SpiderBro Game Bible

Documentation Manual v0.3

Note: Changes in red indicate changes made because of play testing feedback.

# Table of Contents

[Table of Contents](#h.mqjdg2kvyanz)

[Summary of Play Testing Changes](#h.dytzo5n4ax0m)

[Outline](#h.lkjxx4ynnh9z)

[Unity Version](#h.xawmil8wg9q9)

[Player Object](#h.54mwv5wh199u)

[Enemy Objects](#h.1q8lzpvdq563)

[AI Design](#h.nqmf3gdnxyu5)

[Milestone 3](#h.trvex3ucdgc8)

[Milestone 4](#h.31h857ymk048)

[Milestone 4+](#h.mq1g4xlkpn87)

[Map Design](#h.dxh6dvjc4a2j)

[Milestone 2](#h.uxzx83w3x74e)

[Milestone 3](#h.hb1sqm4jrsu8)

[Milestone 4](#h.d59m2picd9fs)

[Version Control](#h.s7tkyniz2e1n)

[Game Controller Module](#h.fq30443ma0ni)

[Implementation in Unity](#h.n90v6kufj7c)

[Title scene](#h.9ot12mup366i)

[Room Scene](#h.idc9yu9m3nfo)

[Game Controller](#h.o6k1uj13p8e)

[LevelStatus.cs](#h.jsknpdn2c411)

[Player](#h.iinvcvg7i3e8)

[HUD](#h.1aulm3qufexh)

[Spawner](#h.heez62ixkn1x)

[Mouth](#h.52b7x3xj3mkn)

[Room](#h.z9lc1tt21sk1)

# 

# Summary of Play Testing Changes

* Nest Spawning System
* Fixing Clipping of Camera
* Fixing Animations
* Additional Sounds/Noises

# 

# Outline

*Goal: Without being killed: prevent enemy insects from entering the human’s mouth, until the timer reaches 0.*

*Context: You are a pet spider who loves and cares for his/her human master.*

*Location: Currently a simple bedroom with furniture, we plan to eventually add many other locations for future levels*

# Unity Version

*5.2.1*

We will be using version 5.2.1f1 (44735ea161b3)

<https://unity3d.com/unity/whats-new/unity-5.2.1>

Please do not use a later version/update your unity version past 5.2.1, without first approving it with the rest of the team. We usually do not need the bug fixes/new features from newer updates to Unity and updating can end up breaking the build and set us back a lot. If there’s a good reason to update, we can discuss that and approve it if deemed necessary - otherwise just stick to 5.2.1

# 

# 

# Player Object

The Player Object is responsible for allowing the player to interact with the game. It represents the player’s avatar as a spider. It has a couple of key features:

* The Cameras - Has both the first person and third person cameras attached to it
* The player model (a spider in this case)
* Hitboxes represent how the player can attack and be a
* Sounds from various movements (footsteps, attack sounds, etc.)
* Player stats such hp, xp, etc.

One change made as suggested by play testers, was to fix the clipping of the spider’s legs with the camera view. We were able to adjust the depth of the clipping plane in order to fix the issue. The player has two possible attacks: a melee attack which damages anything near their head in a cone shaped area, and a projectile attack which fires a small projectile which does a small amount of damage and slows enemy units.

We also have a 3rd attack planned, which allows the player to put down web traps, which act like trip mines and freeze cockroaches in place for a few moments.

Damage amounts are not yet implemented, and currently instantly kill all cockroaches.

Also note due to some feedback, some play testers found that Attack1 (melee) was partially useless do to Attack2 (projectiles) being more useful in many circumstances, as a result we have made the hitbox for Attack1 larger and easier to hit enemies and plan to have it do much more damage then Attack2 in later implementations.

Planned Damage amounts:

* Attack1 (melee): 4 dmg
* Attack2 (projectile): 2 dmg, and slows down the damaged unit for half of its speed for 5 seconds.
* Attack3 (trip mine): 1 dmg to all in blast radius once triggered, all within blast radius, are frozen and trapped for 15 seconds.

The player is also planned to have a total of 50 hp.

# 

# Enemy Objects

The enemy object represents the threat to the player who is the spider and the spider’s master. The goals of the enemy object:

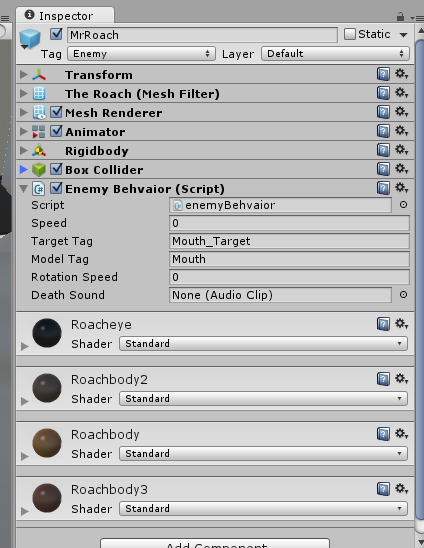
* Move straight into the spider’s master mouth.
* If they see the spider, attack the spider.



The enemy objects are represented by different kinds of insects. These insects have different attributes but the same goals.

* The cockroach, is the normal enemy unit, fast and easy to kill.
* The worm/centipede/beetle, is the heavy enemy unit, slow but difficult to kill.
* The moth, is medium speed, and easy to kill but will fly (if time permitting)

Depending on the enemy type, they will have the same behavior and will eventually use the same script to handle their pathfinding and movement. Enemy objects are prefabs generated by the spawning system.



Much has changed since our initial enemy object design. Currently, largely as a result of play testing, we’ve changed it so that there are still going to be three enemies, but their roles have largely been redesigned.

* The Worker Cockroach - an enemy unit with average speed and health, instead of attacking the human mouth, it will build and repair the closest cockroach nest spawning area. If attacked by the player it will attempt to flee to safety, before resuming its original task. Has a 25% chance of spawning. 5 hp
* The Drone Cockroach - a simple enemy unit with average speed and low health. It simply locates the mouth and heads for it damaging the mouth and dying when it runs into it. It ignores the player. Has a 50% chance of spawning. 3 hp.
* The Warrior Cockraoch - enemy unit with high speed and high health, instead of attacking the mouth the locate the player and attempt to attack the player. They have both a ranged and melee attack like the player. 9 hp. 25% chance of spawning

Attacks:

1. Ranged 2 damage to player
2. Melee 4 damage to player

Each enemy unit spawns from a “nest”, there are a number of invisible “nest nodes” scattered throughout the map. As new worker cockroaches spawn, they will attempt to build these nests by slowly repairing them to full health and thus enabling them to begin spawning. Once a nest has been built, it must be damaged to less than 1 hp in order to disable it from spawning. The nest model will also disappear when this happens. Once a cockroach begins building a nest, a nest model appears and can be damaged. The game starts with just one nest from which all cockroaches spawn. The player can damage nests to prevent them from spawning and if the player destroys all cockroaches nests and all current cockroaches, they will be able to win the game.

**Nest Locations**



**Green Dots -** Potential nest building areas.

**Red Dot -** Root nest (original spawning location).

# AI Design

## Milestone 3

The enemy AI simply locate the coordinates for the target and move towards the location. If they touch with the mouth target, then the enemy object will be destroyed and the mouth will lose health points. The game master object keeps track of the game timer and the handles the win/lose logic. The spawner ai will randomly select a range of coordinates and instantiate more enemies when the enemy numbers are running low. When the game ends, the game controller object will remove all spawners and enemies from the map.

## Milestone 4

The AI will be similar to milestone 2, with some additions with the enemy attacking the player and building nests. See Enemy Objects for more details. Enemies will seek if player is nearby and if they are warrior type enemies; if the player is not nearby the enemy will head towards the mouth target. Worker types will attempt to flee when attack, and otherwise spend their time building nests. Drones keep the implementation from milestone 3 and just damage the mouth ignoring the player. For navigation for a more complex environment, there will be a pathfinding system that will find the nearest node and calculate the shortest path to the mouth using Dijkstra’s / A\* algorithm. The spawners will have an addition to their random range placement by checking the elevation to establish a seamless spawn by using a raytrace to determine its appropriate elevation range.

