Morp Rune

Contents:

S.No	Topic	Page No
1.	Introduction	03
2.	Game Ideas and Critical Analysis	04
3.	Gameplay	05
4.	Asset Design	06
5.	Software Used	08
6.	UML Diagrams	10
7.	Code Snippets	13
8.	References	15

Game Design Document: Morph Rune

Introduction:

Everyone plays casual games which require little to no thinking, some games are just based on muscle memory too. My game is designed to be relaxing and easy to play, once they learn the rules. It follows the basic principle of Rock/paper /scissors with more variants.

The game is simple easy, endless. and requires the user to pay attention to the game.

Fig 1: Popular casual Games

NinJump a popular Casual game

Genre: Casual, endless

Targeted audience: 5+

Game Ideas:

Game 1: Morph Rune

Game is inspired by the classic game Rock/Paper Scissors. The lost theme is introduced in a story. Re'home a young child of a ritualistic tribe loses his way home and is forced by a trickster god to solve multiple puzzles to reach home. He does not realize that the problems never end.

The game will be easy to implement for the web browser.

Game 2: and there I tried to lose her

The game is inspired by stereotypical couples who hate each other.

The player controls a Husband/wife whose goal is to lose his/her partner.

The player must navigate through a pack man like map and place things all around the place to distract his/her for 5 seconds to win the game.

The game will have a time limit by which the player must lose the partner to win

Game 3: I need to lose

The game is inspired sports men who throw games for money.

The player controls an Olympian who must force his player to lose without getting caught.

The player must press the keys which appear on the screen in the correct order in the right time to make his character loose at the sport.

If the player gets caught by the referee trying to cheat he/she will be disqualified.

Game 4: Lost Human

This game is inspired by the movie John wick

In this game player will control a puppy who is searching for his lost master.

All the other pets in the street try to block his path and the puppy must follow a trail to his master.

Game 5: Don't lose the rhythm

This game is inspired by music.

The game will display music notes and the player must detect the pattern and insert a musical note every time a chord is missing.

The game will be easy to implement for the web browser.

Critical Analysis:

The game ideas were discussed with my peers and teacher. They suggested I go for interesting casual game concept. The other games such as I tried to loose her would be too hard to implement through HTML5 without an IDE.

Final Game Selected: *Morph Rune*

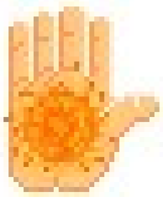
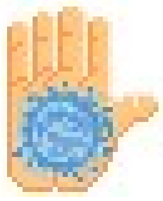
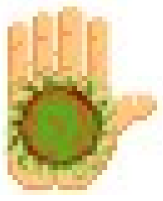
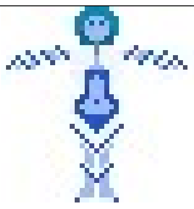
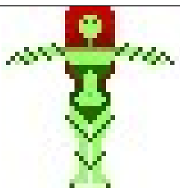
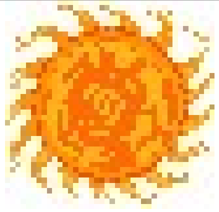
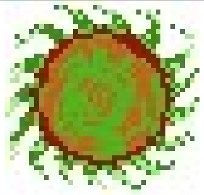
Gameplay:

The player essentially takes control of a rune which can change its shape and properties.

The player must control it and counter “enemy runes “ by taking its shape and changing into an element which beats the “Enemy rune”.

There are three shapes and three properties which the player can change into.

The table Below describes all the possible ways the player can change his rune.

	Fire **Q**	**Water** **W**	**Tree** **E**
Palm **J**			
Idol **K**			
Ball **L**			

Game Mechanics:

The game utilizes basic mathematics and computer concepts for all the functionalities.

Random function in JavaScript was utilized to provide a dynamic and non-repetitive appeal to the game. To detect the interaction between the Player’s rune and “enemy runes” a simple collision system was created.

Asset Design:

Concept Art (Place holders):
Initially Filrect function of the canvas element was used as place holders for the different runes. Different colors were used to represent the Rune types and shapes

After using these assets, for some time I realized that Neo can fly and does not probably require any specific animation frames for this action and hence jump animation was omitted in the final asset design.

Initially the assets were made in photoshop then exported to krita for compression.

Actual Assets:
The were designed in Adobe Photoshop CS6. Pixel Art was used to make the assets as it limits the file size, while still being pleasing or engaging to the player. All the Assets contained three frames of animation.

