

Alpha Release

# Doomsday: Underground Uprise

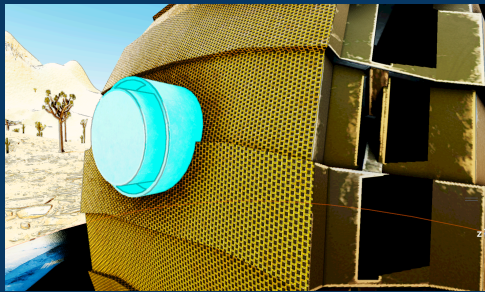


**Technische Universität München**

**Winter Semester 2023/2024**

Games Laboratory

Matija Jajcinovic Hongbo Chen Haorui Tan



## Summary of our Results

We have implemented all key-aspects of our games (Layer 3) which are:

- Buildsystem in the Underground Base
- Tower Defense on Surface
- Physically based Animations for Robots
- In our opinion having a relatively good looking game (which has to be verified by playtesters)
  - by using HDRP
  - custom detailed models
  - using custom shaders
  - using Robotics and Machine Learning for interactive animations

We think that the game stands out by its unique mix of genres and love in the details;

The high effort custom models and interactive animations are definitely a highlight + the base building logic that is both simple and challenging to manage.

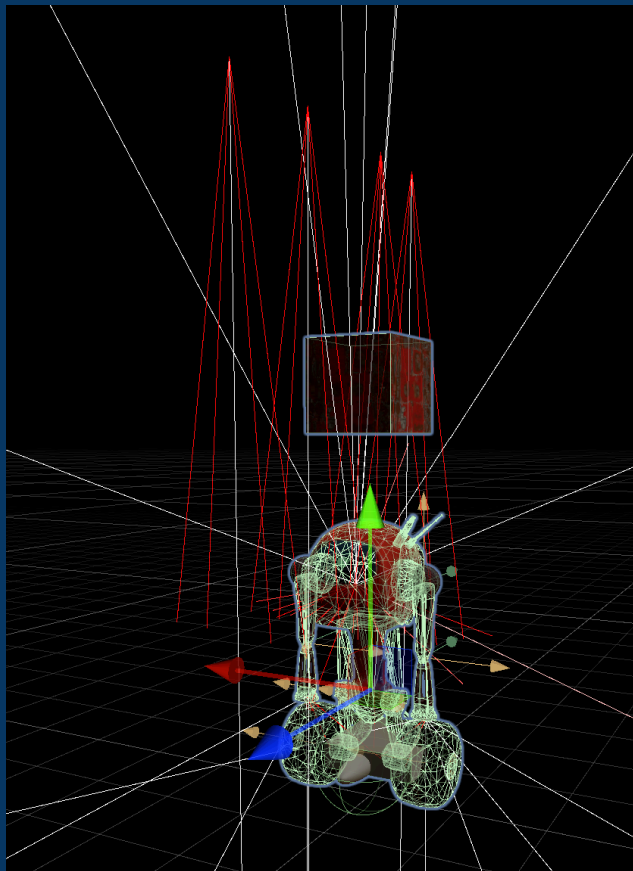
## Worker Robot



The Worker Robot is for now the only ML trained entity in our game (self made 3D model). As a pendulum robot the task of moving while balancing is a tricky problem; the rotation of the tires introduces a torque onto the upper body which can only be counteracted by having the center of mass higher; for movement it has to lean forward to use the gravity for counteracting the mentioned torque. Additionally the terrain has many bumps/is uneven.

The ML uses PPO + LSTM and after 100 million iterations (40 hours) it trained well to move on the terrain. The robot learned to use

its arms as a counterbalance and rarely falls. For the cases it does fall we introduced a PID-controller that only acts if needed (Dot product).



## Task

The robot has to go to a crate/box dispensed by a farm/factory, the box gets picked up automatically by a script, and then carry it to the base.



## Training

The training process was very tedious:

1. Inputting the global position introduced overfitting which causes failure when the terrain was altered at one point by mistake
2. Fails to climb to high mountains bc falling means a big minus in long time reward -> safer to just try to climb slowly
3. Too weak or strong motors caused inefficient learning
4. Sometimes the physics exploded causing the whole training to crash

A similar process was used to make it balance and carry a box glued to its arm. Unfortunately due to some bugs in the transfer of the trained Model to the gameplay scene the box carry model was not used (but it trained well, the robot moved backwards).

## Physically Based Animation - Light Combat Robot

The main feature of this robot is that all moving parts are made with Rigidbodies and move only with force and torque produced by PID controllers.

The gun is made out of several pieces that deform on shooting (kind of gun flexing).

The guns have both vertical and straight recoil, making the robot shooting process visually very nice and naturally looking.



The light attack robot is built in the underground base and dispensed by the player at the surface by opening the build menu (TAB + 2). The Robot chooses the nearest target automatically and shoots with two guns at it.

For now only one Robot and one gun is supported, our high goal is to introduce more types of units and weapons but we decided to make one robot and gun well.

## Crop Farm and Factory

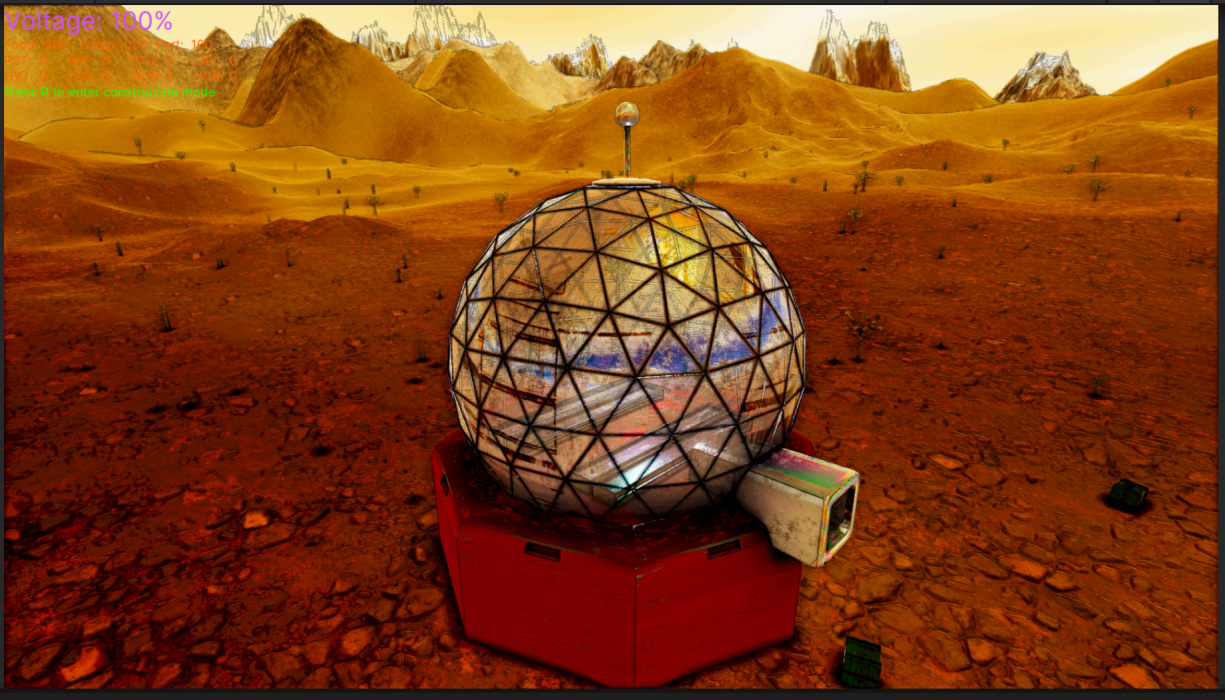
Both dispense boxes every x seconds, which then get automatically assigned to the dedicated worker.

