

Prasad Prabhu and Jeffrey Tan
CSC165 Final Game Project
5/13/23

PDF Guide

1. the name of your game, your names, and your 165 section number(s)

- Game Name: Apocalypse Arena
- Team Member Names: Prasad Prabhu and Jeffrey Tan
- Section: 2

2. at least one image (screenshot) showing a typical scene from your game being played



3. instructions for compiling and running your game, including the network server

1. Firstly, open the CSC165 Server folder, and run compile.bat.
2. Then, run run.bat. (No configuration needed)
3. Next, Open the CSC165 Client folder, right-click run.bat, select Edit, and replace the existing IP address with the IP address that the server is running on (if running the server and clients on a single machine, it will be the same IP address).
4. After that, run compile.bat in CSC165 Client.
5. Lastly, run run.bat to start the game.

Note: After two clients join the game, the server has to be reset for the next pair of clients to join the game.

4. any special device requirements, such as particular input device(s)

- The game plays best on a keyboard and mouse. An Xbox controller also works but only has the basic controls available, and it's not as smooth.

5. how to play your game, including what things happen and how the scoring works

- When you first load into the game, you are given a temporary avatar, and you are placed in a small town with various items in them. In that town are two characters you can pick from, and pick them by walking into them, you will start the game. Once you pick a character, you are teleported into the center of the arena, and monsters will begin spawning (if the network is running) and will continuously spawn and attack the player. For scoring, the overall goal is to survive as long as possible and to level up as high as possible. At around level 30, a "You Win" message will be displayed. Also, for the scoring, the player can kill a monster, and it will drop an XP orb. If the player picks up enough XP orbs, they will level up. At certain levels, the player will get new skills.

6. what player controls are available (what keyboard/gamepad buttons do, etc.)

- Keyboard
 - WASD Movement
 - Mouse Look Rotation
 - Mouse Click Attack
 - C Spawns Monster (for game testing purposes)
 - LSHIFT Teleport
 - QE Zoom in/zoom out camera
 - <> Minimap zoom in/zoom out
 - L Turn the positional light in the town OFF/ON
 - X Toggle XYZ axis ON/OFF
 - 8 Select Avatar 1 (for game testing purposes)
 - 9 Select Avatar 2 (for game testing purposes)
- Controller (Xbox)
 - Left stick Movement
 - Right stick Rotation
 - Button 0 / A Teleport
 - Button 2 / X Attack
 - Button 1 / B Turn the positional light in the town OFF/ON
 - Dpad left/right Minimap zoom in/zoom out
 - Left/Right Bumpers Zoom in/zoom out camera

7. a list of initializations done in the scripting

- baseSpeed
- sprintSpeed
- teleportDistance
- startingHealth
- startingLevel
- startingExperience
- startingSkillPoint
- atk
- fireballLv
- fireballTravelDistance
- avatarOrbiterLv
- orbiterSpeed
- circleLv
- xpGainedPerOrb
- monsterSpeed
- monsterHealth
- monsterAtk
- teleportCooldownTime
- timeBetweenPassiveHealing
- passiveHealingAmount

- upgrade1
- upgrade2
- upgrade3
- upgrade4
- Upgrade5
- Upgrade6
- Upgrade7
- upgrade8
- baseSoundVolume
- maxLevelXP
- arenaInvisibleWallDistance
- icecreamHealthAmount
- terrainY
- terrainScale
- planeY
- planeScale
- grenadeX
- grenadeY
- grenadeZ
- townX
- townZ
- orbiterYHeight
- angelX
- angelY
- angelZ
- ambientLight
- avatarX
- avatarY
- avatarZ
- spotlightHeightAboveAvatar
- gravity
- grenadeBounciness
- rcMin
- rcMax
- orbiterLvl1Scale
- orbiterLvl2Scale
- orbiterLvl4Scale
- rangerAttackForce

8. a brief summary of any changes (or none) that you made to the network protocol

