

Gluttony Run

By Ryan Smith, Tim Phillips, Shane Kusmierz

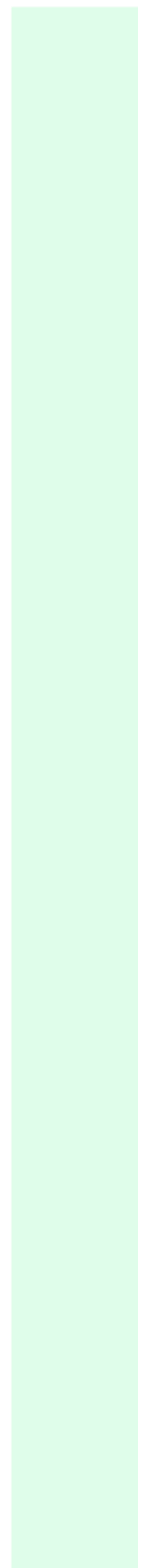


Table of Content

Scope	Pg4
State machine	Pg5
UML Diagram	Pg6
Discord Link	Pg7
Mechanics	Pg8-Pg11
Character states	Pg8
Movement	Pg9
Jump	Pg9
Fart	Pg10
Interactable object	Pg10
Fat indicator	Pg11
Score	Pg11
Game and team status	Pg12
Game Project	Pg12
Shane Kusmierz	Pg12

