Group 5 Presentation 1

Alex, Terry, Matt and Dylan

Game Premise: Horror! Walking! Simulator!

- First Person PointerLock Controls
- Navigating a creepy environment.
- Have to get collectibles.
- A monster is chasing you.
- If it catches you, you’re DEAD!

The Game Mechanic Twist

Twist, Continued

- The monster can’t move when it is being observed.
- Music gets louder and footsteps get faster as monster approaches you.
- When you look at the monster, it freezes and the music stops.
- This creates something of a pseudo-rhythm game, where the speed of footsteps and volume of music indicate to the player how much time they have to turn to observe the monster.
- Player has to balance navigating environment and turning to find the monster. Could become quite challenging when there are multiple monsters or other constraints (e.g. having to blink).

Inspirations

- Low poly 2000's era browser horror games
- Condemned: Criminal Origins
- Indie horror games like The Path
- Clock Tower

Working Theme

You are this guy :

Running from this guy:

What We've Done So Far

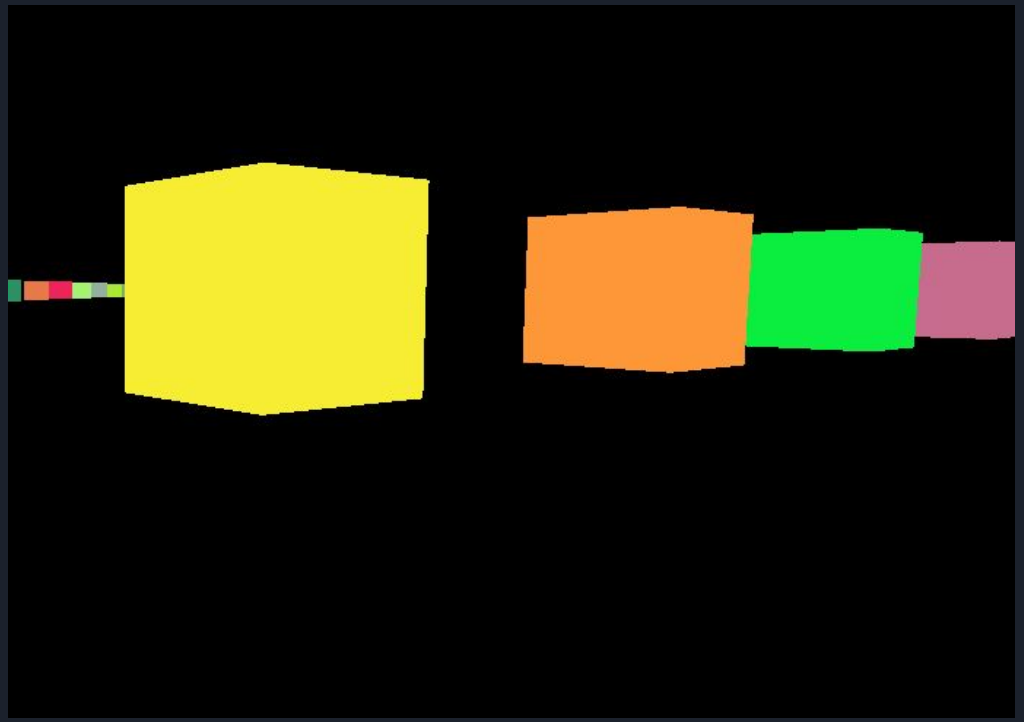
- Implemented Pointer Lock Controls. Borrowed Liberally From this Source Code (Show Demo): https://codepen.io/Gowther/pen/voWgvN

- This code gave us a great framework to start with but had a big issue we needed to solve. (Automatic movement based on "velocity" calculations.)

- So we had to make a lot of changes, mainly to the animate function.

What We've Done So Far

- Filled Environment with a "terrain" of randomly colored cubes. So far they are just there for testing moving around/perspective, but they will also help us figure out collision detection (so you can't just walk through them).



What We've Done So Far

Created a rotating red cone that loves you so much it never stops chasing you. This was to figure out the basic mechanics of making an object follow the player. Eventually it will be replaced by a monster that hates you so much it never stops chasing you.

```
function coneChase(){
    var player = new THREE.Vector3();  //These two lines create an empty vector and copy the current position of the player into it,
    camera.getWorldPosition(player);   //so we can make something chase the player.
    cone.translateOnAxis(cone.worldToLocal(player), .01);
}
```

Lighting & Materials

Light Source: PointLight

Material: MeshLambertMaterial (non shiny material that reflects light)

Flashlight: created camera > created the light > attached light to the camera

```
camera.add(light);
```

What next?

- Models
- Sound bites/music cues
- Dynamic lighting (realistic outdoor lighting for windows, flickering lights, shadows)
- Raycasting for Object Collision and Monster Freezing