Game Engines

Task 1

The game which I am going to make a replica of is **Bubble Struggle.** The game itself is not very complicated and it is only going to have one goal: to destroy the balls without being hit. I have gone over all the software development models and I found that the best model to use in my case is the **Scrum Agile Model**.

Scrum Agile Model

With Scrum I can start working right away. The first thing to do is to provide a visual of how the finalized product (in this case the game) is going to look by designing a few screens from the game using Photoshop/Illustrator. If I don't like some of the visual elements, I can simply change them and make improvements (sprints) which is basically taking a step by step approach. By using the Scrum model I will be gaining continuous feedback and making changes as I go along.

The Scrum model has 3 phases:

Phase 1 is where I create a **product backlog,** which is basically a list of features that the game will have. The backlog will also contain game features which are not necessarily going to be developed, but if implemented, could perhaps make the game better, more challenging and more enjoyable.

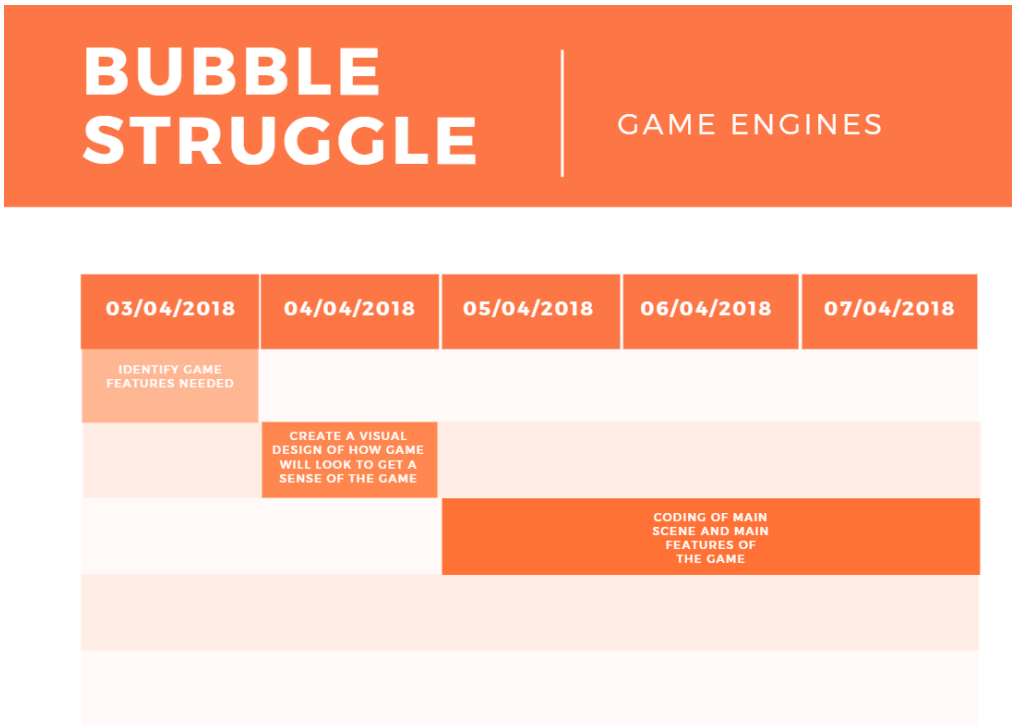
Phase 2 is where I **estimate the time** it is going to take for implementing the game features. Next is **prioritization**, where I will prioritize the game features by most important to least important.

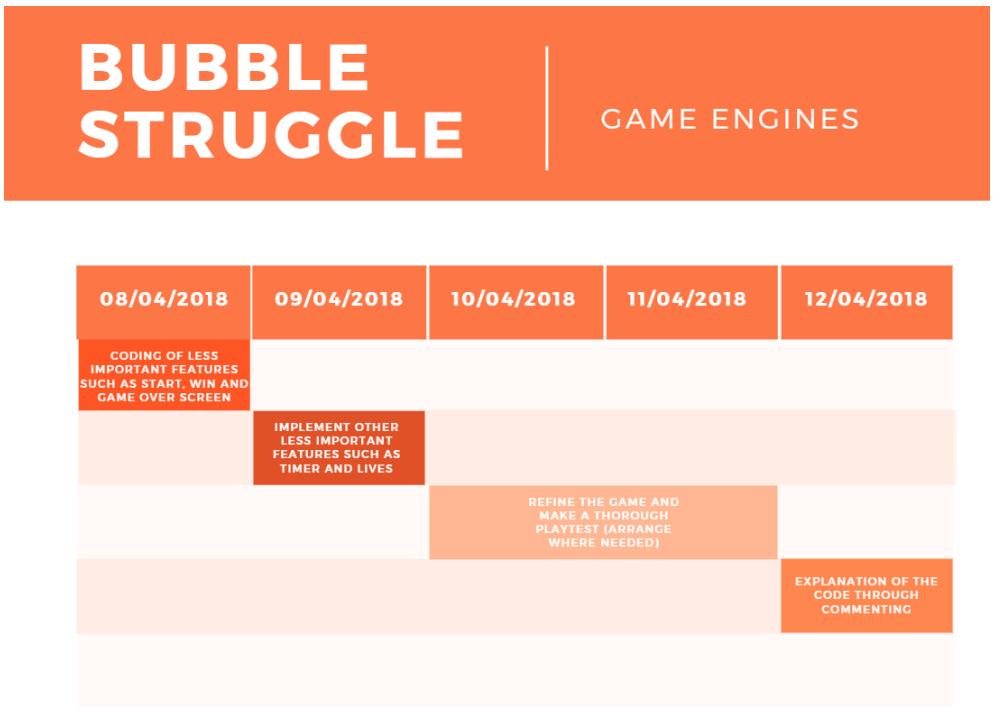
Phase 3 is the **sprint**. This is where I choose the most important features of the game to develop and execute them. Since the first game is done solo and does not require any meetings, I can simply carry out a feature of the game, check that it works accordingly by carrying out a playtest, and if it works, I plan out my next game feature/mechanic to develop. This phase will be repeated until all features are carried out. Each time a feature of the game is carried out, the next one immediately begins.

To sum it all up, Scrum is the most ideal model for me because it is adaptable, customizable and can easily modify this model according to my needs.

According to Clinton Keith, author of the book "Agile Game Development With Scrum", when asked in an interview regarding the development of game projects before and after using Scrum Model, Clinton Keith answered:

We were very disorganized, forming Sammy Studios, growing to hundreds and starting big games. Thousands of daily problems. Scrum improved this by engaging everyone in daily problem solving. (Keith, 2016)





n.b: Any errors/bugs found while coding will be fixed at the same time it was found to maintain an Agile Scrum development

Components and Code Structure

The *main* objects/features for Bubble Struggle will include:

Main Camera – Without any cameras, we would not be able to see the game and would simply see a black screen with the message: "Display 1 No cameras rendering." The settings for the main camera are left at default.

Background – Although a background can be set by simply changing the colour of the Background in the Camera component, I still think it is good practice to simply create an Empty Object and use the Sprite Renderer component to customize the background and this would give more flexibility as well in terms of positioning.

Player – The player will be needed so that he could position himself in the game and try to destroy the balls without being hit. The player would have a Script component which will allow him to move on the X coordinates as well as an OnCollisionEnter2D so that we would know when something collided with the player. The player would also have a Circle or Box Collider 2D to define the area of the collision. Furthermore, a Rigidbody 2D is also required so that it collides with other objects.

Walls – Walls are required to confine the game within a limited space. It does not require a Script, however Rigidbody 2D and Box Collider 2D are needed to restrict the player from going out of the game screen and also acts as a stopper to the balls and the ball destroyer.

Balls – The balls are what brings the challenge to the game. They will have to be bouncy in the game and so a Physics Material 2D is required, which will be attached to the Circle Collider 2D component. A Rigidbody 2D is also needed. The ball will have a Script as well, which will give an initial force to the ball so that it won't stay bouncing in one place. When the ball is hit, it will split into 2 smaller balls until all balls are destroyed. We will know that the ball is hit by creating a tag.

Ball Destroyer – This will have two Scripts. One script would be to control the Destroyer and specify its position and the other one would be for when the Destroyer collides with the ball. The Ball Destroyer will have a Box Collider 2D component and the Is Trigger will be enabled.

Each object in the game will have a transform component for obvious reasons such as to scale, rotate, move and position the object in the scene. For certain objects such as the player and the ball destroyer, I would also need to know their location to set certain constraints as well as attach 1 object with the other and this would not work without the transform component.

The *secondary* objects/features for Bubble Struggle will include:

Canvas with Text – This would be a good practice to use to make the game more user friendly and a canvas with text would be used before starting the game and after finishing the game by either losing or winning. Below are some basic visuals of how they would look (not finalized)

Materials and Textures – These are what makes the game appealing to play and also creates contrast between different game objects.

Lives – Limiting the player with lives could make the game more appealing as well. This however would be more useful for a game with different levels. A Script would be needed with a defined counter, and each time the player is hit by a ball, counter is decreased by 1. A Text component to show the user the number of lives left would also be added.

Timer – To make the game even more challenging, a timer could also be added to the game. This would require vigorous testing however to find the right balance of time that should be given to the player before losing a life. The Script for this would have a timer (example: 10 seconds) and an if statement to show that if the timer reaches 0, the player loses a life. The timer would be displayed by a Text component.

References

Powell-Morse, A. (2018). *Scrum - What is it and how do you use it?*. [online] Airbrake Blog. Available at: https://airbrake.io/blog/sdlc/scrum-what-is-it-and-how-do-you-use-it [Accessed 3 Apr. 2018].

Ashley, Davis (2016). *Agile Game Development with Scrum: Book Review and Interview*. [online] Available at: http://www.what-could-possibly-go-wrong.com/agile-game-development-with-scrum/ [Accessed 3 Apr. 2018].

Scrum Company. (2018). *Scrum in under 5 minutes*. [online] Available at: https://www.youtube.com/watch?v=2Vt7Ik8Ublw [Accessed 3 Apr. 2018].