Ryan Smith

Pg15

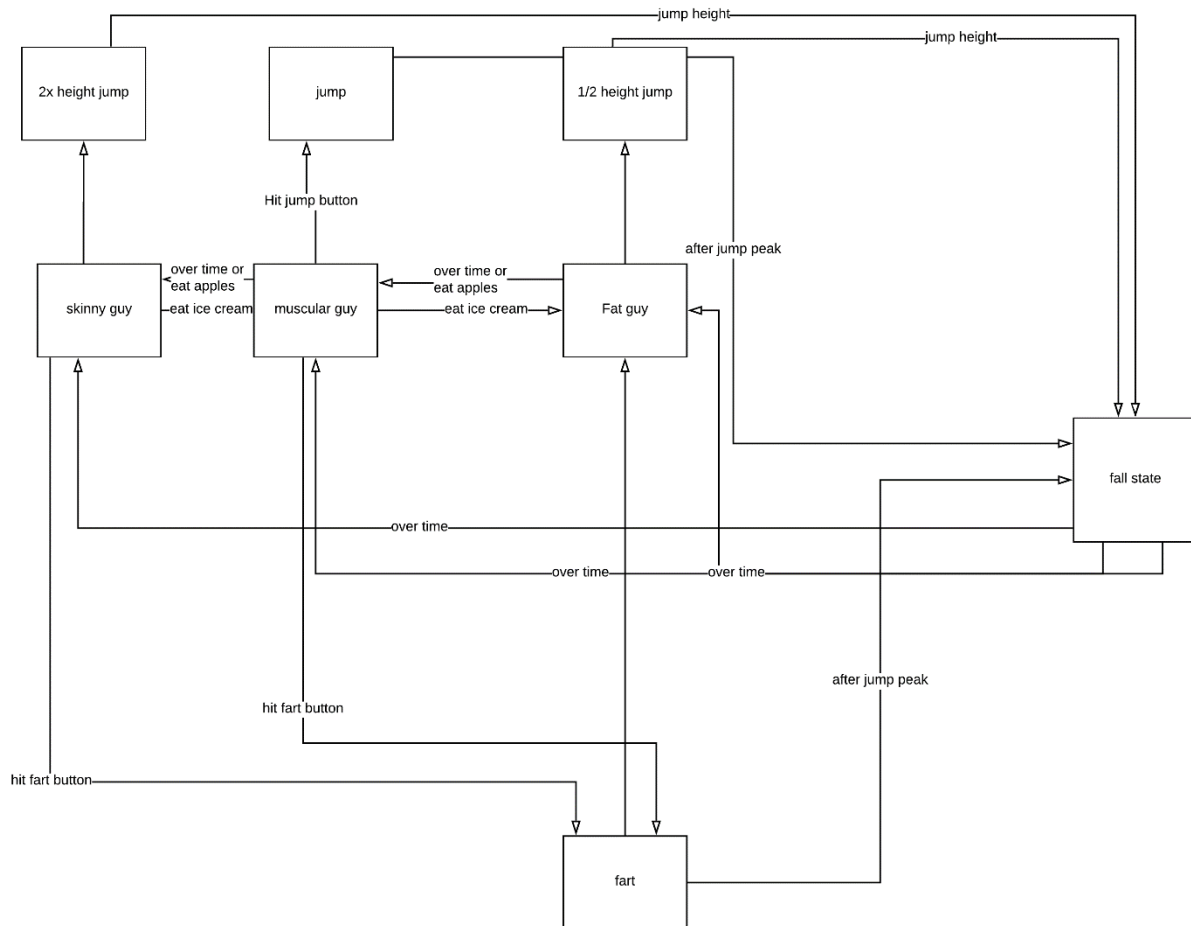
Timothy Phillips

Pg17

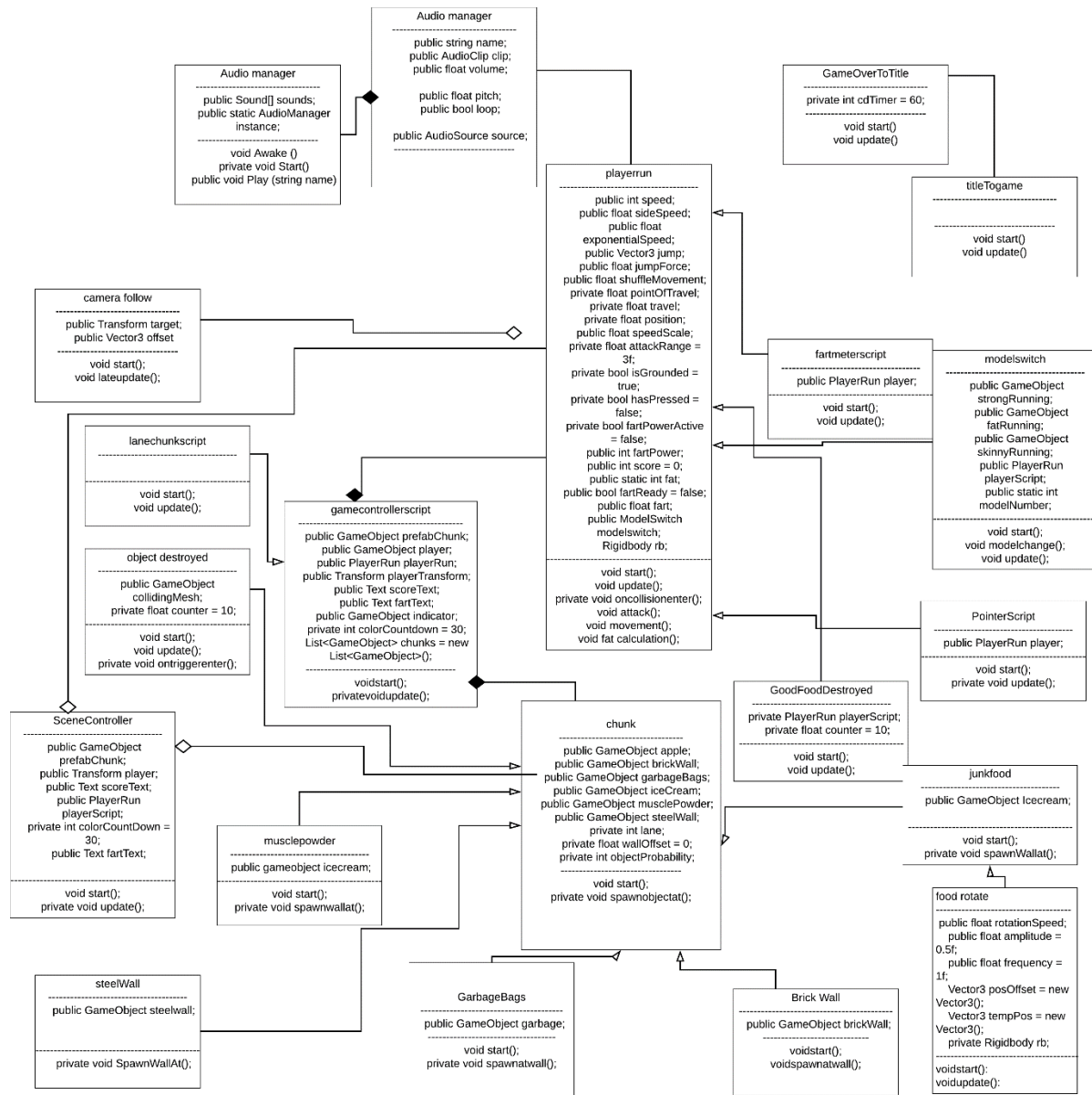
Scope

- The player will be able to switch between three different states.
 - A slow lumbering fat state
 - a fit muscular state
 - a lanky lean state.
- When the player gets too fat or too lean they will die.
- Both the fat state and the lean state will have an obstacle that only they can get over.
- When the player eats enough food, they will be able to use a super fart move.
- Have a fat bar that shifts back and forth depending on how much bad/good food the player has eaten, and how fast the player is running.

State Machine



UML Diagram



Discord link

<https://discord.gg/ssgEaXc>

Mechanics

Character States

The character is programmed to move differently in the three states they are in. the code for the muscular state is:

- If the player is between 0 and 600 on the fat bar the code will set them to the muscular model.
- it will accelerate at 9 units and use the strong running settings to make it look like the guy is in shape.

the code for the lean state is:

- If the player is under 0 on the fat bar, the code will set them to the lean model.
- It will accelerate at 12 units and use the skinny running settings to make it look like he's going super-fast.

The code for the fat state:

- If the player is above 600 on the fat bar the code will set them to the fat model.
- It will accelerate at 5 units and use the fat running settings to make it look like the guys out of breath.

Movement

The character is programmed to move differently in the three states:

- To move the player's Z position is added to itself equal to a speed value times the delta time.
- The value for speed is the lowest for the fat state, in the middle for the muscular state, and the fastest for the lean state.

Jumping

As with running, jumping is going to be different with each state:

- To move the players rb.velocity is added to itself equal to vector3.up times the jump force.
- The jump value is the lowest for the fat state, in the middle for the muscular state, and the highest for the lean state.

Fart

- The fart is the player's special ability is to fart and it will only be activated when they press the F button.
- If the player eats either fat or healthy food, the fart bar will fill up a small amount.

Interactable objects

The food interacts with the player based on whether it is healthy or junk food:

- The apple takes away 100 fat from the fat bar edging it towards the lean state.
- The ice cream adds 100 fat from the fat bar edging it towards the fat state.
- The muscle powder sets the player to 700 if the player is above 600 making them closer to the muscle state no matter how fat they are.
- The muscle powder also sets the player to 300 if the player is under 400 making them closer to the muscle state no matter how lean they are.
- All the obstacles and food spawn from a single spawn point in each lane with a random number generator deciding what spawns.

Fat Indicator

The fat indicator scrolls left or right depending on whether the player is losing or gaining weight.

- The fat bar slowly declines in value over time when not eating.
- The fat bar decreases a lot more when eating apples.
- The fat bar increases a lot more when eating ice cream.
- The fat bar moves towards the middle area when eating muscle powder.

Score

The score will show how well the player is doing in the games:

- The score increases over time as the player runs.

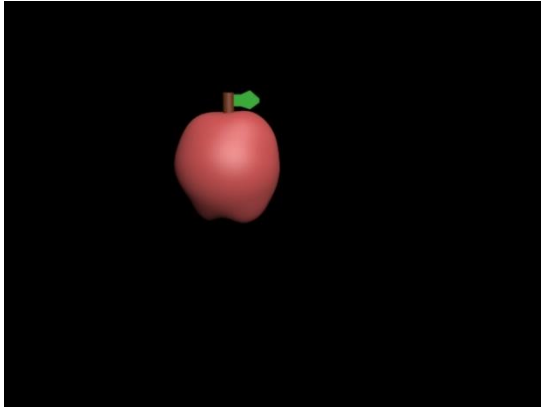
Game and Teams status

Game project

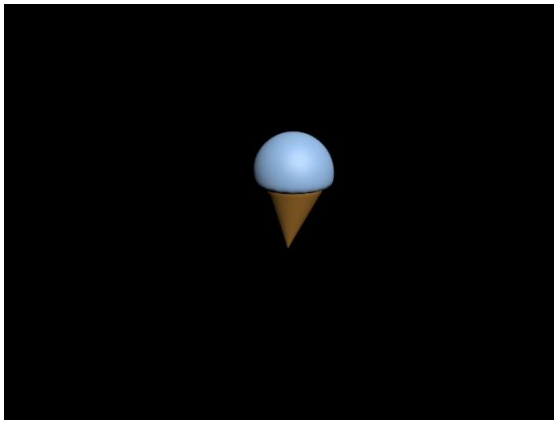
The game's status is finished. We are spawning one object in each lane as well as some streetlights and buildings on the side for the purposes of aesthetics. The UI elements show the players fat meter, fart meter, and score. The players fat meter goes up or down depending on what the player eats or if the player isn't eating. When the player hits either of the extremes or an obstacle they will get a game over and get to start over again.

Shane Kusmierz

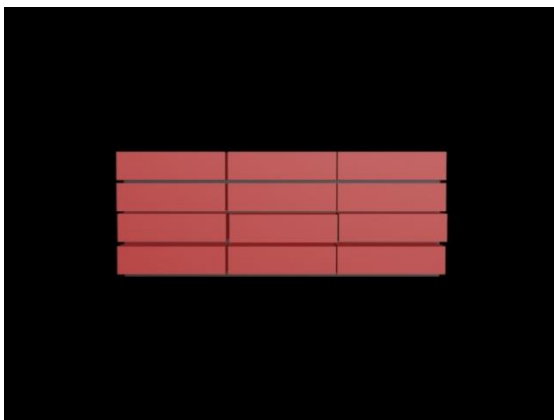
For my job I oversaw environmental art like the food and background designs. I am also in charge of the voice acting for the announcer and how the food interacts with the player. I have modeled the food, muscle powder, and ice cream along with the obstacles: steel walls, brick walls, and garbage bags. I have also coded the interactions with game elements such as implanting the script that destroys food objects. I also am in charge of the design documents formatting, length, and stylistic choice. I placed a script in all the lanes to make said obstacles spawn and another script to make the objects rotate. I also helped Ryan out with some of his scripts such as getting the streetlights and buildings to destroy over time.



