Gesture-Based UI Development Project Documentation

GitHub: https://github.com/Panachaijames/Project-for-Gesture-Based-UI-Development

Team: Panachai Thongvinit/ Pantaree Somcharoen

Introduction

The goal of our project is to develop an application with gesture-based systems. There are many different options for technologies available to create a gesture-based application. After browsing through the options, we decided to go for the android gesture games, including drag, tap, swipe, tilt phone, and shake gestures.

Purpose of the application

The purpose of our application is to build a game with a mini-game within it. Each game will have a different gesture, for Example, the Air hockey game will use the Drag gesture, The Flappy bird will use a tap gesture, the Maze will use the tilt phone Gesture, etc.

UI Design of application

The UI design is different in each game, The Air hockey will naturally have a background of a hockey table. There will be two-goal, one for the red side, or a player, and one for the blue side, an AI, or another player if played on Multiplayer mode. The score will be on the right side with the pause button. See figure 1 below.

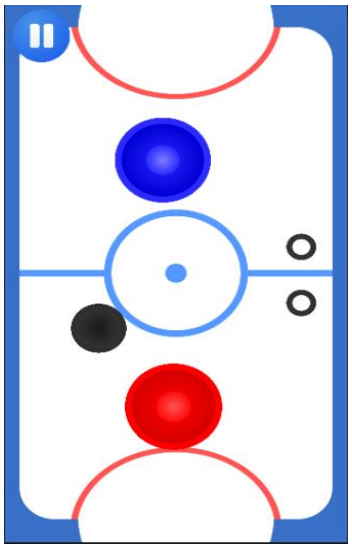


Figure 1

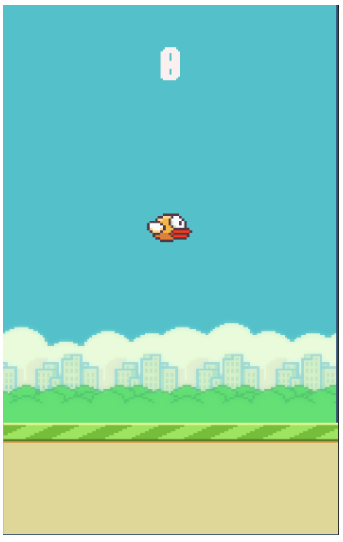


Figure 2

The UI design for Flappy bird has the sky and the far-away building as a background. The ground will move as the animation played. The score will be on the upper middle of the screen. See figure 2 above.

The Maze UI design is pretty simple as it only has a grey background and a pause button on the upper-left of the screen. See figure 3 below.

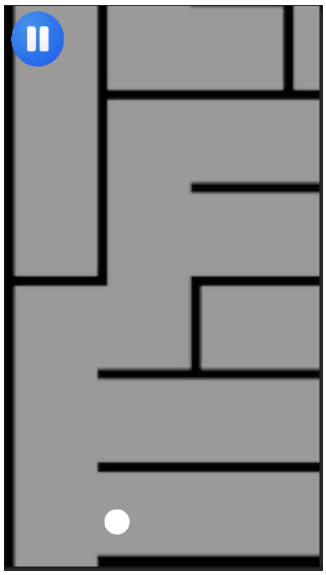


Figure 3

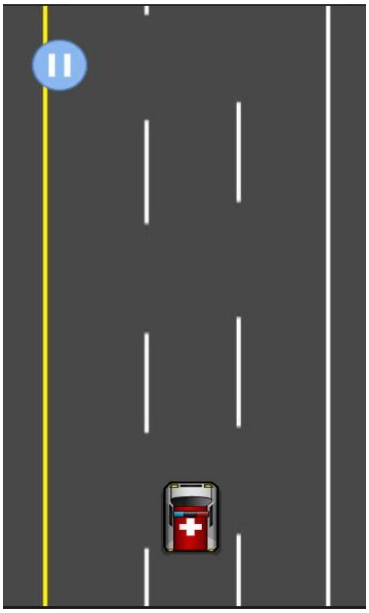
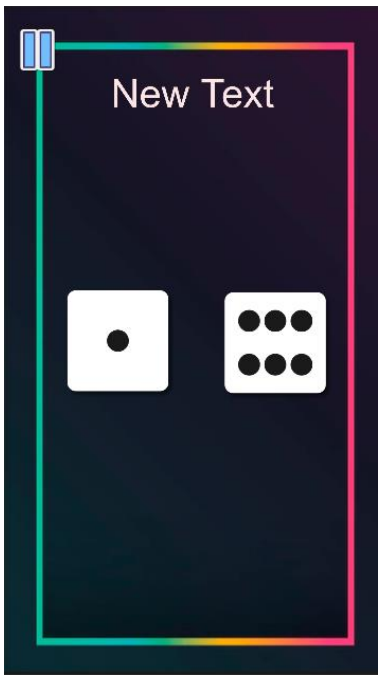


