GAME PROGRAMMING (ONE CREDIT COURSE)

GAME DESIGN DOCUMENT

ON

TAKESHIS CASTLES

BY

ARJUN ASHOK (15Z309)

BALAJI MUTHAZHAGAN T (15Z312)

SANKA GOPI SUMANTH (15Z366)

Project submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE AND ENGINEERING

Of Anna University

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

CONTENTS

CHAPTERS **Page. No**

CHAPTER - 1

GAME OVERVIEW

1.1 GAME CONCEPT:

The game that we have developed is called the “Takeshis Castles”. This is a two-player turn based shooting game wherein the player one will be on one castle and player two on the other castle and both of them will have their own canons to shoot the others castle. So, while shooting the fireballs the player has to hit and destroy the others castle to save his castle. So, to save his castle the player takes up this daring adventure This game was developed for the age group of 8-24 years.

1.2 GAME FLOW SUMMARY:

The player is welcomed by the front screen with an interesting background and a game name that supports the theme. After the intro screen appears, there will be a set of options for continuing into further part of the game. This game also has a tutorial to how to play. The player is displayed like he/she will be on the castle. The player will aim and shoots the opponents castle.at first, player one will get a chance to shoot and then the camera will be move towards player two and player two will shoot now and it repeats continuously until player wins the game. When the player shoots the cannon, it is shown with a path that hits/misses the castle. This is supported with an audio to give the shooting feel to the player. Also, the destruction of the castle is also shown in the game. The status (health) of the players are displayed along with the game. At the end, the players who is having health will be the winner. An appropriate background is played throughout the game to give a feel of the game for the player.

CHAPTER – 2

GAMEPLAY AND MECHANICS

2.1 MISSION STRUCTURE:

No. of Players: Two – players (2)

No. of Levels: 1

Rules:

Level 1:

- The game starts when user clicks the PLAY in the intro screen.
- By selecting PLAY option, the game will begin.
- The player can rotate the canon left and right using the mouse.
- At this level, there are only castles with canons which are ready to shoot the opponents castle.
- The player should destroy the opponents castle by targeting the it using the mouse cursor and clicking on it.
- This would shoot a canon which destroys the castle.
- If it hits on the base of the castle, then the health of the player will decrease.
- However, the player loses if his castle is destroyed.
- In this case, he/she has to restart the game.

Restarting the game:

- If player loses in any of the levels, he/she should start playing from level 1 again.
- If the player finishes all the levels, he/she can restart the game from level 1.

2.2 IMPLEMENTATION OF THE GAME

2.2.1 TECHNICAL FUNCTIONALITIES:

Software used: Processing

Platform used: Intel i5 processor, 64-bit Windows 10 Operating System

Platform requirements: Intel i3 processor,64-bit Windows XP Operating System

Packages used: Minim for audios and video for videos.

The game works on the concept of state transition wherein each screen in the game is a state and the change from one page to another happens by changing the game state. The look and feel of the game is changed by calling appropriate functions as the state changes.

Displaying images:

The images are displayed using the PImage class and the associated functions like loadImage() and image(), The images that we have used are '.png' images. These images are resized according to the need before rendering.

Displaying background images:

The PImage class is made use of here also. The background images are of the '.jpg' format. They are fitted to the size of the screen. This game will work perfectly for a resolution of 1366 x 768.

Playing background audio and other related audios:

For this the minim library in processing is made use of. The Minim class, the loop() and play() are used for the background that plays throughout the game and the cannon shots accordingly.

Displaying messages and status:

The welcome message, scores and the current level are all displayed in the appropriate levels using suitable fonts like Forte, Jokerman and Cooper Black. The size and color of these are also adjusted to the need and situation of the game. Also, they are aligned at suitable positions in each screen. A tip message is displayed in all the levels as a hint.

Movement of camera:

When the player one shoots the ball onto the opponent's castle, then the ball may or may not hit the castle and vice versa. When a particular player has shoot the ball, then the camera will move in such a direction of the ball and then it will move to the other player's castle automatically. This repeats until a player has won the game.