The apple model



The ice cream model

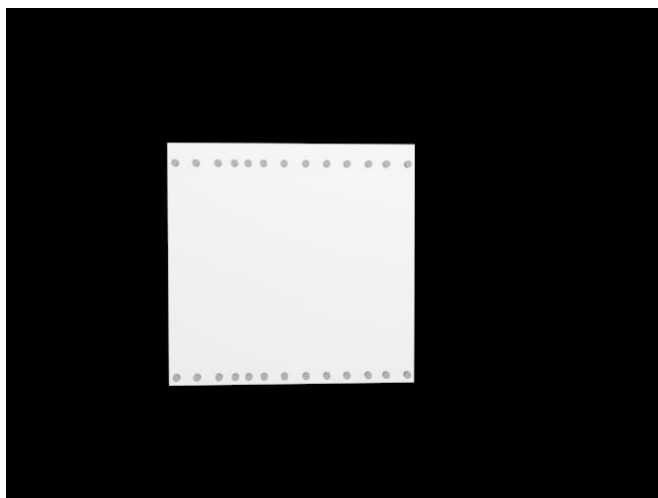


Brick wall model





Muscle powder

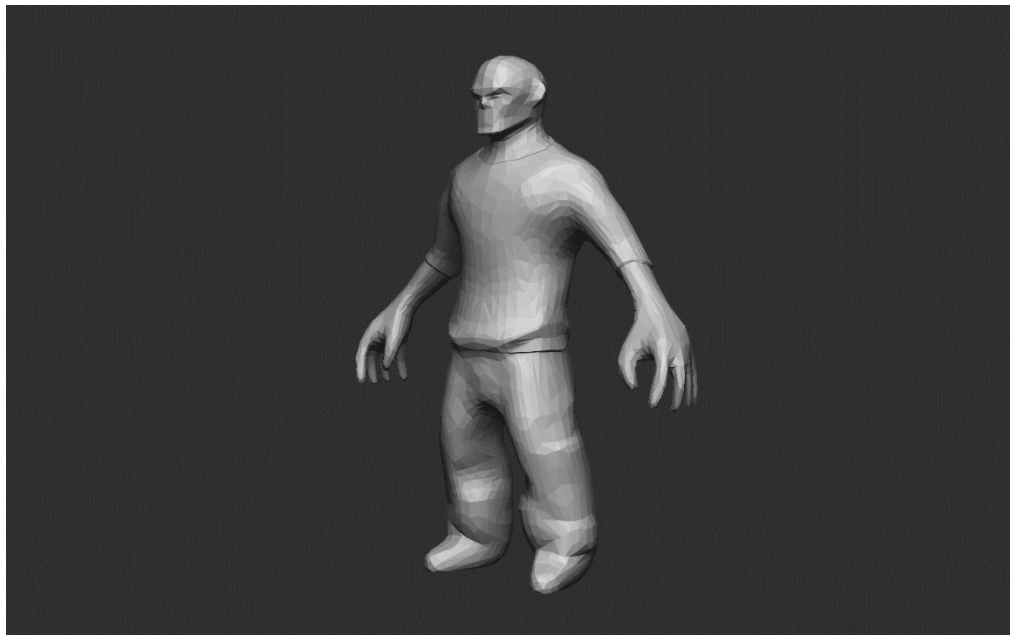


Steel wall



Ryan Smith

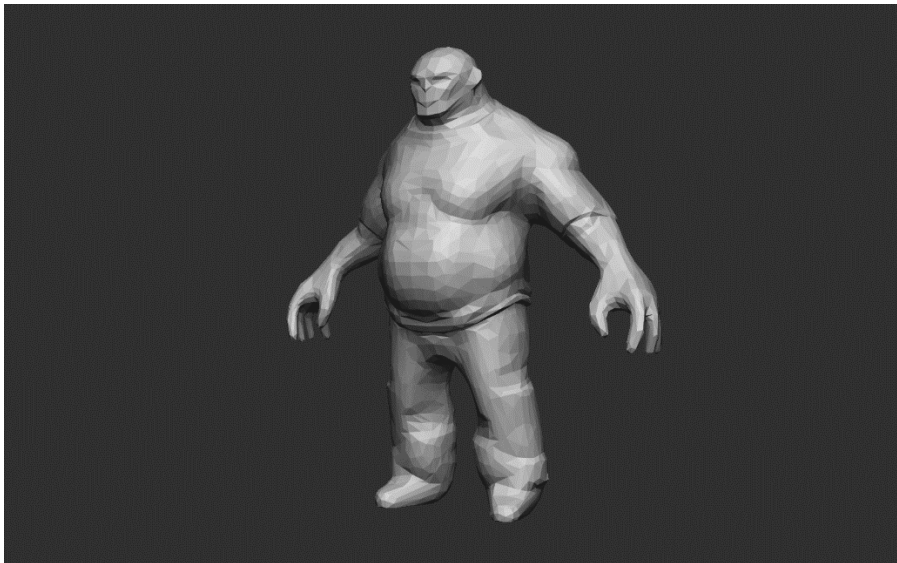
Ryan Smith is the guy in charge of the characters. This means that he is in charge of the mechanics, animation, and rigging along with the voice acting for the character. Some of the things he has made were the characters three different states models, and the code making sure the character does not fall through the floor. He and Tim also made sure that the UI elements for the character corresponded with which state the character in so when the Fat bar is leaning to the right the character is skinny and as it progresses more towards the left going to muscular and then eventually fat. He also has worked with the food destroyed code I made and has it so that now when the food goes past the player it dies after some time passes. This code was passed onto all the other objects, so they will eventually die over time.



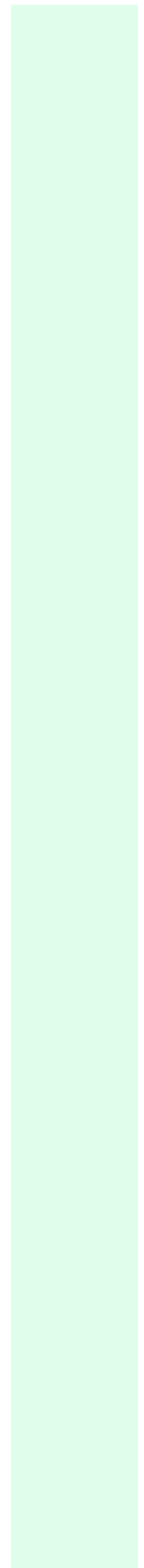
Skinny guy model



Strong guy



Fat guy

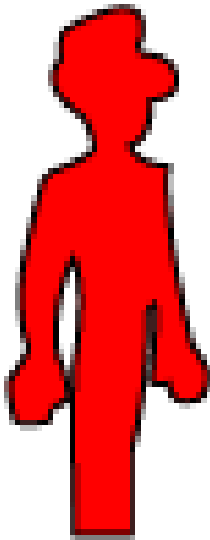


Timothy Phillips

Tim is in charge of all the UI elements and the background music along with being the project manager. What this means is that Tim has programmed the score making it so that it increases slowly as the player progresses through the level, the fat bar making it so that the player's fat bar decreases over time and the state changes to show that. Tim formatted the UI and gave it more of an aesthetic appeal along with making sure that it was properly able to go to both far extremes. He made sure that the distance was able to properly tick up so that the player could get a score and was the one who made a fart input. He also was the one who made the goal lists with our input and kept us on track to nail all the objectives we must hit.



UI 1.0



Lean guy

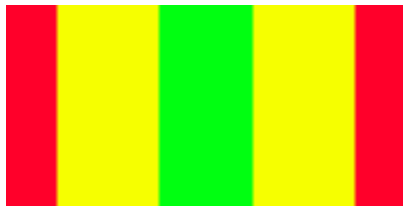


Muscular guy

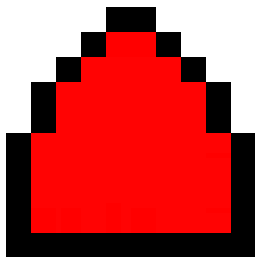




Fat guy



Fat bar gradient



Icon



Fart bar

