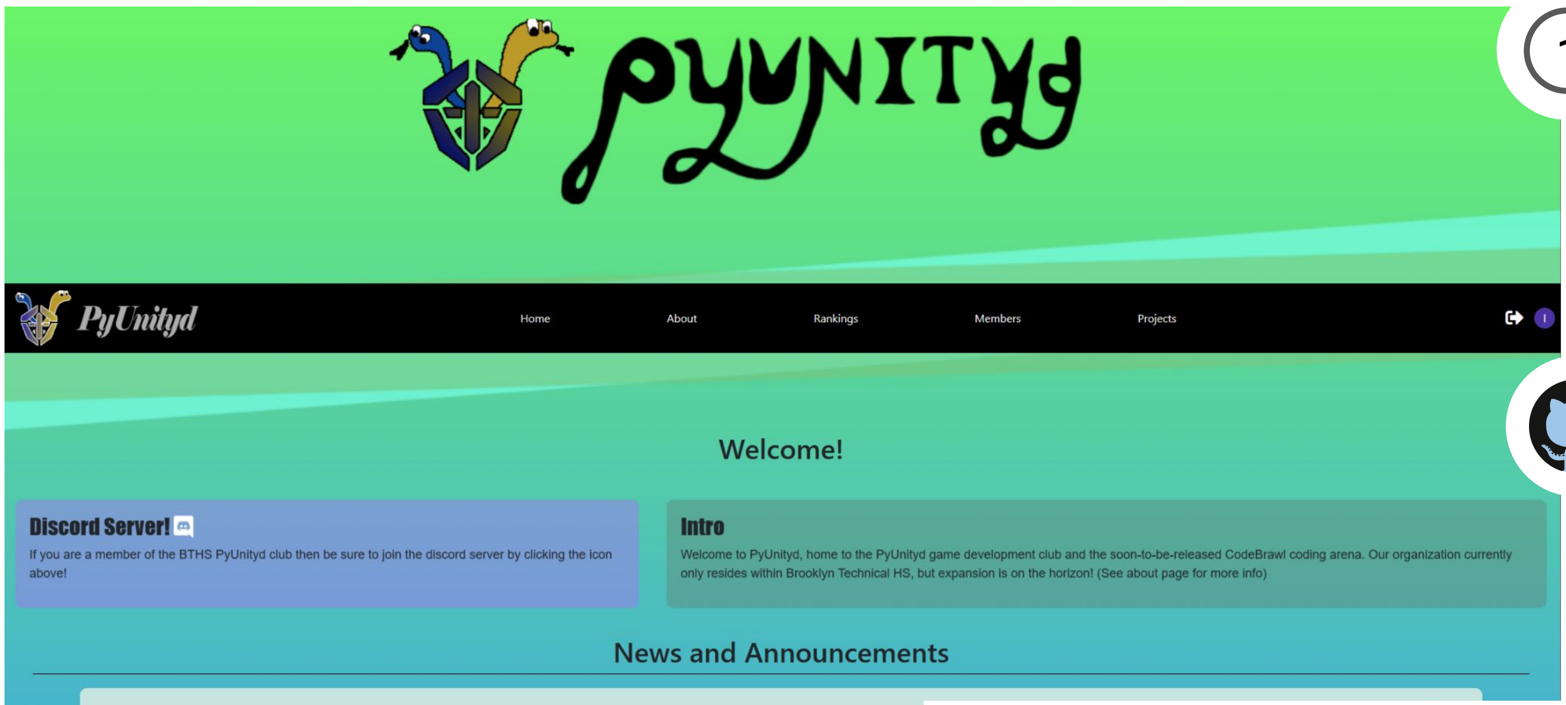
What's Ahead

Ongoing work to port this game to a webapp

PyUnityd.com

www.pyunityd.com

09/2019 - Present



Welcome!

Discord Server!

If you are a member of the BTHS PyUnityd club then be sure to join the discord server by clicking the icon above!

Intro

Welcome to PyUnityd, home to the PyUnityd game development club and the soon-to-be-released CodeBrawl coding arena. Our organization currently only resides within Brooklyn Technical HS, but expansion is on the horizon! (See about page for more info)

News and Announcements

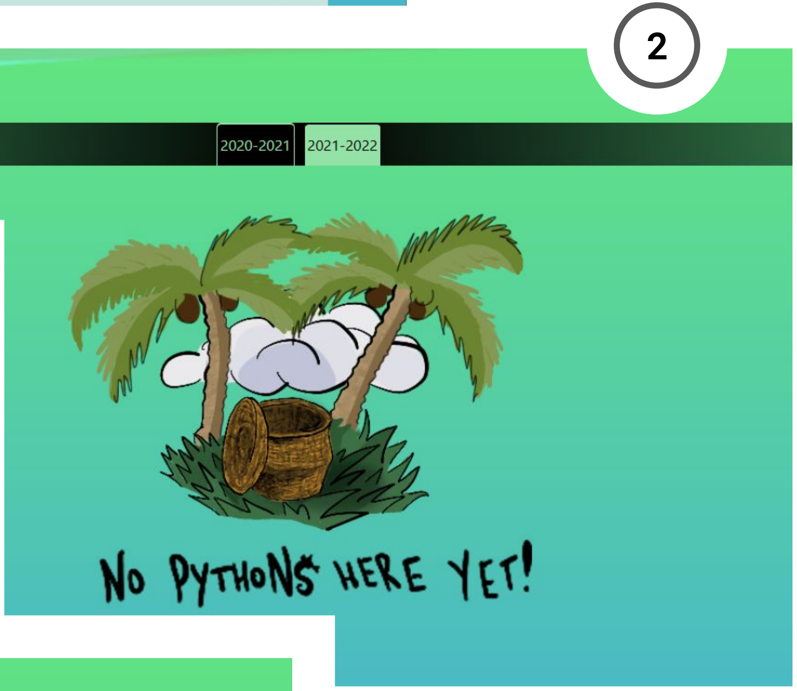
Image Test

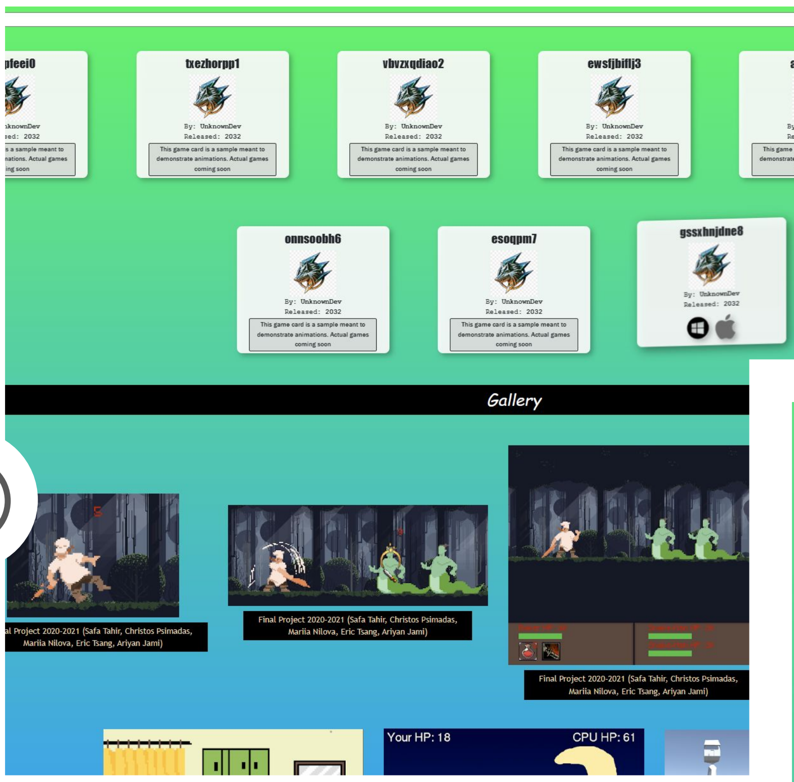
A test article

Test Article 4

A test article

AlekseyPanas/PyUnitydRanking





pfeei0

txezhorpp1

By: UnknownDev
Released: 2032
This game card is a sample meant to demonstrate animations. Actual games coming soon

vbvzxqdiao2

By: UnknownDev
Released: 2032
This game card is a sample meant to demonstrate animations. Actual games coming soon

ewsfjbiflj3

By: UnknownDev
Released: 2032
This game card is a sample meant to demonstrate animations. Actual games coming soon

acwchvh4

By: UnknownDev
Released: 2032
This game card is a sample meant to demonstrate animations. Actual games coming soon

buln

onnsoobh6

By: UnknownDev
Released: 2032
This game card is a sample meant to demonstrate animations. Actual games coming soon

esoqpm7

By: UnknownDev
Released: 2032
This game card is a sample meant to demonstrate animations. Actual games coming soon

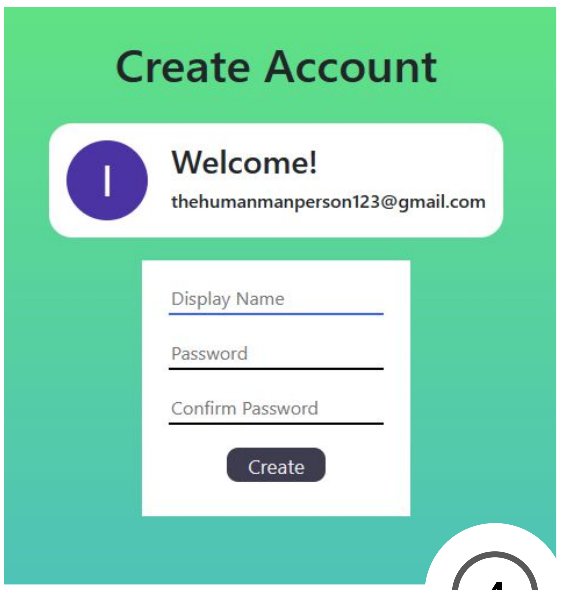
gssxhnjdne8

By: UnknownDev
Released: 2032

al Project 2020-2021 (Safa Tahir, Christos Psimadas, Mariia Nilova, Eric Tsang, Ariyan Jami)

Final Project 2020-2021 (Safa Tahir, Christos Psimadas, Mariia Nilova, Eric Tsang, Ariyan Jami)

Final Project 2020-2021 (Safa Tahir, Christos Psimadas, Mariia Nilova, Eric Tsang, Ariyan Jami)



Create Account

Welcome!

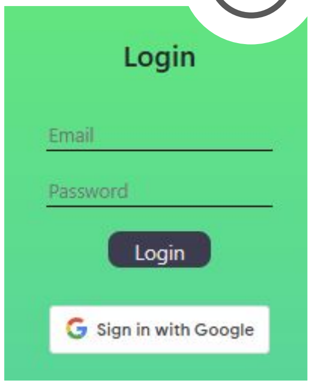
thehumanmanperson123@gmail.com

Display Name

Password

Confirm Password

Create



Login

Email

Password

Login

Sign in with Google

Technologies

- NodeJS
- ExpressJS
- PostgreSQL
- OAuth
- AWS EC2, RDS, Route53
- TLS (SSL)
- Git / Github
- NginX
- HTML / CSS / JS
- Async / Await
- Ubuntu
- Photoshop / Procreate

Purpose

Official website for the Brooklyn Tech PyUnityd Club created to display club membership, host competition results, and showcase club projects.

Contribution

As Founder and President of the club, initiated the decision to create an organization website and built the solo project from scratch

Captions

1. Home page
2. Art created in **Procreate**
3. Project Page with gallery of club members' creations
4. Account creation system with google login option

What's Ahead

After graduating High School, plans were put in action to turn this into a public game development competition platform. That is still an **ongoing** development

Robotics: Team334 Techknights

09/2017 - 05/2020

1

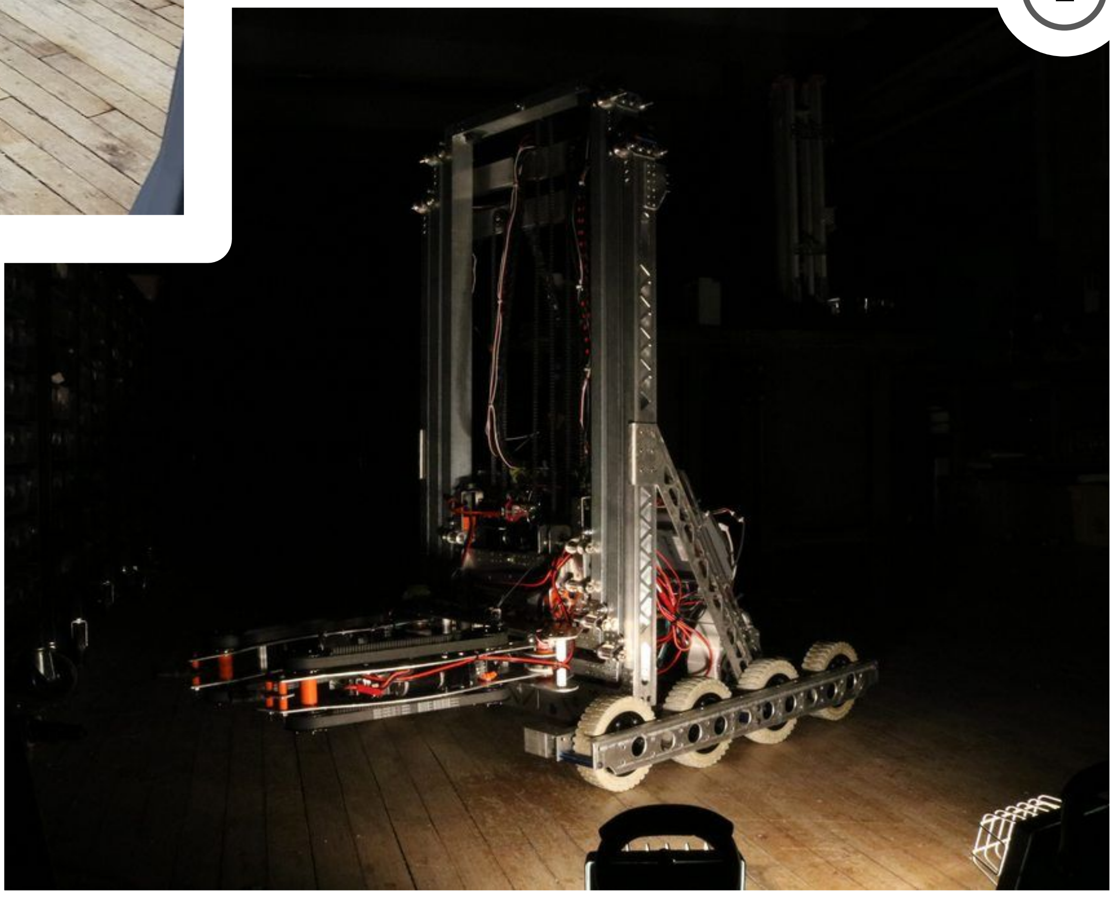
Purpose

Brooklyn Technical High School's Robotics Team (334 TechKnights) participates in an annual FIRST Robotics competition.