## Milestone 4+

Depending on the future feedback and the backlog, the ai will be the same as Milestone 3, having the enemy climb over obstacles and behaviors if there are different enemy types. Warriors will likely have the biggest changes, attempting to dodge player attacks and hit the player while moving, etc. They will also be given patrol routes, and visit places where the player frequently inhabits. If they spot the player while on patrol they break away and attempt to kill the player.

# 

# 

# Map Design

## Milestone 2

A simple boxy version of the bedroom setting to test out player functionality and the majority of the basic game play mechanics.

## Milestone 3

Rough of bedroom setting with player functionality, enemy ai, and the mouth objective in place and any balance changes from the play testing. Some polishing may happen if time permits.

The bedroom setting will more detailed than milestone with a larger room with a bed(bed frame, mattress, pillow), desk(drawers and chair), dresser, cage and more simple clutter objects that will be obstructions for the player and enemy units. Simple materials used in the items in the room and lighting to give a simple representation of the room that will be polished for the final milestone. Some of these planned objects did not happen, but are planned for Milestone 4, mostly due to time constraints.

## Milestone 4

polished version of the bedroom setting with final player functionality, enemy ai, mouth objective. The final map will have more environmental aspects in lighting, sounds, assets, materials, and layout from the comments from the play testers from milestone 3 to enhance balancing. May also make additional maps available if time permits.

We are planning possible additional levels in addition to adding nest nodes/objects, and more detailed clutter. These levels can be different bedroom styles from children room, and teenager room, and other ideas that we still need to discuss. We have also discussed adding webbing from the spider or some other obstruction that gives the player multiple objectives to defend, but slows down cockroach progress to the mouth. Additional obstacles may also prevent the player from destroying nest nodes too directly. The nest will create an incentive to draw the player away from the bed area and towards other locations on the level.

# 

# Version Control

SpiderBro uses Git for version control. The repository is hosted on GitHub:

[*https://github.com/CS3540-SpiderBro/spiderbro.git*](https://github.com/CS3540-SpiderBro/spiderbro.git)

There are several important branches in the repository:

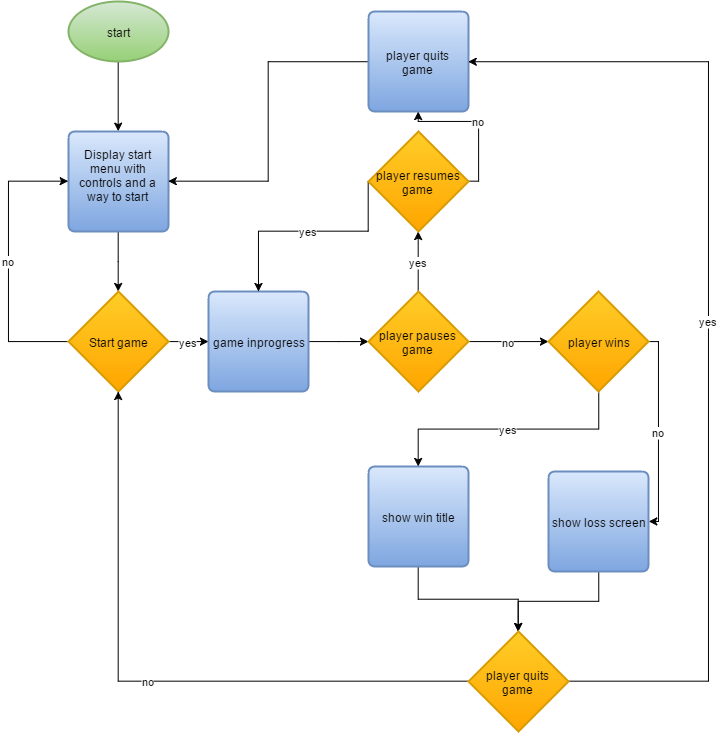
* *master:* The stable branch where only stable, working games are

# 

# 

# Game Controller Module

The Game Controller Module, controls the state of the game. For example, when to display menus, pause the game, how much time should be allowed, etc. It is tied heavily with the respawn module and works with it to control the overall game.

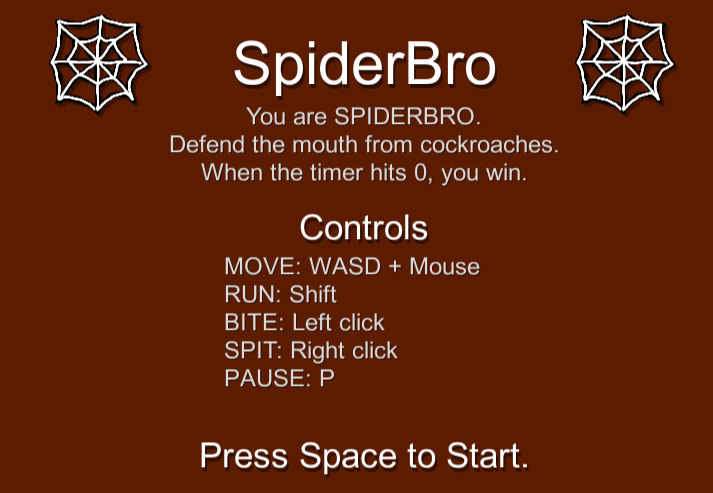
Here’s a basic outline of the Game Controller logic:

# 

# Implementation in Unity

We have two scenes in this game: The Title scene and the Game scene. Each will be detailed separately.

## Title scene



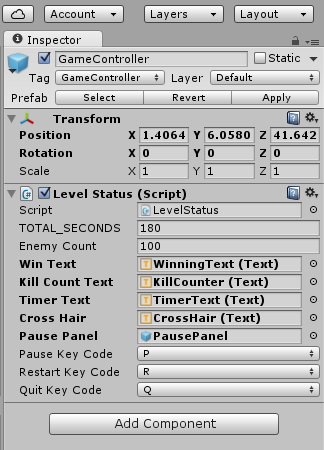
Simple title with game instructions and listing the controls. Implemented on a canvas, using a Panel for the solid background. the Panel contains various text and sprite elements.

Attached to the Panel is TitleScript.cs. In Update, it listens for a spacebar press, and when it’s pressed, it will load the level scene. The name of the level scene is specified in a public variable so it can be easily changed.

## Room Scene

The room scene has many different GameObjects and Scripts working together.

### Game Controller



The game controller is a non-rendered object, which contains several useful pieces of information used by the LevelStatus.cs script. This includes variables for how many enemies to spawn, how many seconds to allow, which controls to use and text to use. Note: One change we made to this was fixing the pause feature, due to feedback from testers.

