Pirate's Secret

Design Document

Project Team
João Lucas Rulff da Costa
Matheus de Sousa Bernardo
Ricardo Xavier Sena
Vinicius Carvalho Machado

I. Game Title

Pirate's Secret

II. Overview

A. General Idea

The game is an arcade/adventure game with a ninja as the main character trying to escape from pirates. It's a side-scrolling game that the main character can basically jump and fire throwing stars. The main objective of the game it's to survive as long as possible. The engineering behind the game will try to make the run and the jumps more close to reality. The levels are about the distance that the ninja travels and the score of the game, the longer he goes, more score he gets and more difficult the game becomes. The score is about the distance plus some extra points by killing enemies. For now, the game will be only single player. The music and the background will be ninja themed. The game can be challenging as the time goes on, and players gets used to the controls pretty fast because it will be easy to play.

B. Gameplay mode

Background Story

14th century. A ninja is captured by pirates. After some days been abducted, the ninja got his way out the pirate ship, and now is running for his life with a secret from the pirates. The pirates are running behind the ninja to capture or kill him so the secret dies with him.

Perspective

Side-scrolling view.

Interaction

The game is basically played with two controls. The arrow UP, the letters A and P.Up will make the ninja jump, the letter A will make the ninja throw a star and P is for pausing the game.

Challenges

The ninja need to survive from the many pirates that can appear on the screen, also have to escape from traps and holes and jump obstacles.

Actions

To pass through the level, the character has to pass through enemies and obstacles. The enemies can be killed with a star and the obstacles can be jumped by the ninja.

III. Core Features

Physics Engine

The game will have some variables and functions to allow the horizontal and vertical (jump) motion, and the combination of them. Besides that there is the physic and collisions behind the throwing star.

Graphics Elements

There will be a background based on context, the ninja character itself, the enemies/obstacles, the stage and the throwing star. Besides that, the lives, the health bar and the score will be shown on the screen.

Collision Detection

The game will check collision in three situations. One is when the ninja throws a star to kill an enemy. Another one is when the ninja collide with the enemy without killing it before. The last collision to check is when the ninja walks on a pirate trap.

Scoring System

The scoring system will be based on the distance that the ninja walks and how many enemies the ninja kills.

Level Incrementing

The game level will automatically increases when the ninja hits a check point based on the distance. For example: when the ninja hits the distance 1000 he will go to the next level.

Opening Screen

The opening screen will open a screen showing the title of the game and a menu with the options: Start Game, Control Menu, Options and Quit Game. The player can select any of these options from the menu.

Sound Effects and Music

A background music will be playing the entire time. The act of killing an enemy, dying or losing some portion of health will have a sound effect as a feedback for the player.

IV. Secondary Features

V. State Diagram of the play

A. High Level State Diagram of Pirate's Secret

Attachment 1

B. Low Level State Diagram of Pirate's Secret

Attachment 2

VI. Internal Economy

Lives:

The ninja starts with three lives and a full health bar, if the ninja collides with a pirate it will drain 50% of the health bar. If the ninja falls on a hole, he loses one life. If the ninja gets trapped he loses 25% of the health bar.

Ammo:

The throwing stars will be infinite with a small delay between shots.

VII. Game Balance

Positive feedback:

There will be two ways of positive feedback; when the player kills an enemy it will be rewarded with score points and visual/sound effects. Also when the player level up it will be rewarded with visual/sound effects.

Negative feedback:

One way of negative feedback is when the ninja collides with an pirate and the ninja loses health, if the ninja reach 0% of health it will lose 1 life. If the ninja falls on a hole he loses one life, if the ninja loses all his lives is game over.

Adjusting game difficulty:

The game difficulty will not be controlled by the player, it will increases accordingly with the distance that the ninja has completed.

VIII. Victory Conditions

Winning the game

The game will continue as long as the players has remaining lives or he/she decides to quit. The goal itself is just to "survive" as far as possible collecting the higher score you can get.

Losing the game

The game is based in lives, the player loses one whether he touches the enemy or falls into holes. If the player loses all lives, it is game over.

IX. Development Environment

Programming language and additional features

Java 8 will be the main programming language. LibGDX framework will be used to support the development of the game.