Fire Element:

Fig 2: Fireball

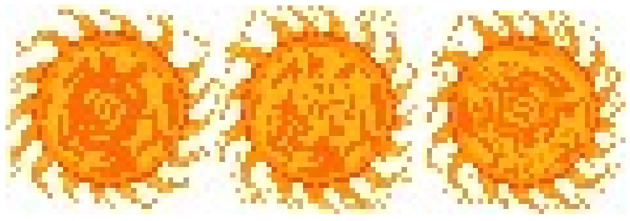
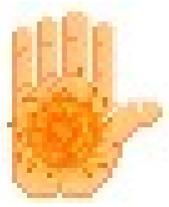


Fig 3: Palm Fireball



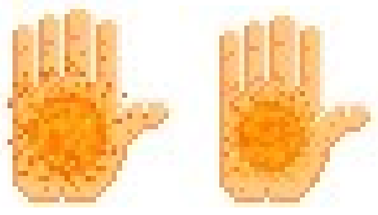
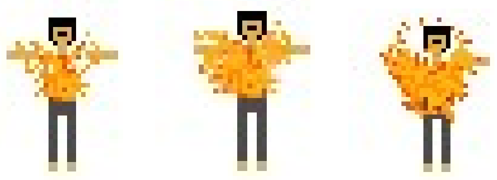


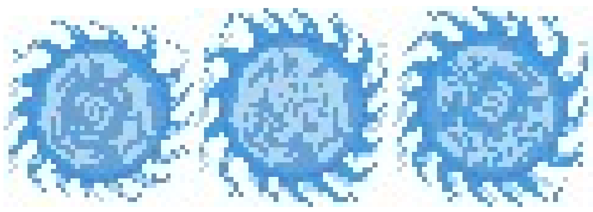
Fig 3: Man on Fire or Human Torch

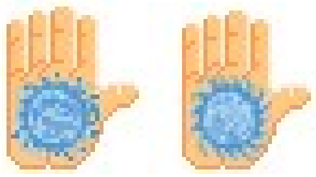


Water Element:

Fig 4: Waterball

Fig 5: Palm Waterball





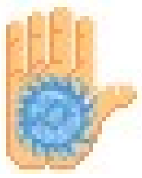
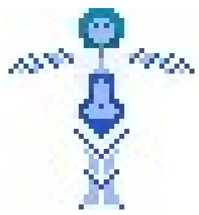


Fig 6: Hydra Maiden (does not have animation)



Leaf Element:

Fig 7: Leaf Ball

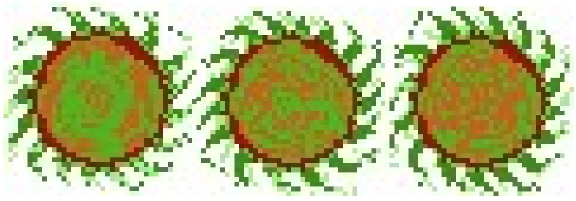
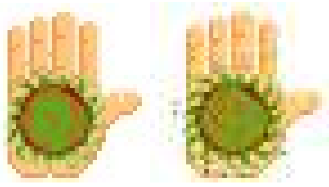


Fig 8: Palm Leaf Ball



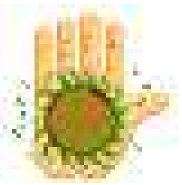
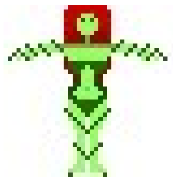


Fig 9: Ivy (no animation)



Software Used:

Adobe Photoshop

The assets were initially designed in photoshop. The output files were quite large and it was difficult to compress them. So the Designed image was imported into Krita which was utilized to republish at smaller size.

