Craig Shefchek
Tyler Crabtree
Spencer Kitchen
CS: 447
Project 2: Group game

Armored Glove

Game Overview:

The game will behave similarly to the arcade game Gauntlet, with our own spin and style on it. The team plans on completing a game where the party of players explore various levels of a dungeon while attempting to fight their way to the end. Throughout their adventure, the party will pick up temporary power-ups to aid them in their quest. Enemies will spawn out of Monster Spawners placed throughout the map. The players must destroy these Monster Spawners to stop the flow of monsters. During the journey, keys will need to be picked up to progress through the various locked doors throughout each level. The subsections below help break down certain components of the game.

<u>How will the game be played?</u>
Player controls:
A-key: Move the player's character left (west).
W-key: Move the player's character up (north).
S-key: Move the player's character down (south).
D- key: Move the player's character right (east).
A-key & W-key: Move the player's character in the northwest direction.
D-key & W-key: Move the player's character in the northeast direction.
D-key & S-key: Move the player's character in the southeast direction.
A-key & S-key: Move the player's character in the southwest direction.
Spacebar: This will trigger a character's light attack.
Enter: This will trigger a character's hard attack.

The character will perform attacks in the direction that the character is facing, (generally, the last directional key pressed). The character will auto orient himself/herself allowing attacks to be made with spacebar or the enter keys.

<u>What types of interactions are possible?</u>
Players will work their way through levels. These levels will contain power ups for the player to pick up (by collision). The player will be able to fight enemies; enemies will take away the

player’s health if a collision happens. A player can defeat an enemy if the a player’s attack collides with the enemy (projectile or melee attack). Also, a player can find keys to then unlock doors to traverse to the next/lower dungeon. When a player gets hit by an enemy, they will lose health. Once a character has ran out of health, they will be unable to play until their teammates revive them by getting to a checkpoint.

What are the visual entities in the game?
Separate entities will be used for different playable characters that the users will have control over. There will be enemies the are attempting to track down the players to “attack” them. There will be Monster Spawners that are releasing new enemies the into the current level until destroyed. Power ups, that will give the player who finds it temporary boosts in stats: increased attack and healing. The player will also need to find keys to be able to progress through the level.

What will the player do?
The main goal is to find the exit to the next level. This generally means attacking enemies, working as a team (if playing multiplayer), and using keys on doors to progress closer to the exit. Once the exit is reached, the next level will be loaded (if the player is not at the end of the game).

What makes this idea interesting, or why do you think this will be fun?
The idea behind the game is simple; make your way through a dungeon to lower levels of the game. The player will have to explore each level. In their explorations the player will come across hoards of enemies to fight and destroy the spawners to prevent more from appearing. Balancing attacking, exploration, destroying monster spawners vs enemies will create a fun dynamic. Also, players will have to search for keys and along the way to add to the overall tension and sense of adventure. Finally. finding different power-ups will add another layer of complexity for the player.

Development Strategy:

Spencer Kitchen developed a grid-based system in his individual project. Our group will be using said grid to build upon to form our game. The system will aid in moving a tracking enemies, as well as attacks and collision. Some modifications will have to be made to account for the larger world coordinates.

The most difficult part of the project will be implementing the networking. Preliminary research into the topic is needed, along with speculation with the implementation of networking. Given that this is the first time for the entire team creating a networking multiplayer game we won’t know what challenges we face until we tackle the issue. However, the rest of the development plan is depicted below.

Development Plan (subgoals):

Develop Client and Server:
Player 1 will host server and the game client. Everyone else will connect to the server through clients.

Develop Grid based map generator:
All maps will be static with the server and every client containing a local copy. This will reduce the amount of data that needs to be transmitted over the network.

Create enemy Ai for tracking player:
Will be a shortest vector algorithm.

Will only track players in a certain range:
Create 8 direction player movement. We want players to move diagonal.

Connect up Grid map with the server and client:
Add in player and enemy and make sure they are updating properly over network

Implement unique game features:
Monster Spawners, power ups for character, items, score, health, stats, etc.

Splash Screens:
Polished screens to guide players from game transitions.

Milestones:

Complete game maps: This is the simplest idea to execute. Most of the game is dependent on the map, so we will finish this aspect first: A rough date for this is November 10th.

Complete character actions: A local game should be able to be playable with one character. Our implementation of a solo game should be able to be completed before networking is implemented. A rough completion date is November 20th.

Implement networking: This will consist of creating a multilayered game over at least two systems. This is the final milestone to be complete, so a rough completion date is November 30th.

Technical Showpiece:

Networking: Smart Server, Dumb Client Model
The server will do many of the calculations and also know the correct current state of the game at any given time. The Client on the other hand will only handle animations based on data received from the server and also register their local game state with the server.
Player 1 will host the server along with an instance of the client. An example is shown in "figure 1."

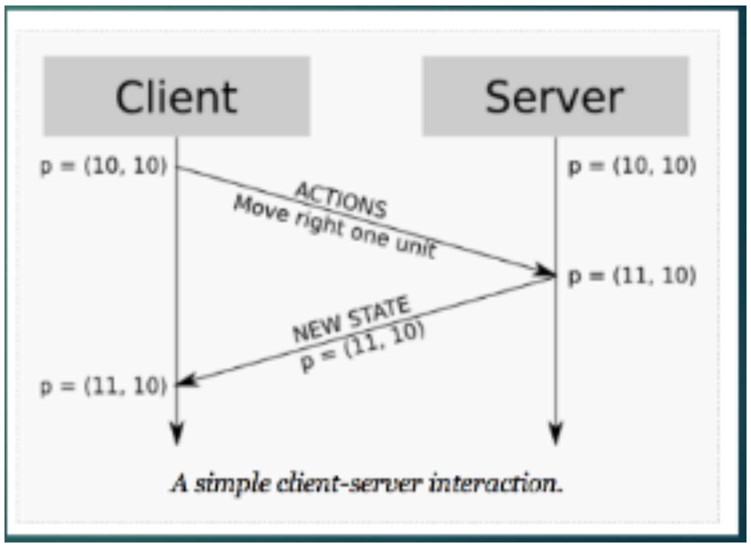

Figure 1

High Bar:

1. Players have multiple types of attacks:
    - Strong attack – Slower, more powerful.
    - Weak attack – Faster, less powerful.
    - Giving the player the choice of what attack that they may want. A slower more powerful attack, forcing the player to “charge” their attack leaving them vulnerable. Or a faster normal attack that does a moderate amount of damage (low bar).
2. Up to 4 player support:
    - For the low bar as a team we only intend on tackling 2 player networking.
3. Up to 4 classes (Knight, Wizard, Archer, Warrior):
    - Each class has unique animations, attacks, and stats. For the low bar we have separate classes for players to select from, but they have no real difference other than appearance.
4. More than 2 power-ups in game:
    - This could be near limitless. Power Ups that do not wear off, different weapons and armor.
5. More than 2 levels in game:
    - This could be done with randomized maps. If time permits maps could be randomly created, allowing for a near infinite game play. Along with the randomized unlimited maps the enemies could get stronger, faster, and may even smarter (Pathfinding).
6. Efficient pathfinding algorithm:
    - The original idea for enemy pathfinding is the shortest vector. The will work but it has issues, enemies will get caught on walls. Implementing a smarter pathfinding element will fix this issue.
7. Spawners that fight back, essentially acting like a level boss:
    - As the levels progress the monster spawners can take on a different form, moving throughout the map, and perhaps even fighting back. Perhaps if the players are far enough away the spawners try to keep that distance, forcing players to track them done. When the players get close enough, the spawner could turn and fight, all while spawning new basic enemies.

Low Bar:

1. Melee attacks: A player will be able to perform a short-ranged melee attack.

2. Long-ranged attacks: The player’s long-ranged attack will be a limited ranged projectile.

3. Multiple levels: Have at least 2 expansive levels. Hard coded levels to ensure entertaining gameplay.

4. Multiplayer: At least a 2 player game.

5. Unique Classes: Visually the different classes will be unique(Knight, Wizard, Archer), but their overall stats and attacks will be the same.

6. Power-ups: At least 2 different types of power ups will be available.

7. Attack booster: This power up will only last a limited time.

9. Healing item: This boost increases health

10.Monster Ai: Path finding will be the shortest vector.

11. Monster move towards players and cause damage based on collision.

12. Health counter: A player will have health points. These will be lowered when contact with an enemy is made.