

1.

Gordon's Pizza Simulator

Mohammad Taufique Imrose, Max Rutherford

CSC 165-01

## 2. Game Screenshots



### 3. Instructions to compile and run:

- To compile and run our game, the first thing you would need to do is go into the CSC165RestaurantSim file. From there, you would want to first, execute the clearTAGEclassFiles batch file, then execute the buildTAGE batch file. Once you've cleared and rebuilt TAGE, you can go ahead and execute the compile batch file, then execute the run batch file.
- For networking, after you run the two TAGE files, you would want to go into the server file, and first run the compileServer batch file, then run the runServer batch file. After you run these two batch files, you can start up your game on two separate clients
- NOTICE: We currently have limited networking capabilities within our game, we had it working well at one point but once we introduced NPCs it bugged out the networking logic. We will try our hardest to fix this before our game demo

#### 4. How To Play Our Game:

When you first load up the game, your player (human model with a chef hat) will spawn outside of Gordon's Pizza. As you spawn, you will see an NPC already walking towards the door of the restaurant. You move using the standard WASD keys (W to go forward, S to go back, A to move left, D to move right). We have mouse control in the game as well to look around, but if you would like to use keyboard controls for that as well, Q is to rotate the character left and E is to rotate the character right. Once you make it behind the counter, there's a few interactable objects you can interact with. There's an oven between the counter and the walk in fridge where you can walk up to it and press F on it to bake a pizza. By now, you may have noticed a list of ingredients in the top left corner. For now, the player can only make one pizza (it's just a cheese pizza), so it will only cost the player one sauce item and one cheese item. Once it's done, the pizza will be seen on the player's shoulder. When it comes to the customer, they will walk up to the counter and wait for you to take their order. Once you get close enough to the register, a prompt will show up telling you to press F to take their order. Once you do this, they will go sit down in one of the chairs, wait for a few seconds, then come back up to the counter. Once you have a pizza ready, you can walk up to the customer and press F to serve them. You'll notice that your cash (top right of the screen) will go up by 10, you made a sale! Be careful though, there's a thief that lurks behind the customer NPC and will come in and steal \$10

from the register. To stop him, you need to catch him before he gets too far from the restaurant and press F to catch him and get your money back. Finally, there's another object on the counter that you can interact with in order to buy ingredients. If you walk up to the computer and press O (it will prompt you), it will allow you to buy any ingredients that you need to make a pizza. If you try to make a pizza without the valid ingredients, the oven won't let you make a pizza. It's a very simple game for now, but we may continue to develop it over time as a personal project as we feel that there's a lot of potential here.

## 5. Controls:

W - Move Forward

A - Move Right

S - Move Backwards

D - Move Left

Q - Rotate Left

E - Rotate Right

F - Interact with Customer, Thief, Oven

O - Interact with Ingredient Order Menu

Mouse Control - Used for rotating the character

Backspace - When in the ingredient order menu, this will close it.

Enter - Starting the game from the main menu

## Gamepad Controls:

We do not currently have these implemented, but we will try to implement them before the demonstration of our game on Monday.

## 6. Lighting Description:

Inside of our game, we have a global lighting setting, three positional lights, and a spotlight light. The global lighting is set to a very dim setting, as the skybox for our game is a sunset and we also wanted to show off our ambient lighting. The first positional light is a lamp that sits above the door of the restaurant, illuminating the Gordon's Pizza sign above our door. The other two positional lights are two lamps inside of the restaurant (one in the dining area and one in the kitchen area) that provide ambient lighting for the customers and the players inside of the restaurant. Our spotlight is a light that shines on the billboard outside of our restaurant that advertises Gordon's Pizza.

## 7. Changes to Network Protocol:

We currently do not have any changes to the network protocol.



## 8. TAGE Changes:

The only changes that we made inside of the TAGE engine were updates to the HUD manager. Because of all of the different interactive objects in our game, we needed to have different prompts show up during the game. The original HUDManager only provided two HUDs. We added 6 more to add up to 8 total HUDs, as when we would try to display prompts on the same HUDs, they would overwrite each other.

## 9. Game Statement:

The genre for our game is Simulation, the theme is everyday life, the dimensionality is 2D even though it is set in a 3D world, and the activities utilized in our game are cooking, exploration, and budget management.

10. Explanation of where the project requirements are visible:

**Physics:** The whole restaurant is protected by physics walls; if the player model tries to walk through the walls, they won't be able to, same for other furniture objects without chairs.

**3D Sound:** The music in the restaurant is played from the speaker object; if you walk closer to it, you hear the sound louder. You can also toggle the music by walking up to the speaker and turning it off/on.

**AI:** There are two AIs in our game. One is the customer AI, who walks into the restaurant, places a pizza order, sits in a chair, waits for a few seconds, and comes back to the counter to pick their order. The other one is ThiefAI, who comes up to register and steals 5\$ and tries to run away.

**Terrain:** Visible terrain is throughout the map, but there is no terrain height map.

**Skybox:** Visible skybox throughout the game

**Light:** Multiple Ambient lights with a visible model on the ceiling, you can walk up to it and see.

**HUD:** Each game's logic is printed on a different HUD. We have a total of 8

**Animation:** Animated player model, slight issue with the speed, so it looks like it's shaking.

**Hierarchical Scenegraph:** When a pizza is cooked, a pizza is added to the player's hand and it goes away when the player walks up to the customer.



## 11. Requirements failed to implement:

- Networking
- Selecting an avatar for the player characters
- Terrain heightmap with physics plain ground

## 12. Techniques That Went Beyond The Requirements

I don't believe we had any techniques that went far beyond the requirements.

### 13. Contributions of Team Members:

#### **Max**

Obj + texture: player, register, mushroom, cheese pizza, pep pizza, sauce can, soda cup

Game logic: Customer AI, Thief AI, Networking (sort of), Controls

Game feature: Serving the customer, stopping the thief from taking money from the register, gaining money on successful serving of pizza, being able to move the player around using 6 different keyboard controls.

#### **Sam**

Obj + texture: Counter, pantry shelf, restaurant, table, chair, signboard, pc, light, oven, speaker.

Game logic: inventoryManager, gameLogic (Bake pizza), Physics Builder, playerController, ordering pizza, placing all the objects, ordering ingredients.

Game features: sound, physics, animation, terrain, HUD.

#### 14. List of Assets We Created Ourselves

- Player model and texture
- Oven model and texture
- Register model and texture
- Mushroom model and texture
- Cheese pizza and texture
- Pepperoni pizza and texture
- Sauce can and texture
- Soda cup and texture
- Restaurant model and texture
- Counter model and texture
- Pantry shelf model and texture
- Table model and texture
- Chair model and texture
- Computer model and texture
- Light model and texture
- Speaker model and texture



## 15. Permissions for Items We Didn't Create

Sunset Skybox Texture was obtained from

<https://opengameart.org/content/xonotic-skyboxes>, and it was created by the user "Calinou". The license to use this skybox is provided here:

<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

The terrain texture was obtained from

[https://polyhaven.com/a/mud\\_cracked\\_dry\\_riverbed\\_002](https://polyhaven.com/a/mud_cracked_dry_riverbed_002), and it was created by Dario Barresi and Dimitrios Savva. The license to use this texture is provided here: <https://polyhaven.com/license>.

The sounds were obtained from,

<https://pixabay.com/service/license-summary/>

## 16. Names of Computers Tested On:

WOLFENSTEIN

MYST

DONKEYKONG