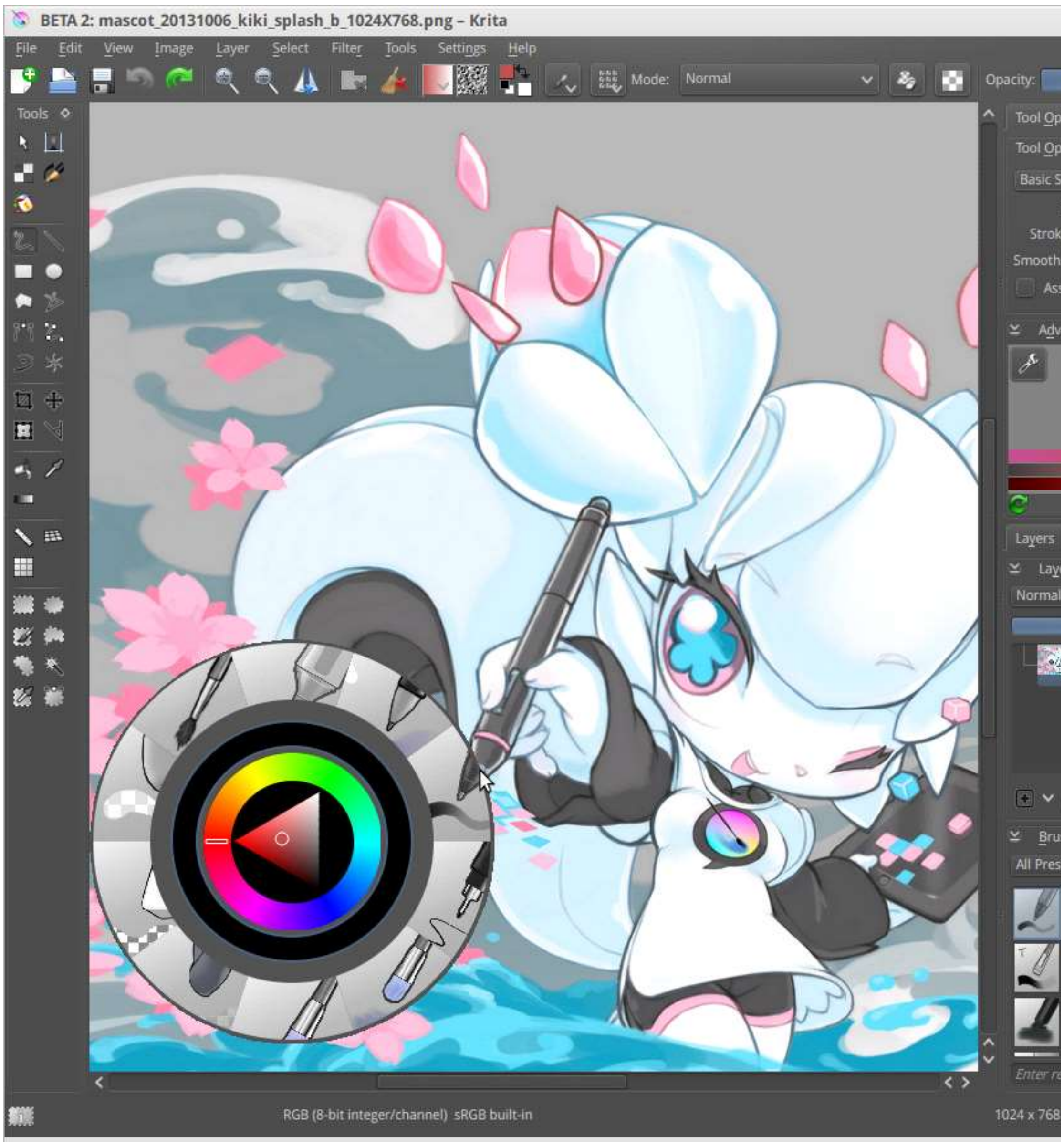
Fig 10: Photoshop CS6

Krita:

Krita is a digital painting application best utilized with a Pen tablet. It is an opensource software.

It mainly deals with Raster Graphics. It is mainly used for creating comics, illustration and concept art. Krita is a lot like GIMP which is also a opensource software.

Fig 11: Kiki - Krita's Mascot and the Interface


BETA 2: mascot_20131006_kiki_splash_b_1024X768.png – Krita
File Edit View Image Layer Select Filter Tools Settings Help
Mode: Normal
Opacity:
Tools
RGB (8-bit integer/channel) sRGB built-in
1024 x 768

