Github repository: https://github.com/Ultimate-Anarchy-Team-COSC345/COSC345.2
Team name: Ultimate-Anarchy-Team
Game Name: COVID-Raider

For beta documentation scroll down

We are using multiple versions of XCode to develop our project. (Xcode 9.2 and 11)

In Xcode we have split the files into subfolders based on what they do for easier understanding such as Assets, ViewController, Template, and Scenes.

The Scenes folder contains GameScene.swift which contains the scene for our game. Currently the alpha will only utilise this scene so it also contains functions for behaviour and controls.

Scenes we will be including in future:

- Main Menu
- Options (sound, brightness, etc)
- Player Select (choose sprite)
- Game Scene
- End Scene (this is where the anarchy happens)

In Assets the Assets.xcassets are where we are eventually going to add our own graphics (characters, items, Non-Player Characters) that we design. Currently we have placeholder graphics for the playerSprite and Non Player Characters, named blinky.

In the Template folder we have the AppDelegate.swift will handle application life cycle events such as responding to the app being launched, and receiving data.

For the Alpha release we have implemented a virtual joystick to control the player, with two buttons A and B, a physics enabled player, an enemy Non Player Character (NPC) that destroys the player on collision, the enemy npc currently follows a fixed path and a basic scoreboard to display the player's score.

The joystick function, by tracking its position in a circle around it's base as an angle in radians. That angle is then put into a switch which holds ranges of angles to convert the joystick into up, down, left and right directions. The player turns to face the same direction as the joystick. The directions are then passed to an SKAction move method to move the player in the appropriate direction.

The player is a basic sprite with collisions enabled to keep it on screen and to detect enemy NPCs. If it collides with an enemy NPC the player sprite will be destroyed using removeFromParent.

The A Button will be used for interaction with objects, such as the carrot or other food items and its function will be to collect them and add them to the shopping cart. For now it only shows a different coloration in when it has been pressed.

The B Button will be used for an options button and a backwards button when navigating the main menu. For now it only shows a different coloration in when it has been pressed.

For future releases we want to implement more advanced pathing for the NPC, possibly something similar to A*. The NPC's behaviours will be written as their own script to enable loading of multiple NPCs with the same behaviours. We also intend to implement a map of the supermarket using a tile-map style system to allow for flexibility of level design.

When the game launches a user will have to choose their player from a main menu. To play the game there will be a Joystick touchpad in the middle of the bottom of the screen for the user to be able to move the player around. Also to grab items in the game, the user will have to tap the item (while pressing the A button) and the object will be worth a certain amount of points every time a player taps an item the scoreboard for will increase. The scoreboard is going to be a list of variables and each variable is worth a predetermined number of points.

We will have a predetermined value that is an acceptable number of points a user can score in the game. If the user exceeds this value they will receive a message at the end letting them know how many people they have disadvantaged with greed. If the user does not exceed the predetermined value they will receive a message letting them know they only got what was necessary. Also based on how well they avoided the AI players there will be a count of how many viruses attached themselves to the player for example if a player collides with 12 AI players then the amount of virus particles caught is 12 million (this could as be made into a percentage).

References that we used to aid in creating this Alpha:
https://www.raywenderlich.com/2399-ios-7-game-controller-tutorial#toc-anchor-002

Beta Version Documentation

The Beta Version of the app began with a ground up rework of the game's mechanics and graphics drawn from scratch(spare one character). The current version of the game contains a player that is static on the y axis and can move from side to side.The aim of the game is to collect resources that move towards you from the top of the screen and dodge infections/enemies.

Currently the game contains three sprites, a food basket, a player and a villain. The food basket is what the player needs to collect to increase score, at this point, the game is won if 250 points are accumulated, and the game is lost if the three lives are lost.

COVID-raider has three scenes: the menu scene, the game scene and the end of game scene. The menu scene is very basic and to start the users touches the word play, from here the game transitions into the game scene which is where the base of the code is. The player is controlled via swipe gestures, where a user swipes left or right on their screen and the player sprite will move according to the swipe. The other sprites (villian and food basket) are spawned at random based on a timing function and two randomised number functions.

The collisions between the player and villain and the player and food basket are done using contact test bitmask this is used instead of collision bitmask as collisions are used so that sprites bounce off each other and contact is used when a node a comes into contact with node b and instead of bouncing off each other, other actions occur.

The background is also a sprite and is formatted to fit the size of the screen. At current the only sprite not designed by our group is the player which was downloaded from itch.io. Also the only sprite at current to not have a physics body related to it.

The end of game scene is where the user gets a message about if the coronavirus killed them or if they killed the elderly. This is still in the early stages and this scene will be revamped for the final release.

The future directions for the final release include sound effects, basic music, a high score function, more polished scene transitions and increasing difficulty during gameplay.

The game will start off at a steady pace and gradually increase speed of entities moving down the screen over approximately 3 minutes. Different levels of difficulty will spawn multiple enemy types that can further apply debuffing effects to the player. Potential effects include slowing the responsiveness of player controls and point reduction. A new beneficial facemask sprite is also planned to give the player a 1- hit shield.

A further graphical improvement will be adding screen flashes when the player collides with damaging entities.

Issues that we ran into.
We struggled to get the artwork working correctly for a while, it was a simple fix but found it hard to diagnose. The pathway for the artwork was being stored on our personal device rather than with the game itself, and was only being referenced.

Another issue was understanding the collisions, we had a very difficult time with the physics body of our artwork, this was fixed by scaling down the physics body to match the size of the sprite and the invisible field around that node disappeared. This issue did cause all the entities to go out of wack and caused a cascade effect on the other spawned nodes.

References for Beta
https://lionheart963.itch.io/4-directional-character