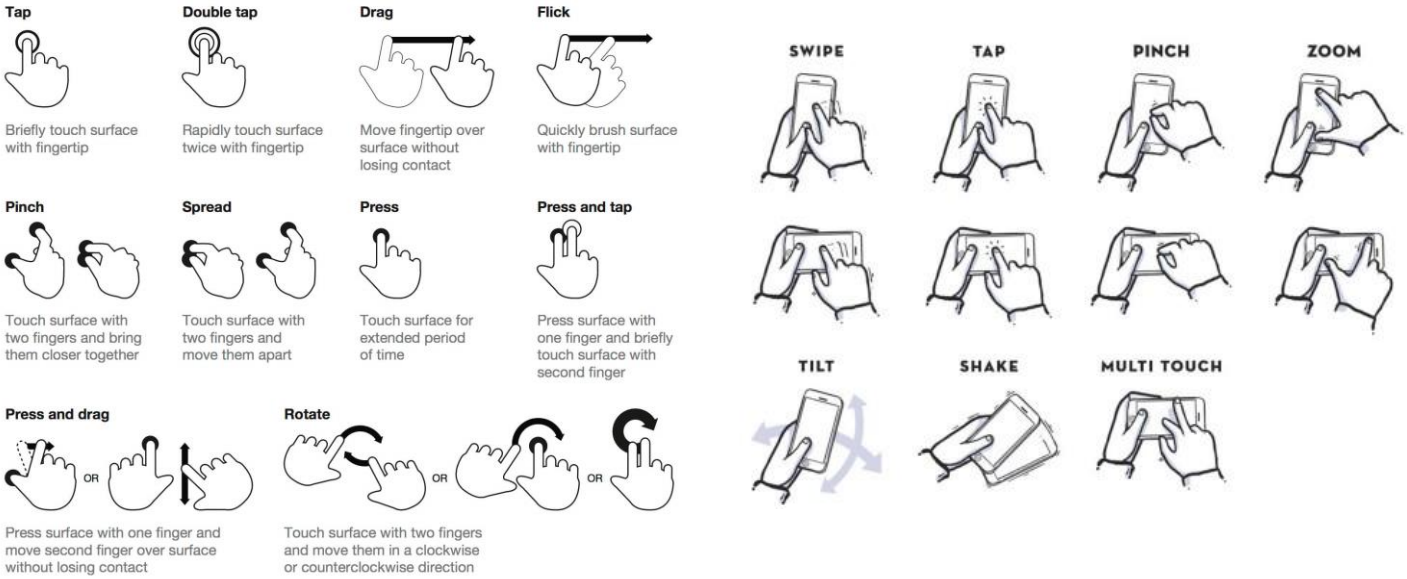
Figure 4

The Ambulance UI design is made referring to the road in real life, there will be a pause button on the upper-left of the screen, a grey road with consist of three-lane for the player to move and the edge line pavement marking on the side indicated the limit in which the player car can go. See figure 4 above.

The UI for dice is also quite simple as it consists of 2 things, the background with a combination of green, purple and dark colour, and the pause button at the upper-left side of the screen. The score will be in the upper middle of the screen.



Gesture identified as appropriate for this application



The gesture that is used for Air hockey games is normally the thumb or Index finger for controlling the strikers through the Drag gesture. The justification for the Thumb finger is that The thumb is commonly used as the main finger for a mobile phone, especially for this game which is played in portrait mode. However, the index finger is also important as it allows the player to have more space, which leads to more control of the paddle. The reason drag control is used in this game is that it allows the player to have more control in direction of the strikers than the other gesture used for android.

The Flappy bird game used the tap gesture and normally used the thumb finger to play as it should be the most stable finger, however it depends on people as some people are comfortable with the index finger to play as well. The reason it requires one's stable finger to play is that if the bird hit anything then the game will be over and it will only make it harder to use other fingers.

The Maze game uses the tilt phone gesture so it did not require one to use a finger except to use them to pause the games. The justification to use the tilt phone gesture is that the gesture provides a new way for the player to interact with the game, furthermore tilt phone gesture is also a more appropriate gesture for this game than the other gesture.

The Ambulance uses the swipe or flick gesture to control the car to go to the wanted lane, so it usually uses the Index finger to play. The reason flick gesture is used in this game is that it is better to use an easy gesture with little movement to change the lane as it could help players play the game more efficiently and made players have a feeling of more control over the car.