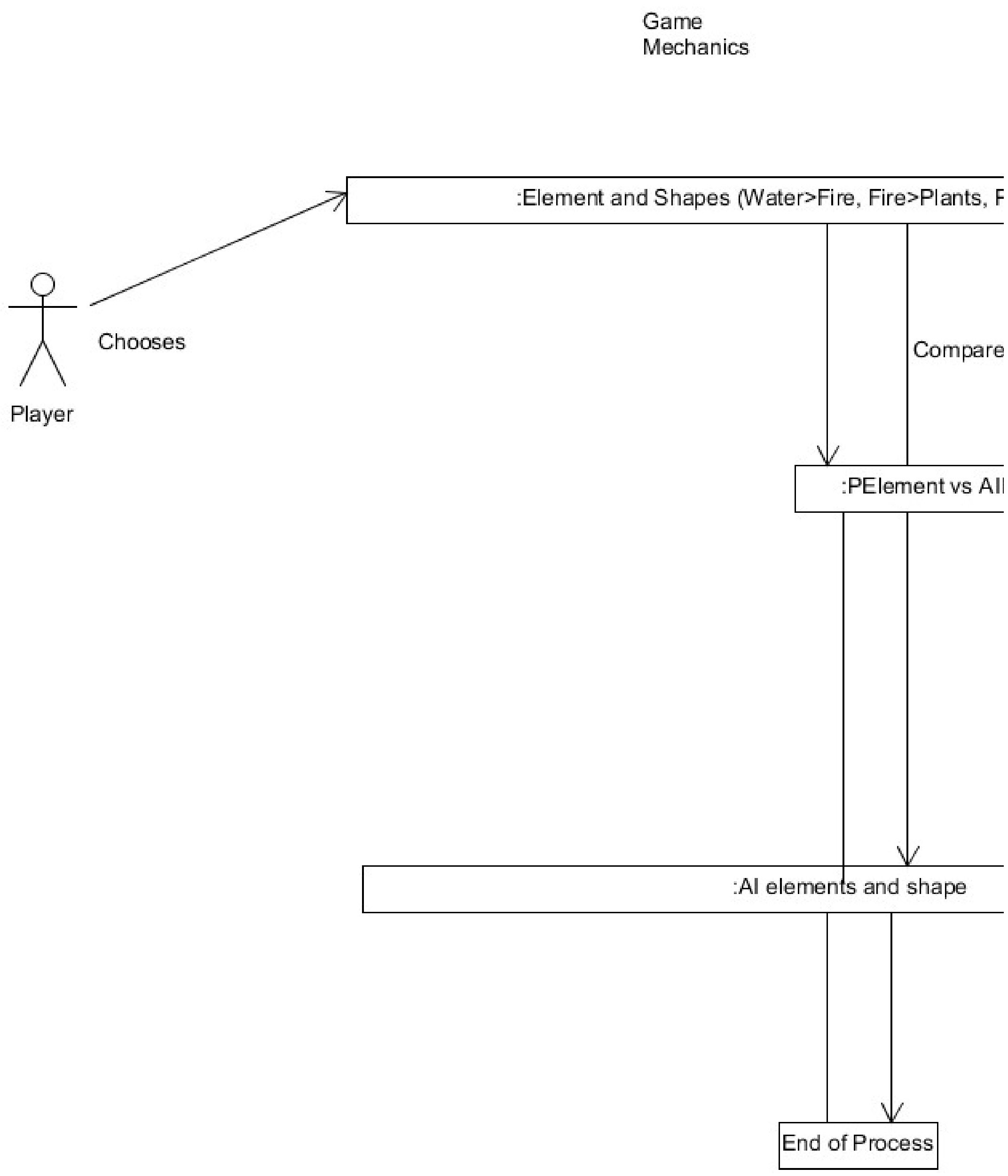
Editey:

It is a Google drive addon which allows creation and manipulation of javascript,Html, Css documents on the drive.

This software was used to enable easy debugging and preview of the code.