Each year, a new game is revealed comprising both autonomous and human-operated phases.

Within a six-week build season, the team designs, constructs, and programs a robot from the ground up to meet the game's challenges.

2



Captions

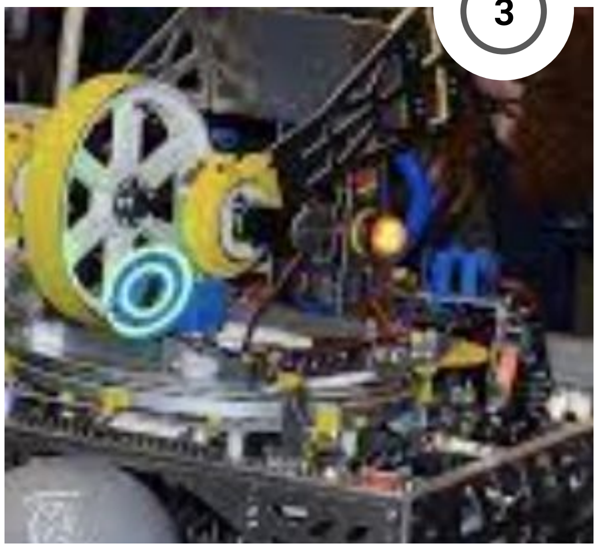
1. Fun side project: a ride-able "seat bot" coded in **C++** and powered by Arduino
2. Team334's robot for the 2019 competition
3. 2015 Robot used frequently for testing code (LED ring for aiding in computer vision tasks)

Contribution

Contributed code for four years, writing computer vision in **OpenCV** with **Python** and the remaining autonomous and teleop code in **Java**.

In latter two years, acted as programmer lead implementing an **Agile** workflow.

3



Technologies

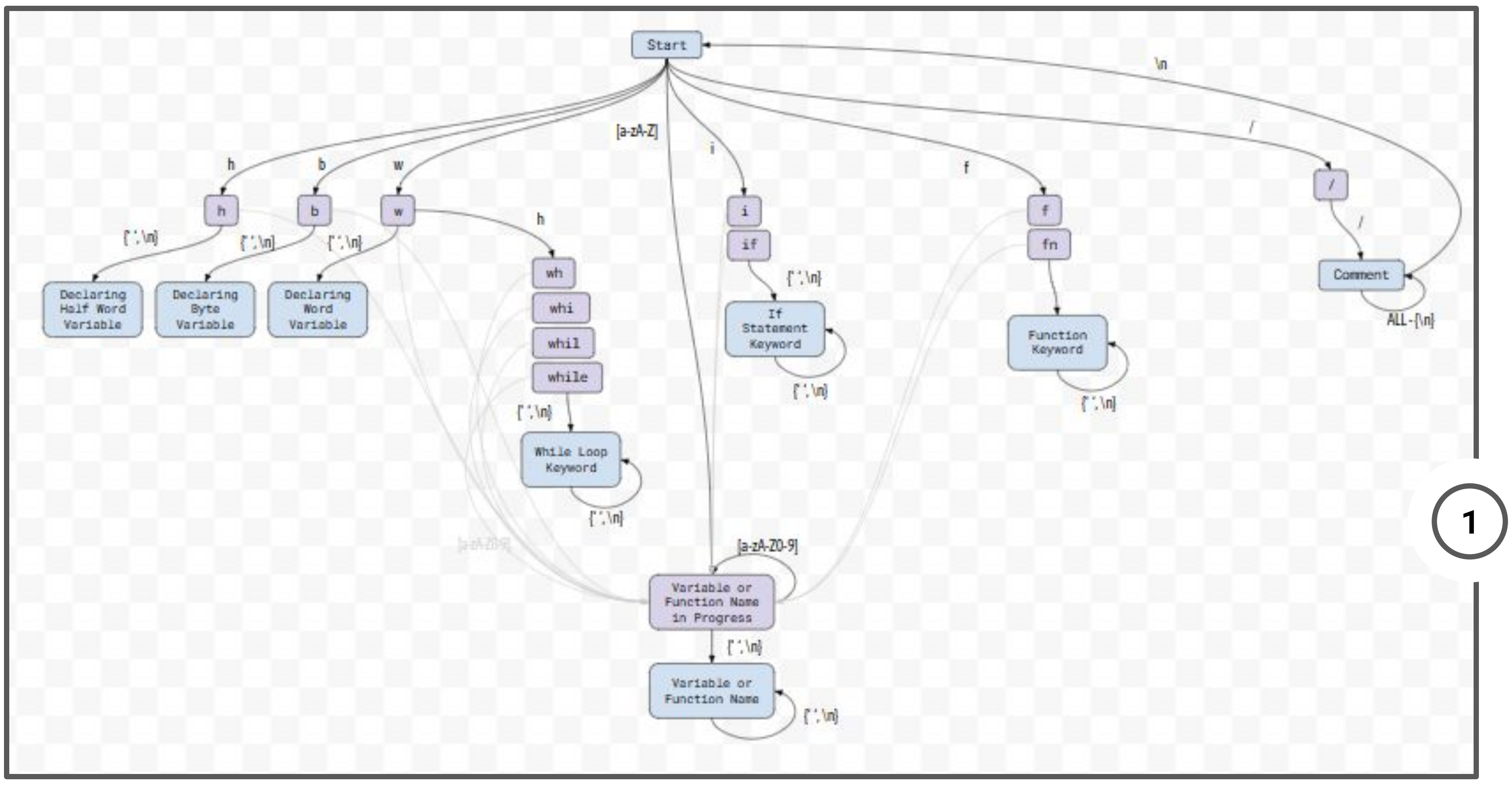
- OpenCV
- Numpy
- Java
- PID Loops
- Jetson NANO
- Arduino
- C++
- Ubuntu
- Robotic hardware & Sensors
  - LiDAR, Gyro, Ultrasonic, Camera, Accelerometer, Potentiometer, Pneumatic systems, Motors
- Python
- ROS
- Git/Github
- Agile (Shortcut)

Team334/R2019

Team334/R2018

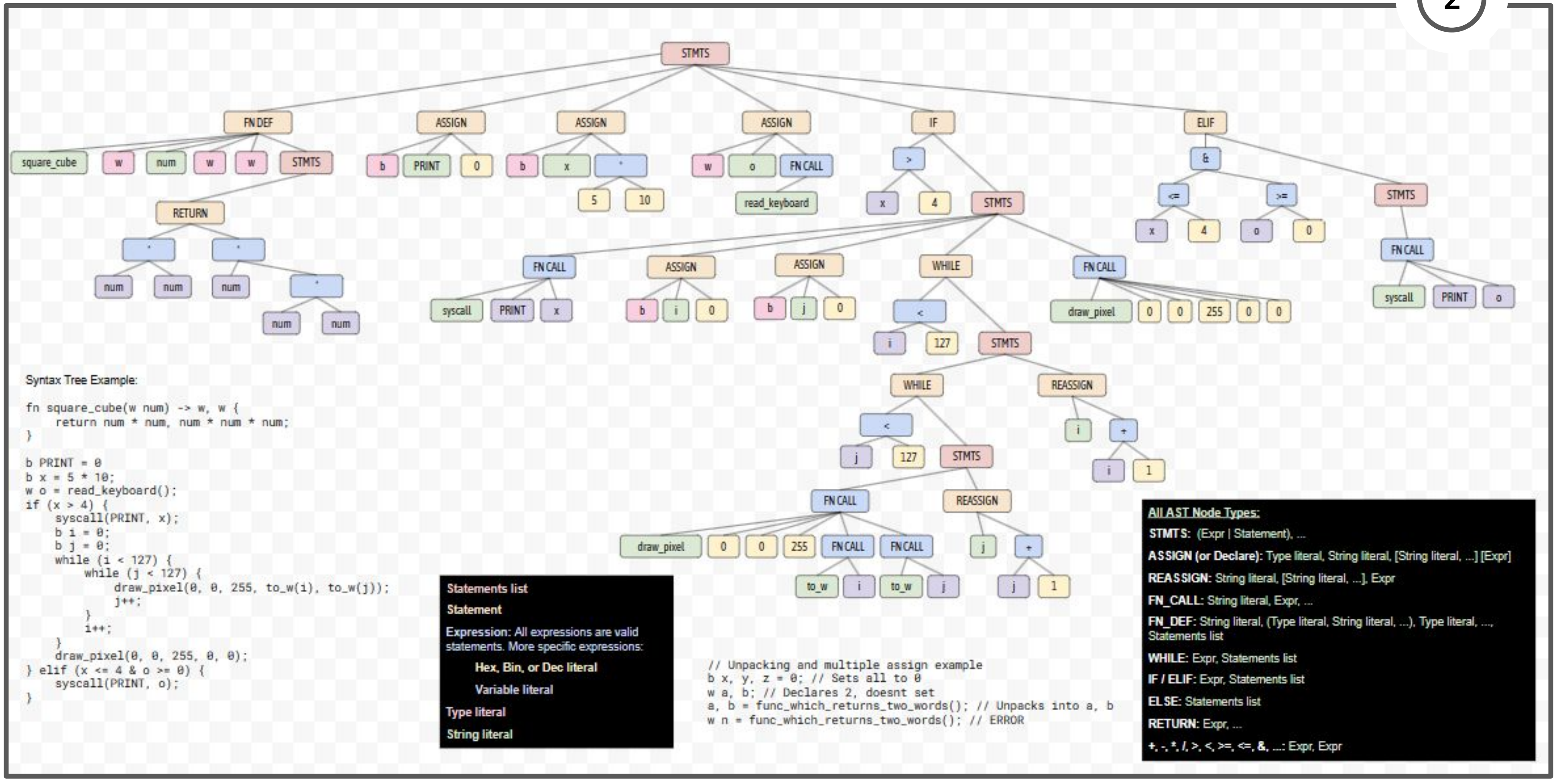
Custom Language MIPS Compiler

06/2023 - Present



1

2



Syntax Tree Example:

```
fn square_cube(w num) -> w, w {
    return num * num, num * num * num;
}

b PRINT = 0
b x = 5 * 10;
w o = read_keyboard();
if (x > 4) {
    syscall(PRINT, x);
    b i = 0;
    b j = 0;
    while (i < 127) {
        while (j < 127) {
            draw_pixel(0, 0, 255, to_w(i), to_w(j));
            j++;
        }
        i++;
    }
    draw_pixel(0, 0, 255, 0, 0);
} elif (x <= 4 & o >= 0) {
    syscall(PRINT, o);
}
```

Statements list

Statement

Expression: All expressions are valid statements. More specific expressions:

Hex, Bin, or Dec literal

Variable literal

Type literal

String literal

```
// Unpacking and multiple assign example
b x, y, z = 0; // Sets all to 0
w a, b; // Declares 2, doesnt set
a, b = func_which_returns_two_words(); // Unpacks into a, b
w n = func_which_returns_two_words(); // ERROR
```

All AST Node Types:

STMTS: (Expr | Statement), ...

ASSIGN (or Declare): Type literal, String literal, [String literal, ...] [Expr]

REASSIGN: String literal, [String literal, ...], Expr

FN_CALL: String literal, Expr, ...

FN_DEF: String literal, (Type literal, String literal, ...), Type literal, ..., Statements list

WHILE: Expr, Statements list

IF / ELIF: Expr, Statements list

ELSE: Statements list

RETURN: Expr, ...

+, -, *, /, >, <, >=, <=, &, ...: Expr, Expr

Purpose

A personal solo project conceived after finishing Computer Organization course out of curiosity to write own person language. The project is **ongoing** in its early stages with a completed grammar parser and 10% completed compiler. The language is low level including only byte, word, and halfword datatypes, array functionality, loops, and if statements.

