

# Pizza Boy: Documentation

---

Winter 2018 CS 432 Final Project

## Introduction

---

### Developers

- Collin Barlage
- Matthew Wagar

### Concept

FPPS (First Person, Pizza Shooter)

## Users Guide

---

### **NOTE:**

*Because of the amount of texture loading we are doing, Pizza Boy may crash a few times before a successful load. However once it has successfully loaded, it should work consistently after that. This error is caused by allocating to much space for the Object class.*

### Installation

- Open the build solution in Visual Studio
- Build the project and run

### Controls

- W,A,S,D – Move around the world
- Left click – Shoot pizza
- Q – Quit game
- Space bar – reset game

## Implementation Details

---

### Main Menu

The main menu / endgame menu was implemented as a single cube underneath the scene. This cube has a 1280x720 texture on it with the menu image. A stationary camera CAM2 is pointed at the cube, and an invisible click box is drawn over the 'start' portion of the menu. When the user clicks this the game begins and the active camera changes to CAM1.

## First Person Camera

The first person camera was implemented by using `glutWarpPointer()` and other glut functions to take control of the mouse and track its movement. Once these parameters are accessible we simply rotated our camera class based on these parameters

## Pizza Projectiles

Pizzas can be shot from the position of the camera. This is implemented by creating 10 pizzas on startup, positioning them below the scene. They are spawned (*moved*) when the user clicks, and animated on the vector of the angle at which the camera was when shot. This vector is created from the FPScamera data. Once 10 pizzas have been shot, the first one that was shot is recycled and respawned. This way the user can shoot infinite pizzas without creating a large amount of geometry.

## Particles

Particles are implemented by simply using `GL_POINTS` when calling `glDrawArrays()` on a vector of vertices. When shooting a house, the colors are randomly chosen from the pallet of the house (red white and brown for the door). The skulls emit black particles

## Collision Detection

Collision is implemented by looping though all projectiles, and then looping through all targets. Simple distance computations are done in each iteration of the nested loop.

## Work Allocation

---

Feature	Developer
First-person camera	Matt
.obj parser	Matt + Collin
Main menu(s)	Collin
Pizza model + textures	Collin
House model + textures	Matt
Shooting projectiles	Collin
Particles	Collin
Collision detection	Matt
Lighting	Collin
Texturing shaders	Matt

## Citations

---

- Libraries
  - [OpenGL](#)
  - [GLUT](#)
- Software
  - [3DS MAX](#)
  - [MAYA](#)
  - [Adobe Photoshop](#)
- Textures
  - All textures from Google Images, however they were Photoshopped to be seamless by Collin + Matt
- Models
  - House created by Matt with Maya
  - Pizza created by Collin with 3DS MAX
  - Skull downloaded from [Free3d](#), however *heavily* altered in 3DS MAX