**Software Requirements and Design Document**

**For**

**Group Tanx**

Iteration 3

**Authors**:

Blake A

Andrew T

Nolan M

Alex G

# Overview (5 points)

Our goal for the project was to develop a browser based reinterpretation of the Wii play game Tanks that uses mouse and keyboard for controls.Essentially, we are created a point-and-click bullet-hell-esque game featuring one user completing different levels by killing all enemies within the level before they get killed. We have a 25 stage game where every level either has a different layout or different amounts and types of enemies the player must destroy. We have made the controls of the game much more usable and less frustrating than the Wii, and also have made the game a little more forgiving to players while still offering a few somewhat challenging levels and features. There are 5 types of enemies in the game, all with their special quirk, including a final boss that is meant to be made quite difficult.

# Functional Requirements (10 points)

*List the* ***functional requirements*** *in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

1. Game Menu: Will present the user with the option to play the game or quit. Should the Player choose “play game”, the app will give the user the option to select a stage or view the controls. After a stage is selected the game will start up. This menu will be developed via html.

Priority: med

1. Stage Selection Menu: Will present the user with an option to choose any one of the 25 stages and a back to game menu option. If a stage other than stage one is selected, the user will be prompted to enter a stage code. The stage codes are presented at the end of each stage so no stage can be played without beating the prior stage. Also will be implemented via html.

Priority: med

1. User Tank: Allow users to control a tank that can move and shoot around the map. Will allow users to use their keyboard to move the tank (WASD) and use their cursor to control the direction in which the tank fires.

Priority: high

1. AI Tanks: Create multiple different AI opponents to provide a challenge to users when playing levels. There are different AI tanks that will have different abilities and will progressively get more abilities as the game goes on. For example, the 1st/tutorial stage may have a tank that doesn’t move or shoot and die on one hit so users can get used to the simple functions of the game. Then, the later levels feature AI that can move faster than the user, have projectiles at higher velocities, take more than one shot to hit, etc.

Priority: high

1. Projectiles: The user tank and most AI tanks must be able to shoot projectiles on command. Projectiles must kill, regardless of where they come from, on impact of a tank. They will also only take a certain number of bounces before they disappear and tanks will only have a limited amount of projectiles on screen that the tank that shot that projectile can have.

# Non-functional Requirements (10 points)

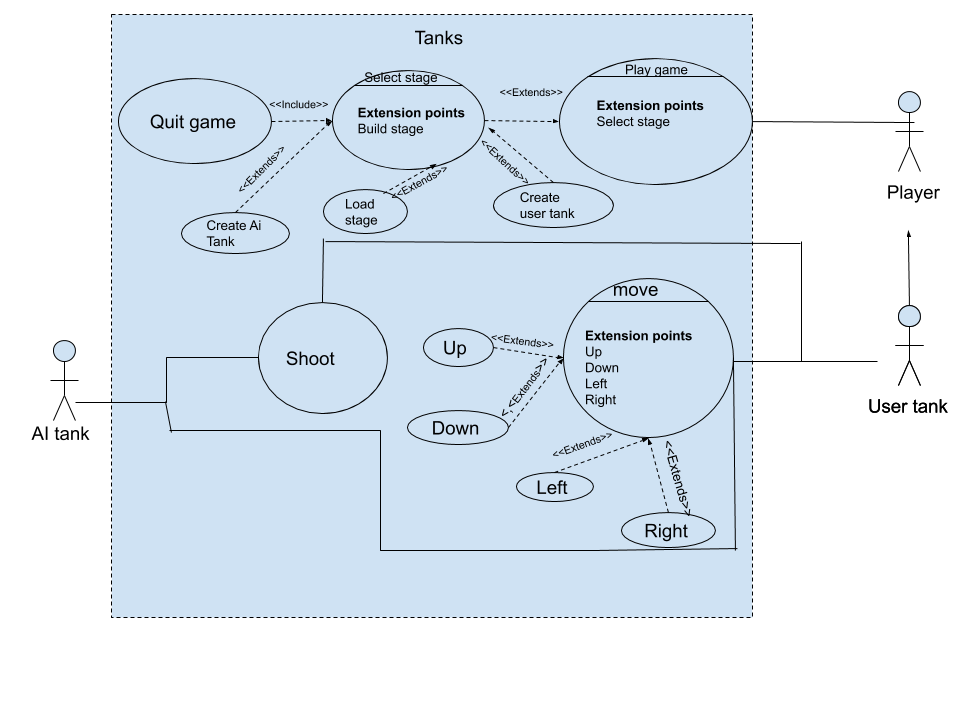
*List the* ***non-functional requirements*** *of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

* The software must run on any HTML5 supported Web Browser
* A user will not be able to access a level until the previous level has been completed
* The levels and menus load within a reasonable amount of time
* The gameplay runs relatively smooth
* Levels/menu screens are killed once the player switches to a new screen, ensure previous screens are not still running in the background