AlekseyPanas/PanaScript

Technologies

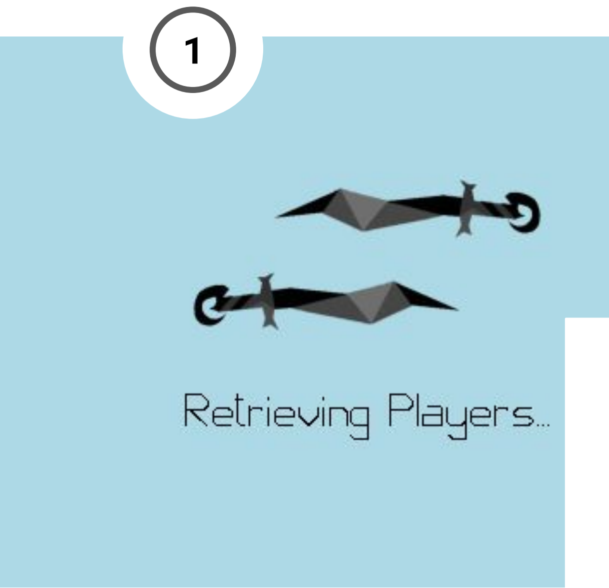
- C
- Abstract Syntax Trees
- MIPS Assembly

Captions

1. DFSA diagram initially used when brainstorming how to parse grammar
2. Syntax tree when brainstorming language syntax and components

Knorr Stats

12/2019 - Present

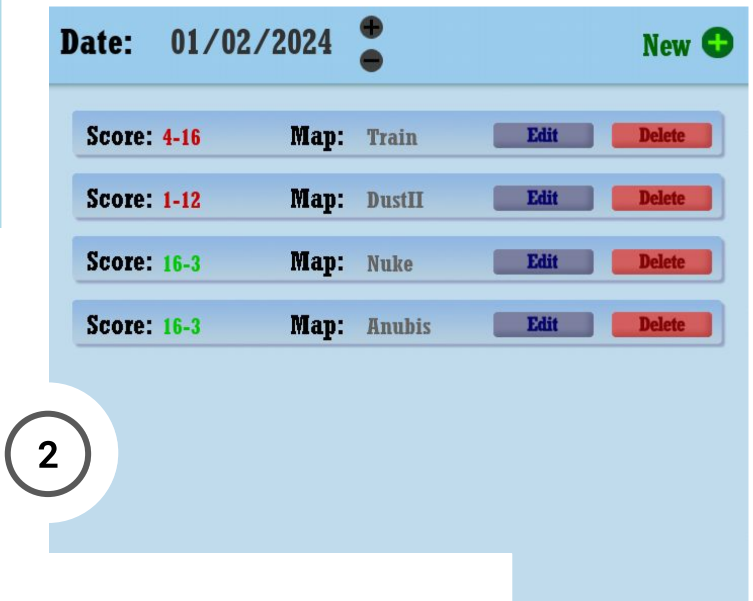


Purpose

Solo-built app created after launching a semi-casual Counter Strike team to track games played and per-player statistics to gauge team performance and analyze areas of improvement

Captions

1. Animated loading screen with custom-made team logo
2. Interface for adding played games
3. Errors when connections fail
4. Interface for inputting game statistics
5. Web frontend for viewing team information (work in progress)



Date: 01/02/2024

Score: 4-16 Map: Train Edit Delete

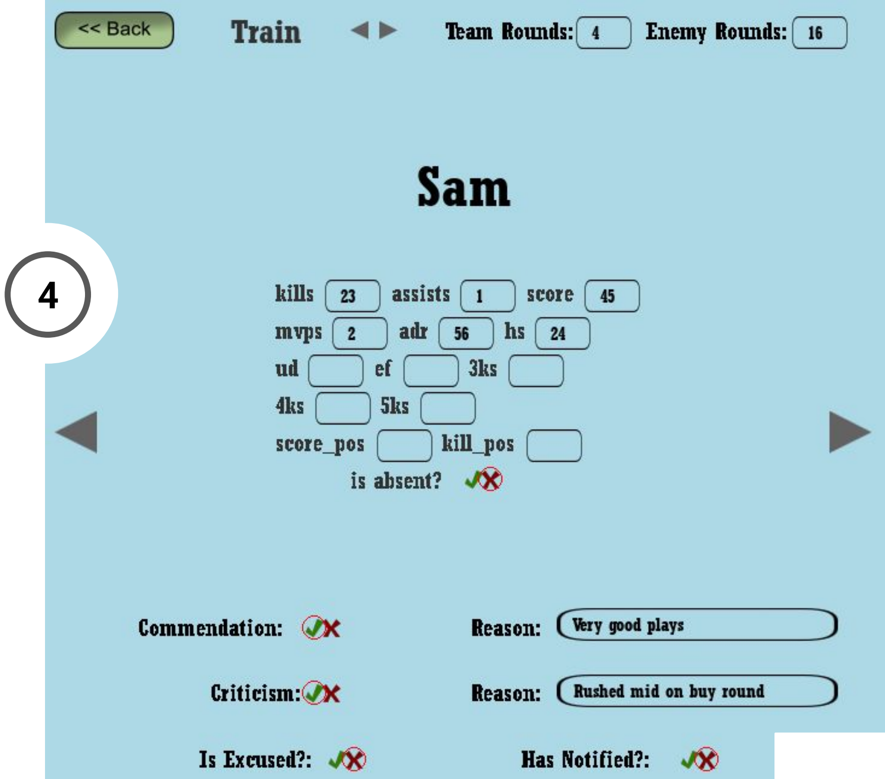
Score: 1-12 Map: DustII Edit Delete

Score: 16-3 Map: Nuke Edit Delete

Score: 16-3 Map: Anubis Edit Delete

Submit

Error: Cannot Connect to Database



Train

Team Rounds: 4 Enemy Rounds: 16

Sam

kills 23 assists 1 score 45
mvps 2 adr 56 hs 24
ud ef 3ks
4ks 5ks
score_pos kill_pos
is absent? ✓✗

Commendation: ✓✗ Reason: Very good plays

Criticism: ✓✗ Reason: Rushed mid on buy round

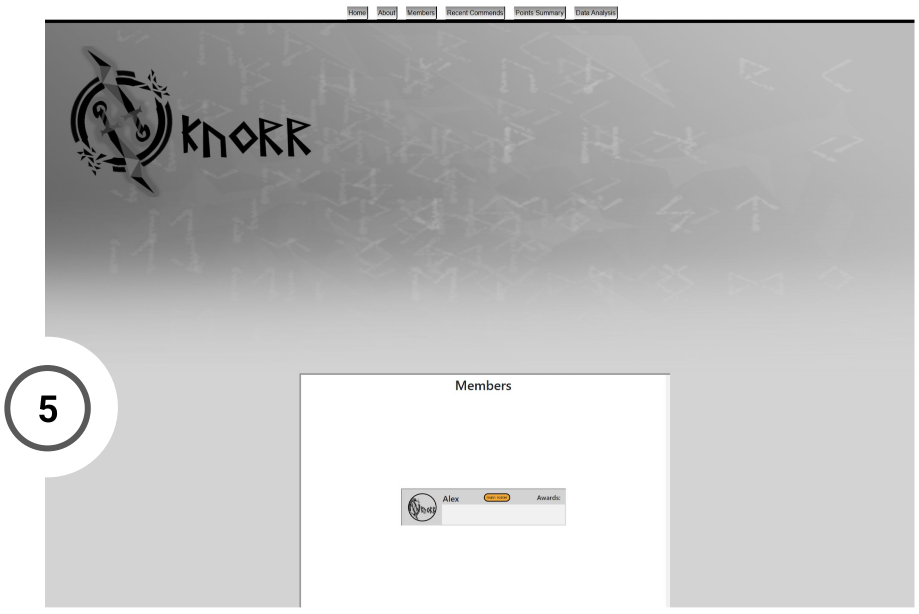
Is Excused?: ✓✗ Has Notified?: ✓✗

What's Ahead

Although the team was dissolved after first two years of Covid, **ongoing** work is being done to implement automatic tracking with a steam bot pivoting towards an open-source release

Technologies

- NodeJS / ExpressJS
- PostgreSQL
- Python / Pygame
- HTML / CSS / JS

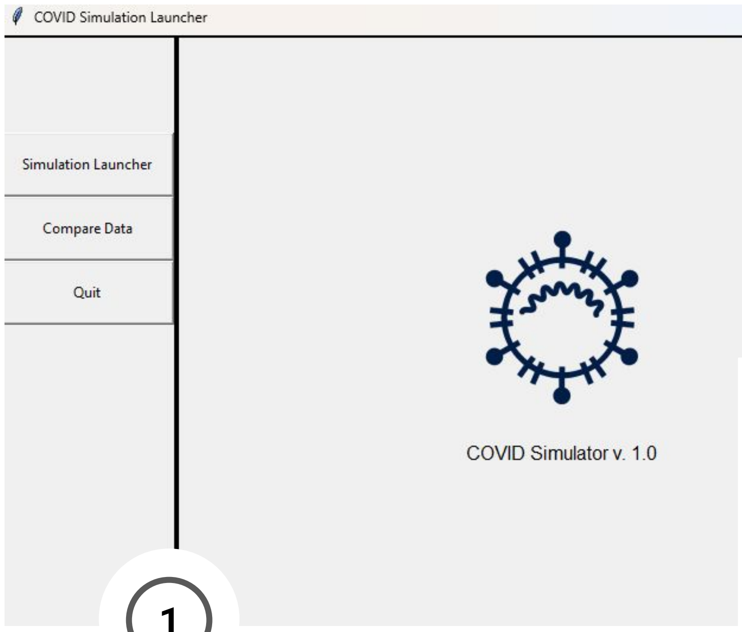


Members

Alex main roster Awards:

CovSim

2021

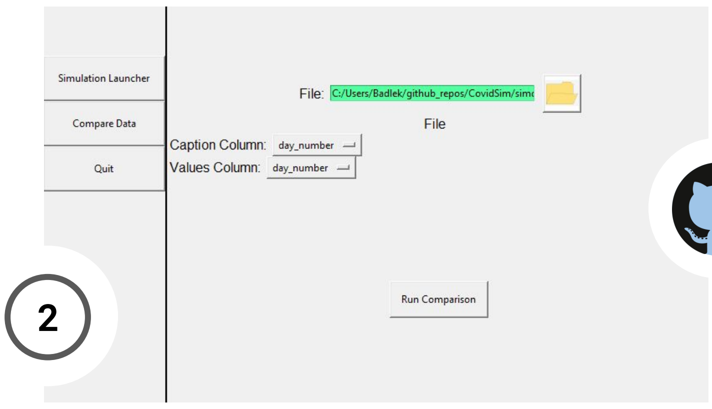


COVID Simulation Launcher

COVID Simulator v. 1.0

Purpose

Four-person first year University computer science project with a theme around Covid. The app runs an infection simulation for an arbitrary city defined with customizable parameters.



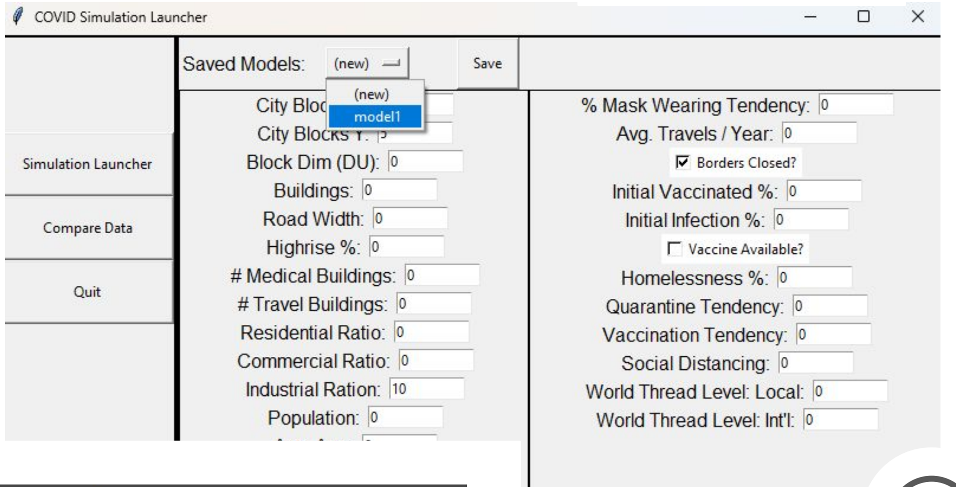
File: C:/Users/Badlek/github_repos/CovidSim/simc

File

Caption Column: day_number

Values Column: day_number

AlekseyPanas/CovidSim



COVID Simulation Launcher

Saved Models: (new)

City Bloc

City Blocks Y: 5

Block Dim (DU): 0

Buildings: 0

Road Width: 0

Highrise %: 0

Medical Buildings: 0

Travel Buildings: 0

Residential Ratio: 0

Commercial Ratio: 0

Industrial Ration: 10

Population: 0

% Mask Wearing Tendency: 0

Avg. Travels / Year: 0

☑ Borders Closed?

Initial Vaccinated %: 0

Initial Infection %: 0

☐ Vaccine Available?

Homelessness %: 0

Quarantine Tendency: 0

Vaccination Tendency: 0

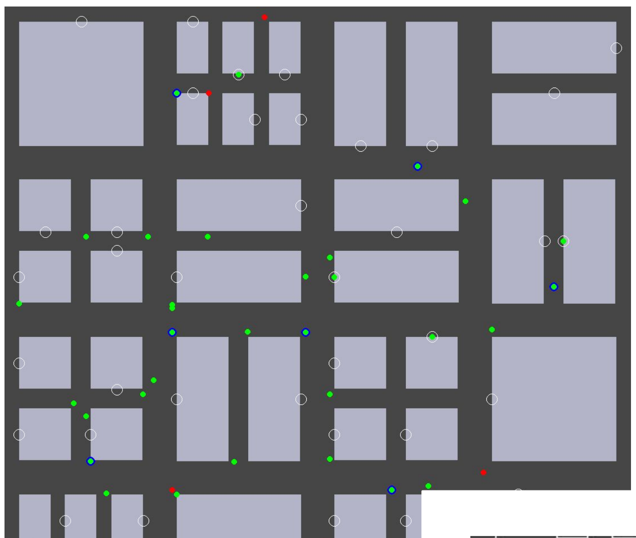
Social Distancing: 0

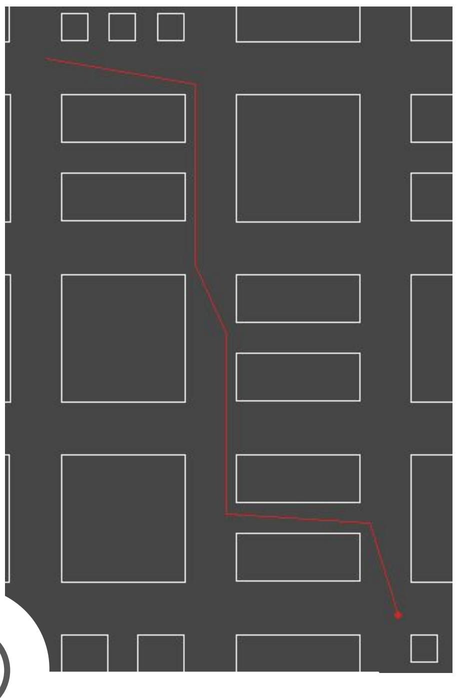
World Thread Level: Local: 0

World Thread Level: Int'l: 0

Captions

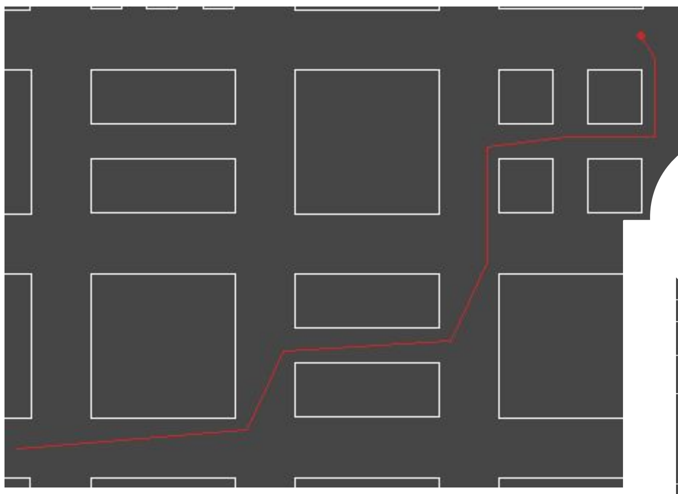
1. Tkinter launcher GUI
2. Load CSV files to view statistics via Pandas
3. Simulation configuration interface
4. Simulation view, dots are people with red as infected and blue as vaccinated
5. Custom-built path finding algorithm

