Task 1

Phase 2

Iterative Model – Rachel Fitz, Francesco Theuma Carabez, Yannick Farrugia

1. **Planning and Requirements:**
   We will start off by brainstorming about some games as well as planning and discussing what we want our game to be about. Eventually we decided that we wanted to make a 3D game similar to *Subway Surf* but with a different purpose. The idea of the game will be for it to be a procrastination game which playfully determines the life of a student. The pick ups would be items which would contribute to good school grades, as well as distracting items which would diminish school grades. We would also like to add a point system which would increase and decrease depending on the items which the player picks up.
2. **Analysis and Design**
   We will then discuss the code breakdown; which code to use to make certain aspects of the game function. Roles will also be given to the group members so that each person would know what he or she is working on. Each group member will all contribute to some coding, but someone will also be entitled to creating the models of the game. The software we are planning to use are Maya for creating the 3D models, and Unity for the actual coding.
3. **Implementation**
   Once we are settled with each person's roles, we will then begin working on our particular tasks. Francesco, one of the group members, will be working on creating the spawners for the random generation of obstacles as well as the game being over when the player touches an obstacle. Yannick will then be working on the movement and jumping mechanics of the player, as well as the point system and timer of the game. Rachel's main focus will then be creating the 3D models of the game; the obstacles, the items, and the character. She will then make the box colliders and on trigger function which cause something to happen when the player collides with the item or obstacle. She will also be working on the menus and switching between scenes in the game.
4. **Testing**
   Each time we'd finish a section of code we would test it by using Unity's play mode. Eventually when the game would be completely finished, we would test it by playing it ourselves and maybe also ask other people to play it and give us feedback on it.
5. **Evaluation**
   When we are done we will then discuss any improvements we can make as well as any changes. If we find any bugs we will also try and remove them.

   Experience from previous project:

   Last semester, the game I produced was a similar version of doodle jump. It is very much different to the game me and my team mates are going to make now. Doodle jump is a jumping 2D game with 2D sprites. Therefore, most of the components I used to make my game functional all included the word 2D, for example BoxCollider2D, OnTrigger2D and OnCollisionEnter2D(Collision2D). However, with regards to controls and mechanics, this semester's game will be quite similar to my version of doodle jump because the player could move the character left and right and we are planning to also do this in this semester's game. A difference is that this time, the player will be able to make the character jump by pressing the space bar; in my game last semester, the character would automatically bounce

each time he lands on a platform. This was arranged by playing around with the physics and gravity of the character as well as the bounciness of the platform. We will also be playing around with the points this time, as well as have a timer. Points will be able to go below 0 this time too depending on what items are picked up. The timer is what automatically switches modes in the game. In my previous game, there was no timer, and points could only increase each time the player landed on a platform.

Game Idea:

We based our idea off the game known as Subway Surfers. In Subway Surfers, you are automatically running in an infinite straight line. This is the case with our game too. In Subway Surfers, since it's a mobile/tablet game, you swipe left or right to move left or right, you swipe up to jump, and you swipe down to crouch. Our game is also going to feature all of this except for the crouch. In Subway Surfers you also collect coins and powerups as you run; the coins can later be used to buy stuff from the game's main menu. In our game one will be able to collect items too and this takes us to the plot of the game. Our game is going to be a procrastination game. The items inside it are procrastination items and school items. Depending on the game mode, procrastination or school, you collect the items you are required to. In Subway Surfers however, you are just escaping from the cops.

Code Needed

Game manager

The game manager will be simple and will contain what happens when the game mode changes.

The Player

The player in this game will have to move from left to right using the arrow keys and use the spacebar to jump. For the movement sideways, we will use the Input.GetAxis ("Horizontal"); so that

we can set the horizontal movement. When jumping we will use Input.GetButtonDown("Jump") so that when the player uses the space bar, he will jump.

Game modes

The game will have two game modes which are Procrastinating and Productive. These modes will be set as an Enum and each mode will trigger a different score. These game modes will switch with each other randomly when the timer becomes zero.

Timer

The timer will be set for 20 seconds. When the timer arrives at 0, the game mode will change and the timer will reset.

Score

The Score increases or decreases according to the game mode and what obstacle will the player hit. For this we will use an if function to set what happens with each game mode. When the game mode is Productive the item with the tag good will give a positive score while the object with a tag bad will give a negative score. While when the game mode is Procrastination the item with the tag good will give a negative score while the object with a tag bad will give a positive score.

Game over

The game over will be set when the player hits an obstacle or the Score is +20. Whenever the player hits an obstacle the spawner stops spawning obstacles and the game over screen will load.

Scenes

The game will have three main scenes: the menu, the game and the game over. We will use the SceneManager.LoadScene to load each scene.

Spawners

The spawners will be used in the game to spawn items and obstacles. The spawner will have a time delay so between each spawned item there will be some space.

Items

The items we will have in the game are obstacles that when the player hits them they the player will loose and the pick-up items which will have tags attach to them and according to the game mode the pick-ups will give either a positive or negative score.