Readme File for COMP 4478 Assignment 2 Parth Joshi (1126914)

[pjoshi4@lakeheadu.ca](mailto:pjoshi4@lakeheadu.ca)

Github Link: <https://github.com/parthjoshi1202/COMP4478_As2_Parth>

Unity Version: 2021.3.20f1

In this game I used Prefabs of the card sprite (blank.jpg) to make a memory game called cartoon character in Unity 2D. I created a 4x4 screen with blank cards. If the user flips two unmatched cards these cards would flip, and user would play another round. If they are 2 matched cards, they disappear. Once all characters are identified, a Game over Scene is loaded with an option to play again where the characters would have different positions as compared to the previous session.

**Controls:** Left Mouse

**Front end:** I had set game scenes, one for the Game which is the main scene and the other being the Menu scene which shows up once the game is over. For the game scene, I had set a background, a Grid Layout Group to organise the 16 cards, a Prefab for the 16 cards to be displayed as well as a Game Controller to facilitate the game logic. The menu scene had a button, which restarts the game, when clicked.

**Back end:** These game objects are connected to C# Scripts. I started with AddButton.cs where 16 copies of the prefab are generated and also given a name along with setting it up in the grid layout called “cardPanel”.

In GameController.cs, I stored the buttons as well as the sprites in separate Lists and arrays. Then I obtained the buttons generated before by tagging them to obtain them easily as well as setting a background of “blank.jpg” for each of them. After that the sprites are positioned randomly in the Grid Layout where are hidden by the “blank.jpg” image, using the RandomCard() function. AddGameCards() function ensures that there are 2 sprites generated for each sprite since it’s a 4x4 screen and there are 8 sprites along with ensuring that only 2 cards are displayed at a time. Now the PickCard() function stores the first and second guess (first and second card clicked), its index and its image. If the user did not guess the card for the first time, the first guess is set to true. Same goes for the second guess and then these are sent to the other function, checkFinish() which checks if the both the guesses match or not. After a second of waiting, if both the guesses match, the buttons (the cards) become unclickable, and the cards disappear. This was possible using the index of that card. If they don’t match, they flip back, and the user selects another card. The checkFinish() function counts the correct guesses the user made, once they are equal to the actual number of guesses in the game, that means once all cards are matched, the scene manager prompts the user to play again. Thus, showing the game over screen and then restarting the game with the new position of sprites generated randomly by the RandomCard() function.

Finally, the Menu Scene has SceneManager.cs script attached where it uses switch case to navigate to the Main Game Scene again.

Here are some screenshots of the game.