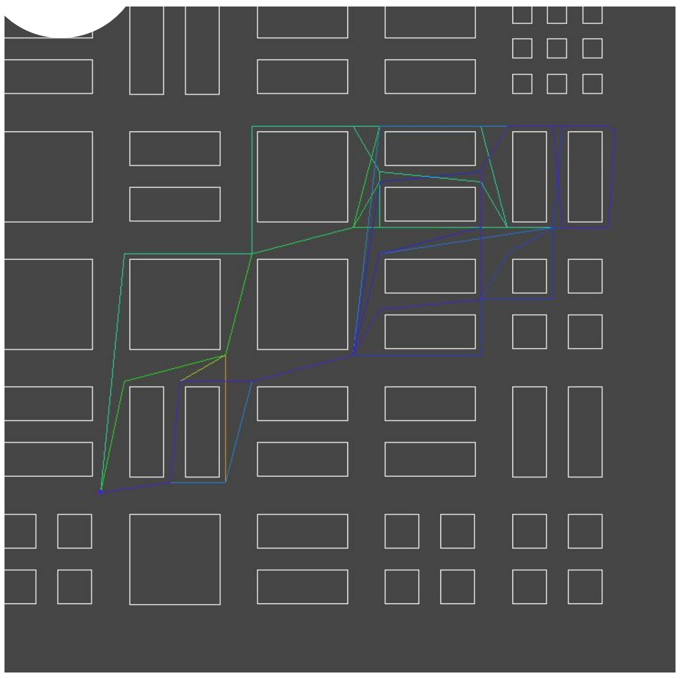




Technologies

- Python
- Pygame
- Tkinter
- Pandas
- Numpy



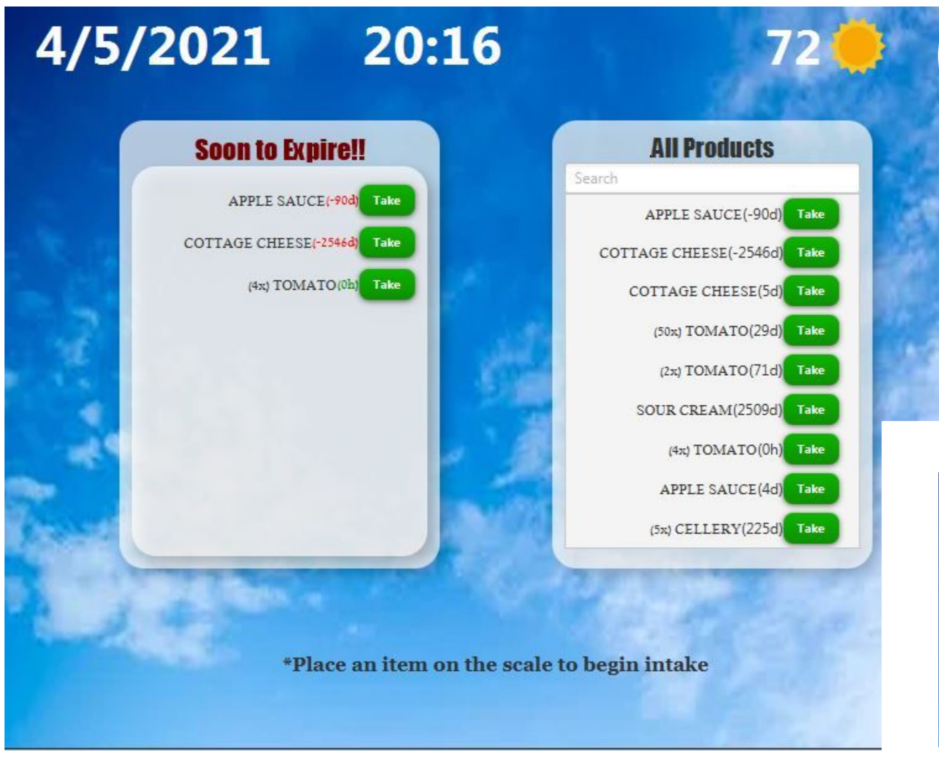


Contribution

- Designed a recursive **pathfinding** algorithm to navigate between an arbitrary set of quadrilaterals
- Using **probability** and **calculus**, wrote core simulation logic to compute citizen infection using time integral approximations over distance
- Wrote logic to generate the city
- Wrote majority of the pygame UI for visualizing the simulation

Advanced Fridge

2020



Soon to Expire!!

- APPLE SAUCE(-90d) Take
- COTTAGE CHEESE(-2546d) Take
- (4x) TOMATO(0h) Take

All Products

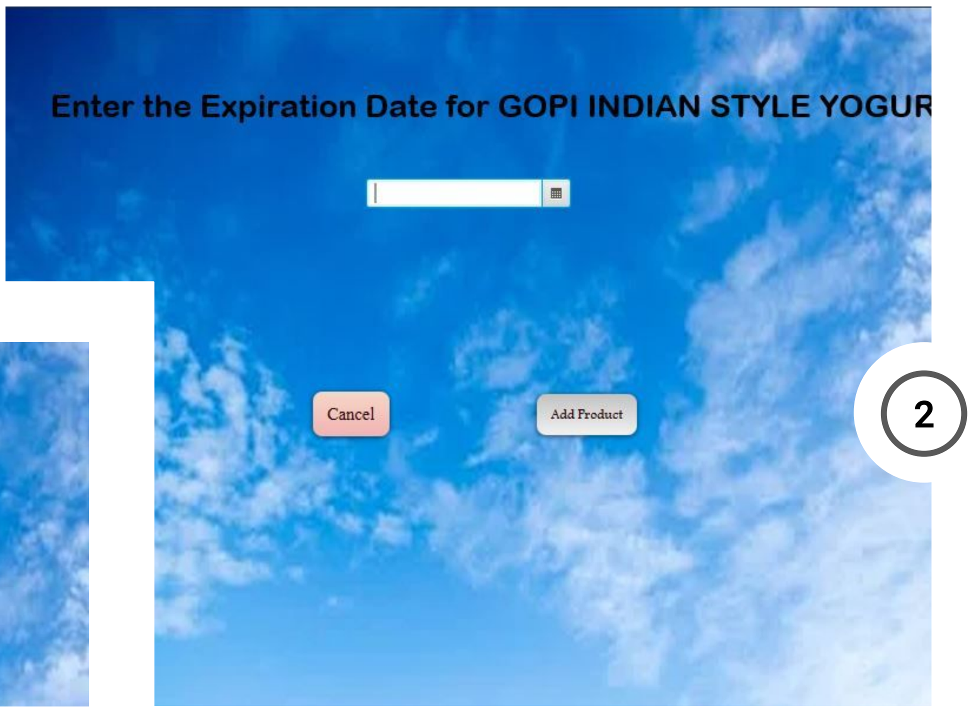
- APPLE SAUCE(-90d) Take
- COTTAGE CHEESE(-2546d) Take
- COTTAGE CHEESE(5d) Take
- (50x) TOMATO(29d) Take
- (2x) TOMATO(71d) Take
- SOUR CREAM(2509d) Take
- (4x) TOMATO(0h) Take
- APPLE SAUCE(4d) Take
- (5x) CELLERY(225d) Take

*Place an item on the scale to begin intake

1

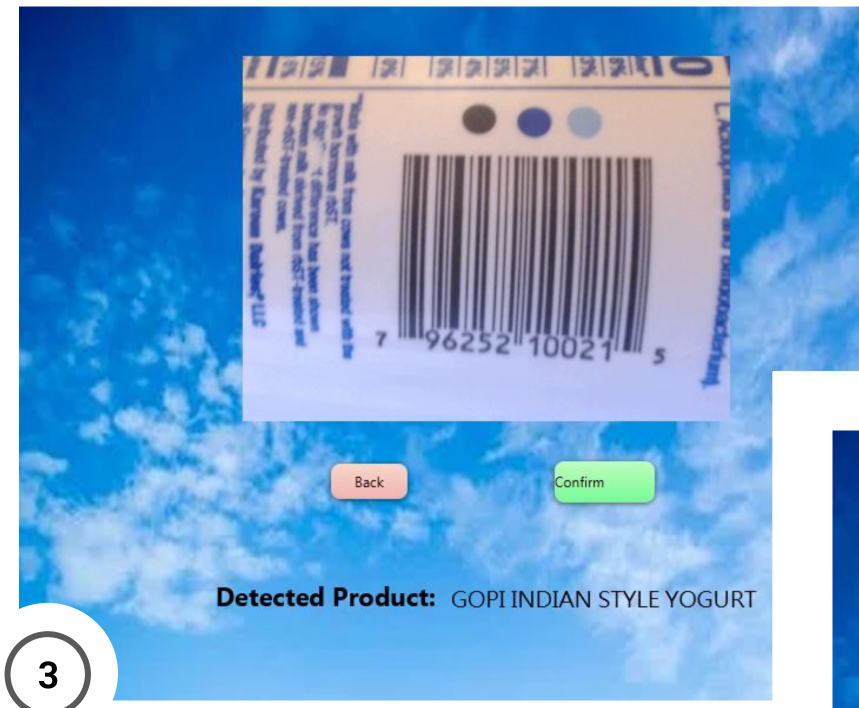
Purpose

A fridge management app developed for a hackathon project (UofT Hacks) with one partner. The app is a proof of concept for an embedded software that would run on a smart fridge allowing you to scan products as you put them in to track expiration dates and suggest recipes



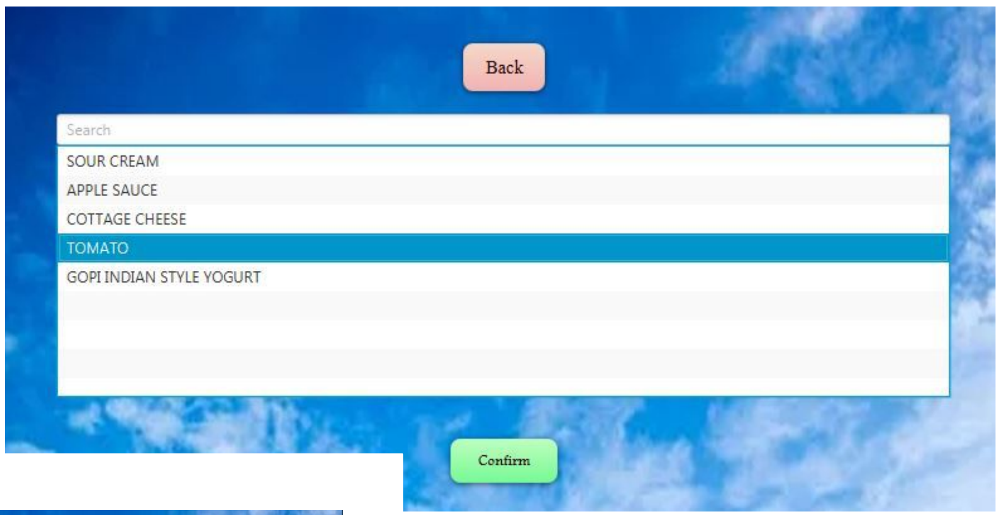
Enter the Expiration Date for GOPI INDIAN STYLE YOGUR

2



Detected Product: GOPI INDIAN STYLE YOGURT

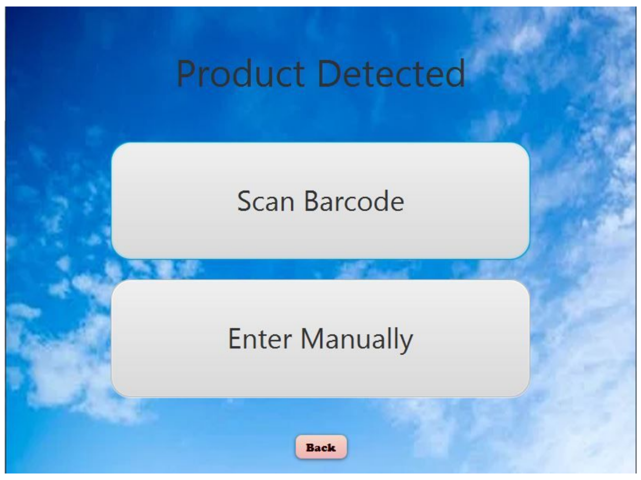
3



- SOUR CREAM
- APPLE SAUCE
- COTTAGE CHEESE
- TOMATO
- GOPI INDIAN STYLE YOGURT

Captions

1. Main screen displaying products
2. Manual expiration entry if not deducible from vision code
3. Barcode product detection



