Alomakoth

Final Package

Nathan & Sarah

20 June, 2018

ICS3UI

Table of Contents

**Program Specifications Page 1-8**

Initial Specifications…………………………………..1-2

Work in Progress Reports…………………………………..2-6

Release Schedule…………………………………..6-8

**Main Components Page 9-15**

Program Overview…………………………………..9

Disclaimer/Known Bugs // LIST ALL SOURCES IN CODE…………………………………..9

Journal…….………………..10-14

List of Sources……………………..14-15

User Manual……………………………………….…………Within Documents labelled “ USER MANUAL”

Game……………………….Within Release 2.4, within core, within scr, within gdx.game

**Others Page 16-19**

Scratches…………….………………..16-17

Notes for Future Programmers…………………..17-18

Favourite Lesson………………………………………..18-19

Program Specifications

**INITIAL PROJECT SPECS:**

[**https://github.com/NathanRJohnson/Fighting-Game.git**](https://github.com/NathanRJohnson/Fighting-Game.git)

**PROGRAM OVERVIEW:**

Alomakoth is a two player fighting game similar to Street Fighter. Players will choose avatars from a selection and battle against the other user until one person wins. There will be health bars, different background to choose from, basic and special attacks,

**DESCRIPTION OF USER INPUT:**

General Options:

* Use the mouse to select buttons to start the game, select background, pause the game.

Player 1:

* A and D (player 1) to move the character left and right respectively.
* S to crouch.
* C to block.
* W to jump.
* WASD to select characters in character menu.
* To use basic and special attacks is space bar

Player 2:

* LEFT and RIGHT(player 2) to move the character left, and right.
* DOWN key to crouch
* P key to block.
* UP to jump.
* Directional keys to select characters in character menu.
* To use basic and special attacks is the enter key.

**DESCRIPTION OF PROGRAM OUTPUT:**

* Buttons: to start game setup of characters and select background, pause game and play.
* Start screen, character selection screen, background selection screen and the actual fight screen.
* Health bar for each character.
* Special attacks will have effects (of somesort, to the best of our abilities).
* Characters on either side of the screen.
* Display winning character.
* Character names already given displayed on the top of health bars

-------------------------------------------------------------------------------------------

**Work in Progress**

**Major Developments/breakthroughs:**

*Nathan:* From the beginning of the project, I’ve pretty much only had success. I’ve managed to get gravity to work, taking inspiration from Daniel Shiffmans “*The Nature of Code ”* <http://natureofcode.com/book/>. I created two scratches, one that implements a basic attack for each player, an knockback system, and life points, and a second that is in the middle of testing the blocking feature.

Sarah: A major development I experience would be learning how the ability to switch screens between the main game and implemented screens using a button to go back and forth between 2 screens. This was only possible with the help of “Extending the simple game” article from <https://github.com/libgdx/libgdx/wiki/Extending-the-simple-game>. With the help of Kieran’s project and this website I gained the knowledge on how to the codes work and how to set up the basis of the game based on implementing screens. I have a scratch for the screen switch method as well.

**Major Challenges/Setbacks:**

*Nathan:* The only thing that I would want to change is to move the “basicAttack” and the “block” method into the fighter class. Also, I would like that the Fighters jump a little higher.

Sarah: A major challenge and/or setback I’m experiencing at the moment is having more than a one button on the game’s main screen. When I click the start button it will go to the ScrBckgd screen, which is wrong because no code is present for the screen to switch when the start button is clicked. As well as if you click near the bottom of the window the screen will switch as well even with no button image present there. Presently, this has turned into a scratch to try and fix the code.

**Any Modifications to your specifications/release schedule:**

* The “S” key and “DOWN” key will be used for blocking, rather than crouch.
* Player Selection and Background selection is now switched to release 3.4 and 3.5.
* Buttons for background, instructional screen, and player selection will be present on the mainscreen and clickable; but nothing will be loaded within the screens until later.
* Background music is now release 3.35
* Tournament mode is now release 3.9

**Description of Scratch/test program:**

**Describe the generic concept you needed to test out:**

Sarah; I tested out the concept if you click the button from the game menu then the screen should switch to the screen set. For instance, if the background button is clicked, the screen will switch the screen of background selection.

**Source any website/book that helped you with that concept:**

Sarah: To get an idea of what to do I took inspiration from Daphne’s code of her former game POLYGONE: <https://github.com/DaphneLai/POLYGONE-Final/tree/master/core/src/gdx>

Like her, I made a boolean in the button class to check if the click of the mouse is within proximity of the button image. If that is true, the screen will switch to the screen loaded.

**Describe the code and the lesson that you learned from it:**

Sarah: From her code I learned that you can create public booleans to test and integrate in your main code. By creating a boolean in the button class, I can clean up more code and make it easier to use “isClicked” boolean in any other screens without needing the load the same code over and over again in different screens. In addition, I learned how to set the boundaries that the mouse clicks that allows the boolean to be true.

public boolean isClicked(float fX, float fY, float fW, float fH) { // inspired by daph's code https://github.com/DaphneLai/POLYGONE-Final/tree/master/core/src/gdx

if (Gdx.*input*.getX() > fX && Gdx.*input*.getX() < fX + fW) {

if (Gdx.*graphics*.getHeight() - Gdx.*input*.getY() > fY && Gdx.*graphics*.getHeight() - Gdx.*input*.getY() < fY + fH) {

return true;

}

}

return false;

}

}

I learned how to code so that the problem of UI coordinates of the mouse and the coordinates of the image that start in the bottom left of the window to work together. As you must subtract the mouse coordinates from the window size to get a common area for the button and mouse vertically. Horizontally, the coordinates shouldn’t be must of a problem.

**Describe any challenges that you enjoyed in integrating this scratch code into your major project:**

Sarah: The code works for a single button which is great; but unfortunately, when integrating more than one button, the two buttons clash and screen switch isn’t successful. As mentioned before, if the start button is clicked, the screen will switch to the background screen. Which is wrong since only the background button should load the background screen. If you click the bottom of the window the screen will switch as well. This problem is still present now.

**Peer Assessment:**

Nathan: Although Sarah did not start on the project immediately, she used her time productively to watch video tutorials. She has been working diligently on creating a button system for screen switching, and even worked on it it at home during the weekend to get it to work..

Sarah: I find that Nathan has done a lot of major programming for the game and has worked very hard in making the game a success working on player attacks, blocking and more. Though, I’m trying very hard, I’m not as successful in problems and feel I’m holding the game back the team from advancing in release as quickly as it can go if I can get the screens and buttons to work.

-------------------------------------------------------------------------------------------

Release Schedule: Alomakoth

Team: Space Monkey Mafia

|  |  |
| --- | --- |
| **Release Name** | **New increment features of this release ( or scratch specifications)** |
| Alomakoth 1.0 | Player 1 creation and load |
| Alomakoth 1.1 | Player 1 movement with WASD keys |
| Alomakoth 1.2 | Player 1 Jumping & Gravity capabilities |
| Alomakoth 1.3 | Player 2 creation and load |
| Alomakoth 1.4 | Player 2 movement with directional keys |
| Alomakoth 1.5 | Player 2 Jumping & Gravity capabilities |
| Alomakoth 1.6 | Creation of start screen |
| Alomakoth 1.65 | Creation of background selection & button  (You can switch screens, but nothing will be loaded within) |
| Alomakoth 1.7 | Creation of instructional screen & button (You can switch screens, but nothing will be loaded within) |
| Alomakoth 1.8 | Creation of player selection screen  (You can switch screens, but nothing will be loaded within) |
| Alomakoth 1.9 | Load back and menu key |
| Alomakoth 2.0 | Create player 1 & player 2 basic attack |
| Alomakoth 2.1 | Create block for player 1 & 2 |
| Alomakoth 2.2 | Create health bar & detect lost health |
| Alomakoth 2.3 | Character Animation |
| Alomakoth 2.4 | Load finished game & Music & Button sounds |
| Alomakoth 2.5 | Load pause key |
| Alomakoth 2.6 | Controller Compatibility |
| Alomakoth 2.7 | Create background selection |
| Alomakoth 2.8 | Create player selection |
| Alomakoth 2.9 | Special attack: Super Attack (all players) |
| Alomakoth 3.0 | Combo attacks |
| Alomakoth 3.1 | Load Super Attack Orb |
| Alomakoth 3.2 | Tournament Mode |
| Alomakoth 3.3 | 4 player mode |

Changes:

* Creation of background, instructional and player screen switched to release 1.65, 1.7, 1.8 respectively
* Controller compatibility changed to release 2.8
* Added the capability of going back a screen by clicking “b” and to start menu when you click “m”. This is release 1.9
* Background music and button effects are changed to release 2.5
* Player selection and background selection moved to release 2.85 and 2.8 respectively
* 4 player mode changed to release 3.3
* Tournament mode changed to release 3.2
* Basic attack of both players changed to release 2.0
* Block of both players changed to release 2.1
* Health bar of both players created is now release 2.2
* Animation Release 2.3

-------------------------------------------------------------------------------------------

Main Components

**Program Overview**

Welcome to Alomakoth! Alomakoth is a two player that with automatically given characters. A high quality background is automatically chosen as well. Alomakoth is a fighter game similar to Street Fighter. Two users keep fighting until one of their characters loses all of their health. Then, one of them is declared a winner. There’s health bars, music, very little sound effects, and basic attacks

NOTE: Player and Background selection is not going to be present until later. Buttons for background “bckgnd”, and player selection “p” will simply be screens for users to look at characters and background more in depth.

-------------------------------------------------------------------------------------------

**Disclaimer/Bugs**

1. The first known bug is with the buttons “Start” and “Backgnd”. These two buttons are png’s, but the image is bigger than it seems to the visual eye. We tried our best to crop the excess transparent background, but it isn’t sufficient. Thus, if you click above the start button, the screen will switch to the play screen, ScrPlay. If you click in between the start and backgnd button, the screen will switch to the background screen, ScrBckgd. If you click below the backgnd button, the screen will switch to the background screen.
2. The second known bug is if you fall of the edge of the given platform with the character in the game, you can get back on the platform and keep fighting. Essentially, we wanted to make it not possible, then the player loses; but, due to time crunch and priorities we couldn’t fix the bug in time.
3. Third known bug is if you finished the game and one player is declared the winner. If you return to the start menu and try to attempt the game again, you will see the winner of the previous game displayed. There’s no restart.

-------------------------------------------------------------------------------------------

**Journal**

**May 28 2018**

**Nathan:**

* Created Animations for Black Belt character on Piskel
* Created PVector Class
* Created hit detection between the fighters and the platform

**Sarah:**

* Started learning code for keyPress and Mouse Clicking
* Confused on how to work intellij :(

**May 29 2018**

**Nathan:**

* Recreated the project so that it is properly formatted
* Updated the PVector class to work with floats rather than doubles
* Remade the Fighter class to work with PVectors
* Created applyForce() method
* Created the force of gravity

**Sarah:**

* Still confused and practicing screen menu changes

**May 30 2018**

**Nathan:**

* Added Gravity, now in version 1.5
* Added the ability to fall of the platform

**Sarah:**

* Got more than one picture to load and controls to work. Going to work 0n multiple screens next time.

**May 31 2018**

**Nathan:**

* Created a new fighter sprite
* Started Basic attack scratch

**Sarah:**

* Working on multiple screens and I have no idea what I’m doing, but I’ll keep going :0
* Looking at websites to try to have a deeper understanding

**June 01 2018**

**Nathan:**

* Got Fighter hit detection to work :)
* Unfortunately it registers multiple times

**Sarah:**

* Worked on screens and got it working
* A single button is working as well
* Goal: working multiple buttons

**June 04 2018**

**Nathan:**

* Created a delay timer between attacks so you spam the attack button and hit 6 times a frame
* Worked on creating a knockback effect on someone who is attacked to reduce the effectiveness of spamming attacks

**Sarah:**

* Got switching screens concept working
* the only problem is getting the if the mouse presses on the button in scrplay it will go back to the main start menu

**June 05 2018**

**Nathan:**

* Fighters are now sent backwards when hit
* Started Working on blocking
* TODO: set the input as damage rather than fighter

**Sarah:**

* Screen switching was a success and the scratch is finished
* This includes using buttons to switch between the main start menu and play menu

**June 06 2018**

**Nathan:**

* Made the knock back direction based on which direction the attack is coming from.
* Continued to work on blocking

**Sarah:**

* Started release 1.6.
* Got the start button and title button loaded

**June 07 2018**

**Nathan:**

* We lost a slow-pitch tournament

**Sarah:**

* I started release 1.65
* Made a second release schedule with modifications
* Ran into problems with more than 1 button because of proximity

**June 08 2018**

**Nathan:**

* Finished the basic blocking scratch
* I think a reversal move would be really cool, if you attack someone who is blocking
* You can now not move when attacking

**Sarah:**

* Finished the WIP
* Discovered bugs with the buttons
* The two buttons are hit when you click a single button.

**June 11 2018**

**Nathan:**

* Add another platform and BG to assets
* Added screens, just how Grondin likes it
* Finished Release 1.6, yay!

**Sarah:**

* Frustrated :( Couldn’t solve overlapping button problem.
* Tried fixing the hit detection

**June 12 2018**

**Nathan:**

* Created and finished a health bar scratch that displays Player 1 and Player 2’s health bars, updating them as they take damage.
* Created a sprite sheet for the black belt fighter, that includes punching and blocking.

**Sarah:**

* Finally got buttons working. Reduced image size to have a smaller image
* Added vector 2d for the mouse coordinates.

**June 13 2018**

**Nathan:**

* Created an animation scratch
* The walking animations work, however I want the punching and blocking animation the play when I hit the button and stop once it’s finished, regardless of whether I am holding the button down or not.

**Sarah:**

* I’m putting every piece of code accumulated together
* Formating to Mr Grondin’s liking of screens
* Hopefully finish soon

**June 14 2018:**

**Nathan:**

* Totally Finished the Animation Scratch, blocking is now a toggle, and punching plays once when pressed.
* Tournament tomorrow!

**Sarah:**

* Finished release 1.65
* Ghost buttons are gone. I had to put the inputprocessor in each screen.
* Scratches are integrated
* Screens implemented
* That took a long time

**June 15 2018:**

**Sarah:**

* I got release 1.7 done with the instructional screen and button loaded
* I got partially 1.8 done with the player selection screen and button loaded.

**June 16 2018:**

**Sarah:**

* I changed the release schedule for convenience sake.
* Thus, I got release 1.9 done with the back and menu keys being “b” and “m” respectively
* I got release 1.95 done with the background music and button clicking sound effects
* Game looks cleaner :D

**June 18 2018:**

**Nathan:**

* Compiled all the code into a complete game
* Health Bars need to rescalled, need to find a new image for the VS logo
* Want to add a win condition if the enemy gets knocked off the edge

**Sarah:**

* I will start the final few documents and manuals of the game
* There’s a looooot to do

**June 19 2018:**

**Nathan:**

* Wrote about my favourite scratch program
* Added an overlay that pops up when one of the players has won
* If you fall of the platform, you now lose
* If you block, the distanced you are knocked back when being hit is halved

**Sarah:**

* I finished a majority of the user manual// final packages
  + Scratch Descriptions
  + Known Bugs
  + Notes for Future programmers
* Nathan filled out the favourite lesson
* Major progress in the user manual
* Unfortunately, not all packages are officially named yet until we finish the game

-------------------------------------------------------------------------------------------

**List of Sources**

Key Input:

<https://stackoverflow.com/questions/23693063/libgdx-key-pressed-held>

<https://www.youtube.com/watch?v=Huifd-C2KrI&list=PLS9MbmO_ssyCZ9Tjfay2tOQoaOVoG59Iy>

Screen Switch:

<https://stackoverflow.com/questions/31222541/what-is-the-difference-between-game-screen-and-applicationadapter-in-libg/31222987?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>

<https://stackoverflow.com/questions/21382913/how-to-resize-a-sprite-in-libgdx?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>

<https://libgdx.info/basic_screen/>

<https://github.com/libgdx/libgdx/wiki/Extending-the-simple-game>

Gravity:

<http://natureofcode.com/book/>.

Button Concept:

<https://github.com/DaphneLai/POLYGONE-Final/tree/master/core/src/gdx>

<https://www.youtube.com/watch?v=Huifd-C2KrI&list=PLS9MbmO_ssyCZ9Tjfay2tOQoaOVoG59Iy>

-------------------------------------------------------------------------------------------

Others

**Scratch Descriptions:**

1. **Combat Scratch**

The combat scratch isolated the concept of basic attacking and blocking. Whenever one of the fighter sprites attacks the attacked sprite will lose 10 points in health and if they block in time the attacked sprite will only lose 5 points in health. This applies vice versa for both main sprites, spfBagLogic and spfBlackBelt. spfBadlogic is the sprite on the very left using the WASD keys and spacebar, AD is moving left to right, S to block and W to jump (this isn’t presently in this scratch) and space bar to attack. spfBlackBelt is the sprite on the very right using the directional keys and enter key, left and right arrows for moving left to right, down arrow for blocking, and up arrow for jumping (this isn’t presently in the scratch) and enter key to attack. Description of health damage and which player punching is described in the text box of intellij.

1. **Animation Scratch**

The animation scratch was to isolate the concept of switching sprite images whenever the character moved, punched or blocked. In this case, whenever the user moves sprDude, sprDude moves its limbs to portray walking. When the user presses the enter key for attacking, the character switches to a sprite of punching. The punching animation includes “effects” of a white arch coming off to emphasize strength. When the user presses the down directional key, the animation switches to blocking. This blocking has the animation of a “shield” like object.

1. **Health Scratch**

This health scratch isolated the concept of losing health whenever a character and/or sprite was hit. For simple reasons, this scratch was stripped down to whenever you clicked using the mouse on the left key the player health bar on the left will lose health. Using the mouse you clicked on the right key the player health bar on the right will lost health. This isolated concept was to test the visual aspect of the health bar with the decreasing red bar and getting the correct code to correspond the right animation.

-------------------------------------------------------------------------------------------

**Note for Future Programmers**

To future programmers, to make your life easier make sure you first off learn how to implement screens and extend a main game class. You must learn how to create a GamMenu that extends a game. Then, proceed to implement screens and learn how to switch through those screens. This is a very important first step so that life is easier later. So, that when you’re working on scratches and putting all your code into you’re shared game copying and pasting code will be way easier. As well as, doing this later will hold back your progress within the game. Thus it is **very** important to get this basis down first. This was what held us [*Alomakoth*] back in time and brought us to a sticky situation of copying and pasting code while trying to prevent the program from crashing.

To make a extended Game class, create a new class with the appropriate name. Then, within the line of code of “public class [insert your class name]” add in “extends Game” at the very end before the starting curley bracket and make sure to implement all methods:

public class GamMenu extends Game{

To make a screen, make a new class and within the line of code of “public class [inset your class name]” add in “implements Screen” before the starting curley bracket and make sure to implement all methods:

public class ScrBckgd implements Screen,InputProcessor {

Whenever you want the implemented screen to connect with the main GamMenu, make sure to call it in the GamMenu [red box example] and import it a the very top of the code [blue box example]. In this case a lot of screens are imported that’s why it doesn’t specify, but once you name the Screen you want to import, intellij should correct and import the class for you.

-------------------------------------------------------------------------------------------

**Favourite Lesson (Nathan):**

My favourite lesson this year was learning how to animate my fighter on the screen. It’s something that I’ve never attempted to do, and it produced a satisfying result. The entire process was a lot of fun, and provided some unique challenges. The first task was to create a sprite sheet. Using piskel, I had some fun drawing a wacky character and his walking, punching, and blocking frames. Using Kieran Halliday’s code, I incorporated my sprite sheet into my animation scratch. However, my sprite sheet came with some interesting problems to solve. Firstly, all of Keiran’s code was focused on making a character walk. Mine had to incorporate blocking as well as punching. I want my block to work as a toggle; when the key is pressed the fighter starts the block animation, and holds the last frame until the key is released. The punching I wanted to work only on the initial key press, and have the entire animation carry out only once, ignoring the length that the key was held. For these to problems, I had to learn about debouncing and frame manipulation. Another unique problem that I addressed in the main code, rather that in the scratch itself, was determining the direction that the fighter was facing. A true master never turns his back on his enemies, so I had to make sure that depending on where the fighter was in relation to his enemy, the directional keys would have to do different things. If the fighter is moving away from his enemy, he walks backwards, if he walks towards his enemy, he faces him. In conclusion, it was a problem that came with unseen challenges, that ended with a satisfying result.

-------------------------------------------------------------------------------------------

THE END