Github repository: https://github.com/kea5555/Ultimate-Anarchy-Team
Team name: Ultimate-Anarchy-Team
Currently our game is called COSC345.2 and this will change when we confirm a name. Potential names include COVID-Raider, Doomsday Preppers, Panic Buy, Run COVID Run. In general we are creating this game bottom up, and so the name will be a finishing touch.

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.sks 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