Technologies

- Java / JavaFX
- Google Zxing
- OpenCV

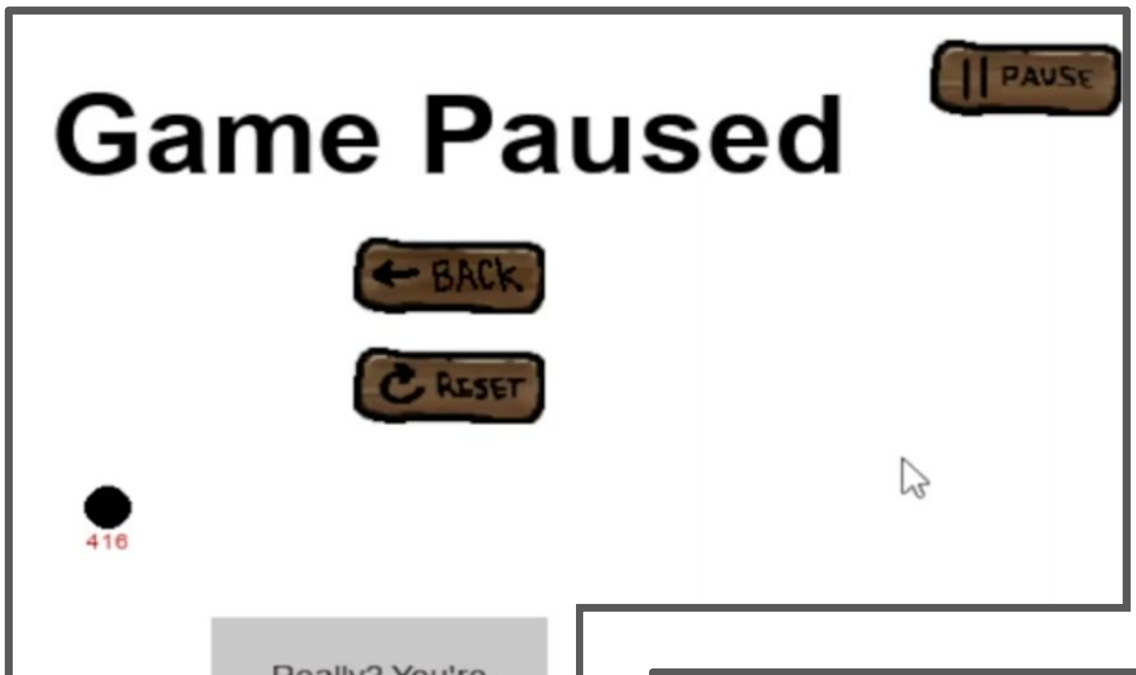
Contribution

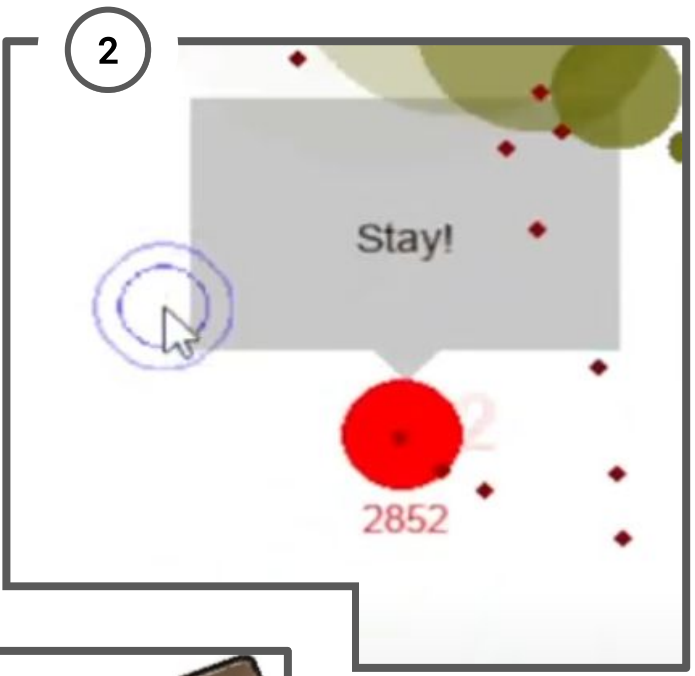
Wrote approximately 60% of the UI, and worked collectively with partner to solve the computer vision component of detecting barcodes and reading expiration dates

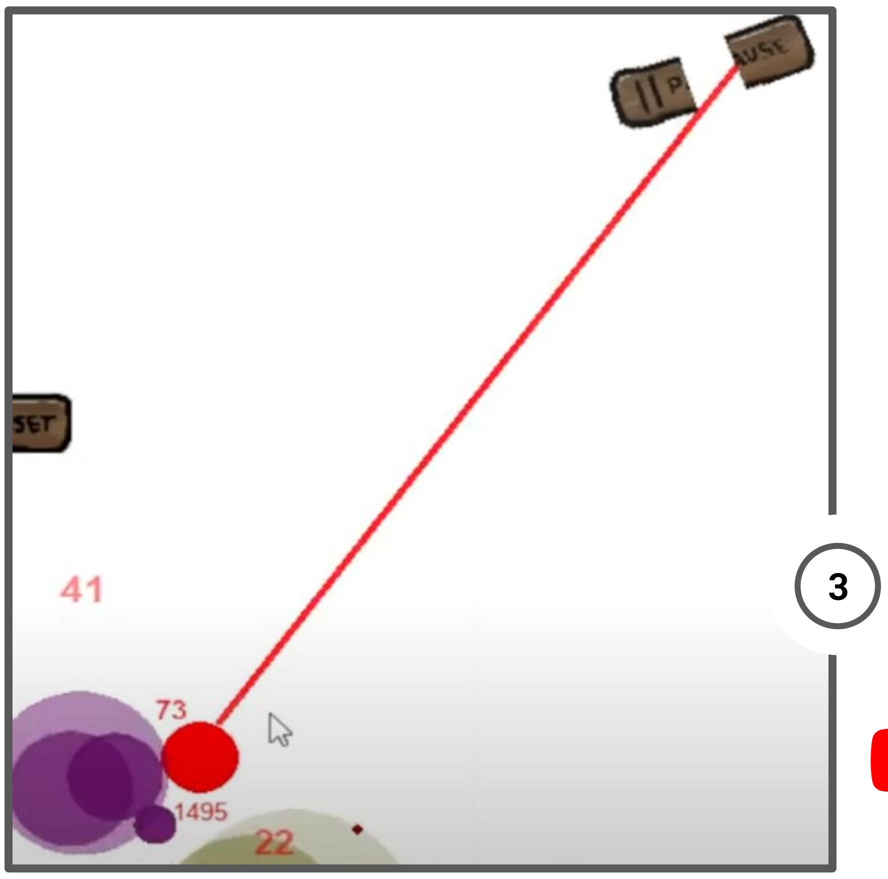
AlekseyPanas/AdvancedFridge

Game: “Breaking the Fourth Wall”

2021







Technologies

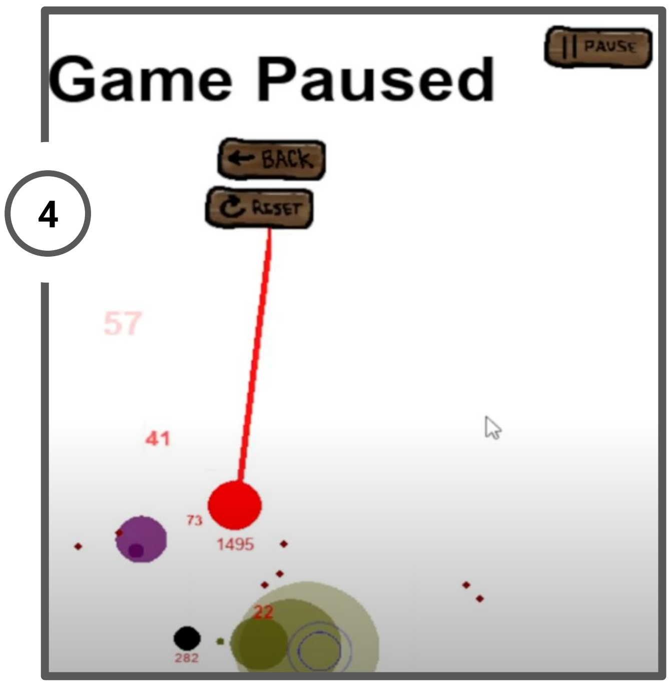
- Python
- Pygame
- Photoshop

Game Demo

Purpose

A two person mini project created together with PyUnityd Co-president as a coding example and inspiration source for PyUnityd club members during a hackathon themed after Breaking the Fourth Wall

The game involves an arcade-style boss fight where the boss uses the pause and reset menu buttons to their advantage. The boss also occasionally traps your mouse pointer in a specific location



Captions

1. Boss aware of game being paused
2. Boss traps mouse cursor
3. Boss destroys pause button
4. Boss knocks reset button out of pause menu so that the player has to avoid it

Contribution

Wrote the entire game with the exception of the boss’ shooting AI. Also designed and created button assets in Photoshop

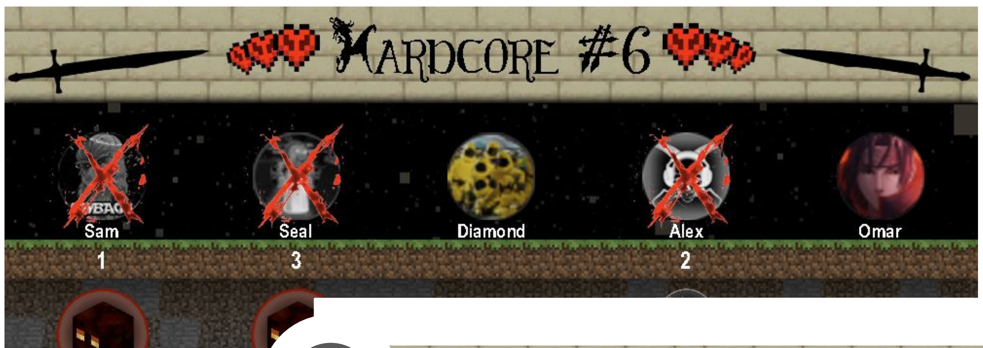
Challenges

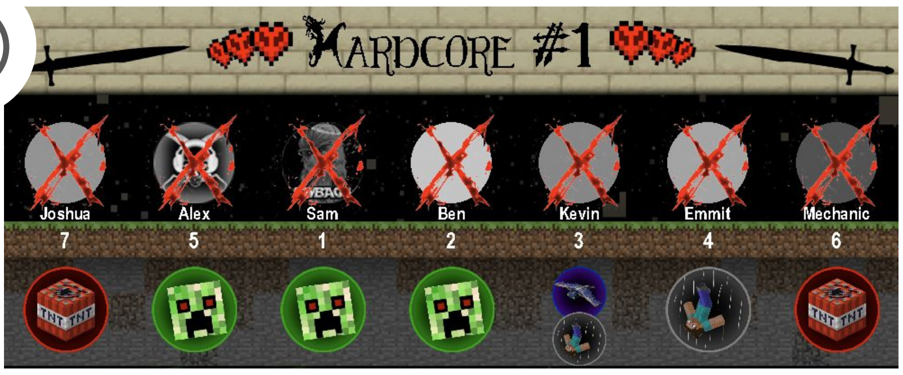
Using Pygame meant that the speech boxes, pause menu, and other key features had to be implemented from scratch.

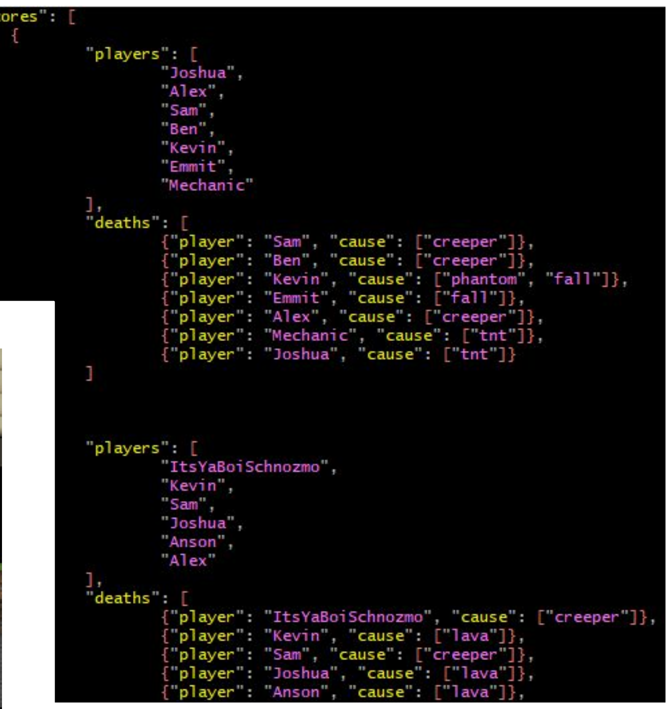
AlekseyPanas/BreakingFourthWall

Various Small Projects

2019 - 2022







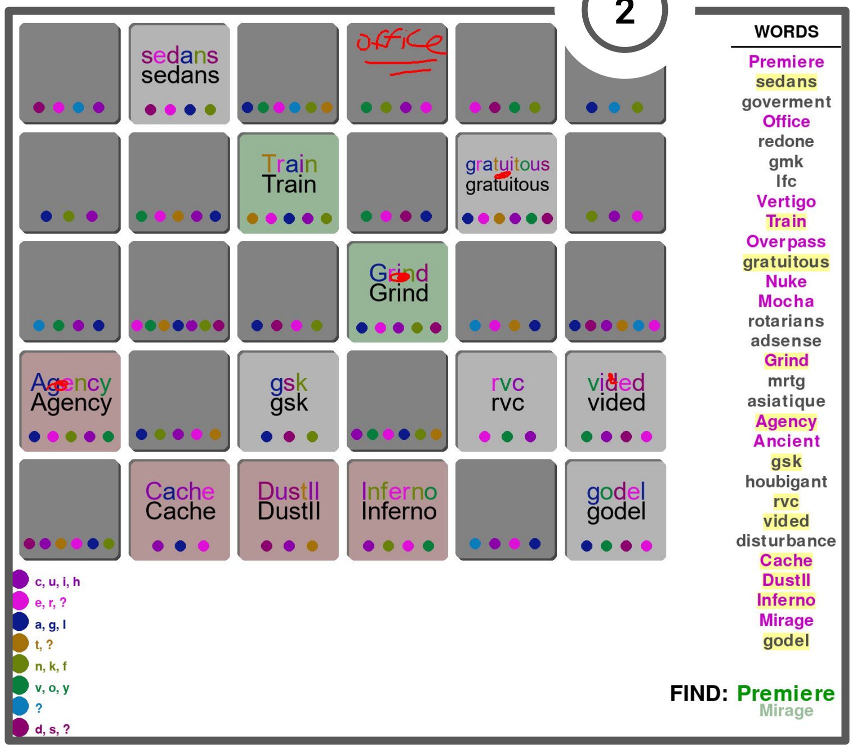
```
{"hardcores": [
        {
                "players": [
                        "Joshua",
                        "Alex",
                        "Sam",
                        "Ben",
                        "Kevin",
                        "Emmit",
                        "Mechanic"
                ],
                "deaths": [
                        {"player": "Sam", "cause": ["creeper"]},
                        {"player": "Ben", "cause": ["creeper"]},
                        {"player": "Kevin", "cause": ["phantom", "fall"]},
                        {"player": "Emmit", "cause": ["fall"]},
                        {"player": "Alex", "cause": ["creeper"]},
                        {"player": "Mechanic", "cause": ["tnt"]},
                        {"player": "Joshua", "cause": ["tnt"]}
                ]


                "players": [
                        "ItsYaBoiSchnozmo",
                        "Kevin",
                        "Sam",
                        "Joshua",
                        "Anson",
                        "Alex"
                ],
                "deaths": [
                        {"player": "ItsYaBoiSchnozmo", "cause": ["creeper"]},
                        {"player": "Kevin", "cause": ["lava"]},
                        {"player": "Sam", "cause": ["creeper"]},
                        {"player": "Joshua", "cause": ["lava"]},
                        {"player": "Anson", "cause": ["lava"]},
```

