COMPSYS 302 PROJECT PLAN

Freedom to Etruaruta

MARCH 25, 2017

1.0 Brief Background:

Adil Bhayani *(Computer Systems Engineering / Computer Science):* Enjoys programming and have been dedicated to learning as many languages as I can to be the best of my ability. Has knowledge of coding in Java, C and Python alongside some knowledge of basic and web and app development.

Sakayan Sitsabesan *(Computer Systems Engineering):* Currently studying Computer Systems Engineering, main interests are the Internet of Things, low level computer architecture, microcontroller programming. Has previous experience using Git, C# and developing Mobile Apps.

2.0 Outline of the System:

Aim of this project is to implement a pong like game in Java using OOP principles and the MVC pattern. The game will involve at least one moving ball that can destroy walls, on higher difficulty settings multiple balls will be in play to make defending your warlord harder. Players will be able to control a paddle(s) which can be used to reflect the ball. The objective of the game is to destroy the other warlords while defending your warlord from enemy fire.

In the year 2600, four races are fighting over the planet Etruaruta, a planet orbiting around Alpha Centauri B located 4.37 light-years from Earth in the southern constellation of Centaurus. The planet once occupied by an un-civilized race called the Etruarutians has now been found to a be a major source of Qeflinda, a major source of energy that much of the galaxy is now dependent on. The Galactic Empire, Martians & the Earthlings have just landed on Etruaruta. The story of who gains control of this now valuable planet forms the plotline of this game.

2.1 Extra Features:

- Multiple Game levels (each have different ball speeds/number of balls) as the story progresses levels get more difficult.
- AI Demo Mode
- High Score system
- Two pedal mode which either follow or reflect each other.
- Skill tree for the player which is unlocked as the story progresses.
- Cinematic clips after a game is finished or as story progresses.
- Difficulty settings to allow user to have a challenging and interesting experience.
- Power ups which spawn randomly on the field and can be triggered when the ball hits them:
    - Flaming Arrows: Ability to throw explosive flaming arrows at enemy bases which causes the destruction of many bricks in the blast radius.
    - Pedal sizes (giant and tiny): Either applies to the warlord that hit the power up or to the warlord's enemies.
    - Crazy ball: When a sense of randomness gets applied to the ball(s) causing them to move erratically.

- Ball control: Allows a warlord to control the movements of the ball(s) allowing them to direct it to their enemies.
- Ghost pedals: For a short amount of time pedals become invisible.
- Replicating ball: Creates another instance of the ball mid game which is initially moving at a 90-degree angle to the ball that triggered it.
- Lottery: Randomly chooses one of the other power ups from above.

3.0 Provisional Schedule:

Week	Task
2	Project plan draft
2	Preliminary work on project including research of similar games
3	Incorporation of Junit and implementation of basic classes
3	Finalise project plan
4	Learn how to use JavaFX and incorporate it into project
4	Creation of game screen with movable pedals
4	Welcome, Help, Options & Exit Screens
4	Implement basic game functionality
5	Work on additional features (as stated above)
6	Cleaning up code and last minute fixes

4.0 Foreseen challenges:

- Understanding the process of game design and the planning that is required to create a game.
    - A lot goes in the making of a game even if it is only just 2D. Things that we must consider include but are not limited to, the storyline, luck and strategy, game physics and controls. There is also the concept of testing. Since we will be following test driven development it is important that we create the initially game structure so that it meets the provided test cases and then possibly creating our own test cases to ensure the game has no major bugs.
- Creating a challenging yet interesting experience for the user.
    - Getting a feel for other similar games to see what form of difficulty is appropriate. Possibly having three different difficulty settings to allow the user to choose. To make the game more interesting the storyline can be shown through cut scenes.
- Creating an immersive and interactive storyline which is suitable for a 10 to 12-year-old kid.
    - Running the story by kids (our brothers) who are 10 years and 15 years old. Understanding our target market will help us create a more appropriate game and to tailor it to the players.
- Ensuring the code is written so that it meets proper OOP design.
    - The theory that we have learnt from Compsys 202 will come into use to ensure that we can successfully overcome this challenge. Initially we will create a class diagram to see the relationships between the classes. This will allow us to understand things like inheritance, aggregation, composition etc. Once we understand those relationships it becomes easier to implement the classes and minimises the need for code refactoring late

5.0 Appendices:

Screen diagrams:

Figure 1: Initial Menu Screen

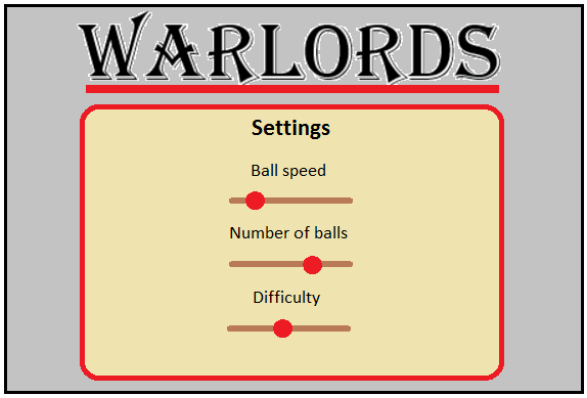

Figure 2: Settings Menu Screen

Figure 3: Help Menu Screen

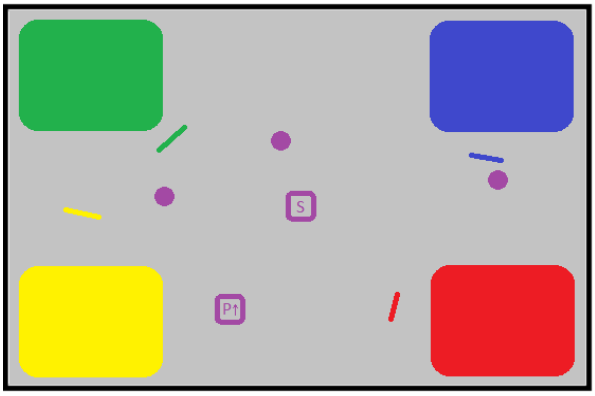

Figure 4: In game screen

Class diagram

=: