**Mighty Kingdom development log 24/1/2019**

Thursday 24 12:30pm  
  
Just started this project as I have just finished another project and didn’t have time earlier. I am setting up public a git repository through Github.  
  
I have setup a repo at: <https://github.com/Matteoson/MightyKingdomTest>

Having a look at the brief. The main tasks to do are as follows. I have added my initial thoughts on how to accomplish each task in blue  
  
What do I need?  
- A surface for the character to move on – easy, I’ll create a flat plane  
- A character that can move and shoot (a basic projectile is fine) – I’ll use a voxel obj model I have  
- A camera that follows the character around – if the camera is a child of the player character it should track nicely   
- Two joysticks on the screen to control the character – look into the UI and input management system to do on screen controls as this is something new for me

The requirements for the character’s movement are as follows:  
- 2 virtual joysticks (bottom left and bottom right on screen) – no problem seems logical, need to learn how to do virtual joysticks  
- Left stick moves the character in the direction of the stick input – left moves got it.  
- Left stick faces the character in the direction of the stick IF there is no input on the right  
stick – OK. The logic is left stick faces character if is the character is not currently shooting.

- Right stick faces the character in the direction of the stick and automatically fires the gun|  
when any input is detected on the stick. – Ok character is always facing the direction it is shooting  
  
**Approach:**  
  
Okay I’ll first start a new unity 3d project, create a plane and a player / character object and see if I can find a nice model for it.

Created gameObject called it player, tagged it as player and gave it a collider. Imported an Obj of a knight from Magica Voxel

Created a Plane object and called it ground, gave it a collider and disabled gravity.  
  
Started looking online for tutorials on touch controls as this is new for me.  
  
Watched one which relied on a free joystick asset. Didn’t go with this solution as at this stage wanted to learn how Unity does touch controls and not rely on an asset.

After watching a few videos such as the ever popular “Brackeys” video. I understood how to write my own code for a joystick input but for this project I decided to go with the free asset to save time. Brackeys video (<https://www.youtube.com/watch?v=bp2PiFC9sSs>) gave me some guidance.   
  
I downloaded ‘Joystick Pack” a free asset – needed a more recent version of Unity 2018 for the asset to work so started downloading Unity Hub and a newer version   
  
While the new version of Unity was downloading worked on the shooting aspect. Imported a bullet model and made a bullet prefab. Also made a shot spawner as a child of the player object.  
  
Created a game world boundary and relevant scripts so stray bullets outside of the game boundary will be taken care of.

Switched to a Unity 2018 version and installed / imported the Joystick Pack asset by Fenerax studios  
  
Started a player movement script (in retrospect should be called player actions or controller) this is the script that listens to the joysticks input and converts those inputs into actions for the player. ~~Created two Enums to handle all the player actions and the relevant functions. I put player’s actions into functions because then its easy to have multiple input methods into the game.~~   
  
Realised that the movements needed to be analogue and 360’ degrees so I used the floats Joystick.Horizontal and Joystick.Vertical to create a vector3 for the rigid bodies velocity. This created movement.   
  
To get the character to face the direction I used the Quaternion.LookRotation method and passed through the rigid body’s velocity. I made a vector3 variable of idleRot which stored the last rotation which is used to make the character face the last direction it moved in. To clear the rb.velocity I used an IF and ELSE statement based on the joystick movements.  
  
Since the bullet movement script always shoots in the forward vector and the bullet is always fired from the character facing forward if the rotation of the character works when a bullet is fired it will shoot in the direction the character is facing.

To get the camera to track the character. I created a script for the camera which contains an offset and a public gameObject of the character the camera always follows the position of the camera plus the offset. This setting always makes the character the centre of the screen. This might not always be desirable for all games.

Removed the game boundary idea for destroying bullets and gave each bullet a lifespan which can be set in the prefab.

Put a textured material on the ground plane otherwise you can’t see the camera tracking.  
  
The biggest challenge I found in this task for getting the rotation and movement controls working as set out in the task.  
  
Wanted to test on Unity Remote on my Iphone BUT I didn’t have the IOS dev installed for my version of Unity so skipped it – would have been fun.  
  
Finished 5:30pm 24th of Jan.  
  
**Some sources I used to help me solve these problems:**  
  
<https://unity3d.com/learn/tutorials/projects/2d-ufo-tutorial/following-player-camera>

<https://answers.unity.com/questions/1401834/how-to-get-my-player-to-face-the-direction-its-mov.html>  
  
<https://docs.unity3d.com/ScriptReference/Quaternion.LookRotation.html>  
  
<https://www.youtube.com/watch?v=pMUuwwO9B3w>

<https://www.youtube.com/watch?v=bp2PiFC9sSs>

Matthew Vecchio