**MonsterChase(UNITY)**

1. **IMPORT** all files into assets. For characters and other visual entities use SPRITE files for animations
2. **SPLIT THE SPRITE FILES**… select the files -> Inspector -> Sprite mode -> multiple -> Apply -> SPRITE EDITOR -> Slice -> automatic ->apply… Rename the split characters if you want
3. Drag character to the scene…
4. ADD **ANIMATIONS** AND SET **ANIMATOR**… Create folder Animations, player animations and Player 1 animations -> right Click-create - animator controller.. name the files ‘**Player 1 Controller’**
5. Add “Animator” Component to the player, and drag the ‘Player 1 Controller’ to Controller drop down.
6. Got to Animation tab and create a **new clip “idle**” … drag and drop standing still pose of the character
7. “Add new clip” and **create walking**.. drag and drop all walking animation poses of characters… You can set the speed of walking by 3Dots->FrameRate
8. In “Animator” tab set “make a transition” to and from Idel-Walk… that is right click on idel->**make a** **transition**-> click walking and vice versa
9. On the left side select Parameters-> + -> Bool -> type “**Walk**”.
10. Click on the arrow from idel->walk and in Condition select “walk” and set value as **true**
11. Click on the arrow from walk -> idel and in Condition select “walk” and set the value as **false**.
12. Add **Box2D** collider and **Rigidbody2D** components.
13. Drag and drop the player from hierarchy tab to ‘**Prefabs’** Folder to save the instance/Copy of the character.. keep updating it by Inspector->Overrides->Apply All
14. Create a Empty called ‘**Game BG’** and inside ‘**Background holder**’ and ‘**ground holder**’
15. Using ‘Ctrl D” to duplicate and **‘v’** to **snap** the background to previous one create large background and similarly ground.
16. For each item, set the **sorting layer / Order in layer**… It is present in Sprint Renderer of each game item. All characters will have ‘player’ sorting layer which we create and the background items will have ‘background sorting layer which we create, and to create visual hierarchy in that use order in layer… 0 is the last layer and higher number means higher layers.
17. Create a folder” **Scripts**” and add a c# script and attach it to the players.
18. To **remove the delay** in input for the player go to Animator-> select the arrow from idle-walk or vice versa-> inspector -> setting -. Time duration =0
19. To decrease the speed of the movement of animation either reduce the **frame rate** to 24 as in step 7… Or Animator -> walk -> Speed (keep it between 0-1 to reduce speed)
20. To adjust the height and speed of jump change **mass** and **gravity scale** values in RigidBody 2d in player Component.
21. Select all the ground items and give it a **tag** in inspector tab as ‘**ground’**.
22. Create **CameraFollow script** to write code so that camera follows the player
23. Give players the **tag** as “**Player**”
24. Set the min and max values to the position of last frame possible
25. Now do steps 2,4,5,7,19 For the 3 kinds of **enemies**
26. Make sure that all the players have the “**Freeze Rotation of Z axis**” turned on… else they will **flip**.. that is All player and enemies -> inspector -> RigibBody2D -> Constraints-> FreezeRotation.
27. For Ghost set BodyType as “**Kinematic**” so that **gravity** does not affect our ghost and it floats in air.
28. Now we need to **randomly spawn** our enemies
29. Create a Empty called **Spawner** and **left** and **right** inside that… set the left empty lo left extreme position and right to right extreme position
30. Create c# script called **MonsterSpawner** and attach it to Spawner item in hierarchy
31. Write the scripts and **attach** the **ghost prefabs** to **MonsterRerences** and left and right items to **LeftPost** and **RightPos**.
32. Because the monster are getting stuck due to BOXCOLLIDER we remove that and use **capsule collider** for all the monsters.
33. Create **Enemy tag** for the Monsters so that we can use it to **destroy** the **player**
34. In Capsulebody2D of **Ghost**, check **isTriggered** as true, Now because isTrigger is enabled Ghost will **pass through enemies** and all game objects.
35. to handle the **collision of other enemies**, goto edit->project settings -> physics 2D and scroll down to see a matric of layers… Add another layer called Enemy in Inspector tab. Now we can see enemy in the matrix
36. **Matrix** indicate which elements can collide with what elements. So if all are ticked every element collides with every element. Soo Uncheck Enemy/Enemy… Now no element with the layer enemy can **collide** with other enemy layer enemy.
37. Now to **collect the enemy objects** which are **out of bounds**. we create empty “**Collector Holder**” and left and right holders.. we give rigidbody 2D to left and right holder. Then we create a script to destroy the enemy objects and attach them to left and right holder (we use isTrigger type of collision detection here).
38. Now all these are saved in Assets->Scenes-> GamePlay
39. Create another **scene** by File->NewScene-> New 2D and save as **MainMenu** to build Mainmenu
40. Canvas->inspector->Canvas Scaler->UI Scale Mode -> “Scale with screen size”->resolution = **1920 by 1080.** (When we ship on mobile devices, this helps unity to scale the UI assets to fit and look took)
41. Canvas->inspector-> RenderMode -> “**screenspace-camera**” and drag and drop main camera to the render camera drop down… this will fit the canvas to the main camera.
42. Using anchor present the square thing in “Image->inspector->Rect Transform” we can **anchor** our elements to a particular location so that it wil stay there irrespective of weather we change our orientation or reslution.
43. Set the BG image and use **Stretch** in anchor pre-sets. set the mood and add the txt. Use buttons and add player images to them.
44. Create **MainMenuController** script and also an Empty game obj… Attach the script to it..
45. Button->inspector->**On Click()…** its empty b default… drag the Main Menu Controller here and from the function drop down ->Main Menu Controller -> playGame() {this is our function in script}… Hence now whenever we hit the button this function will be executed.
46. For any scene to be added in the game make sure that you Add Open Scene… ie “File->Build Settings -> Add **Open Scene**”… Make sure all **your scenes are added** here.
47. Now mostly code is written in GameManager and MainMenuController.
48. In unity generally… Awake() is first called, then OnEnable() and later on Start() is called.