Project Specification

Team3
Maverick Chen, mzc
MinSun Park, minsunp
Jiangtian Qian, jiangtiq

1. The product backlog: a complete list of all functionality (i.e., the actions) of your project, and an English description of each action. We strongly suggest that you organize these features into groups/modules based on related functionality.

Home Page:
a) Login: Users who already registered can login using their user id and password.
b) Register: New users can register using their email addresses. They can choose their user id, which cannot be the same with the existing user id. The fields to complete are: id, password, email, name, and more optional fields.

Choose Team Page:
After a user has logged in, they will be directed to a page where they can choose their teams. A user can choose one of two teams, which are each comprised of two players. The READY button is disabled if you choose a team that already has 2 members in it, or when you haven't selected any team. The game starts when all four players have clicked on READY button.

Game screen:
The game screen opens once the game starts. There are two main panels in game screen: one where the player walks around a maze (travel panel) and the other one where all your information is displayed (info panel).

In Travel Panel, a player can:

a) Shoot using mouse clicks:

1) Players can shoot other players. The player's health points decrease when they get shot. When one's health point is zero, he needs to stay still and wait for several seconds to recover.
2) Players can shoot little robots. Killing robots will add bullets to players weapon collection or HPs for the players.
3) Players can shoot the castle in the other team's base. A team loses the game when their castle is destroyed.

b) Move around the maze: Players can move around and find new land in the maze using up and down arrow keys.

c) Collect bullets: Users can pick up bullets from the ground when they step on the bullet during their travel.

The Info Panel displays:

a) Health Points bar
b) Your ID
c) Your team name
d) Your custom maze map, which is constantly updated depending on where you've travelled, or when you've met your teammate and shared the map.
e) Number of bullets you currently have.

WIN/LOSE page:
When the game is over, it shows if you've won or lost, along with the details about your HPs and the number of times you've been attacked. Player can click button to save the score if he wants.

Work Division: We divided up our whole project into separate roles we mainly play.

a. Setting up initial multiplayer game project framework, take care of everything regarding multiplayer options. - Maverick
b. Creating art assets necessary for the game (robots, castle, designing maze, character art, wall art, ...) - Everyone
c. Implement overall game flow - choosing a team, keeping track of HP and bullets (stepping on HP or bullets increments these), determining winning or losing, sound effects - MinSun
d. Implement user interaction - shooting mechanism, character movement with arrows, travelling updates current maze - Jiangtian

2. *The first sprint backlog: a complete list of the functionality you will complete during your first sprint, and how that work is allocated among your team members.*

Overall, we will have a working application with a simple character and simple maze, where different players can walk around using arrow keys. There is no winning or losing, no shooting down little robots and castles yet.

a) Create node js application for a multiplayer game - let multiple people play the same game at the same time, with real time updates. (Maverick)
b) Create simple character asset and design 2 mazes. (Everyone)
c) Design the web application framework. Player can choose a team and start a game. A player can walk around to collect HPs and bullets on the ground. (MinSun)
d) Player pressing arrow keys (up and down) lets them walk around the maze. Walking around updates their maze on the right. (Jiangtian)

3. The name and Andrew ID of the product owner for the first sprint.

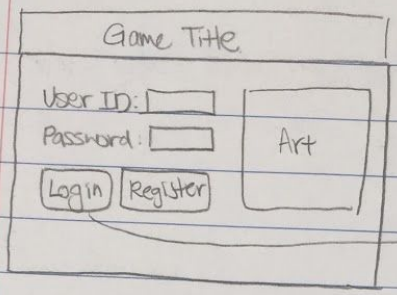
Product owner: MinSun Park (minsunp)

4. A complete implementation of the data models used by your application. This may be written in as Django models, SQL, or some equivalent style implementation if you use another framework.

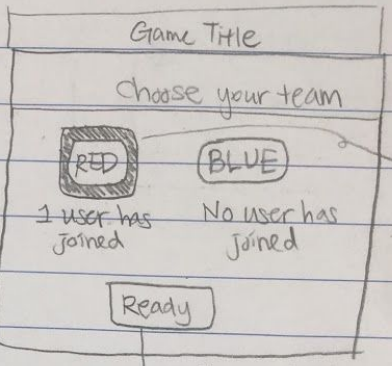
Refer to User.js. Since this will be a real time game, we will only be storing general information about the user on the database for the purpose of logging in and maybe storing number of wins and losses if time permits. Most of the player data such as position and health will just be stored on the game server at runtime for better performance.

5. A complete set of drawn wireframes or HTML mockups for your application, for all non-trivial views within the application.

Home Page:



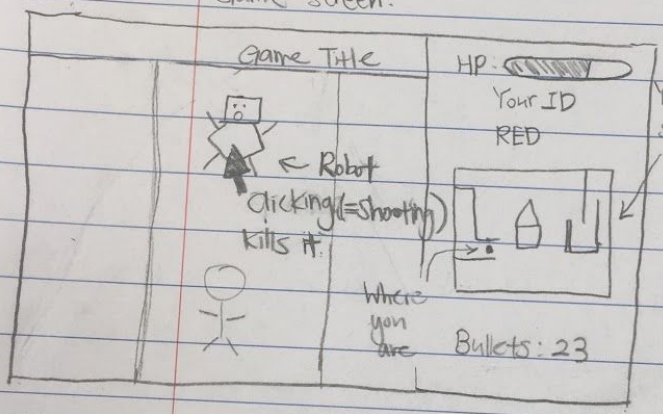
Logged In:



Clicking on a ~~team highlights~~ it.

Game starts when there are exactly 2 members in each team. The <Ready> button is disabled when you haven't selected any team, or the team you selected already has 2 members and all of them clicked on "Ready" button.

Game screen:



Killing robots gives: HP or bullets

↑ ↓ Control movement w/ arrow keys.

HP logic: keeps decreasing as time passes. You should pick up HPs.

When you reach the castle:

3 stages of destruction.