Shooting the castle:

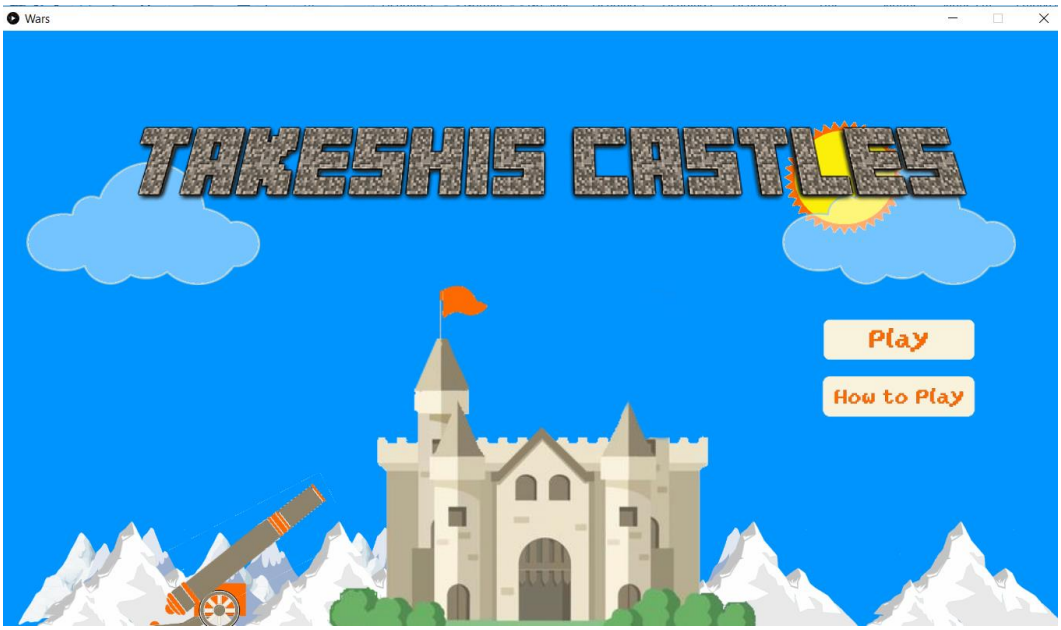
When the person targets the castle and shoots at it, the position of the ball is identified using the mouseX and mouseY. It is then checked if the x,y point collides with any part of the castle. This is found if the x, y point lies within the castle's space. Now, a straight line is generated from the middle of the canon to the castle to show the path of the ball. Upon hitting, an explosion is shown with a different music and an image. Once the castle is hit, the corresponding player will loose the health. If the canon shot is a miss, it moves beyond the size of the screen and it disappears.

Loosing Health:

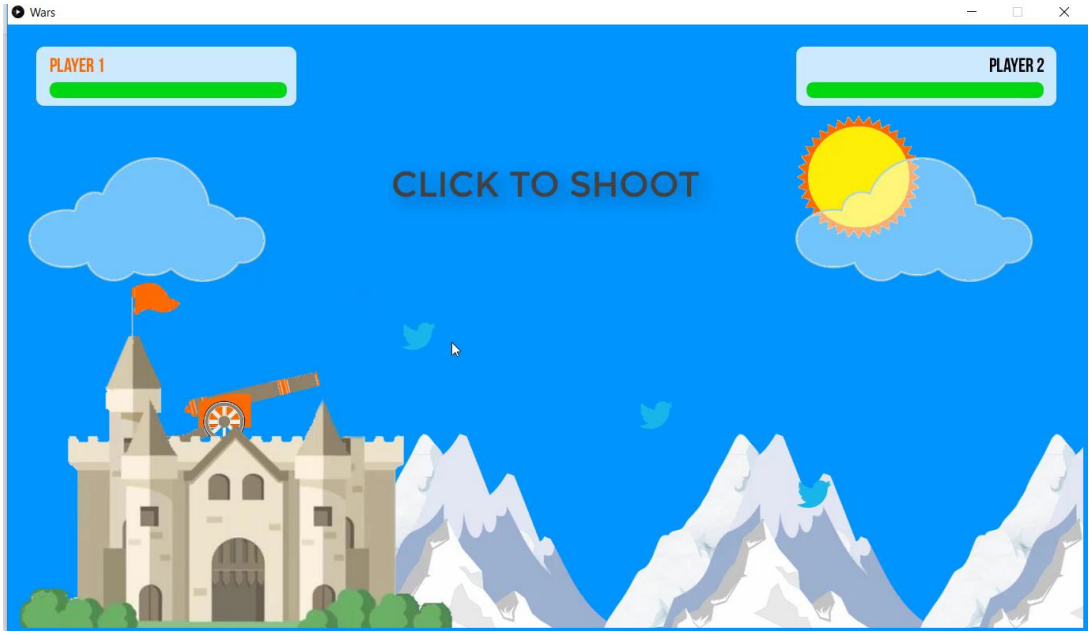
When a castle is hit with a ball, then the corresponding player will lose some part of the health and if a castle is hit for six times, then the player will lose all the health and will also lose the game.

