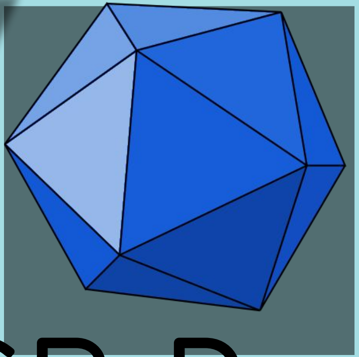


RGB Rush

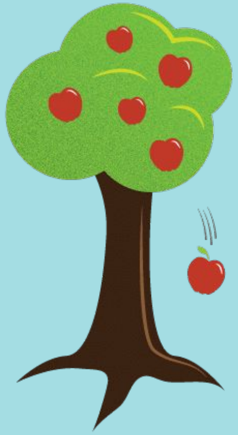
Victor, Jonathan, Claire, and Georgia

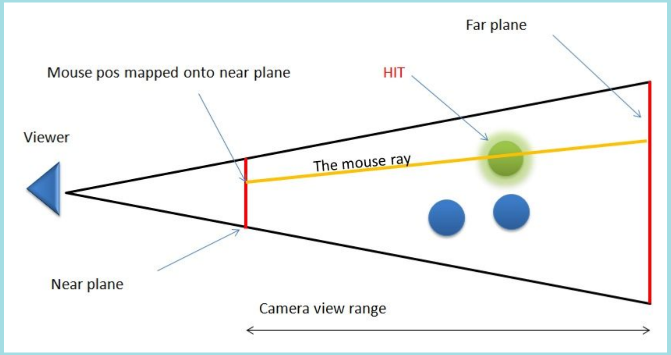
Storyline of our Game!

- Childhood
  - Piano Tiles and Subway Surfers
- 3D Obstacle Course
  - Custom Meshes, Spheres, Cubes, Cylinders, etc
- Key: Color switching!
  - Allows bypassing obstacles of same color

Our Features

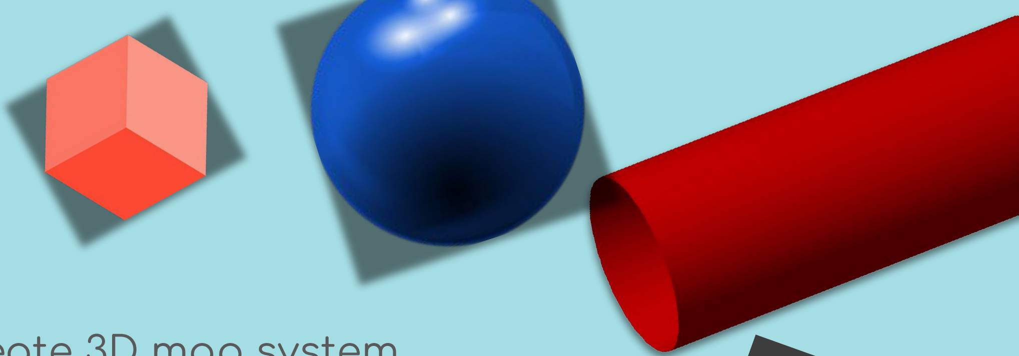
- Obstacles
- Mouse Picking
- Collision Detection
- Shadows
- Physics (Jumping, Gravity)
- First Person Camera
- Text Display
- Interactivity

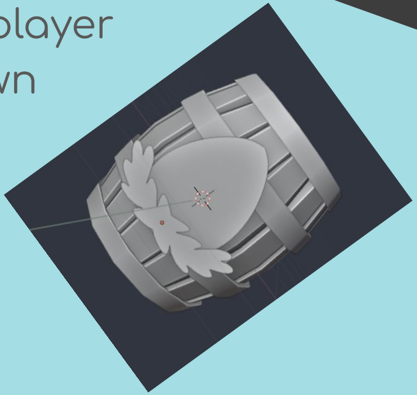


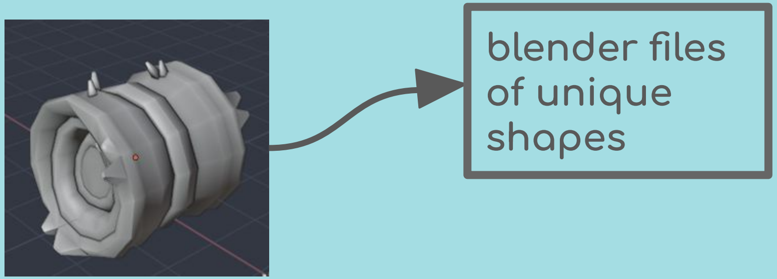


Map/Obstacles

- Infinite Skybox Map
  - Integrated tiny graphics to create 3D map system
  - Moving trees and lanes to give illusion of player movement
- 6 main **obstacle** types—cube, sphere, cylinder, hay, tv, barrel
  - Objects dynamically spawn and move towards player
  - After 5 seconds, objects are deleted and respawn

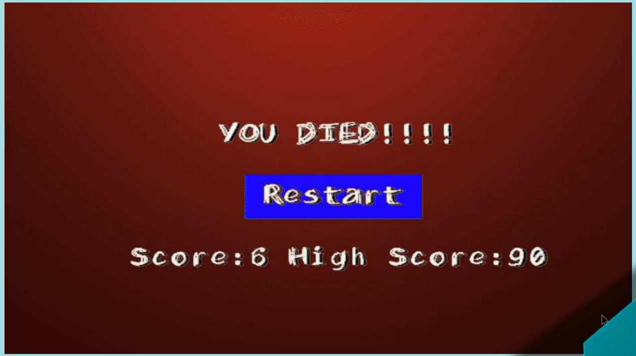






Mouse picking, Player

- Creating a world space ray based on user mouse
  - Normalization
  - Change into Projection -> Eye -> World Space
- Finding intersection with the lenses and restart button
  - Extend ray to lenses
    - Use x and y to confirm intersection
- Player
  - Sliding using interpolation
  - Jumping using vertical velocity
  - Falling using gravity and delta time

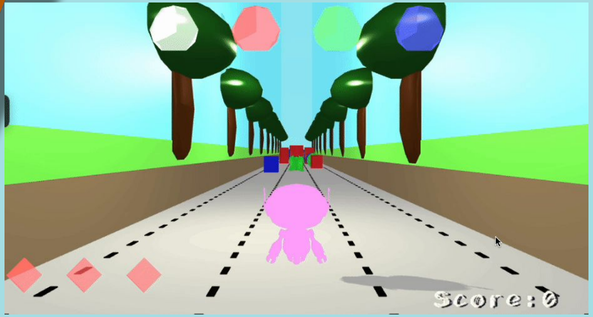

YOU DIED!!!!

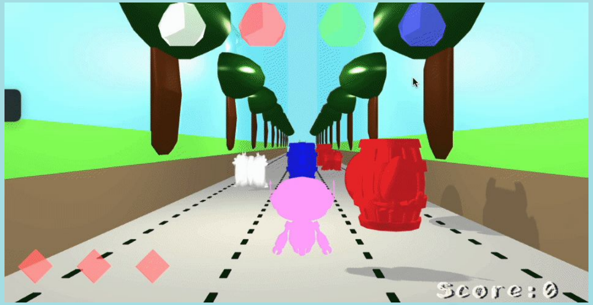
Restart

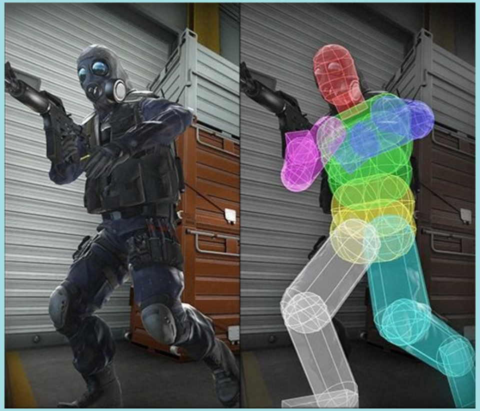
Score:6 High Score:90

Collision Detection

- FPS Games
  - Hitbox
- Method: Axis-Aligned Bounding Box (AABB)
  - Min/Max corner
  - Rectangle Hitbox
  - Non-Spherical
- Spherical Collision Detection
  - Radius
- Applies
  - Jumping
  - Player vs Obstacles

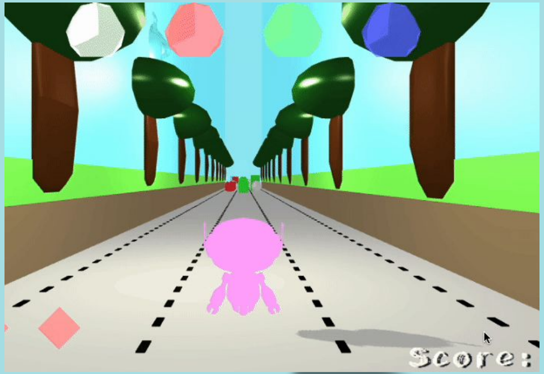






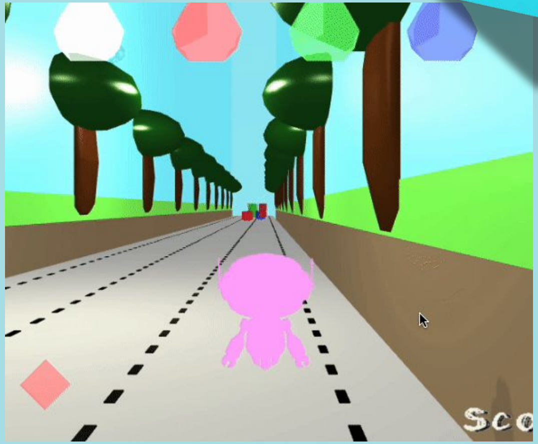
Shadows

- Credit link (Robert Lu) - https://github.com/Robert-Lu/tiny-graphics-shadow_demo/blob/master/examples/shadow-demo.js
- Light's perspective
  - Do a first pass without displaying shadows
- Camera's perspective
  - Do a second pass and display shadows
  - to be able to see if area should be in shadow or not
  - logic pulled from lecture (Two Pass Z-Buffer)
- Have a global texture initialization and modified the the game's architecture in order to support the above

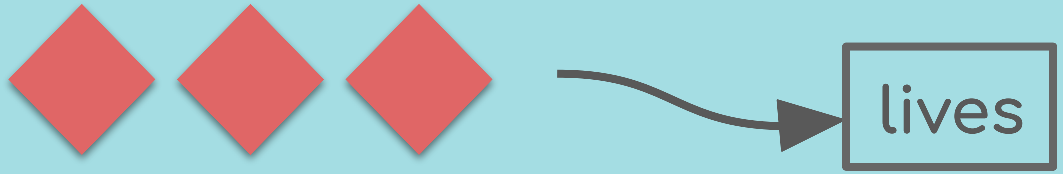


Physics - Jumping

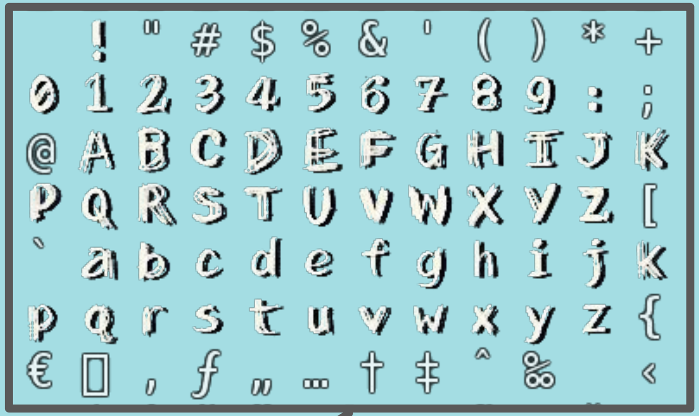
- Each frame
  - if player jumps, set a <u>vector velocity</u> value instantaneously
- Each second
  - apply acceleration due to gravity (-10)
  - using delta time
- Simulated
  - Background moves
  - When ground is hit, set **landed** to true



improved
Playability



- **Lives:** decrement when obstacle is hit
- **Score:** increment when player passes obstacles
  - cubical: +1, spherical: +2, tall: +3
- DeathScreen
  - appears when all lives are lost
  - shows all-time **high score** & **score**
  - uses text-demo (text.png file was re-drawn for a *scary* font)
- Restart button
  - also uses mouse picking to detect if player restarts, which resets game



Challenges

- Giving the illusion that moving dashes and trees are infinite
- Collision Detection for Non-Spherical vs Spherical
  - Accuracy
- Shadows Implementation
- Text representation on screen

If we had more time...

- We actually implemented everything we wanted to in our midway demo!
  - <u>**With more time, we would:**</u>
    - make the thematic elements make more sense
    - further fix the aliasing issue in movement
    - make it look more obvious that you've been hurt when you hit an obstacle

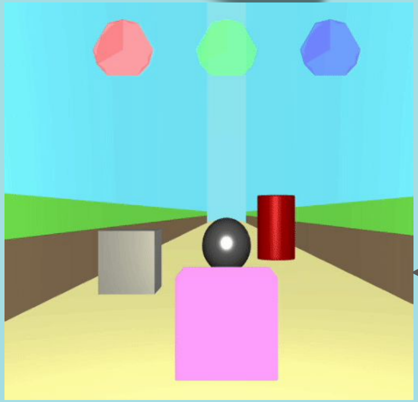
DEMO

Conclusion

A playable, interesting game!

- Lenses as “power ups”
- Player has lives
- Can restart the game
- Has a high score and score

Thank you!



How our game looked at midway demo!