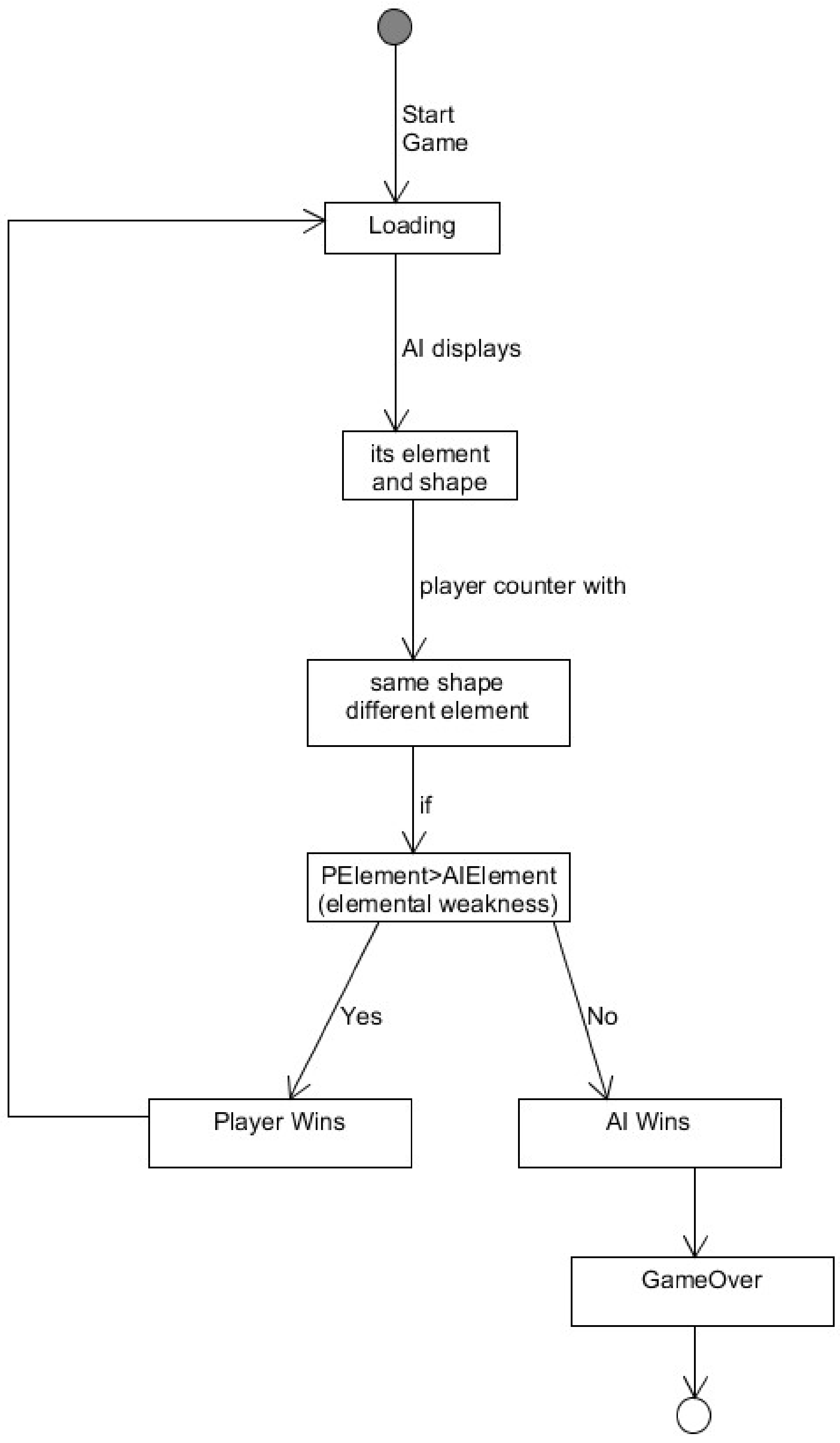
UML Diagrams:

Collaboration Diagram

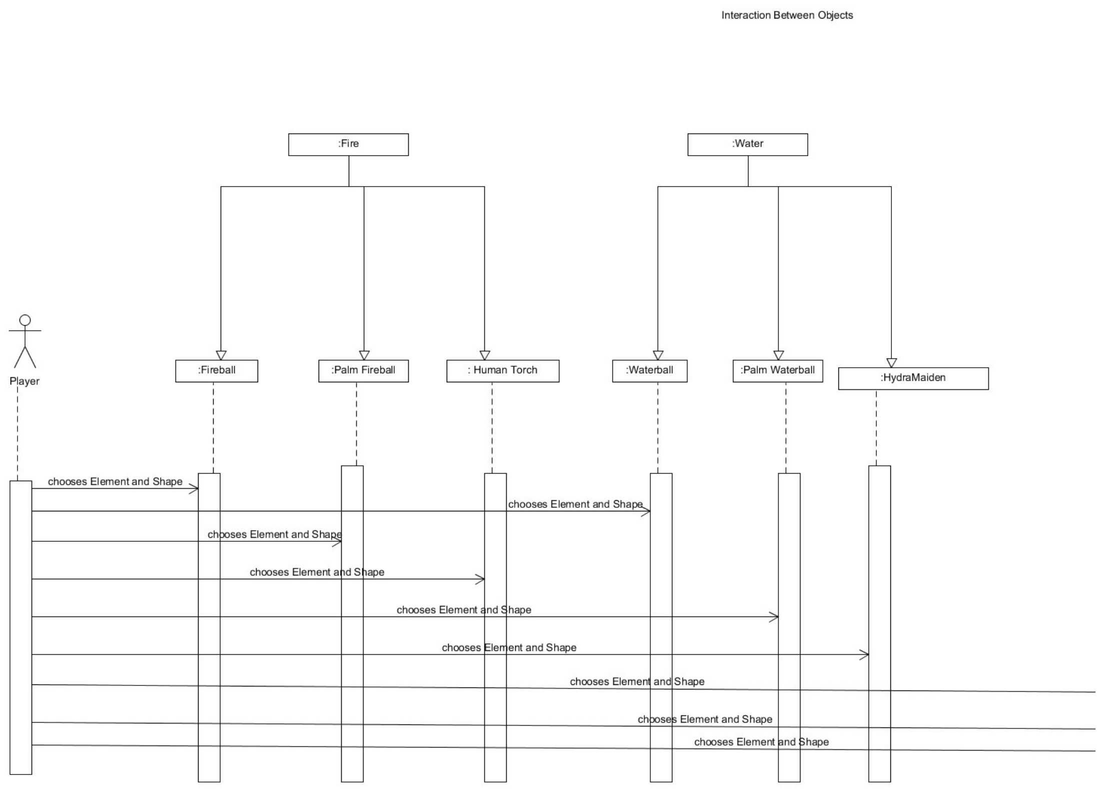


Activity Diagram

Activity Diagrams

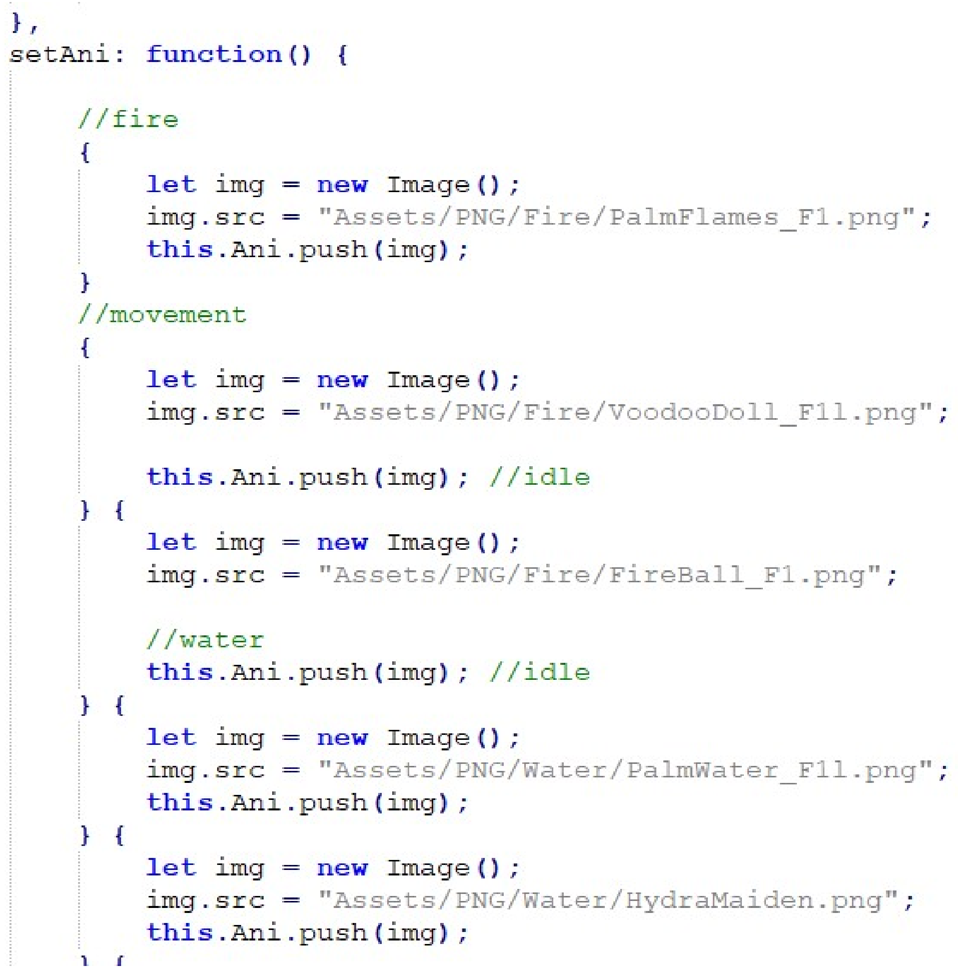


Interaction Diagram:


Interaction Between Objects
:Fire
:Water
Player
:Fireball
:Palm Fireball
: Human Torch
:Waterball
:Palm Waterball
:HydraMaiden
chooses Element and Shape
chooses Element and Shape
chooses Element and Shape
chooses Element and Shape
chooses Element and Shape
chooses Element and Shape
chooses Element and Shape
chooses Element and Shape
chooses Element and Shape

Code Snippets:

Fig 12: Importing Images to Animate



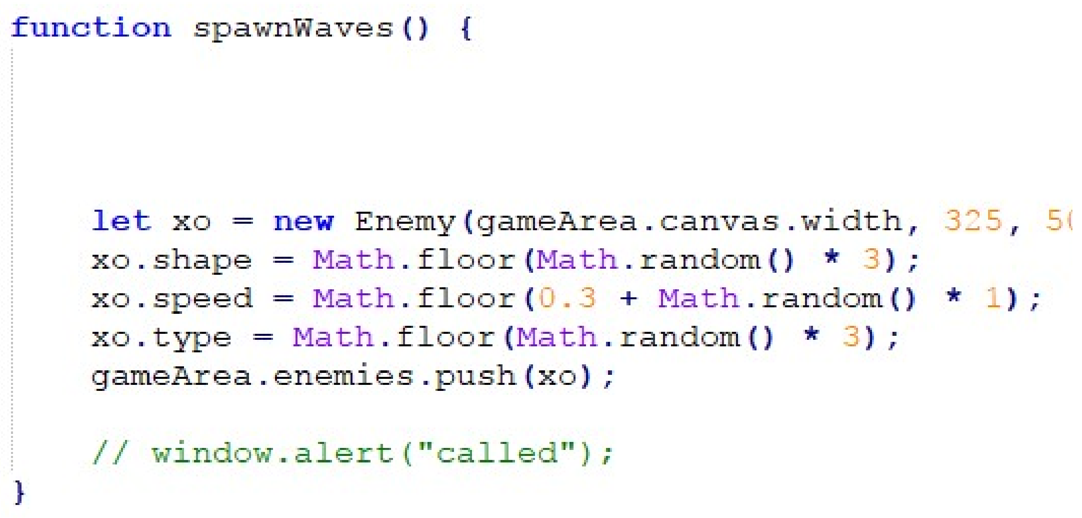
```
},
setAni: function() {

    //fire
    {
        let img = new Image();
        img.src = "Assets/PNG/Fire/PalmFlames_F1.png";
        this.Ani.push(img);
    }
    //movement
    {
        let img = new Image();
        img.src = "Assets/PNG/Fire/VoodooDoll_F11.png";

        this.Ani.push(img); //idle
    } {
        let img = new Image();
        img.src = "Assets/PNG/Fire/FireBall_F1.png";

        //water
        this.Ani.push(img); //idle
    } {
        let img = new Image();
        img.src = "Assets/PNG/Water/PalmWater_F11.png";
        this.Ani.push(img);
    } {
        let img = new Image();
        img.src = "Assets/PNG/Water/HydraMaiden.png";
        this.Ani.push(img);
    } {
```