IDE

The IDE that will be used is Eclipse Mars version 4.4.

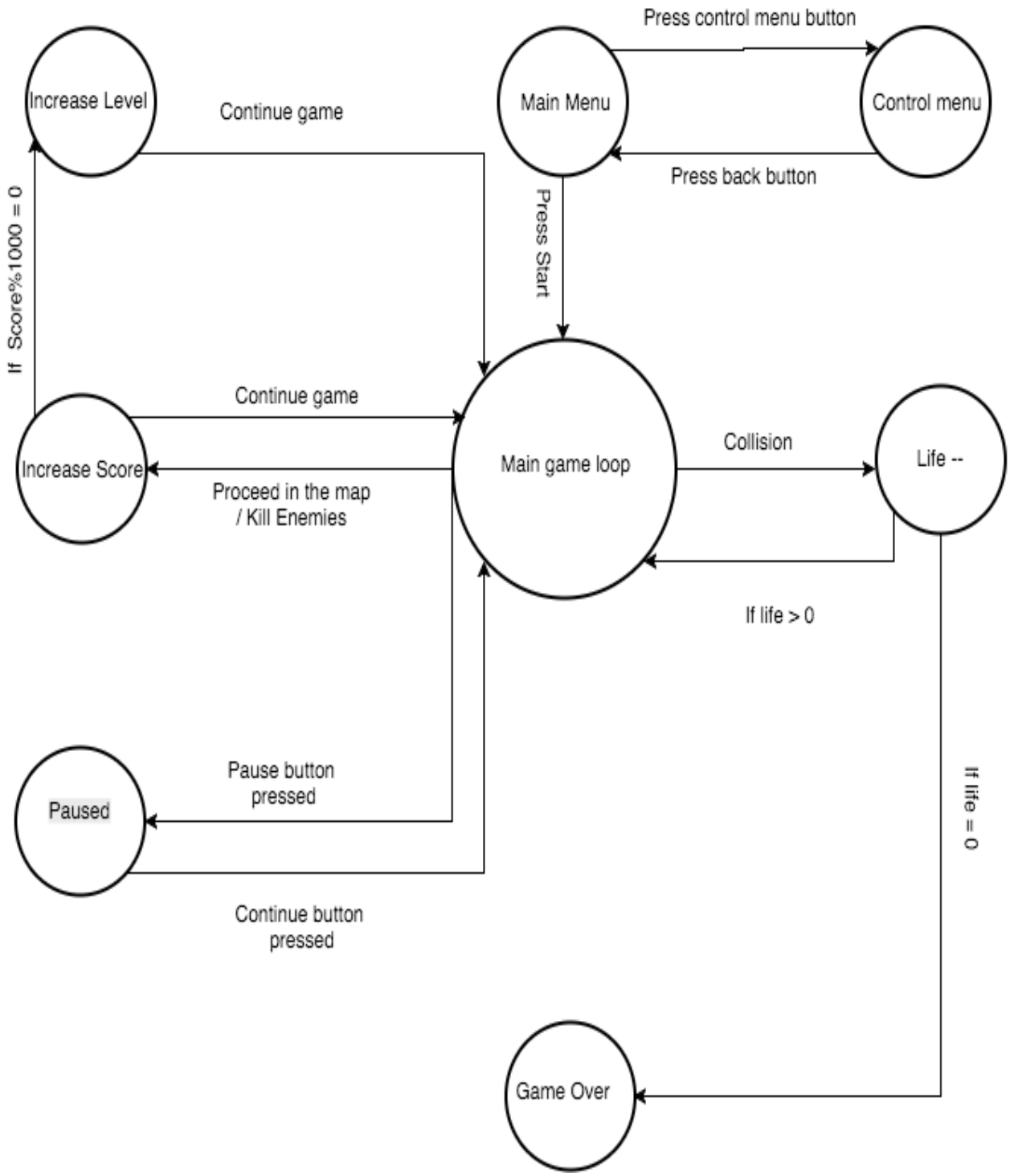
Additional Tools

For help creating the stage we will use Tiled map editor. For image editing we will use Adobe Photoshop.

