Revised Design

080008650

December 16, 2011

Listed below are the differences between the implemented game and the previous design document. The differences exist primarily due to a time constraint. The features mentioned in the original design are intended to be implemented in future development unless explicitly mentioned in this document.

Environment

The game uses a map file present on the phone storage. If the map file is not found the game creates and uses a default map. The map does not contain any spawn points or power-ups. Music and sounds effects are also not present.

Opponents

Both human and AI opponents are available in game. The game offers are single and a multi-player player mode. The single player offers a game versus one, two or three AI controlled players. These opponents use a simple AI protocol which has been adjusted to be easily defeated by the human player. The AI opponents move around slowly to allow the player to hit them more easily. The opponents also have reduced accuracy.

The multi-player allows two or more human players using Android phone clients. The game requires a dedicated server running on the network. The maximum number of players is determined by the server configuration. The clients have to know the address of the server to connect. These features are implemented as a starting point and in future can be improved for easier connections to the server as well as remote configuration.

Rules/Mechanics

Player control and line of sight

The analog stick design is used to control the character. The distance between the touch point and the centre of the analog pad determines the acceleration of the character. The analog pads were seen as too small for most of users fingers, hence if the user's finger moves outside of the analog pad, the direction is preserved however the input is capped by the maximum acceleration set within game.

The direction the character is facing is determined by the input from the right analog pad. When the user releases the right analog pad, the character fires the currently equipped weapon.

The line of sight mechanic has not been implemented. The future development will feature games with mode allowing to have the fog of war turned on and off.

Health and lives

The game features each player having five lives at the start of the game. Once a player has been hit with a bullet, a life is lost. Once all of the lives have been lost the play is dead. The original design featured both health and lives. This was mainly decided to support use of different weapons with varying damage. Once the different weapons are implemented different health modes can be introduced.

Weapons

Only the initial basic weapon, the pistol, has been implemented. This has proven to be sufficient by itself to provide a simplistic game mode, forcing the player to focus on dodging and perfecting aiming as opposed to relying on additional weapons and power-ups.

Scoring and goals of the game

No score is tracked in game. The winning player is the last one to survive, hence the goal of the player is to survive the match.

Architecture

States

The states of the game are kept similar to the original design. The current states feature:

- A title screen allowing the player to choose between single player and multi-player.
- An AI selection screen allows the player to select the number of AI opponents and hence start the game.
- A join screen allows the player to select and address of the server to connect to and create a connection to it.
- A lobby screen displays the number of players joined, which of them are ready to join, and allows the player to set himself as ready.
- The game state presents the player with the game which lasts until only player is left alive, directing the player to the score screen.
- A score screen displays the winning player.

Game loop

The single and multi-player modes share the same loop.

- The game is updated
  - The current game state is updated
  - Update the players position, taking in an account of the map obstacles
  - If the movement packet delay as been reached then send out a movement message
  - If the player has fired, send out a fire message
  - Update the opponents of the game, if the player is a remote connected player and has sent a message, carry out an appropriate action
- The game is rendered