Fig 13: Function to Spawn the elements



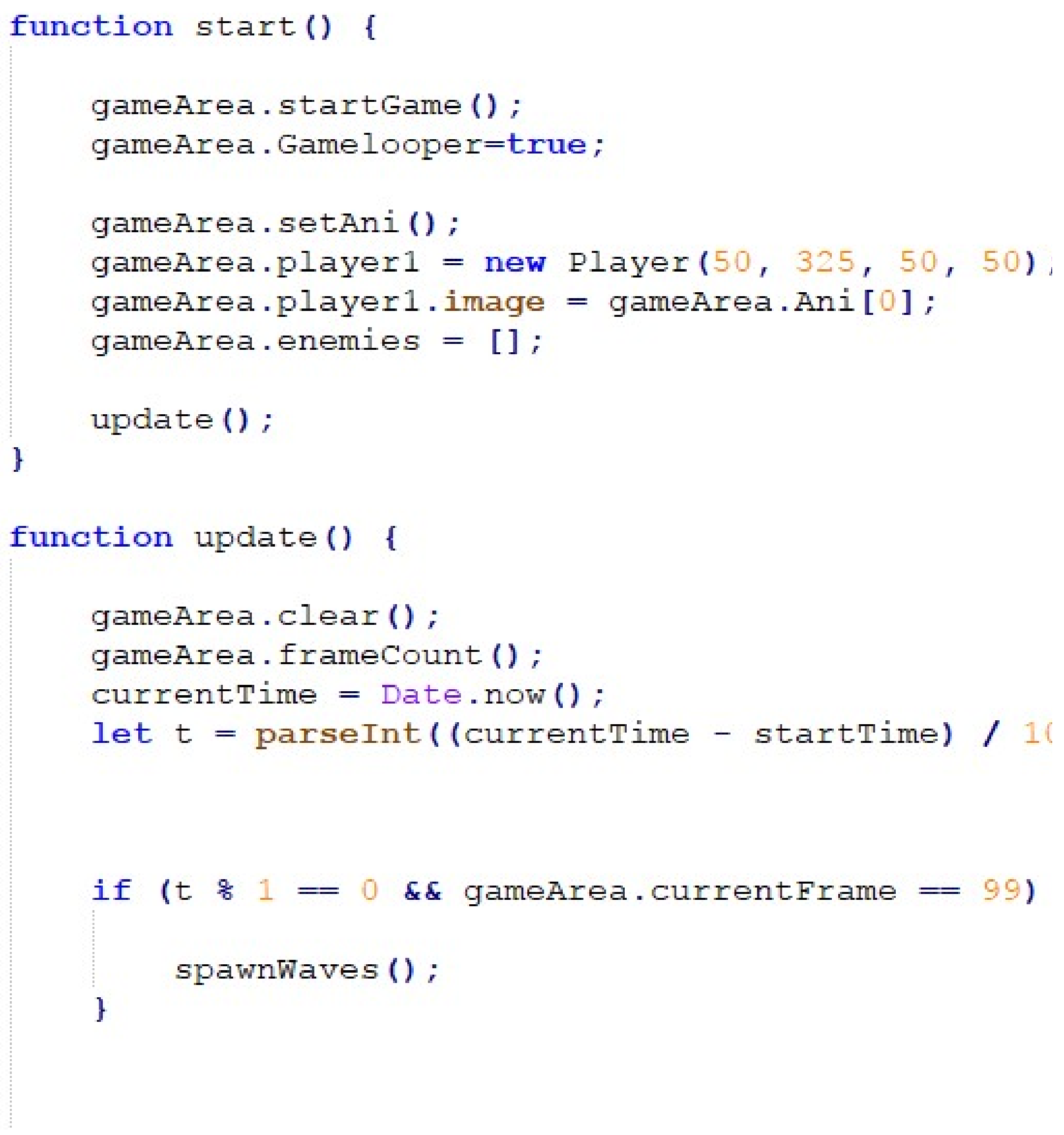
```
function spawnWaves() {



    let xo = new Enemy(gameArea.canvas.width, 325, 5
    xo.shape = Math.floor(Math.random() * 3);
    xo.speed = Math.floor(0.3 + Math.random() * 1);
    xo.type = Math.floor(Math.random() * 3);
    gameArea.enemies.push(xo);

    // window.alert("called");
}
```

Fig 14: The start and update functions



```
function start() {

    gameArea.startGame();
    gameArea.Gamelooper=true;

    gameArea.setAni();
    gameArea.player1 = new Player(50, 325, 50, 50)
    gameArea.player1.image = gameArea.Ani[0];
    gameArea.enemies = [];

    update();
}

function update() {

    gameArea.clear();
    gameArea.frameCount();
    currentTime = Date.now();
    let t = parseInt((currentTime - startTime) / 1



    if (t % 1 == 0 && gameArea.currentFrame == 99)

        spawnWaves();
    }


```

References:

- Kazakov, D. and Tvrdý, L. (2017). Krita. Krita Foundation.
- Knoll, T., Knoll, J., Narayanan, S. and Williams, R. (2012). Adobe Photoshop. Adobe
  Systems.