1

Purpose

Various small solo Pygame projects completed over the years

2

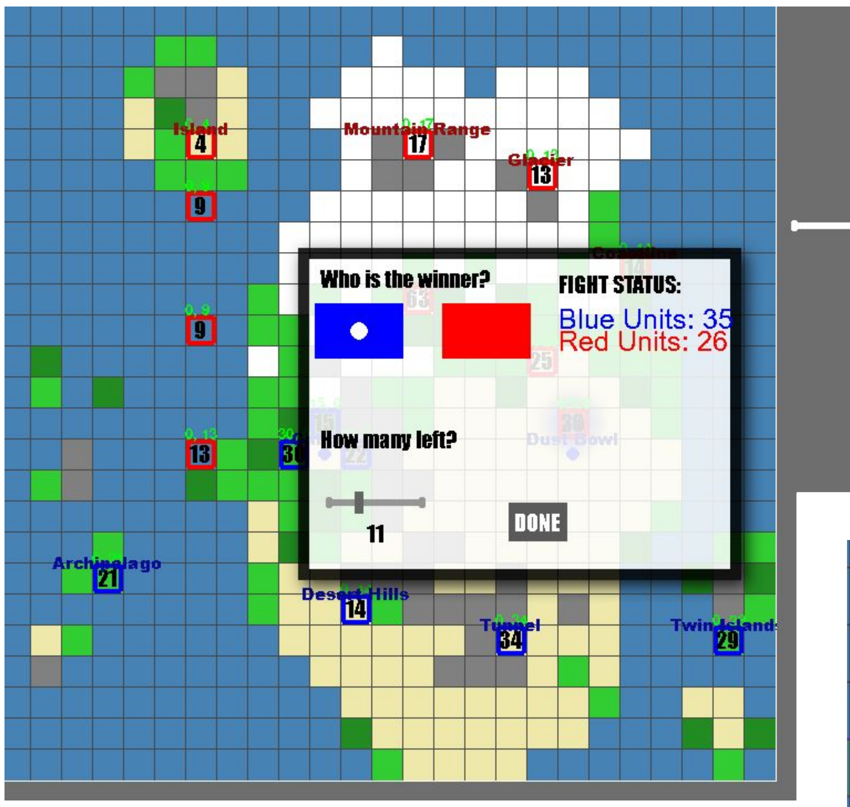


Captions

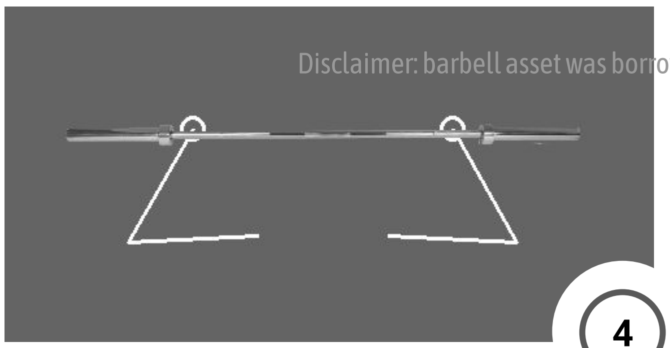
1. **[2020]** Converts **JSON** death logs from Minecraft Hardcore playthroughs hosted with friends into a visualization (using **OpenCV** for image filters and **Pygame** for drawing and exporting)
2. **[2022]** Puzzle game initially created to make map choosing more fun when playing Counter Strike with friends. Purple words from the word list cycle at the bottom right with each revealed tile and will turn green if revealed while prompted. Colored dots help indicate letters and their order for hidden word tiles.
3. **[2019]** A small top-view turn-based combat game where you control units and upgrade bases
4. **[2020]** 2D inverse kinematics where the arms deform to reach specified locations.

Technologies

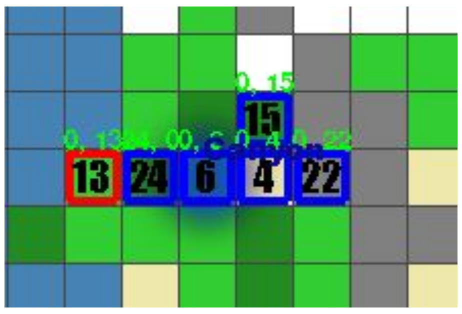
- Python / Pygame
- JSON
- OpenCV



3



4



AlekseyPanas/Hardcores

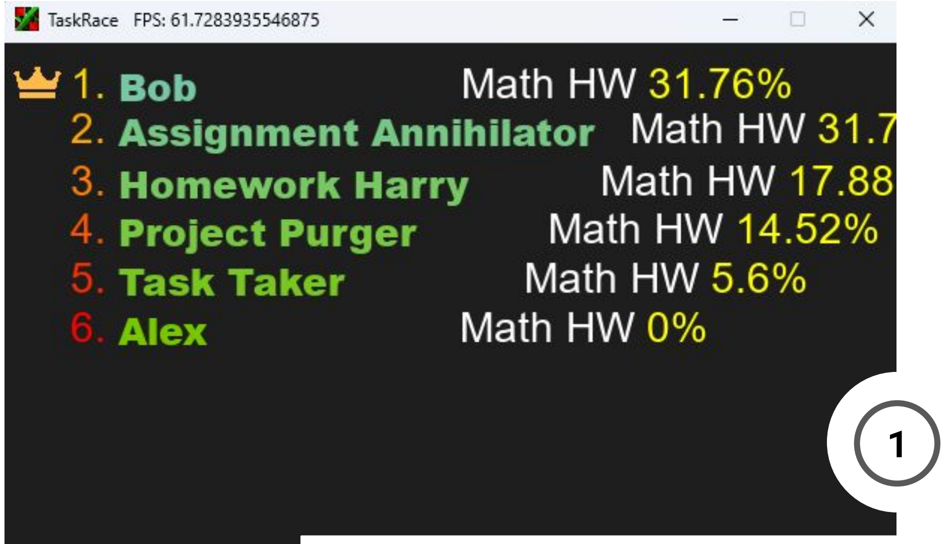
AlekseyPanas/RavenfieldSim

AlekseyPanas/LiftingSimulator

AlekseyPanas/CSMapChooser

Task Race

2020



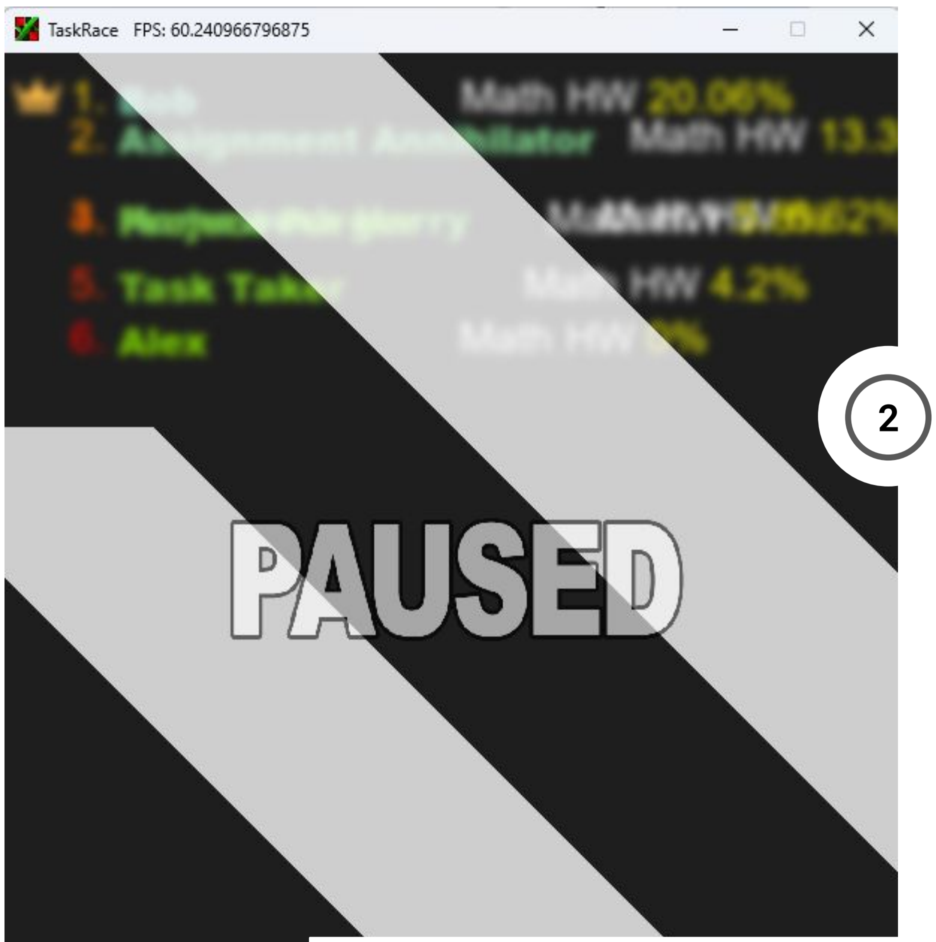
```
TaskRace   FPS: 61.7283935546875
1. Bob                        Math HW 31.76%
2. Assignment Annihilator     Math HW 31.7
3. Homework Harry             Math HW 17.88
4. Project Purger             Math HW 14.52%
5. Task Taker                 Math HW 5.6%
6. Alex                       Math HW 0%
Time: 0:05.31
```

1

Purpose

Implemented as a solo project based on an idea from childhood, TaskRace helps motivate doing mundane tasks by turning it into a race.

After inputting a set of tasks and their estimated time of completion, the app launches with a set of opponents who are supposedly doing the same tasks in some alternate universe, and you are competing against them. The premise is to get chores done faster *(works from experience)*



```
TaskRace   FPS: 60.240966796875
1. Bob                        Math HW 20.06%
2. Assignment Annihilator     Math HW 13.3
5. Task Taker                 Math HW 4.2%
6. Alex                       Math HW 0%
PAUSED
```

2

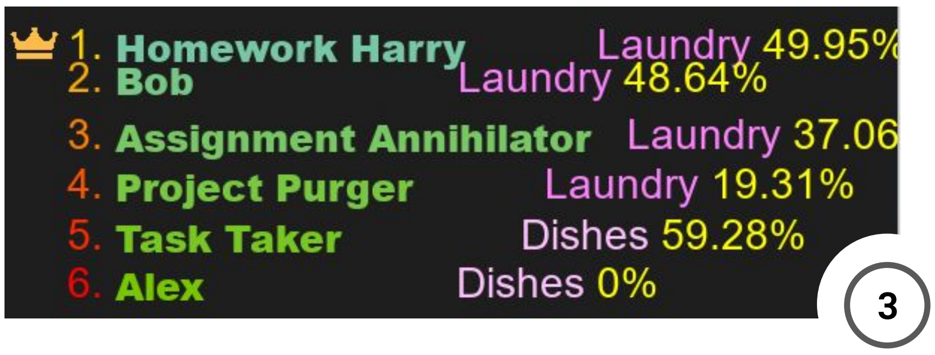
Captions

1. Main app screen showing active percentages
2. Blur effect on pausing
3. Animated position switching

AlekseyPanas/TaskRace

Technologies

- Python
- Pygame
- PIL (Pillow)



```
1. Homework Harry           Laundry 49.95%
2. Bob                    Laundry 48.64%
3. Assignment Annihilator   Laundry 37.06
4. Project Purger           Laundry 19.31%
5. Task Taker               Dishes 59.28%
6. Alex                     Dishes 0%
```

3

What's Ahead

While I have already extensively used this app personally, there are plans to expand it to polish it as a finished product:

- Port to a web app
- Add a more sophisticated settings menu for configuring tasks and opponents
- Add matchmaking to go up against real opponents who have a task list of similar estimated length
- Add auto time estimation for common tasks

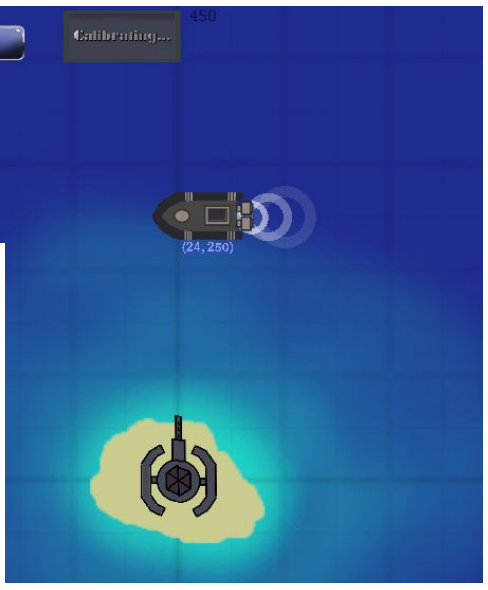
Polar Defense

2019

Purpose

Submission for an open-ended extra credit assignment for a calculus class in high school with a task to develop an educational piece of media on a subject from the curriculum

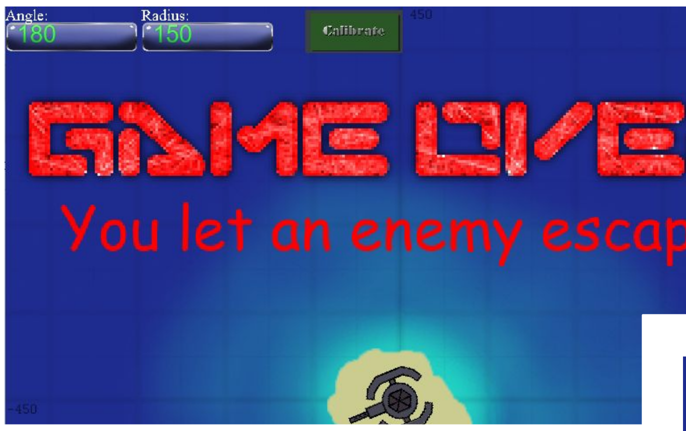
Choosing polar coordinates as the topic, this small game was developed, requiring the player to convert cartesian coordinates to polar to fire the turret at various targets.