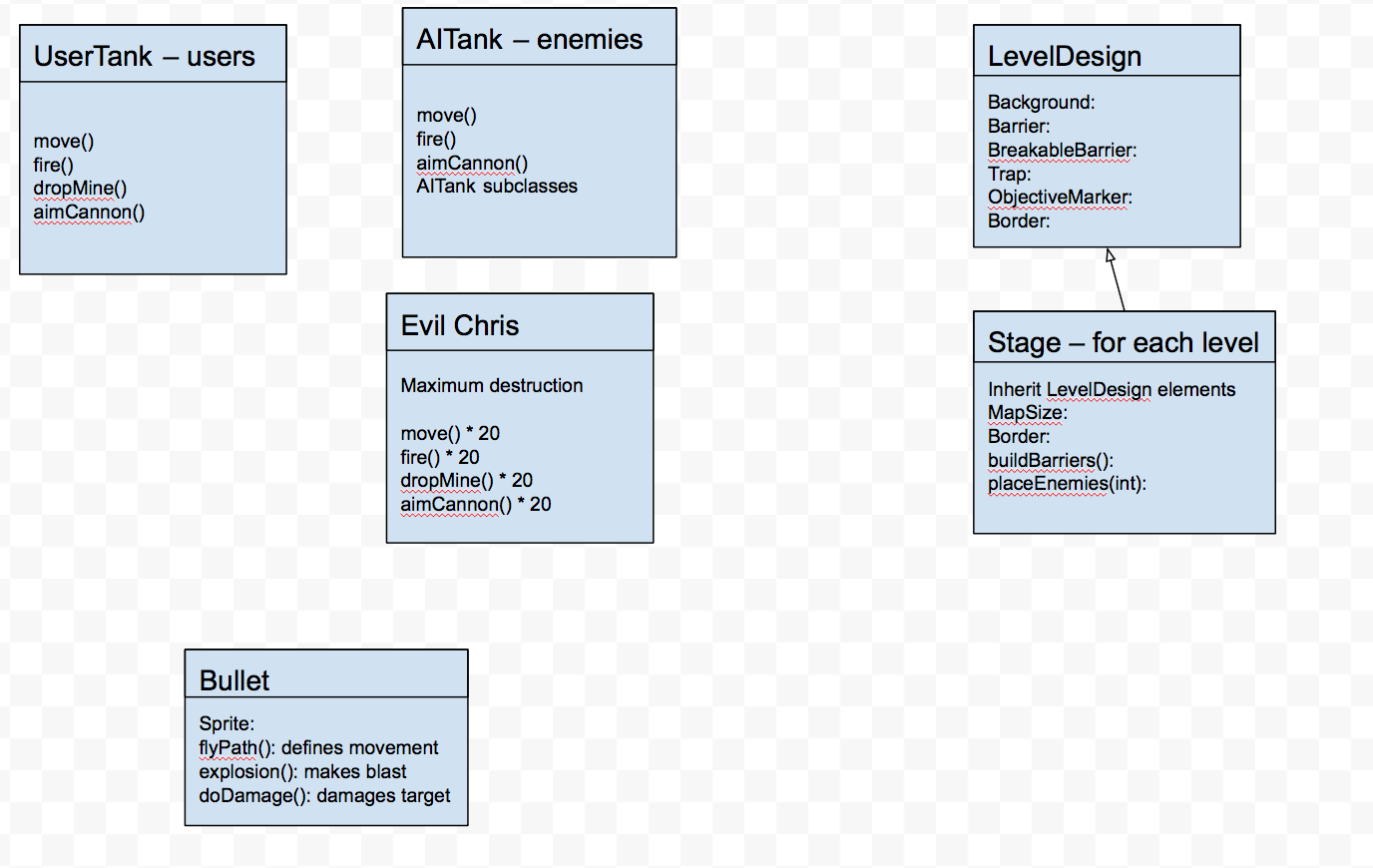
# Use Case Diagram (10 points)

*This section presents the* ***use case diagram*** *and the* ***textual descriptions*** *of the use cases for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*

***Textual descriptions of use cases****: For the first increment, the textual descriptions for the use cases are not required. However, the textual descriptions for all use cases discovered for your system are required for the second and third iterations.*



# Class Diagram and/or Sequence Diagrams (15 points)



# Operating Environment (5 points)

The system should be able to run in any operating system, as we worked on both Windows 10 and the latest Mac OS. It can run on any laptop or computer as long as it is functional and has a web browser and can connect to the internet. The application does require a server like XAMMP to be installed so the game can run on systems browser and transition levels through the html files we built. That being said, one must also have a browser that can run javascript code, which I presume almost every browser used in this era has the ability of doing so.

# Assumptions and Dependen

We’re assuming that there will not be any major issues implemented into the Phaser framework during the duration of our project as the entire implementation of our project is dependent on the functionality of this framework. We also assume users are running our project on a desktop that can support a browser game and a system that can support a local server. We also will assume that XAMPP has no framework updates or problems when hosting our project on a local server.