To simplify both the gameplay and programming every factory and farm spawns one dedicated worker which cannot be destroyed.

The farms are being built by the player at fixed positions; the player can choose if they need Food or Scrap more.



lay 1 Full HD (1920x1080) Scale 0.95x Play Focused





## 3D-UI

3D UI are just 3D objects with colliders; Advantages are that they interact with the scene's light and "feel" different when using.

Disadvantages are the problematic adaptation to different screen ratios and worse performance.

### Switch Button between Surface and Bunker

The switch button has an animation for hovering over it and pressing it. It's used to quickly switch between the two scenes, memorizing the last position it was at each base, using a light to showcase which one is currently on.

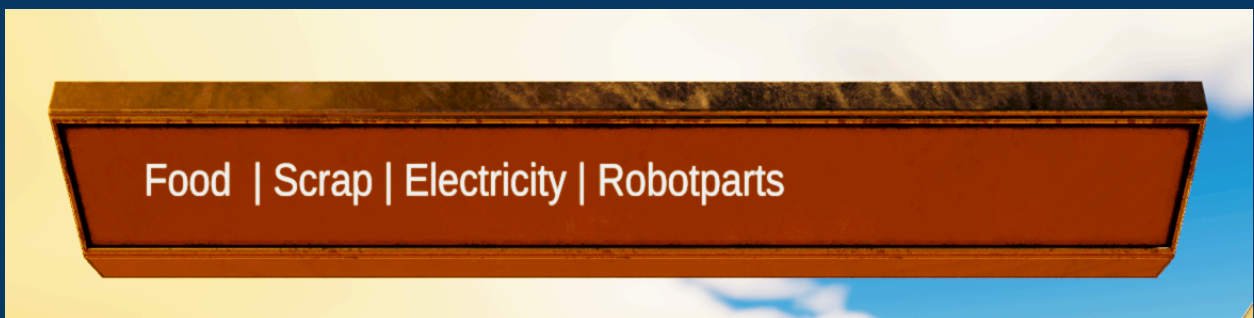
The model is self made and textured (except the image).





## HUD

The rest of the information is displayed simpler with plain text.



## Custom Shaders and Graphics

We designed with Shader Graph a couple of shaders, using shader graph and HDRP.

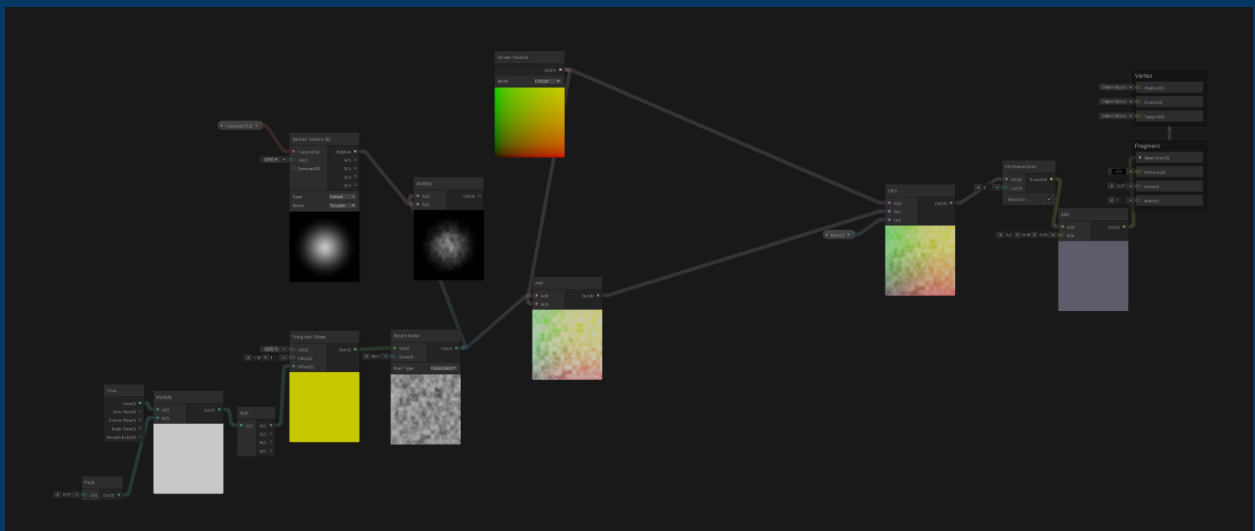
## Cartoon Shader

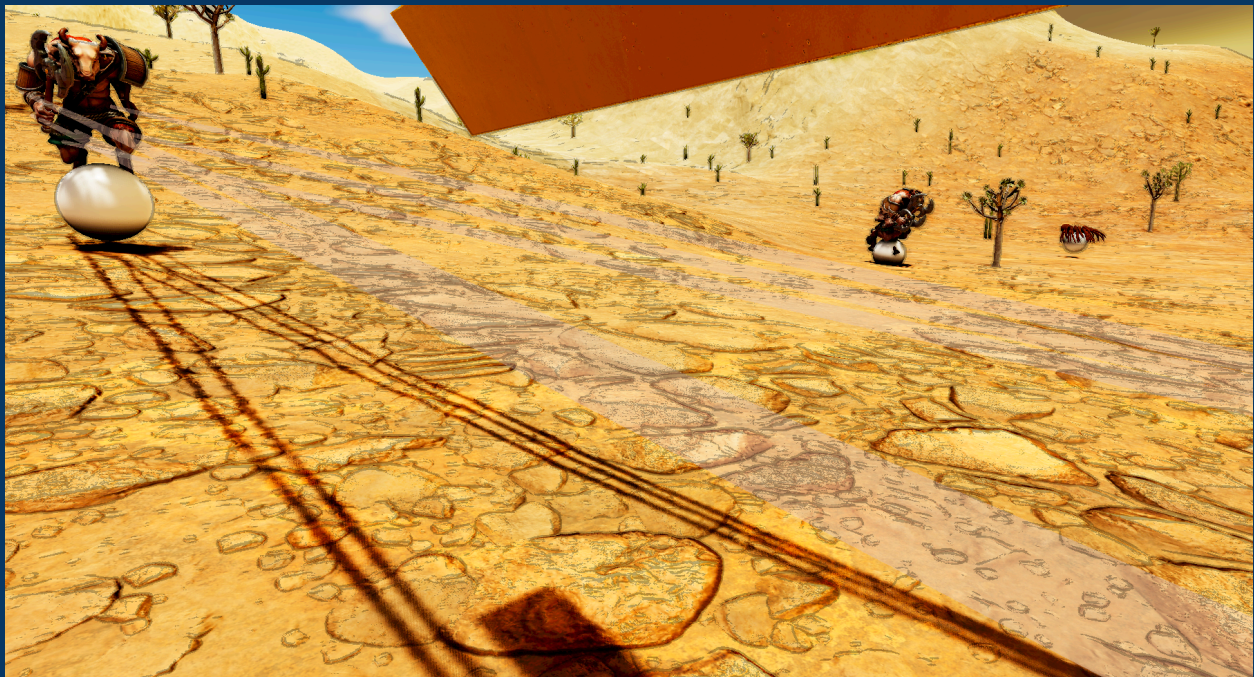
It is a very simple post processing shader using Robert Cross Edge Detection to give some cartoony vibes. Setting it up took hours due to bad documentation on how to use custom post processing shaders.



## Gun Trace

To see where the guns of the robots are shooting we made a simple transparent white noisy shader that despawns after one second. The gun's flash is an Unity made asset.



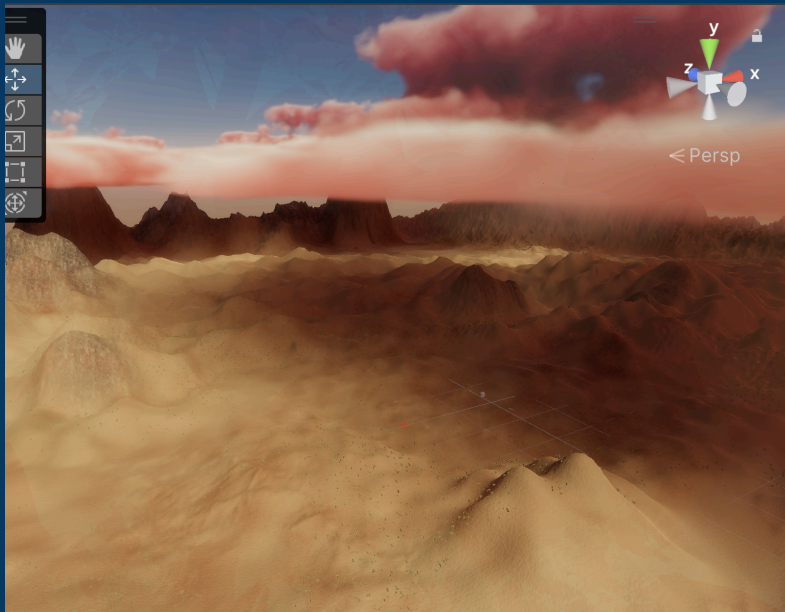


## More Detailed terrain

We have expanded the first draft of the terrain; the enlargement of the site has contributed a lot to the visual fidelity bc a too small terrain just didn't had any natural bounce;

To boost performance we used camera occlusion baking and reduced the amount of vegetation (bought assets).

Another problem that we encountered was that git does not support terrains (declared as .asset); a first fix was to manually copy it. We tried to solve the performance problem with an Asset that uses Imposters (ML algorithm that makes a low poly LOD) but it caused light bugs in our scene. Also Baking together multiple Objects to reduce the render calls did not help bc terrain do it already it seems.



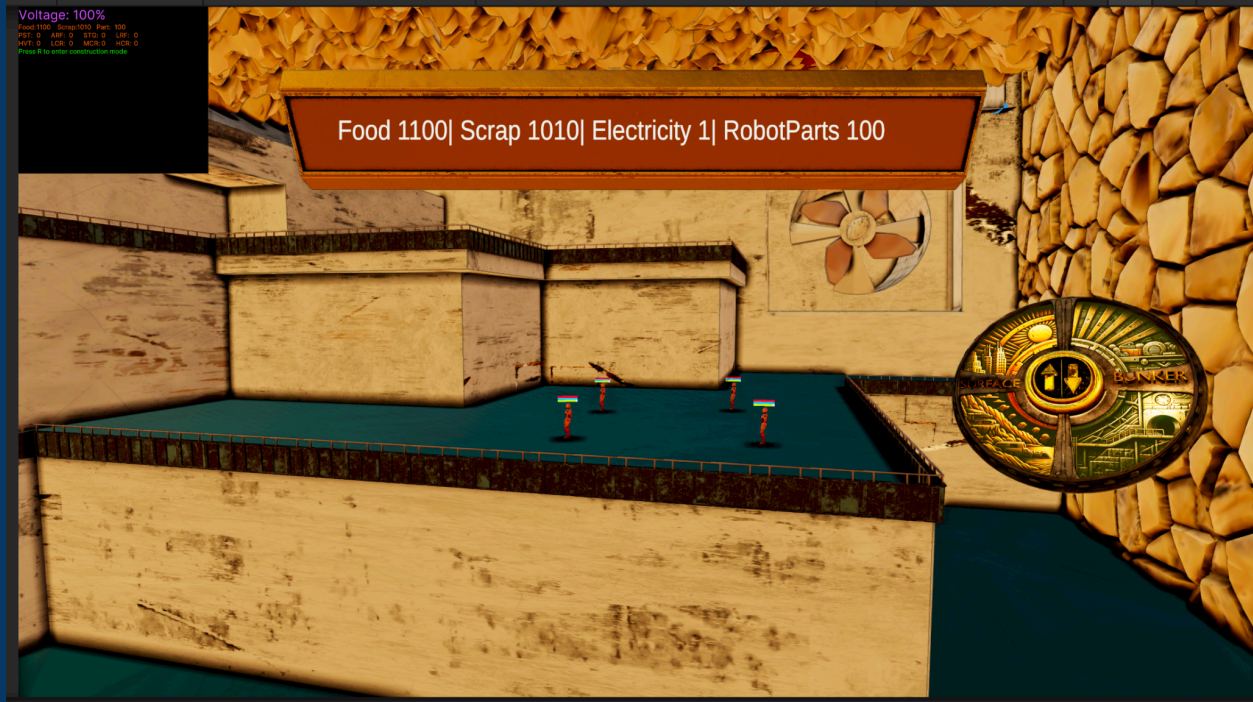
## Volumetric Clouds

To get a more dramatic scenery we used Unity's built in clouds.



## Bunker Model

The custom bunker model is again made in a cartoony yet somewhat realistic style; it is carved into stone; A big ventilator is slowly rotating, the vinyl floor is dirty and used up, and a rusty fence keeps the poor bunker people from falling. Unfortunately we had little time left to add more details like maps, litter, cables etc. The Texturing was buggy due to a bad use of UV maps.



## Unit Selection

The units can be selected one by one (no group selection due to time constraint and small number of robots). A selection mesh appears. The robots orient towards the clicked point, using a custom combination of Navmesh and physics (again PID and custom model). A high D value gave the result for following waypoints

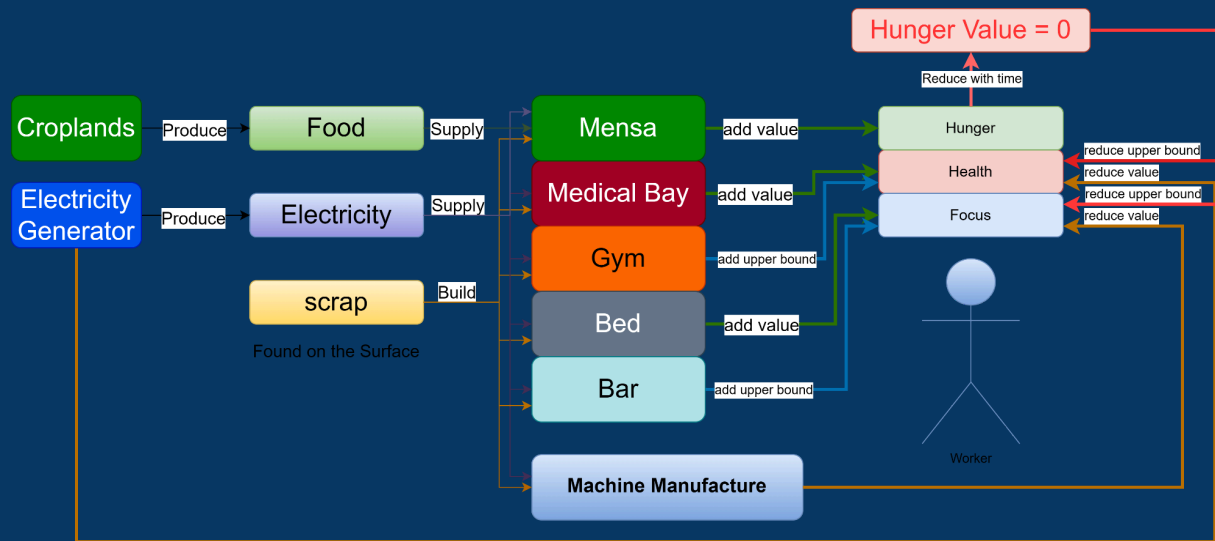


## Underground Game Play

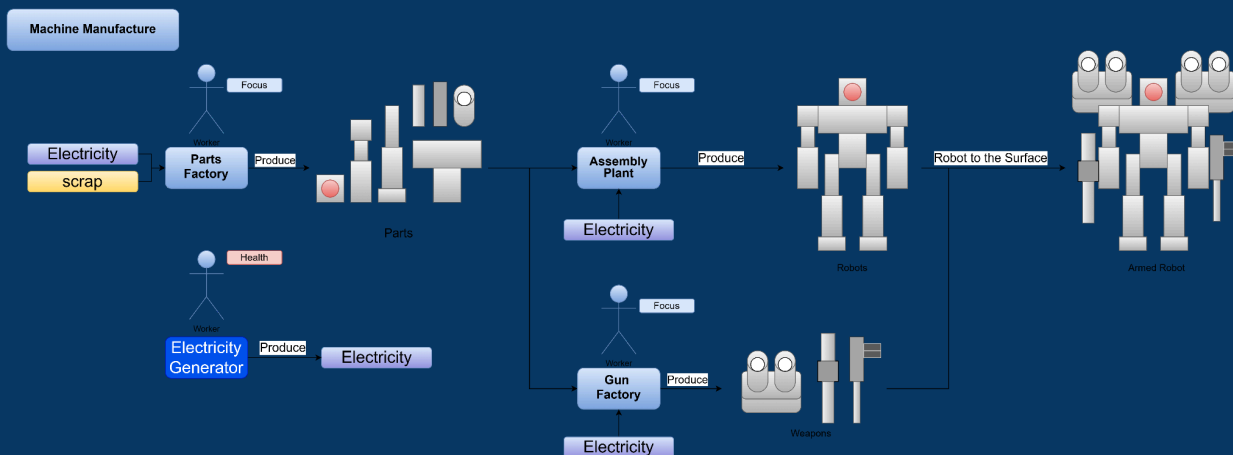
let the player to customize the underground basement that supports the people survive from the doomsday by using RTS construct system to building the underground basement and commanding underground citizens to utilize limited resources to build survive facilities and defense equipments to win the game

- Left click and drag to select people then right click the buildings to deploy them
- Right click the factory when no people are selected to select the equipment to produce at this factory
- Right click the Electricity Generator to release the people who work in the Electricity Generator