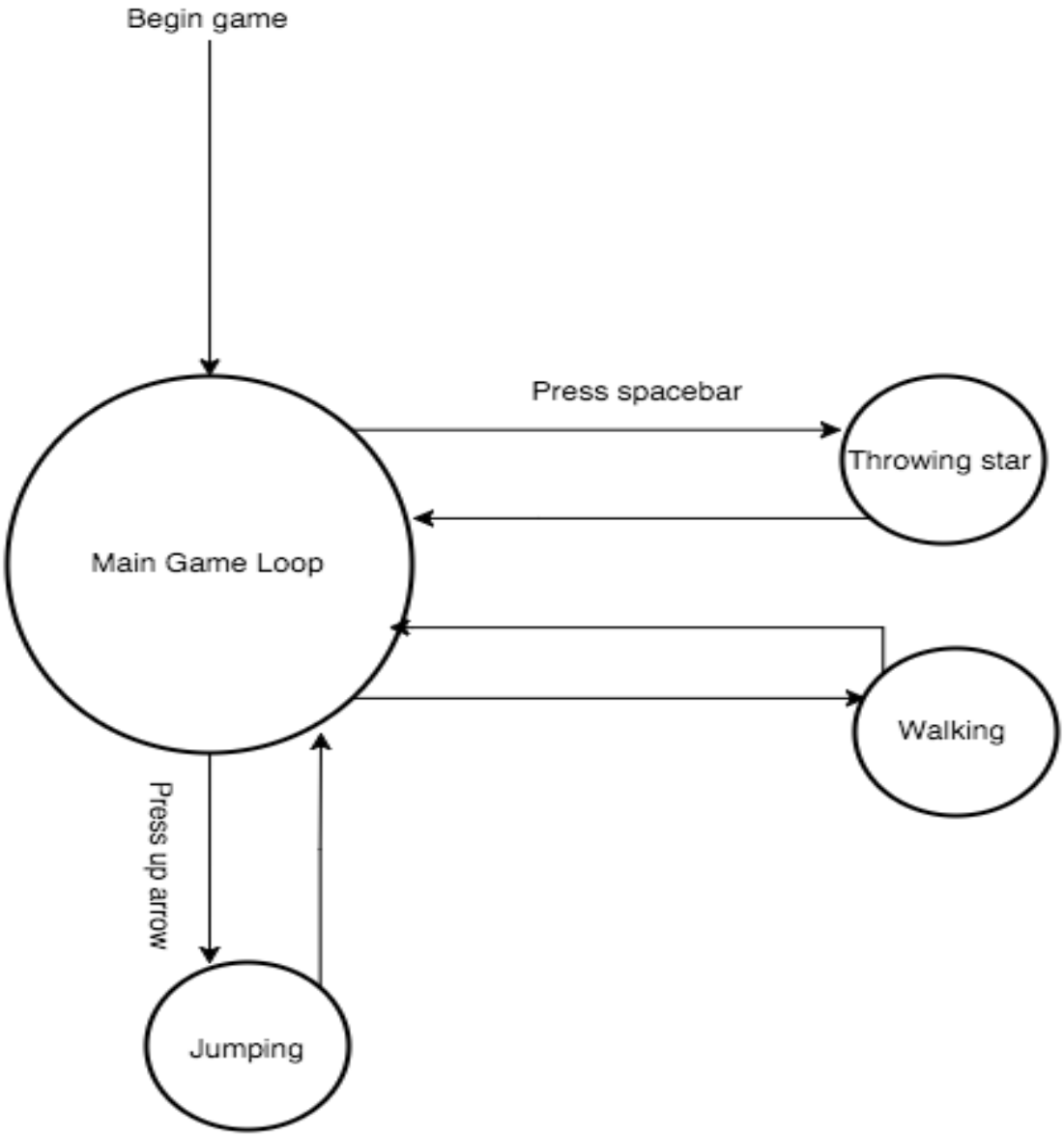
Version Control System

The control system that will be used is Git with a remote repository on Github. We will create a repository for the code and any document or assets.

Attachment 1:



Attachment 2:


Begin game
Press spacebar
Throwing star
Main Game Loop
Walking
Press up arrow
Jumping