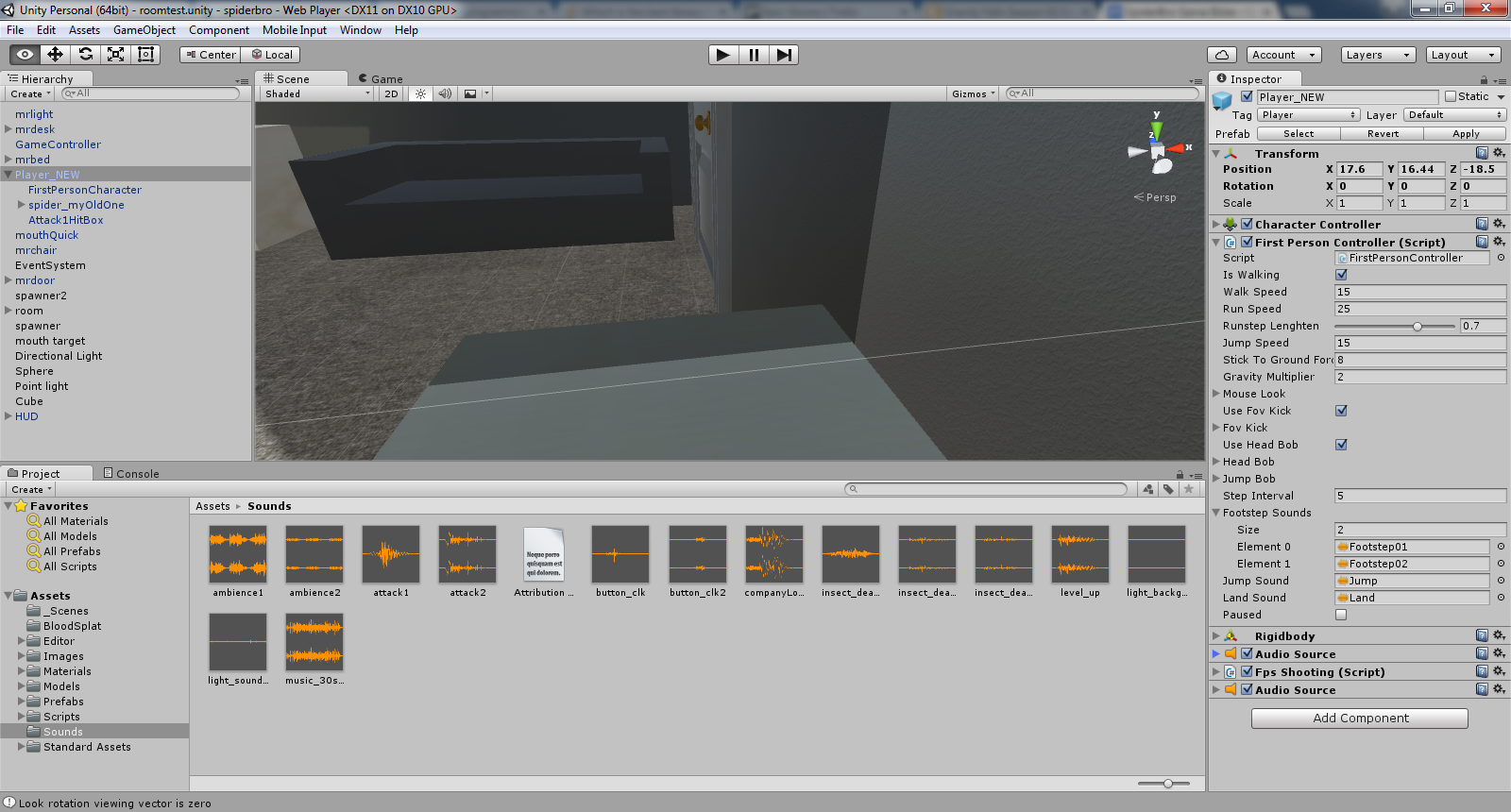
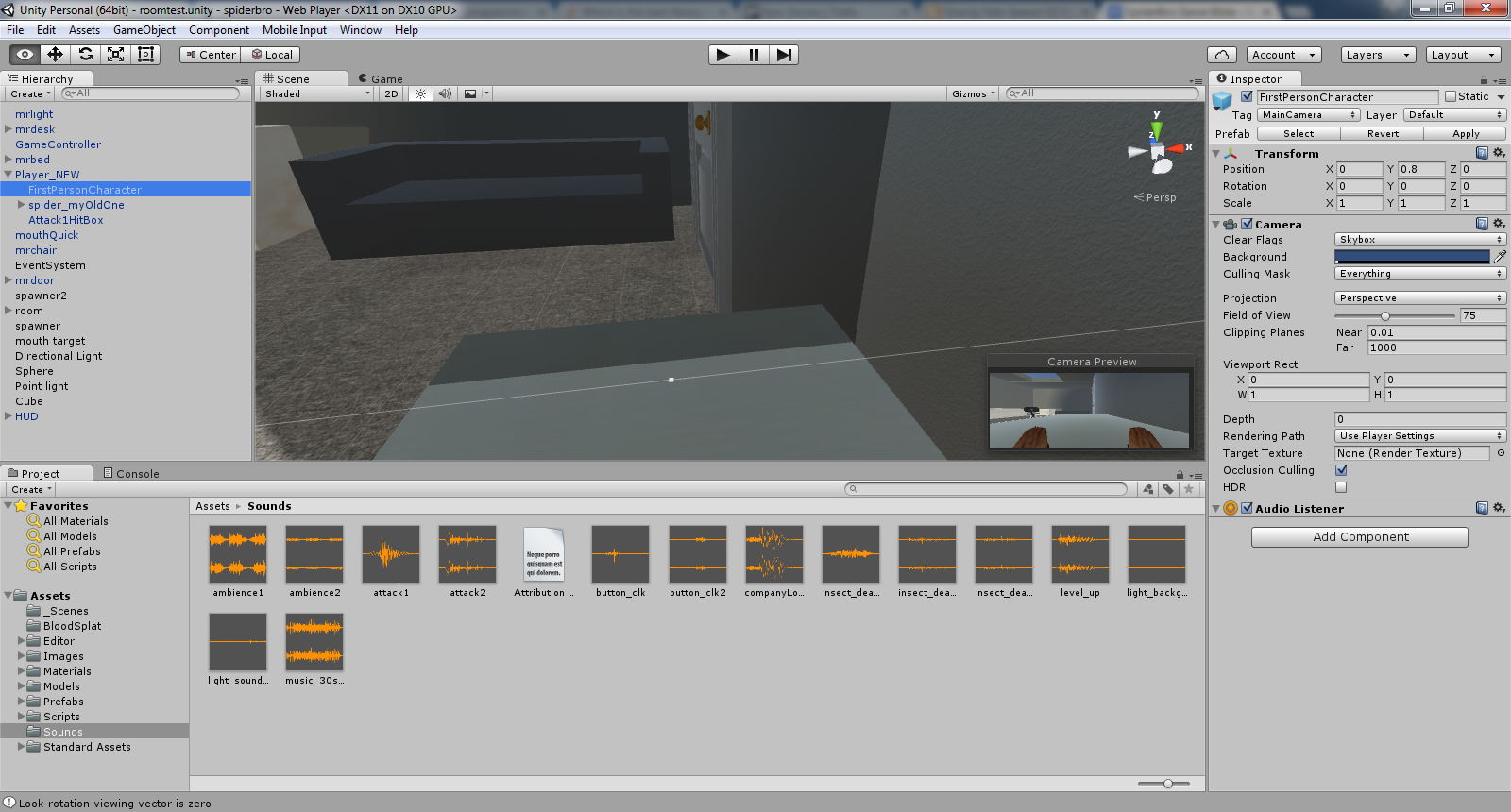
### LevelStatus.cs

A script responsible for managing the following tasks:

* Pausing and unpausing the game
* Restarting the game
* Quitting the game
* Updating the Kill Counter UI
* Displaying Game Over text (win and lose conditions)

## Player

The Player Object is available as a prefab, but should be added to the scene to function.

The Player Object as pictured above, has several components to it, with several variables and functions attached. The FirstPersonController script is a primary feature that handles both speed, jump speed, gravity effects, sounds, bobbing and camera motions and general game input controls for moving the character. A spider model is also part of the player object, which allows the player to be rendered and have a shape to collide with other objects in the game. The FirstPersonController contains a camera object with several settings such as FOV, Clipping planes/culling masks, depth, etc. Controls how the player sees the world.

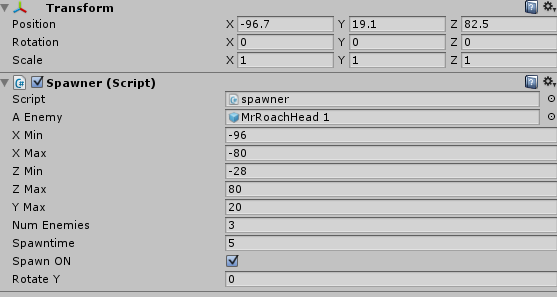
## HUD



The HUD hierarchy contains all of the on-screen UI elements in the game. This includes the following:

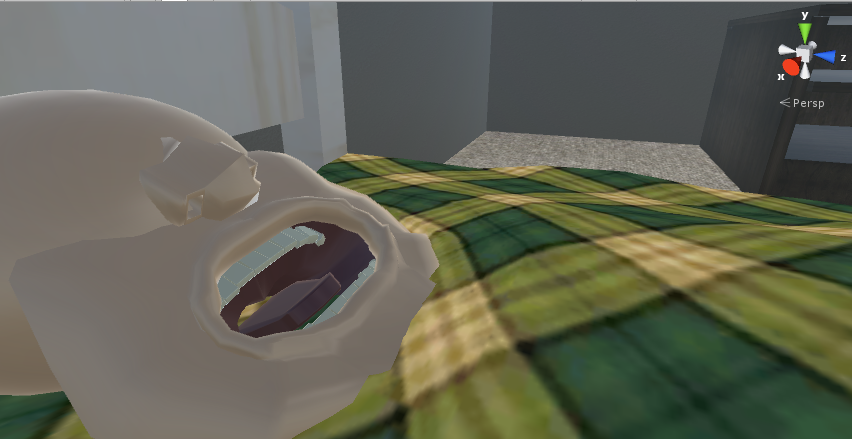
* *Health Slider:* Green bar showing what percentage of health the Mouth has left.
* *Timer:* Displays how much time is left before the round is over.
* *Kill Counter:* Displays the number of cockroaches that the player killed.
* *Spit Slider:* Blue bar indicating the spit ammo remaining for the player. Regenerates slowly.
* *Crosshair:* A simple crosshair showing where the center of the screen is, used to aim the spitting projectiles.
* *Winning Text:* The text to display when the game has ended. Hidden until game over.
* *Pause Panel:* A panel that shows up when the game is paused. Shows directions to either unpause, restart, or quit to the main menu. Hidden at first.

## Spawner



This spawner game object is run by the Spawner.cs script that takes multiple parameters to set the range of “randomly generated” spawns of enemies. Enemies are placed within a range of coordinates on the map and then head toward the mouth from that location. There is a time interval setting that controls how frequently they spawn. This allows them to attack from multiple angles and directions at once, giving the player a better challenge.

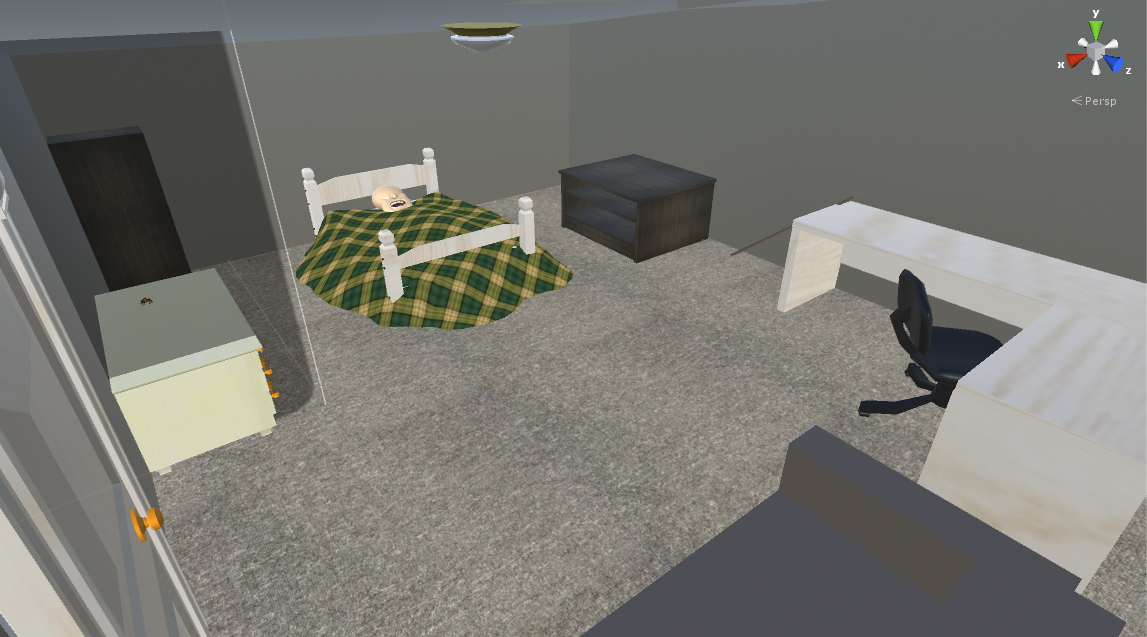
## Mouth



A model created by Greg in Maya modeling software. Contains a “mouth target” that is transform object that is the target point for the AI. This mouth object is a combination of multiple 3d shapes.

The mouth is the point that the player must defend. If the player lets 20 cockroaches enter the mouth, it’s game over. The mouth has a set amount of health that can be adjusted by developers and defaults to 20.

## Room



Mainly composed of cubes with textures and rigidbody colliders, with a few models. Enemies spawn near the surface of the white desk with the chair and then head toward the mouth on the bed. The player spawns by default on top of the white dresser to the right of the bed.