# Buildings effects and People's status



# Equipments produce process

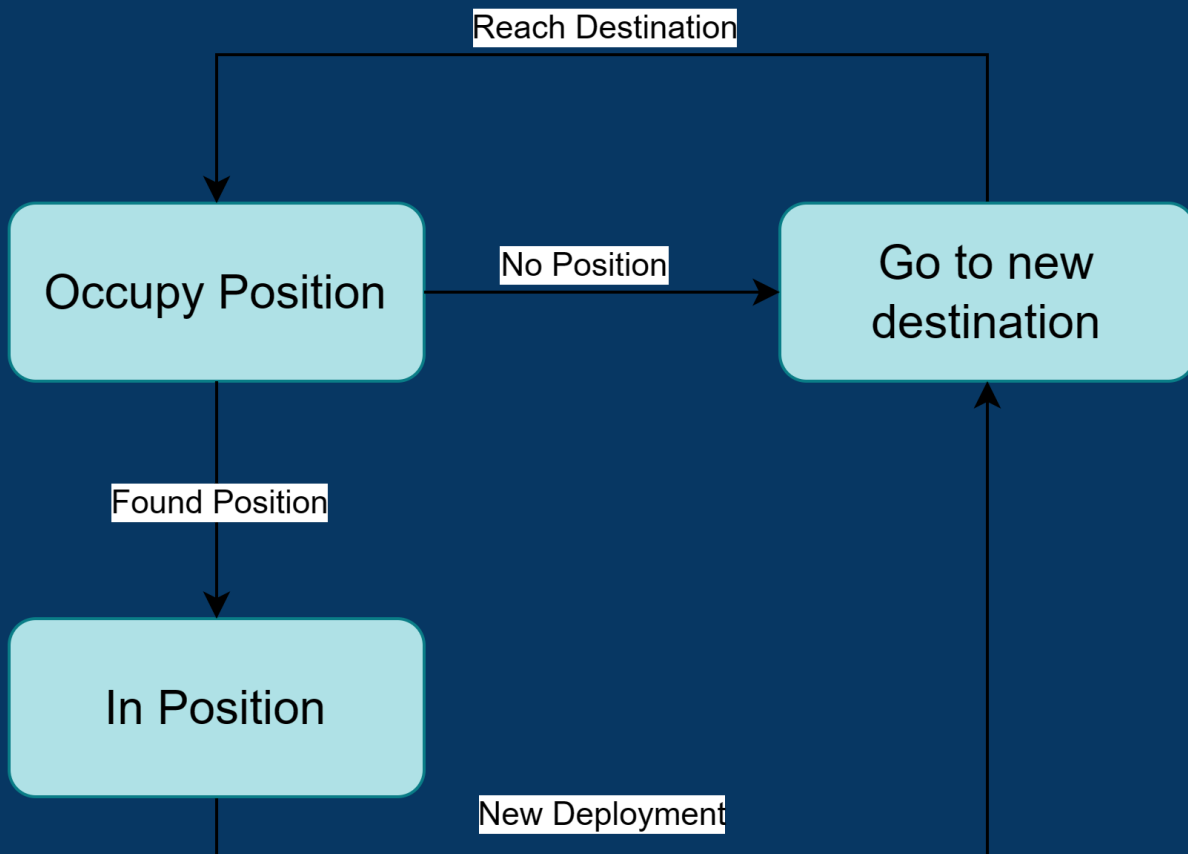


# Highlights

1. Simulating the authentic physiological state of people living in a restricted space which means they will feel hungry and feel tired when keep working and will suffer when don't get enough supply
2. smart deployment logic for player to make the game easier to operate

- a. Players don't have to send the task of the people one by one by hand rather just select a group of people then right click the task building, then the selected people will automatically distribute themselves to all the same kind of task buildings
  - b. automatically search the same kind of empty buildings when the employed building is fully occupied
  - c. automatically replace the tired people when sent to workplace
3. automatic self-supply logic of people
- a. When there is no work for the people to do, then the people will automatically follow the self-recover logic by themselves and don't need to wait for the player to deploy them. In this method, the complexity brought by the people's physiological simulation will be significantly relieved
  - b. achieve this by using looping status to drive the people to go to the destination





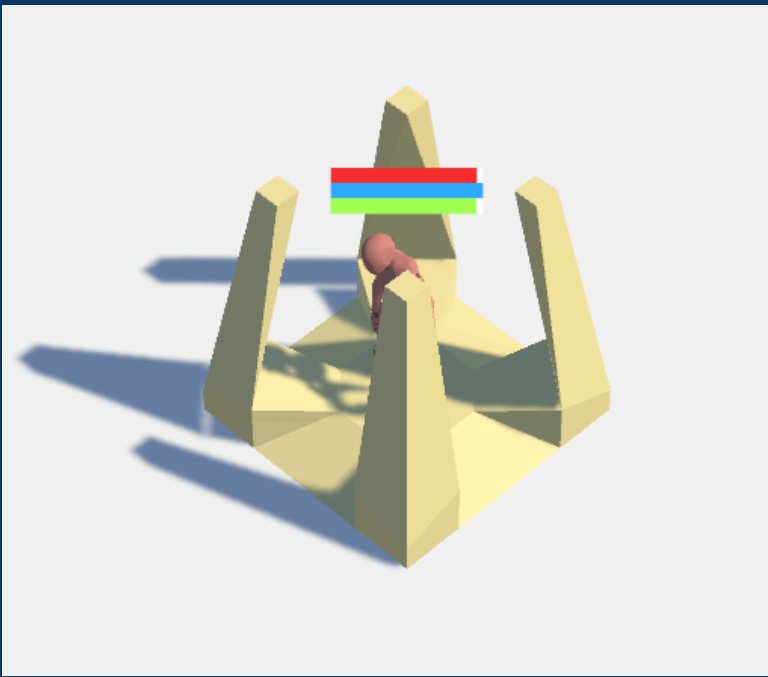
c. using ring queue data structure to store the instructions of self-resupply



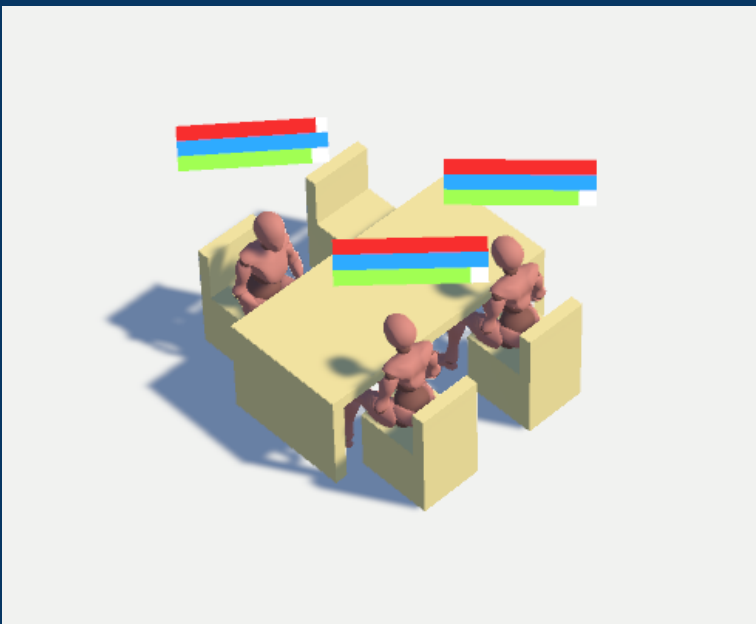
## Underground Basement Facilities Modeling and interaction

Implemented the 3D models of the underground basement facilities and the relevant animation controller used for interaction between people and facilities

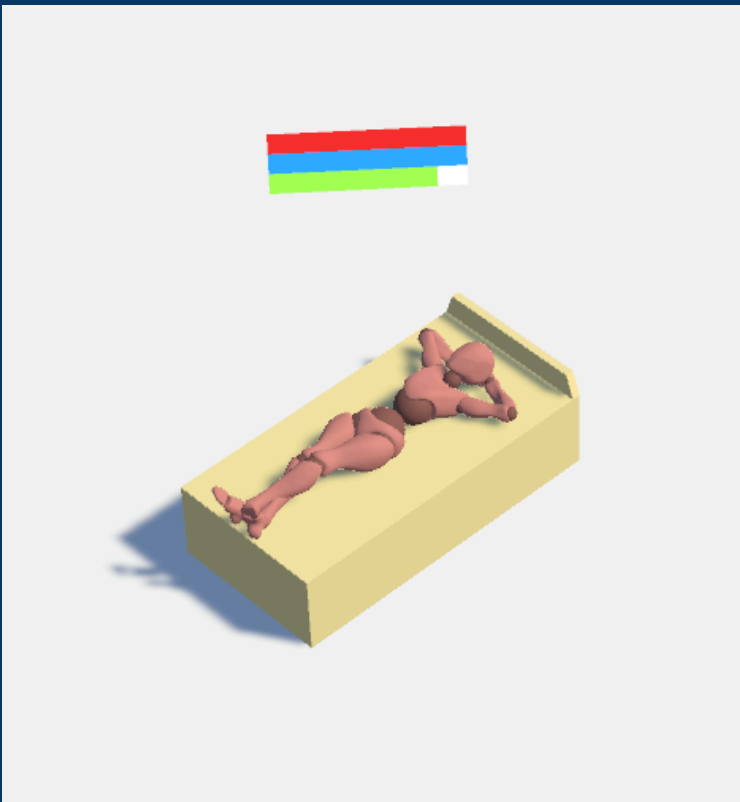
1. Electricity Generator



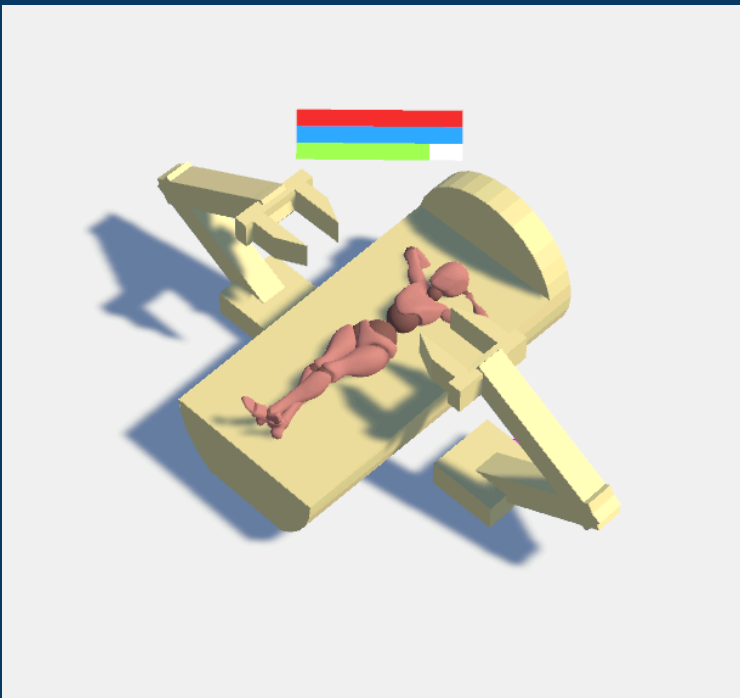
2. Mensa



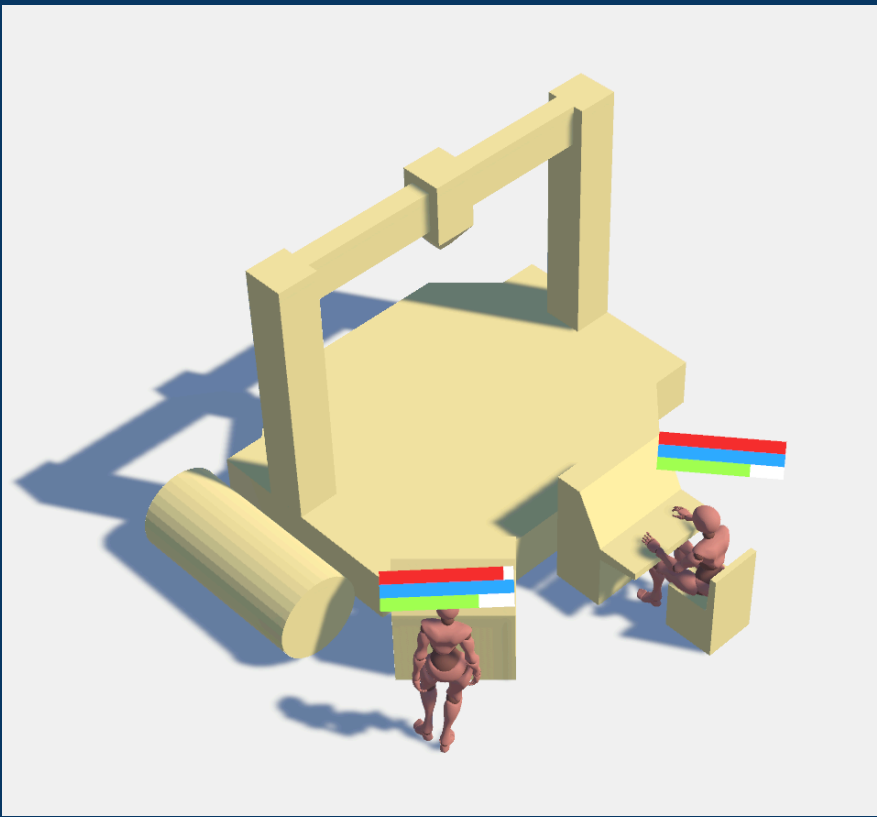
3. Bed



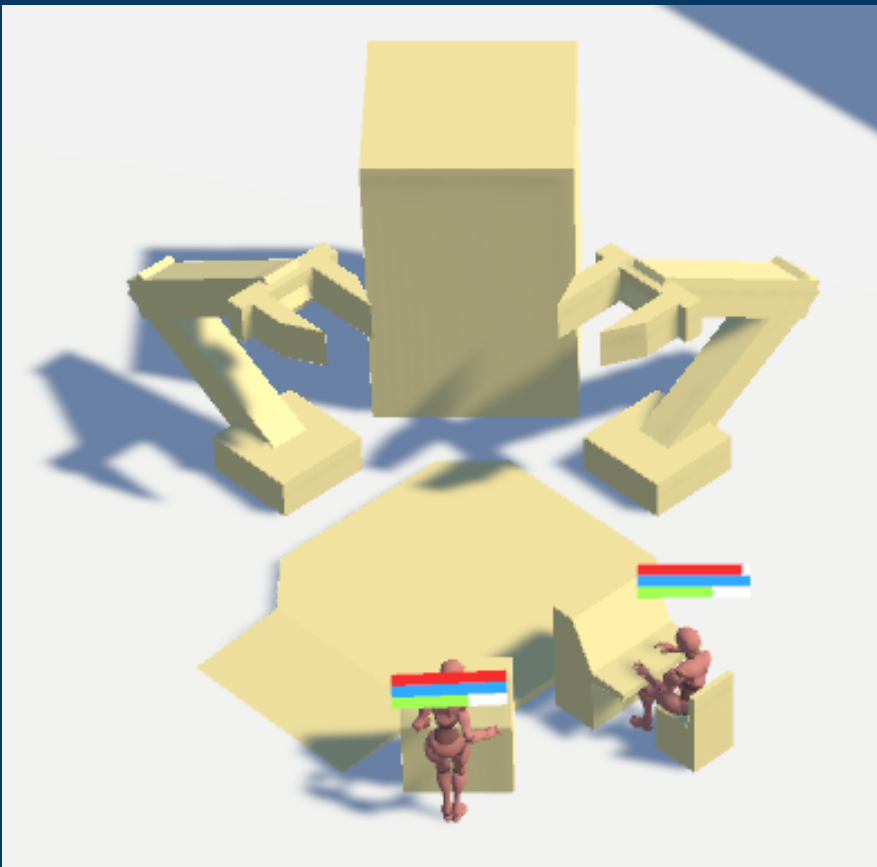
4. Medical Bay



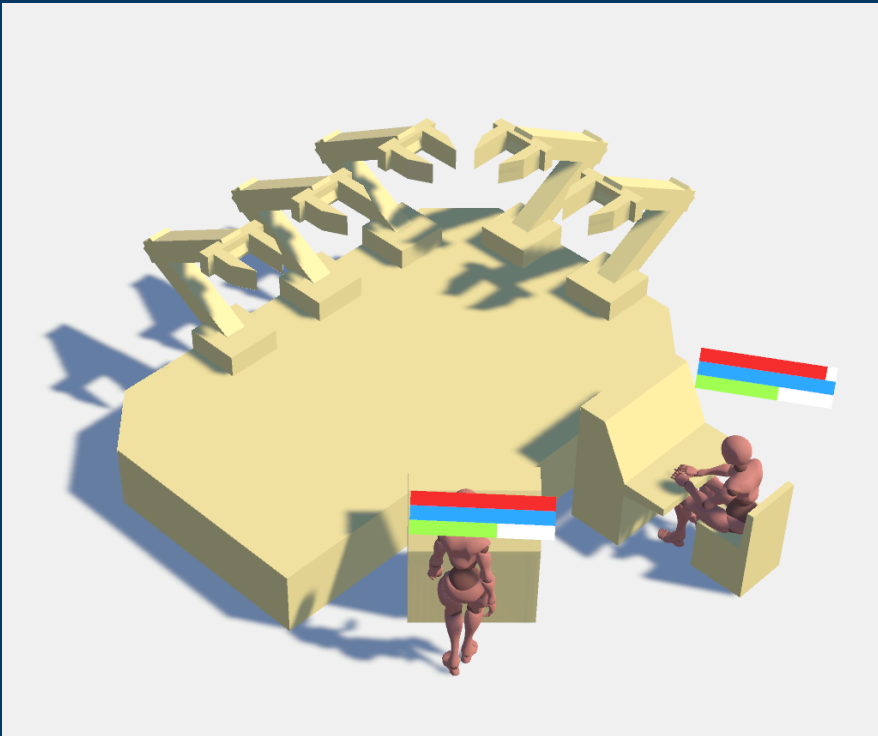
5. Part Factory



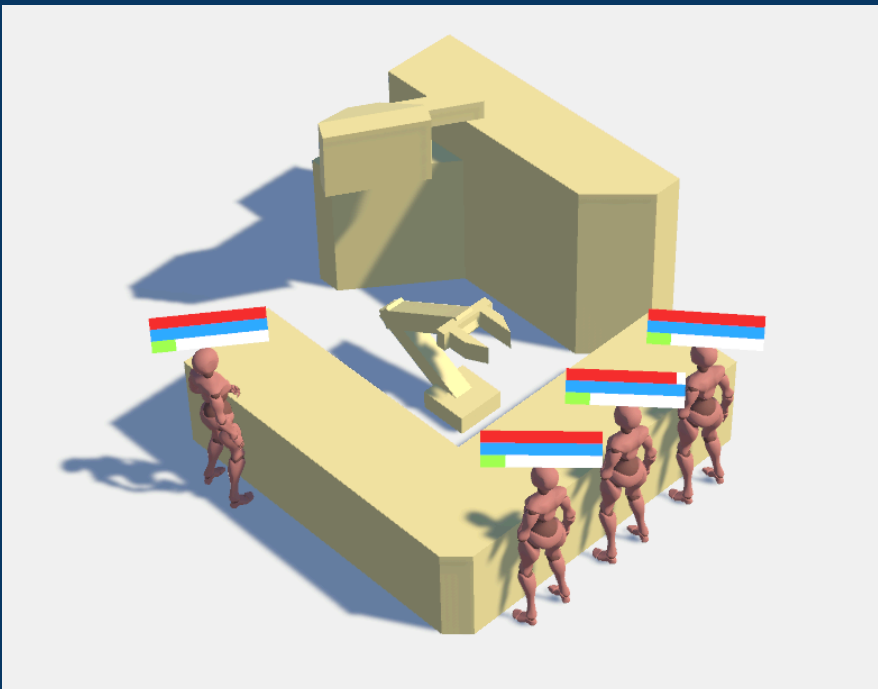
6. Assembly Factory



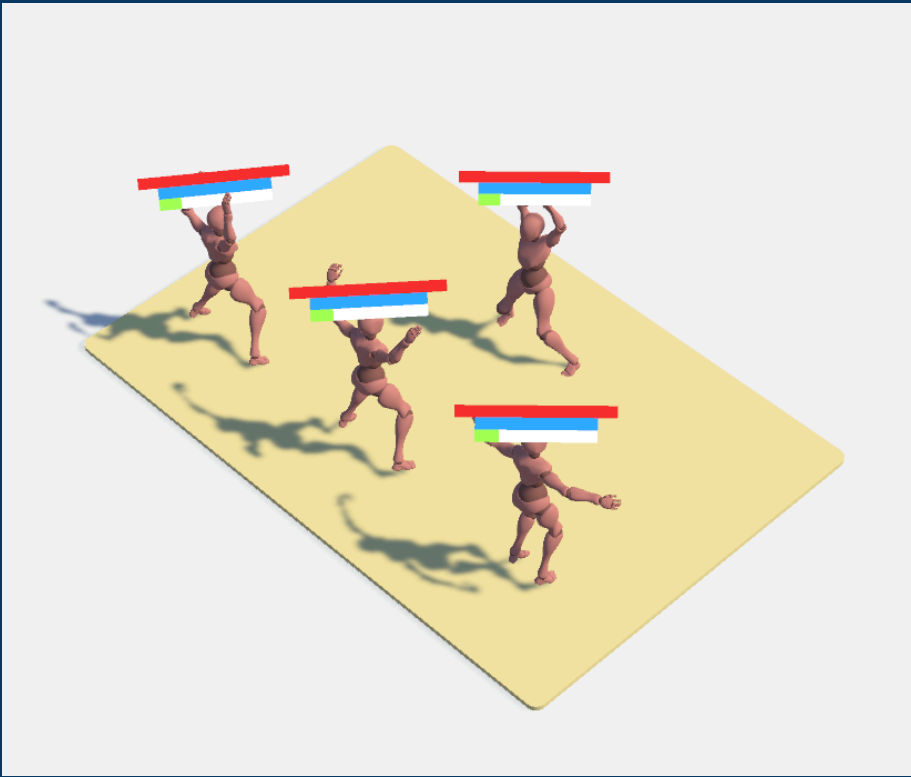
## 7. Gun Factory



## 8. Bar



## 9. Yoga mat



10. Population Generator



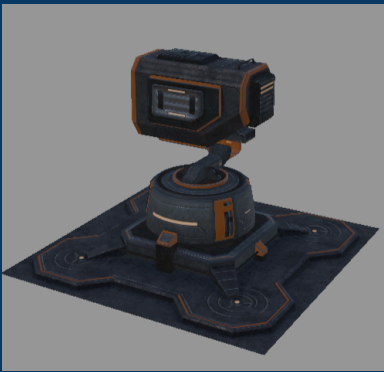
For Tower defense part, we have implement:

-Tower:

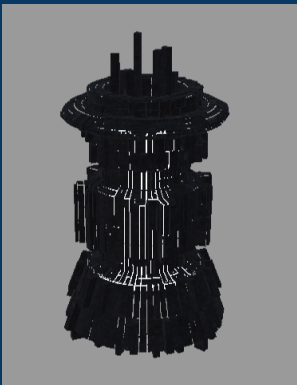
-Laser tower (custom model)



-Cannon tower



-Healing Tower



-Crystal Tower





-Enemy

- Radiated Barbarian



-Bug



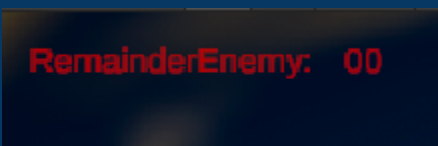
-Spider



-Zombie



-Wave System:



Game starts -> 2min countdown to the first wave:  
Quantity on each side: (same quantity on left and right)  
Spider: 1 Zombie: 2 Barbarian: 1 Beholder: 2

->After all monsters are killed->Countdown 2 minutes for the second wave:

Quantity on each side: (same quantity on left and right)

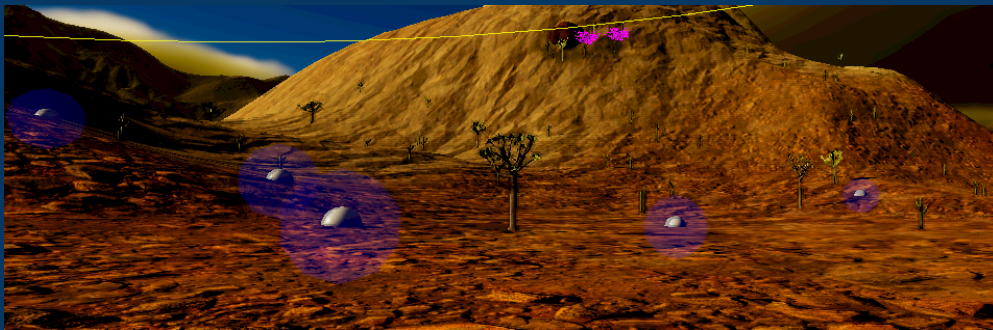
Spider: 2 Zombie: 3 Barbarian: 2 Beholder: 3

->After all monsters are killed->Countdown 2 minutes for the third wave:

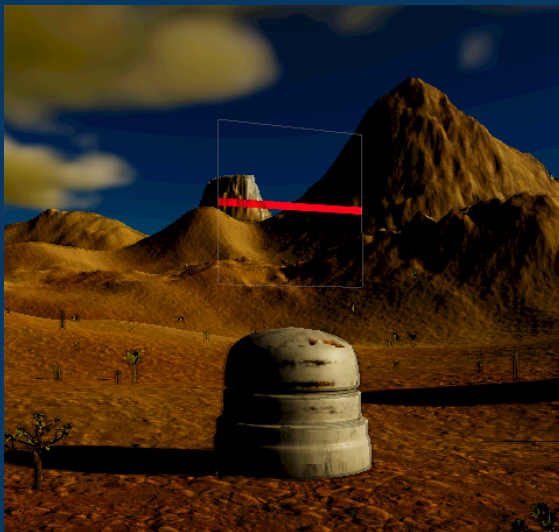
Quantity on each side: (same quantity on left and right)

Spider: 3 Zombie: 4 Barbarian: 3 Beholder: 4

(Infinite loop, the number of each monster is +1 each time)



-Win loss Condition (If health go 0 => Game over)



-Surface construction table

- Tab + 1: Tower



-Tab + 2 : Farm



-Tab + 3 : Robot

Voltage: 100%

Food: 1200 Scrap: 1040 Part: 100  
PST: 0 Light Combat Robot RF: 0  
WV: 0 LCB: 0 MCB: 0 HCB: 0  
Press R to fire, Press F to Drop, Esc to



0

Middle Combat Robot



0

Heavy Combat Robot



0

Pistol

Assault Rifle

Shotgun

Long Rifle



## Next Steps

We did a lot of simplifying steps to make the game mechanics possible in the given time frame, like no free positioning of the Surface Buildings, an automatic aiming by the robots, automatic Worker Robots etc., but which all turned out to be an interesting feature and make the game more unique.

The next steps will be implementing the 4 Layer; Adding some more content and polishing, like additional guns, heavy assault robots and more ML content like melee Robots.

Balancing will also be an issue that we will try to tackle after the player testing/evaluation phase.