The dice application uses a shake gesture, therefore it doesn't need any finger to play. The reason the dice application used this gesture is that it is easy for the user and it is perfectly fit for replicating the action of rolling the dice.

Hardware used

Using a mobile phone gives a high level of playability for the user; there is a strong likelihood that the user already knows how to play the game, as opposed to the other option, where the user would need to learn the controls to play the game correctly. Because of the large number of mobile phone users, mobile phones offer handheld mobility and a strong development kit. Moreover, mobile phones can easily be tested and improved in further development.

Architecture for the solution

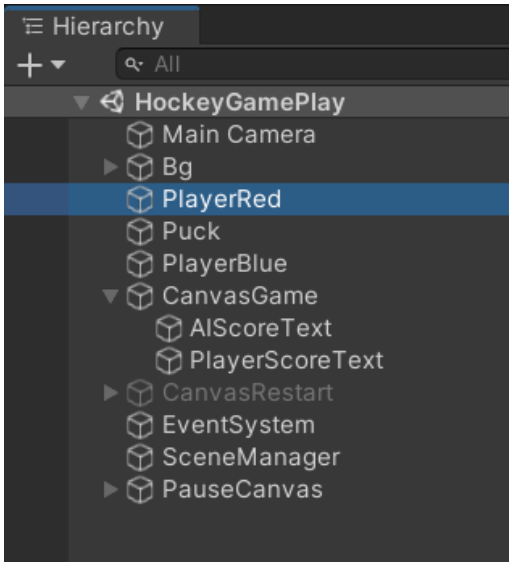
Unity is an Entity-Component (EC)-based development platform in which all entities are instances of GameObject and the features that allow them to be "visible," "movable" and so on are provided by classes extending Component. [1]

The structure of the application is in a 2D environment with a selection of games Hockey, Flappy bird, Maze, Ambulance and Dice.

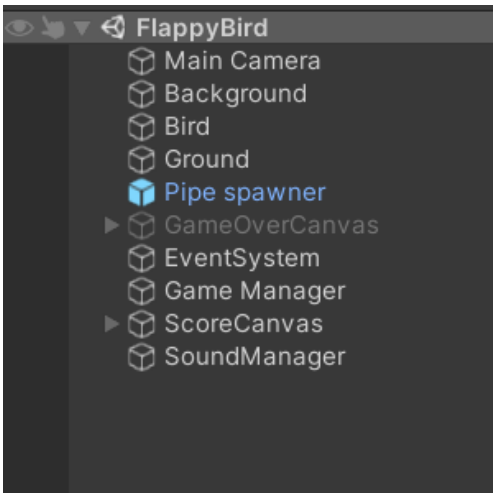
Hockey

The main camera is pointing to the hockey table from above. The game has 3 game objects PlayerRed, PlayerBlue(AI) and Puck. Each game object has RigidBody and collider component with the Puck has a physical material inside the rigidbody for the bounciness when it collides with the wall.

For the input system, we implemented the touch input which references from [2]. TouchPhase.Began checks when we begin to touch the screen and if we are touching the paddles(PlayerRed or PlayerBlue) [3]. After checking the paddles will follow players' fingers in a drag gesture.



Flappy Bird



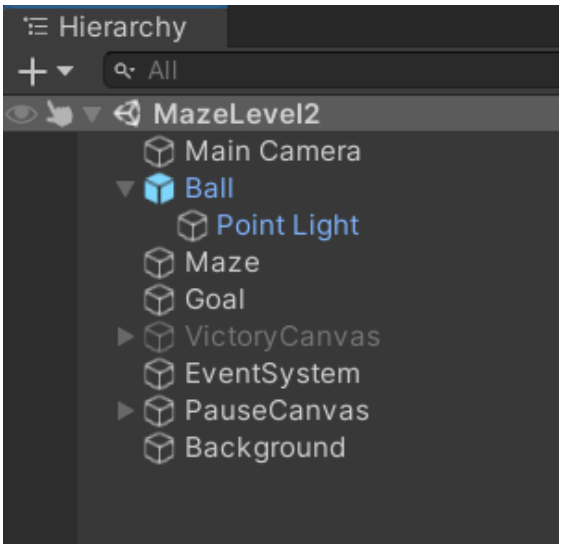
The main camera is looking to the right of the bird. The bird object has a Rigidbody and Collider. The pipe spawner will spawn pipes every 2 seconds, and the height of the pipes will randomly place at the 0.75 range.

The touch input system is using the tap gesture, in which every tap will make the bird fly up. The velocity of a bird flying depends on how rapidly the input is [4].