- All client (ghostAvatar) location, orientation
- All client (ghostAvatar) info (class, health, level, skills level info)
- Server send change Skybox message to all client every 30 seconds
- Server respond to Need NPC (Mini zombie) request from the client
- Server respond to Need NPC (Boss) request from the client

Monsters (mini zombie):

- NPC info (location, target client)

Boss:

- NPC info (location, target client)
- NPC controller behavioral tree check condition (isNear, isNotNear)

9. a list of changes and additions that you made to TAGE

- Additional methods were added to the GameObject and Camera class
- BounceController class was added to the nodeControllers folder
- CameraOrbit3D camera controller was added to tage
- Added additional HUDs in HUDmanager class to display text.

10. a statement indicating the (1) genre, (2) theme, (3) dimensionality, and (4) activities utilized in your game (see week 1 notes [chapter 00] for examples)

- Genre
 - Action. 3rd person shooter
- Theme
 - Modern Coliseum/Arena + Zombies

- Dimensionality
 - Player motion: 3D
 - Object/NPC motion: 3D
 - Camera Motion: 3D (part chase controller and part orbit controller)
 - World Dimensionality: Ground
- Activities
 - Combat
 - Exploration (has the ability to walk around unconstrained)

11. an explanation of where (in the game, not the code) each project requirement is satisfied

- External Models
 - Prasad: created the ranger (larger enemy) and icecream model/UV mapping
 - Ice cream models can be found inside the arena. Ranger can be found inside the arena if the server is running.
 - Jeffrey: created all other custom models (Avatars/Monsters/Town/Skills/etc)
 - Can be found throughout the game.
- Networked Multiplayer
 - The game is playable by at most 2 players. When two players join the game, they are both initially placed inside the town, they can then select their characters, and their selection will be shown to the other player. The players can see each other's avatars and their position/orientation.
 - The game can run in single-player mode when the server isn't running. NPC, AI, and Physics cannot be tested if the server is not running.
 - Players can select their character avatar at startup while in the town by touching one of the avatars or pressing the "8" or "9" keys. The game will start (monsters will start chasing, and the player will be bounded inside the arena) only if the player touches one of the avatars in the town to select their character and start the game.
- Scripting
 - There is a script file in the assets folder named InitParams.js and a helper ScriptController class. The script file initializes a large number of game attributes
- Skybox
 - When you first join the game and are placed into the town, the avatar is in "ghost mode" and can walk through walls. To view the skybox, simply walk through the town walls, and you will be placed outside and can view the skybox. Do not select the avatars inside the town, as that will teleport you inside the arena and prevent you from leaving.
- Terrain
 - When you first join the game and are placed into the town, the avatar is in "ghost mode" and can walk through walls. To view the terrain, walk through the town walls, and you will be placed outside. Do not select the avatars inside the town, as that will teleport you inside the arena and prevent you from leaving. The terrain can be found outside the arena, and you can move up and down the

terrain. Press “8” or “9” to equip the avatar model, which can make movement more realistic. The terrain is not used for gameplay.

- Lights
 - Global ambient light is always present, and there are two additional lights
 - A positional light is placed inside the town at the location of the lamppost. This light can be turned on/off by pressing the “L” key.
 - A spotlight is placed above the player and pointing downwards. This light will always follow the player.
- HUD
 - Is visible at all times when playing the game.
- 3D Sound
 - The game includes multiple 3D action-specific sounds as well as background sounds.
 - Background
 - An ambient bird/wind audio can be heard at all times and is a background sound.
 - Action/3D sounds
 - Dog barking: can be heard inside the town, positioned at the dog location
 - Footsteps: can be heard when moving, positioned below the avatar
 - Turning the light on/off: sound is positioned at the lamppost but can be heard over a long distance.
 - Attack: can be heard when attacking, positioned at the avatar location
 - Icecream pickup: can be heard after picking up the ice cream, positioned at the avatar location
 - Ranger enemy: can be heard inside the arena when starting his attack, positioned at the ranger location
- Hierarchical Scenegraph
 - The town/house is a hierarchical object (the player first spawns inside the town)
 - Angel skill ability is a hierarchical object (obtained at level 7)
 - Flaming orbiter disks ability is a hierarchical system (obtained at level 5)
 - Purple circle “prestige status” is a hierarchical object (obtained at level 20)
- Animation
 - Avatars and monsters have animations
 - Avatar: the feet and hands will move
 - Monster: the hands move
- NPCs
 - In the game, Rangers and Monsters are non-playable characters (NPCs) that follow predetermined behavior patterns. The Ranger, for instance, is programmed with a behavior tree that instructs it to patrol the arena until a player approaches. Once a player is detected, the Ranger's behavior changes: it becomes aggressive, enlarges in size, emits an attack sound, and attacks the player. To perform its patrolling behavior, the Ranger follows a circular path around the arena. This is achieved by first converting its Cartesian coordinates to polar coordinates, calculating its path based on the polar coordinates, and then

converting back to Cartesian coordinates. This process results in the Ranger rotating around the arena in a circular motion.

- In contrast to the Rangers, Monsters in the game are controlled using simpler methods. They spawn in random locations and start chasing a targeted client. The server determines the Monsters' spawn locations, and randomly selects a target client. This information is attached to a message and sent to all clients through the SendSpawnMonsterMessage function, which is called every 5 seconds on the client side. This synchronization of the Monsters' movements and target among all clients ensures a consistent gameplay experience for everyone.
- The ranger and monsters can be found inside the arena when the network server is running, and the player has properly selected their avatar in the town.
- Physics
 - The ranger's attack implements physics and uses a physics engine. The ranger's grenade attacks will bounce off other grenades, the floor, and the avatar.
 - The game uses JBullet's ability to detect which objects have collided. When the game detects the collision between a grenade and the player, the player's health will be decreased.
 - Physics can be tested by triggering the ranger's attack by getting close to the ranger inside the arena. Physics can only be tested when the network server is running since the ranger won't appear unless the server is running.

12. a list of the requirements that you weren't able to get working

- All requirements are working

13. any technique you used in your game that goes beyond the requirements

- Day/Night cycle
 - When the server is running, the skybox will change every 30 seconds.
- Model texture paint via Blender
 - After UV-unwrapped the models, some models were painted directly using Blender texture paint. For example, trees, dog, and ghost.
- Increasing Difficulty System
 - Every 30 seconds, the speed of the monster will increase slightly. There are a total of 10 difficulty levels, where each level has increased monster speed. And it is controlled by the server side so that every client will have the same difficulty.

14. the contributions of each team member, including who designed which model(s)

- Prasad:
 - General Gameplay mechanics
 - Sound
 - Terrain
 - Lighting
 - Skybox
 - Scripting
 - HUD

- Physics
 - Models: Ranger (bigger enemy) and icecream
- Jeffrey:
 - General Gameplay mechanics
 - HUD
 - Skybox
 - Hierarchical SceneGraph
 - Networking
 - AI
 - Animation
 - Models: Every model except the ones made by Prasad and built-in TAGE models.

15. a list of items that you created yourself (models, textures, heightmap, etc.)

- All items in the game except for TAGE shapes were created ourselves.
 - Items we have made:
 - All sound
 - Heightmap
 - Animations for avatar and monster
 - Models/Textures for avatar, enemies, skills/abilities, items in the starting town location (house, fence, tree, dog, lamp, etc).

16. evidence of permission to use any item (models, textures, etc.) not listed in #15

- All models were either handmade or a built-in TAGE shape. All animation, textures, sounds, etc., were made by either Jeffrey or Prasad.

17. which RVR-5029 lab machines (at least two – it's networked!) on which your program was tested and is known to work correctly on.

- METALSLUG and METROID

18. Additional Notes

- The top-level package name has been renamed from MyGame to "finalProject". Also, "a3" has been renamed to "CSC165 Client".