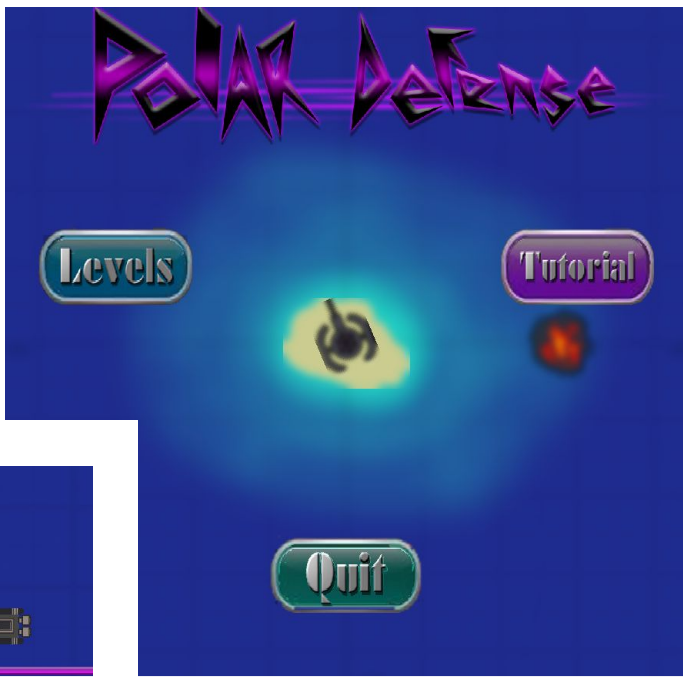


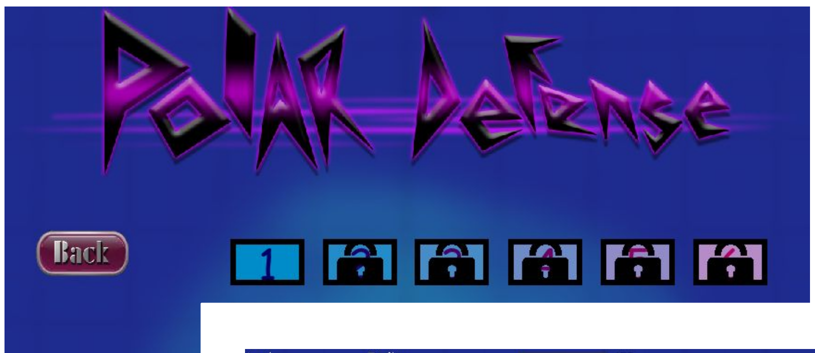
Disclaimer: Explosion, level complete, and game over assets borrowed

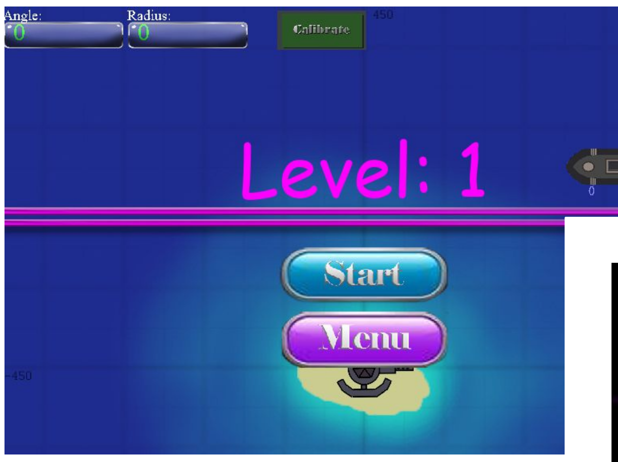
Technologies

- Python
- Pygame
- Photoshop

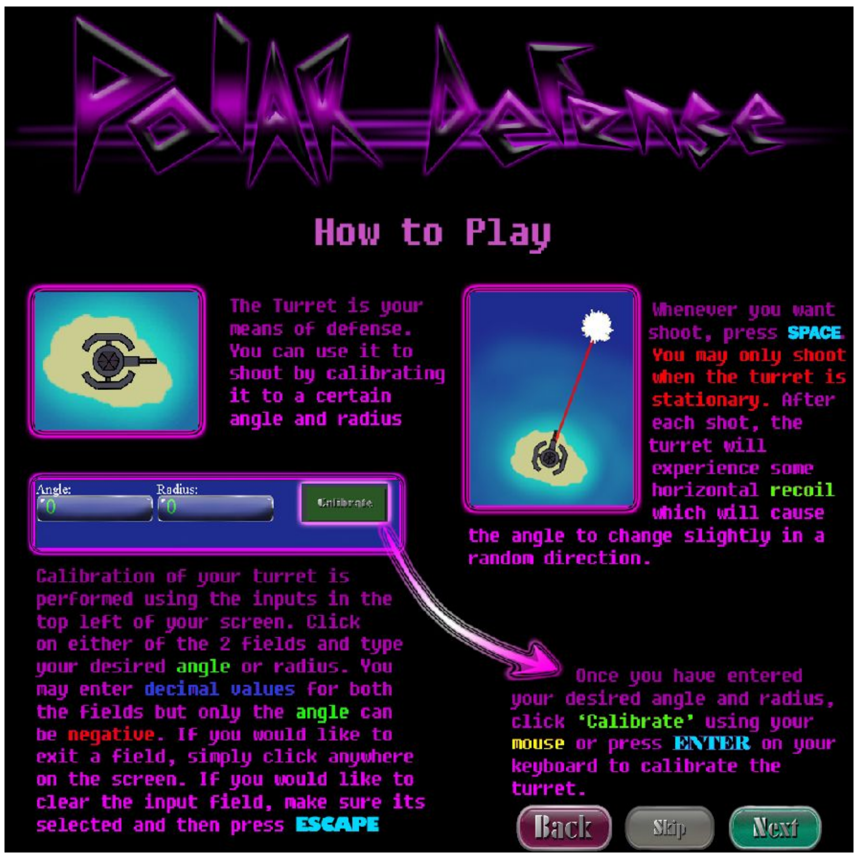








AlekseyPanas/
PolarDefense.git



How to Play

The Turret is your means of defense. You can use it to shoot by calibrating it to a certain angle and radius

Whenever you want shoot, press SPACE. You may only shoot when the turret is stationary. After each shot, the turret will experience some horizontal recoil which will cause the angle to change slightly in a random direction.

Calibration of your turret is performed using the inputs in the top left of your screen. Click on either of the 2 fields and type your desired angle or radius. You may enter decimal values for both the fields but only the angle can be negative. If you would like to exit a field, simply click anywhere on the screen. If you would like to clear the input field, make sure its selected and then press ESCAPE

Once you have entered your desired angle and radius, click 'Calibrate' using your mouse or press ENTER on your keyboard to calibrate the turret.

Contribution

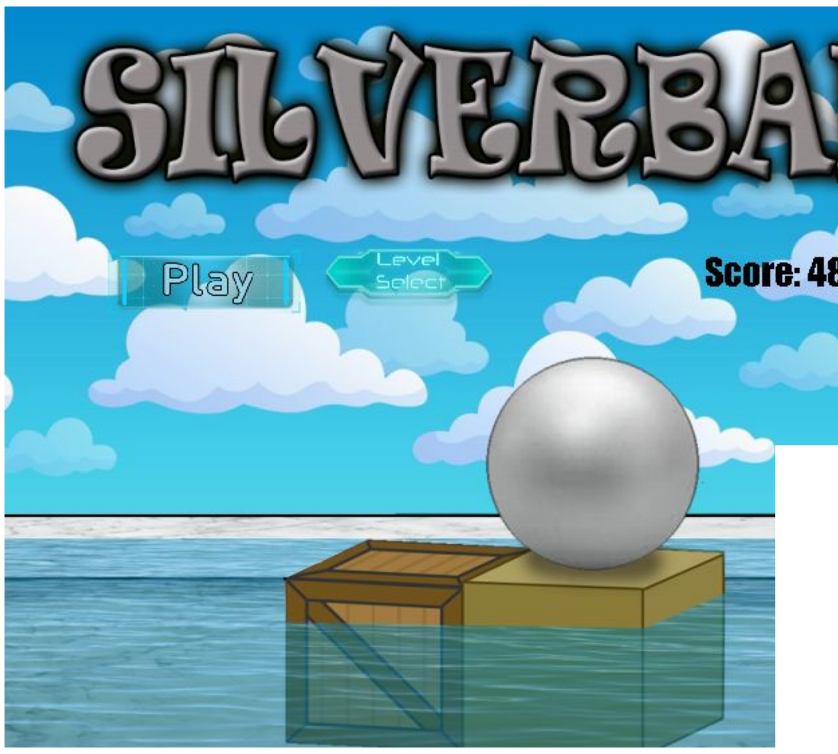
Wrote all the game code while a partner created the boat, turret, and buoy assets.

The remaining assets, including the tutorial screen, were done collectively

Silverball

Game Demo

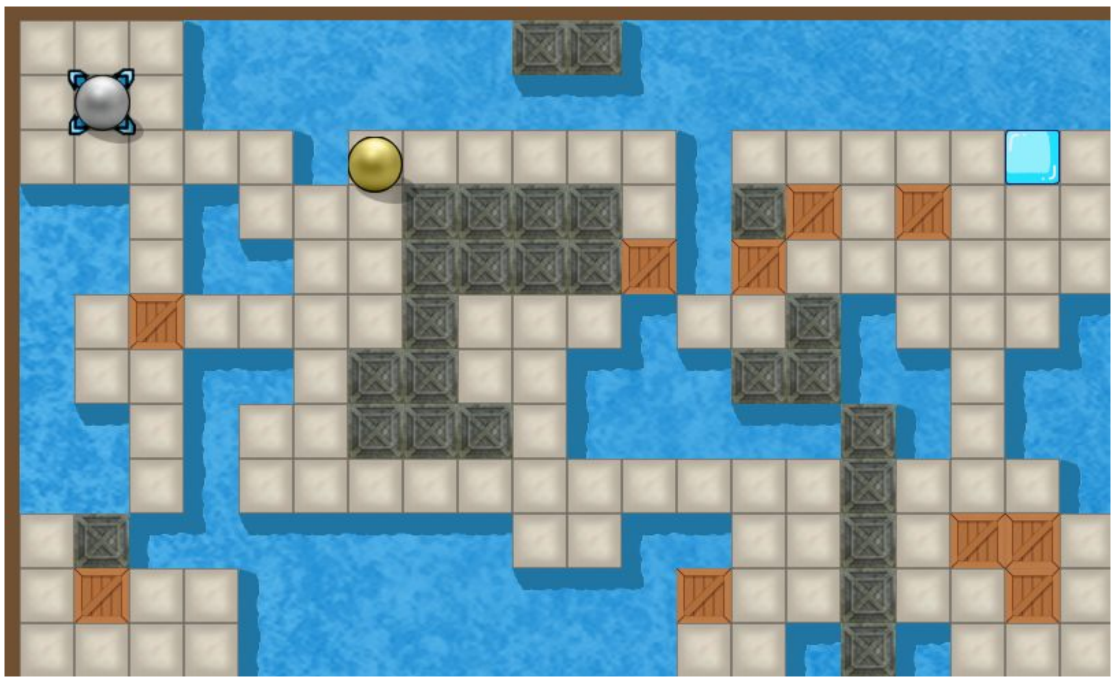
2018

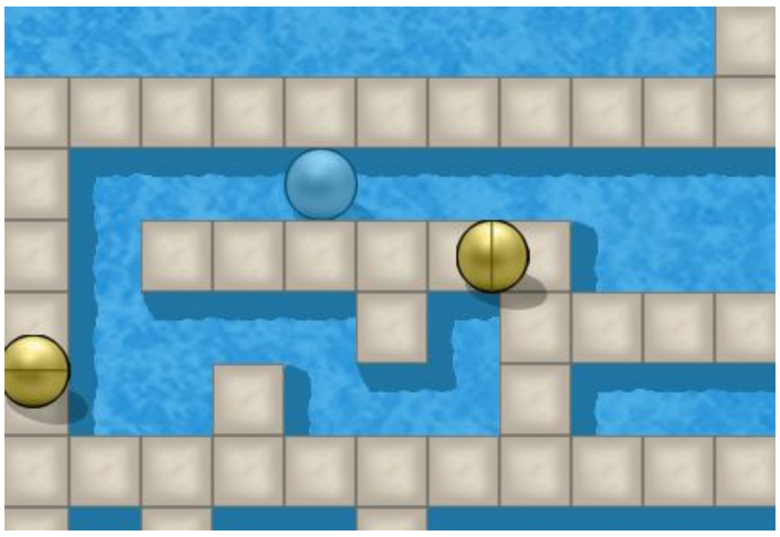


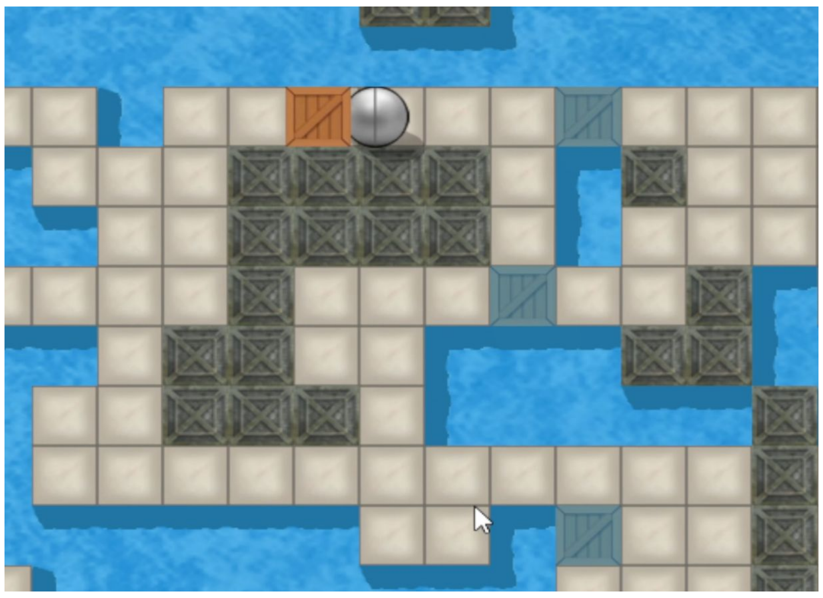
Purpose

Clone of a Miniclip flash game called "SilverSphere" developed as a birthday present for my mom since she enjoyed that game and could no longer play it once flash got taken down.

This solo project features new levels and a simple level editor over the original game





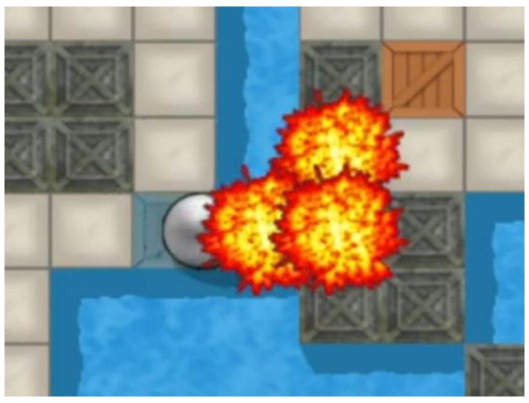


Disclaimer: Majority of assets were borrowed from google images or the original SilverSphere Miniclip game

Technologies

- Python
- Pygame
- Photoshop

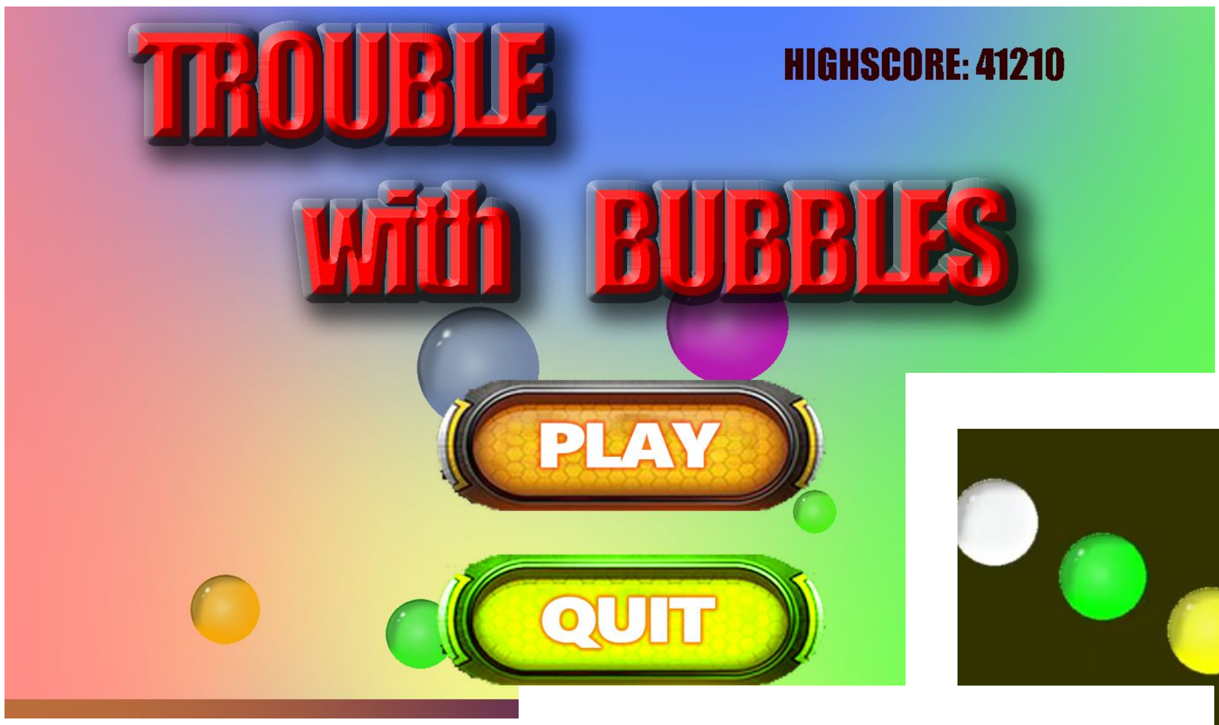
AlekseyPanas/Silverball

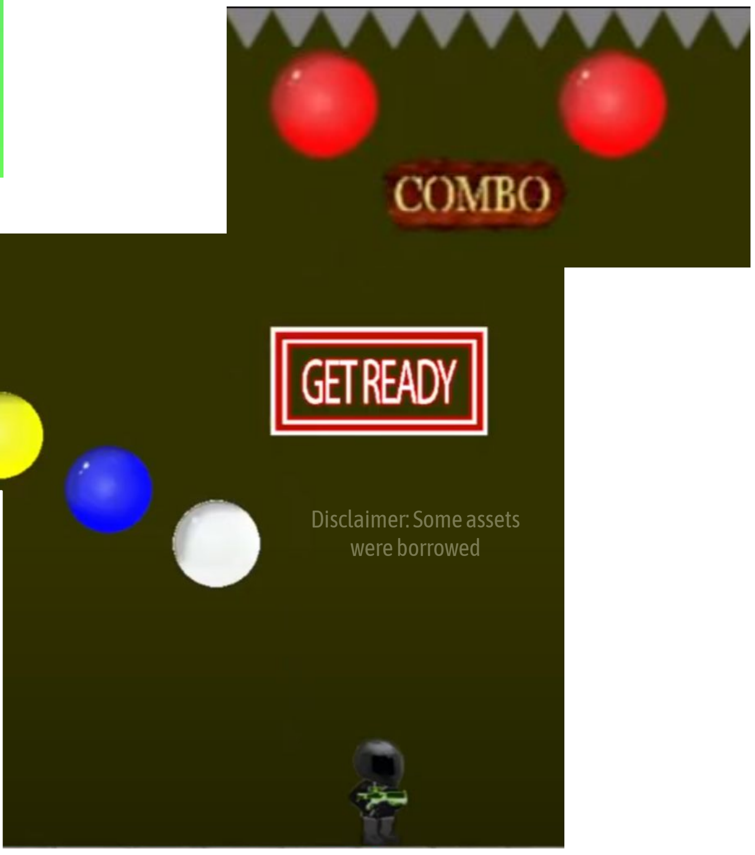


Trouble with Bubbles

Game Demo

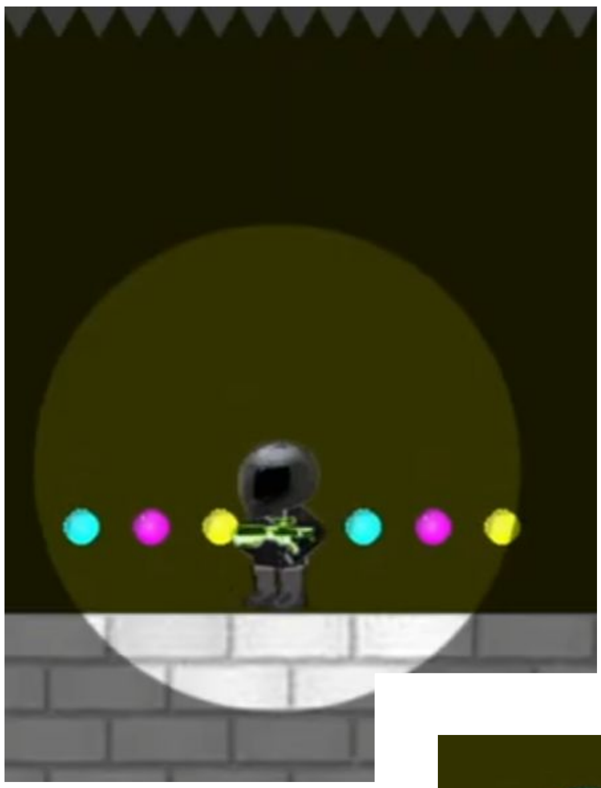
2017





Purpose

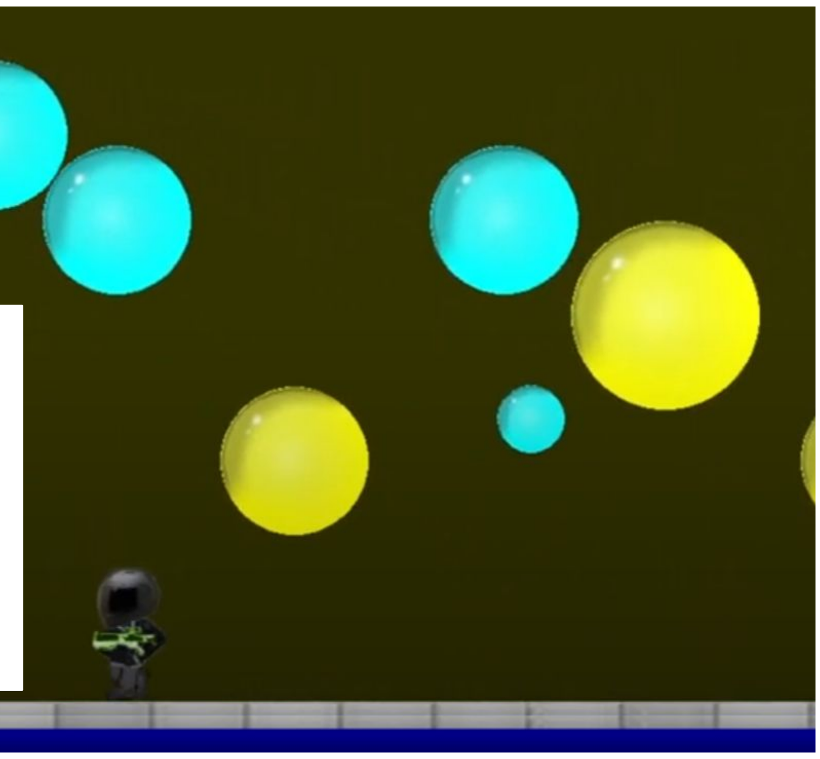
Clone of a popular miniclip flash game called "Bubble Trouble" created as a solo personal project because I beat the original game and wished to make additional more challenging levels

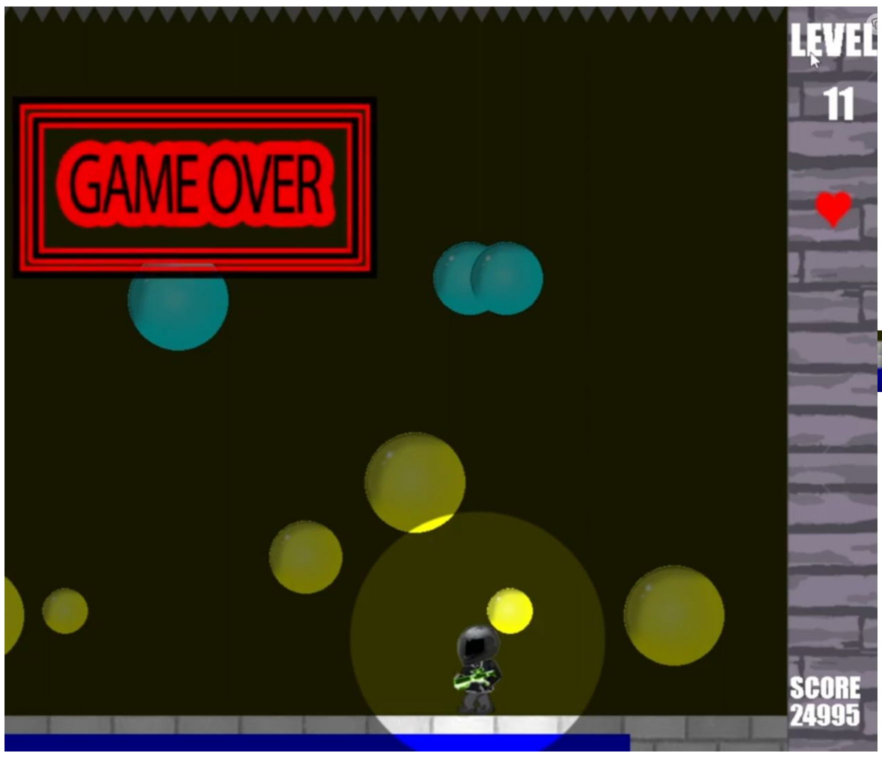


AlekseyPanas/ TroubleBubbles

Technologies

- Python
- Pygame
- Photoshop




GAME OVER
LEVEL
11
SCORE
24995