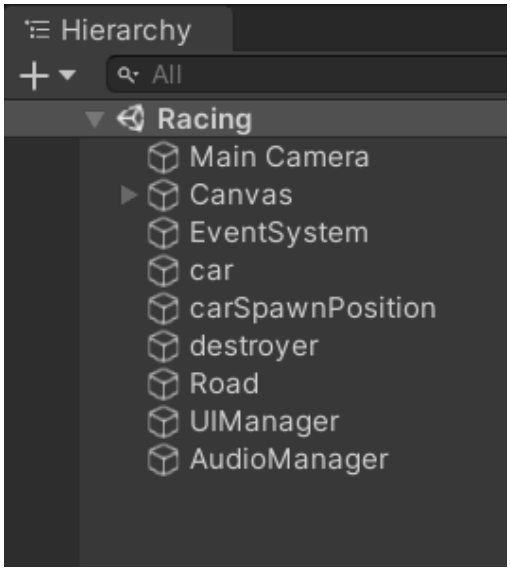
Maze

The main camera in the game will follow the ball while moving in the maze. The ball object contains Rigidbody, collider and Trail renderer. The player has to reach the winning point in order to move to another level.

The input system for this game consists of the input of acceleration with both the x and y-axis. So, the player can tilt the phone in both directions to keep the ball rolling. The gesture that is used in this game is the Tilt gesture. [5]



Ambulance



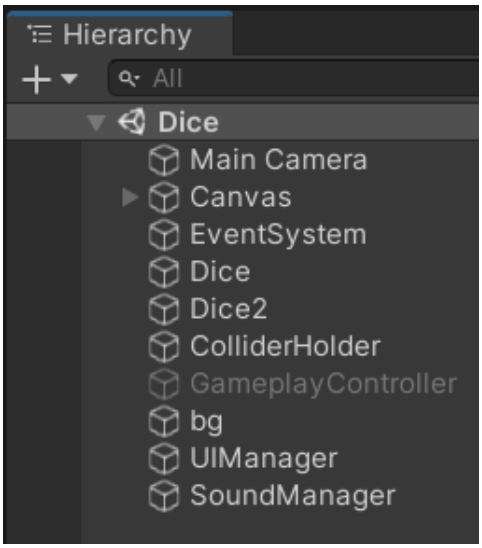
The camera is pointing to the ambulance car on the street from above. The car object has Rigidbody and collider. The car will move at a speed of 10f with a minimum position on the x-axis is -1.6 and a maximum position is 1.6. The enemy car spawner will randomly spawn cars every 0.8 seconds.

The gesture that is used in this game is the swipe gesture, which will detect 2 directions right and left. It will detect if the user swipe left or right by keeping the value in the variable when the user starts to touch and compare it to another variable which has the value when the user releases their finger from the screen. If the variable at the start is more than the value at the end, then it means that the user swiped left or else the user swipe right.

Dice

The camera is looking at the sides of two dice with a dark background. Two dice objects have a collider and rigidbody. ColliderHolder contains a physic material called BouncyMaterial that takes care of dice bounce when hitting the collider.

The gesture used in this part is the Shake gesture which used input.acceleration to indicate the direction of the dice. The rigibody.addForce function to apply the force for the dices to go in that direction. [6]



Conclusions & Recommendations

There are a lot more mobile gestures that are not covered in this application which are interesting. We learned that from Unity touch input, it is not limited to only one gesture but can be enhanced to many gestures for example rotate and flick. Moreover, mobile gestures are commonly a part of our life, that we sometimes do not even realise. That, makes us realise how important and useful it will be when we graduate from ATU. This opportunity was brilliant, it can be applied in not only to the mobile game but to the mobile itself. Overall, we have achieved most of our objectives set at the beginning even though there are some bugs and errors in the project.

References

[1] https://www.toptal.com/unity-unity3d/unity-with-mvc-how-to-level-up-your-game-development

[2] https://docs.unity3d.com/Manual/MobileInput.html

[3] https://docs.unity3d.com/ScriptReference/TouchPhase.html

[4] https://docs.unity3d.com/ScriptReference/Input-touchCount.html

[5] https://docs.unity3d.com/ScriptReference/Input-acceleration.html

[6] https://docs.unity3d.com/ScriptReference/Rigidbody.AddForce.html

https://www.makeuseof.com/mobile-gaming-is-the-future-reasons-why/

https://www.gamesradar.com/why-mobile-games-are-so-popular/

https://www.justinmind.com/blog/tap-or-swipe-mobile-gestures-which-one-should-you-design-with/

https://thenextweb.com/news/how-to-implement-gestures-into-your-mobile-design