Production Model

The production Model we will be using for this game, is the Spiral Model.

The Spiral Model consist of the following

- Planning
- Risk Analysis
- Engineering
- Evaluate

It is called a spiral because we start of from the planning stage, where we planned as a team what would we be doing as a game. We decided to go with a simple mechanic that has Jump, move left and Move right for the player and Enemies moving from left to right to cover their area.

After deciding what game mechanics will be used, we continued to choose the sprites that we need and the so that we could start prototyping. We decided to go with free 2d sprites that where downloaded from various websites. After gathering all the information and we planned everything we continued to the Risk Analysis stage

This stage was used so that we could see what risks where there and how we could overcome them. After creating a prototype of the game with a simple box and simple rectangles for the environment we noticed that the game was going to be short, therefore we found an alternate solution by creating a longer map which continued inside a café instead of just being in a jungle.

We started the coding and while coding we tested each part once the code of the part is ready. We start with the simple mechanics and continued to added enemies and other objects.

As for the evaluate part, we gave the game to our friends which they identified other flaws that we might not captured and started all over again by planning how would we change these flaws, creating a prototype of that flaw, coding and testing it and yet again releasing it.

This model is called the Spiral model because these steps are done over and over again, until the product is good enough.

In other project we had created games using the Waterfall Model which was not ideal because if a bug was found at the end it was quite hard to fix because we had to start over again, this spiral model is good because while coding one can test as well therefore for games this is quite good.

Game Mechanics

As for the Game mechanics, as mentioned before we will be using a simple mechanic which the player can jump and move from left to right in a 2D Side scrolled environment.

I first started to code the left and right action which when the player press [<-], he will move to the left, but when the player presses [->] he will move right. To jump the player will need to press the Space bar. For the animation I created frames which would implicate the moving player and even for the Jump and Landing.

To check if the player was grounded, I created a script and an empty object which was a child of the Character and was placed at the bottom, and scripted that if that empty object is colliding the tag called ground, I would set the ground to true. When the ground was set to true this means that the player was on the floor so that I could active the animation.

As for the code I would be using the:

```
(Input.GetAxis ("Horizontal") > 0.1f)

(Input.GetAxis ("Horizontal") < -0.1f)

(Input.GetButtonDown ("Jump") && grounded)
```

For the enemies I had would be creating 2 empty objects which would be named leftTag and rightTag. The Enemy object will be scripted so that he could go from point to point and if the player would collide with the enemy from the sides, damage will be dealt to the player, but of the player would collide with the enemy from the top, the enemy would die.

The code I will be using for the damage done to the player is:

```
//This code will work with
player.Damage (1);

public void Damage(int dmg){

    curHealth -= dmg;

}
```

```
public int curHealth;
public int maxHealth = 100;
```