2018

Fresh Avocado Analysis

JEANLUKE ZAMMIT, JAMES DIMECH, ANDREW WORLEY

BA IN INTERACTIVE MEDIA & GAME ART LEVEL 6 YEAR 2

Fresh Avocado Analysis

Introduction

Fresh Avocado is a traditional 2-D platformer-puzzle game with the player being a normal everyday Avocado. The aim of the game is to help keep yourself fresh by collecting batteries to power up the fridge in the house. The purpose of the game is to influence players, mainly children, to relate to the avocado emotionally and to eat healthy and always eat fresh fruits and vegetables. The player has two "enemies" to look out for in the game, the first being the bumble-bee which when the player approaches, will begin tracking you and attempting to hurt you. The second is the time itself, If the player takes their time attempting to complete a level, the time will continue to rise, and the avocado will decrease in health resulting in a not-so-fresh and not-so-happy avocado.

Software Model to be implemented

The Waterfall development model will be used for the production of this game due to its linear nature allowing for the making the overall development more convenient for the developers as well as for the project itself.

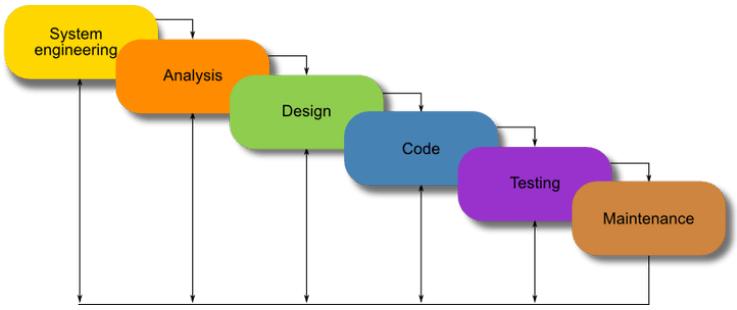

Figure 1 - WaterFall Model *(Powell-Morse, 2016)*

Requirements

The initial phase for development which requires a list of the requirements of the application that the developers will be creating. (Powell-Morse, 2016)

For Fresh Avocado there are six key features which need to be implemented. First, is creating the levels that the players would traverse. Secondly, movement needs to be established with basic controls and jumping. Third is implementing the hazards in the game, such as the enemies, the time limit and the kill plane at the bottom of the level in case the player falls off the stage, Fourth, is the health system which will determine the Avocados Freshness. For the fifth, batteries will then be implemented to allow players to unlock the end of the level. Finally, for the Sixth, we will implement the time limit and its ability to cause damage over time.

2.Analysis

The next phase is to view all the 'potential requirements' that have been listed and implement each requirement in order of importance. (Powell-Morse, 2016)

- As mentioned in the first phase section of the development model, the first feature that needs to be set up is the level design. Paired up with that are the controls which will enable the sprite to be moved around.
- Following that are the enemies, which need to have the ability to take health away from the player upon contact as well as ability to move in a fixed pattern in order to keep the process simple. Along with this, the Health system will then need to be added, alongside the ability to lose health upon contact with an enemy.
- Once the level design, navigation, enemy and health have been implemented the batteries will then be implemented along with the activation/unlocking of the levels end goal.
- Finally, the time limit should be implemented in order to motivate players to finish the level as quickly as possible.

Design

The third phase is where the developers would take all the important factors which were planned in the analysis section and initialize preparation for the project's development. Such factors could consist of assets, scripts, and the programming language amongst others. (Powell-Morse, 2016)

The game will be coded and built in the Unity Game Engine and would be coded by using the C-Sharp (C#) programming language. Sprites and other visual assets will be designed and produced using an application called Pixilart due to the game's 8-bit art style as well as Pixilart's proficiency with said art style.

Coding

The fourth phase is where all the required assets, logic and features will begin being implemented and coded in order to fulfil the project's demands and bring it to life. (Powell-Morse, 2016)

Testing

The fifth phase is where testing is conducted, as well as debugging and fixing any errors found. This phase is a repetitive cycle due to the unpredictable amount of errors which would be encountered and need to be fixed. (Powell-Morse, 2016)

Operations

The final phase of the development where the product is finalized, compiled and exported. The final step in developing a game is to create a "build" which will represent the final product. Most projects perform maintenance and fix other bugs and add new features to the product in the form of updates to keep the software fresh and running smoothly. (Powell-Morse, 2016)

References

Powell-Morse, A., 2016. *Airbrake.io.* [Online]
Available at: https://airbrake.io/blog/sdlc/waterfall-model
[Accessed 01 05 2018].