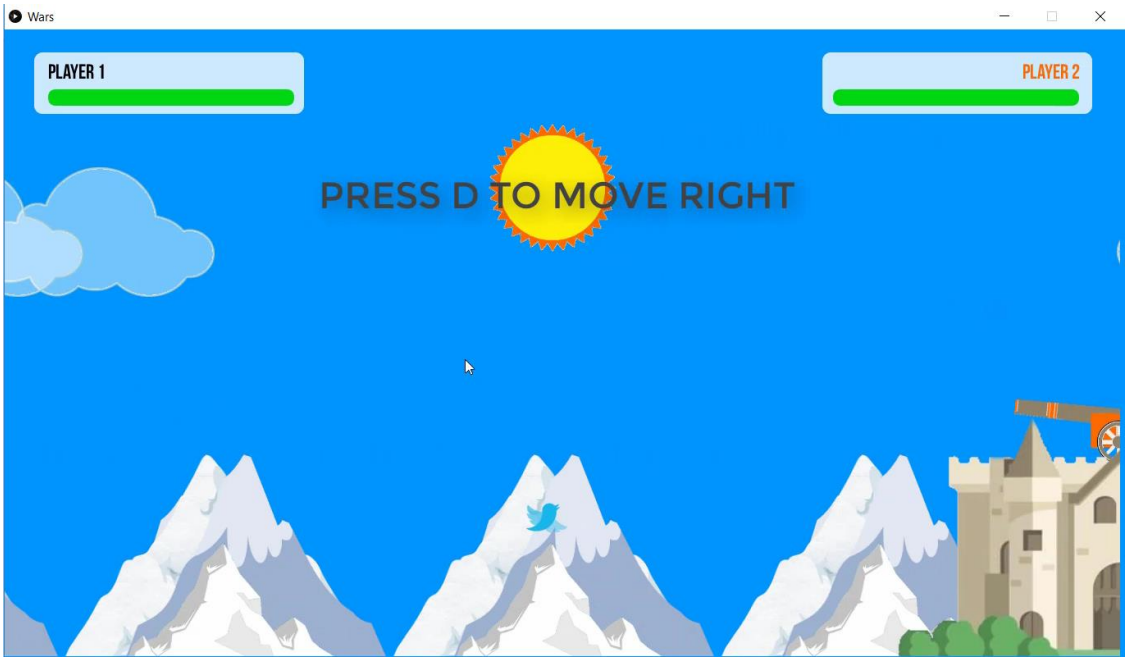
2.2.2 SCREENSHOTS:

START SCREEN

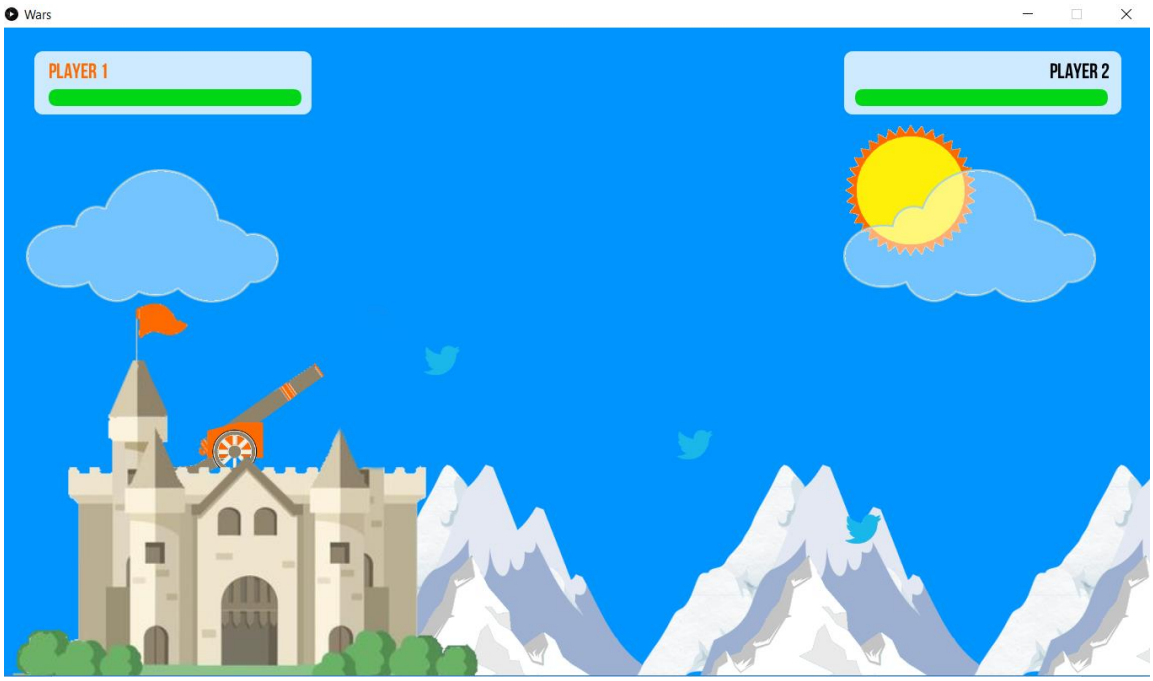


TUTORIAL INTERMEDIATE SCREENS

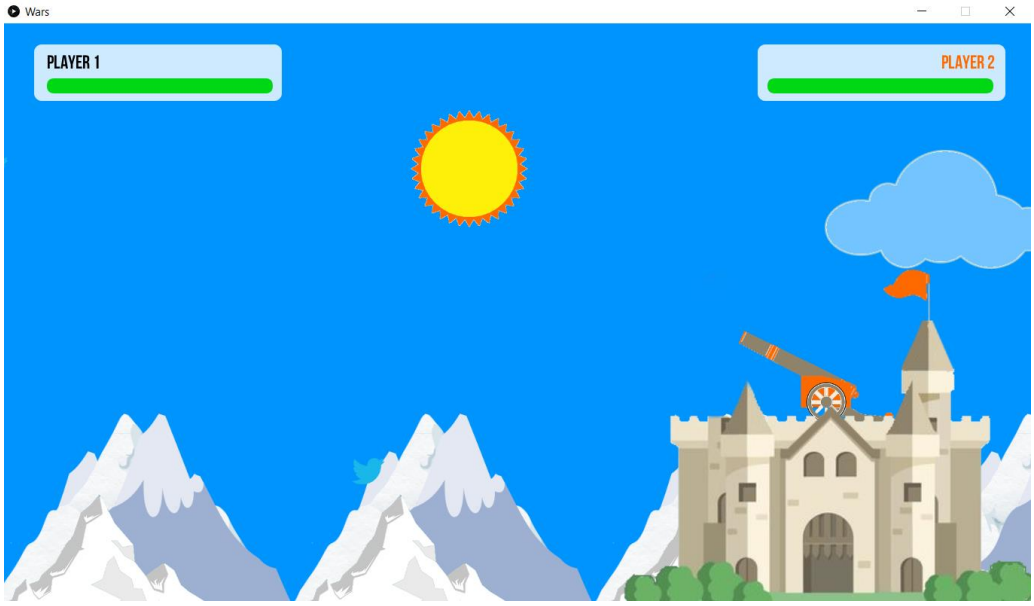




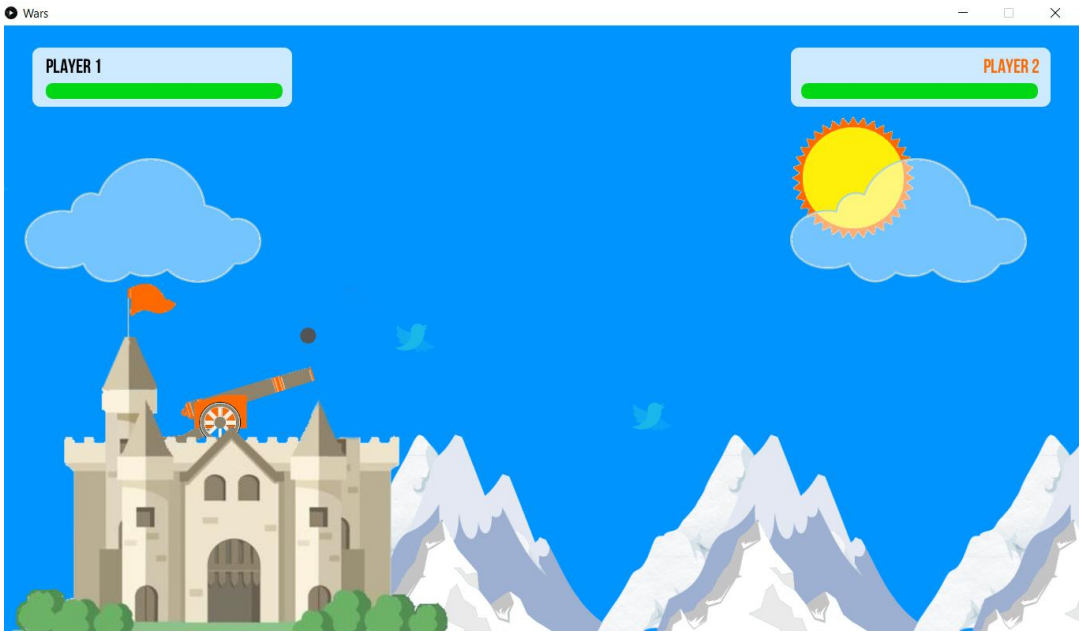
PLAYER-1 SCREEN



PLAYER-2 SCREEN

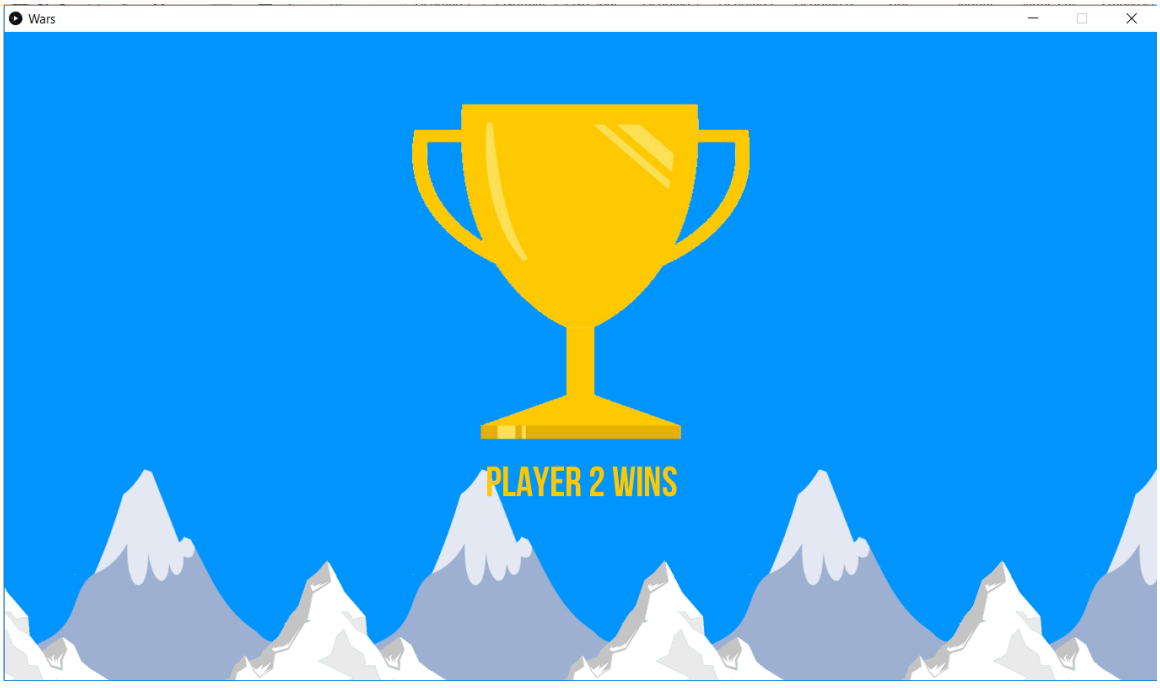


PLAYER-2 HITTING WITH A BALL ON PLAYER-1



PLAYER-1 WINING

PLAYER-2 WINING



2.3 TEAM MEMBERS' CONTRIBUTION:

The development of this game was an entire team effort. Two to three hours of a day for a week and additional effort by everyone in certain parts of the code resulted in this game.

1) Arjun. A - Introscreen,video(tutorial)
2) Balaji Muthazhagan T - Game Mechanics, Projectile motion
3) Sanka Gopi Sumanth - Sound Engineering,Document

CHAPTER – 3

REFERENCES

1) Learning Processing: A Beginner's Guide – The Coding Train, a YouTube channel.
2) Introduction to Game Development using Processing by J.R.Parker, University of Calgary.
3) http://www.processing.org