

# **Top-down ARPG Camera & Character Controller Mobile**

## **(Version 1.5)**

### **Summary**

Thank you for purchasing this package. (Version 1.5)

The demo scenes provided are simple demonstration on how to use the scripts. It can help you if you are struggling with some features and not sure how to fix them. There is a setup guide included in this document for more detailed help.

There are 9 mobile prefabs you can use:

- "1.a. Mobile ARPG Player" Normal top-down camera and uses touch to move character (Like Diablo).
- "1.b.Mobile AROG Follow Cam Player" Follow camera and uses touch to move character (Like Diablo).
- "2.a.Mobile Axis top down Mouse ARPG Player" Normal top-down camera and uses left and right joystick to move.
- "2.b. Mobile Axis top down Mouse ARPG Follow Cam Player" Follow camera and uses left and right joystick to move.
- "3.Mobile Axis ARPG Player" Normal top-down camera and uses left joystick to move.
- "4.Mobile Axis Turn ARPG Follow Cam Player" Follow camera and uses left joystick to move.
- "JumpButton" a button example, in some demo scenes used for jumping.
- "LeftJoyStick" left joystick for player movements, with example image.
- "RightJoyStick" right joystick for player movements uses Unity stock joystick image.

There are 2 camera scripts (normal top-down ARPG camera and follow camera), 4 player move scripts and joystick script (C# version of Unity's joystick script).

## **Scripts**

### **The Mobile top-down ARPG camera (MobileARPGCameraController.cs)**

A Script for an ARPG (Diablo, Path of Exile, Torchlight) style camera that also allows zoom with pinch. It contains code that makes objects transparent if they are between the camera and target. The objects material needs to be a transparent shader so we can change the alpha value. Just drag the script on the main camera and set the parameters. Camera can be a child of the target.

Parameters:

- Target, the target the camera should look at and rotate around. You need a target (Character) for the camera to work.
- StartingDistance, distance the camera starts from target object.
- MaxDistance, the max distance the camera can be from target object.
- MinDistance, the min distance the camera can be from target object.
- ZoomSpeed, the speed the camera zooms in/out of the target.
- TargetHeight, the amount from the target object pivot the camera should look at.
- CamXAngle, the camera x Euler angle. This is the angle the camera looks at from the top.
- FadeObjects, enable objects of a certain layer to be faded.
- LayersToTransparent, the layers where we will allow transparency.
- Alpha, the alpha value of the material when player behind object.
- PinchZoom, enables pinch zoom.

Zoom in and out by pinching with 2 fingers.

### **Follow camera (MobileARPGFollowCameraController.cs)**

A Script for follow style camera that also allows zoom with pinch. It contains code that makes objects transparent if they are between the camera and target. The objects material needs to be a transparent shader so we can change the alpha value. Just drag the script on the main camera and set the parameters. Camera can be a child of the target.

Parameters that's different from former script:

- RotationDamping, how fast it should rotate to target angles

Zoom in and out by pinching with 2 fingers.

## Mobile ARPG move script (MobileARPGPlayerMove.cs)

This script controls the player movement. Simple ray cast gets the target location and ignore layers provided, and then the player moves to target location stopping if it hits something. Can be easily modified to work with Unity Pro Nav Mesh or path finding projects. This script requires Character Controller script, also uses an Animator but is not required.

Parameters:

- MoveSpeed, character movement speed.
- RotationSpeed, how quick the character rotates to target location.
- DistanceError, the distance where you stop the character, distance calculated between the difference of target position and character position.
- Gravity, gravity for the character.
- RayCastDistance, the ray casting distance of the mouse click.
- LayerMask, the layers the mouse click (ray cast) should ignore.
- GuiPositions, a list of rectangles the HUD or GUI is in, will stop ray cast at that rectangle.
- GuiPositionsXFromRight, a list the same size as GuiPositions. If quiPosition i (any qui position) calculates from screen width set i to true.
- GuiPositionsYFromBottom, a list the same size as GuiPositions. If quiPosition i (any qui position) calculates from screen height set i to true.

Move the player by touching the screen on the desired location.

## **Mobile 2 joystick move script (MobileAxisMousePlayerMove.cs)**

This script controls the player movement. Use the left joystick to move forward, back, strafe left and right, and right joystick to rotate the player.

This script requires Character Controller script, also uses an Animator but is not required.

Parameters:

- MoveSpeed, character movement speed.
- RotationSpeed, how quick the character rotates with mouse movement.
- Gravity, gravity for the character.
- JumpSpeed, jump height of the character.
- MyJoystickLeft, the joystick to move the player.
- MyJoystickRight, the joystick to rotate the player.
- MobileCameraController, the camera script being used.
- JumpButton, the gui texture for jumping.

Works best with follow camera, but this also works for top-down camera.

## **Mobile joystick player move script (MobileARPGJoystickPlayerMove.cs)**

This script controls the player movement. Use joystick to move in the desired direction.

Parameters:

- MoveSpeed, character movement speed.
- RotationSpeed, how quick the character rotates with mouse movement.
- Gravity, gravity for the character.
- JumpSpeed, jump height of the character.
- MyJoystick, the joystick to move the player.
- MobileCameraController, the camera script being used.
- JumpButton, the gui texture for jumping.
- MyRotationObject, the object we will be rotating when moving.

Works best with top-down camera.

## **Mobile joystick turn move player script (MobileAxisTurnPlayerMove.cs)**

This script controls the player movement. Use joystick to move and turn the player.

Parameters:

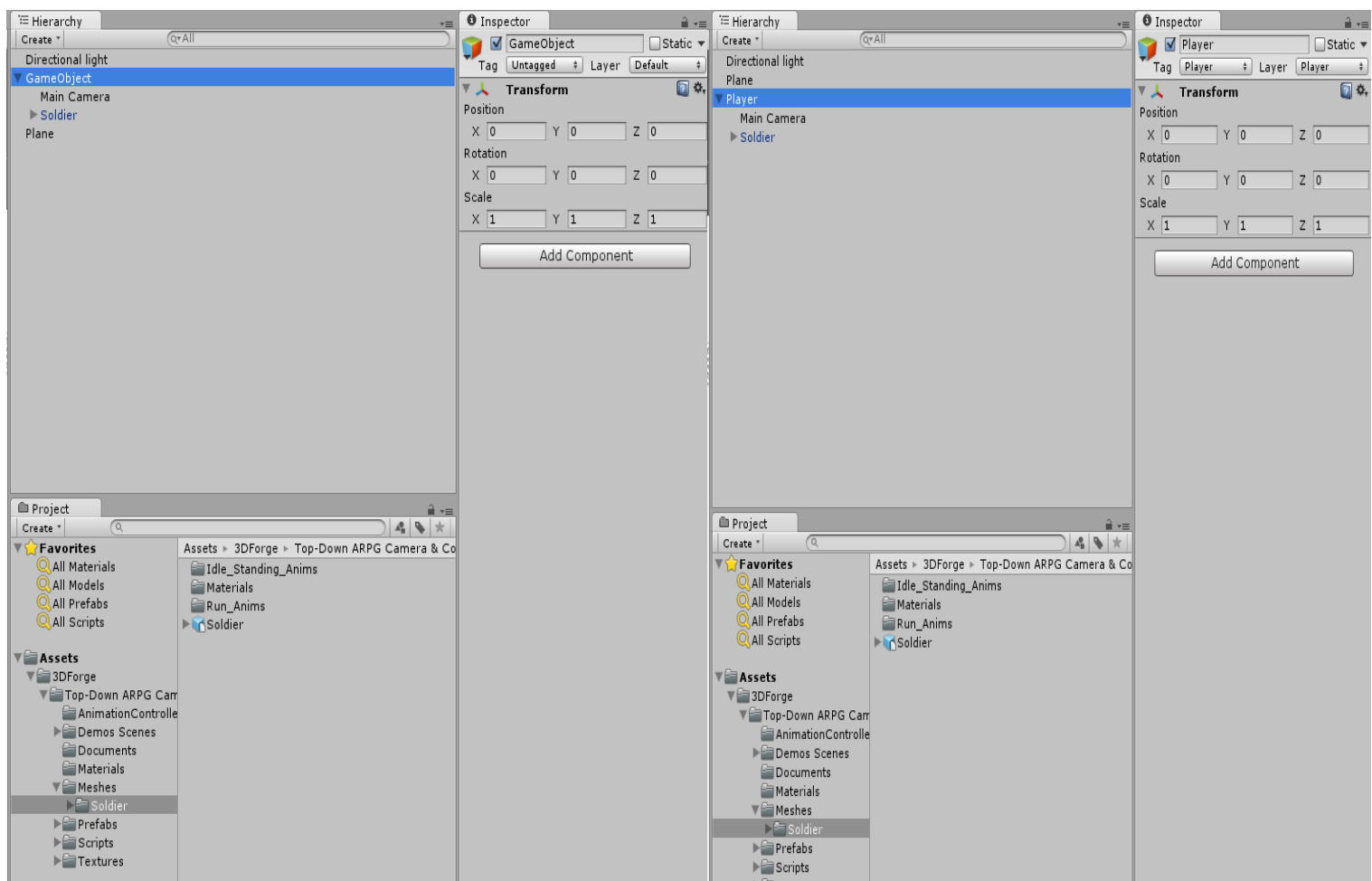
- MoveSpeed, character movement speed.
- RotationSpeed, how quick the character rotates with mouse movement.
- Gravity, gravity for the character.
- JumpSpeed, jump height of the character.
- MyJoystick, the joystick to move the player.
- MobileCameraController, the camera script being used.
- JumpButton, the gui texture for jumping.

Works best with follow camera, but can work with top-down camera.

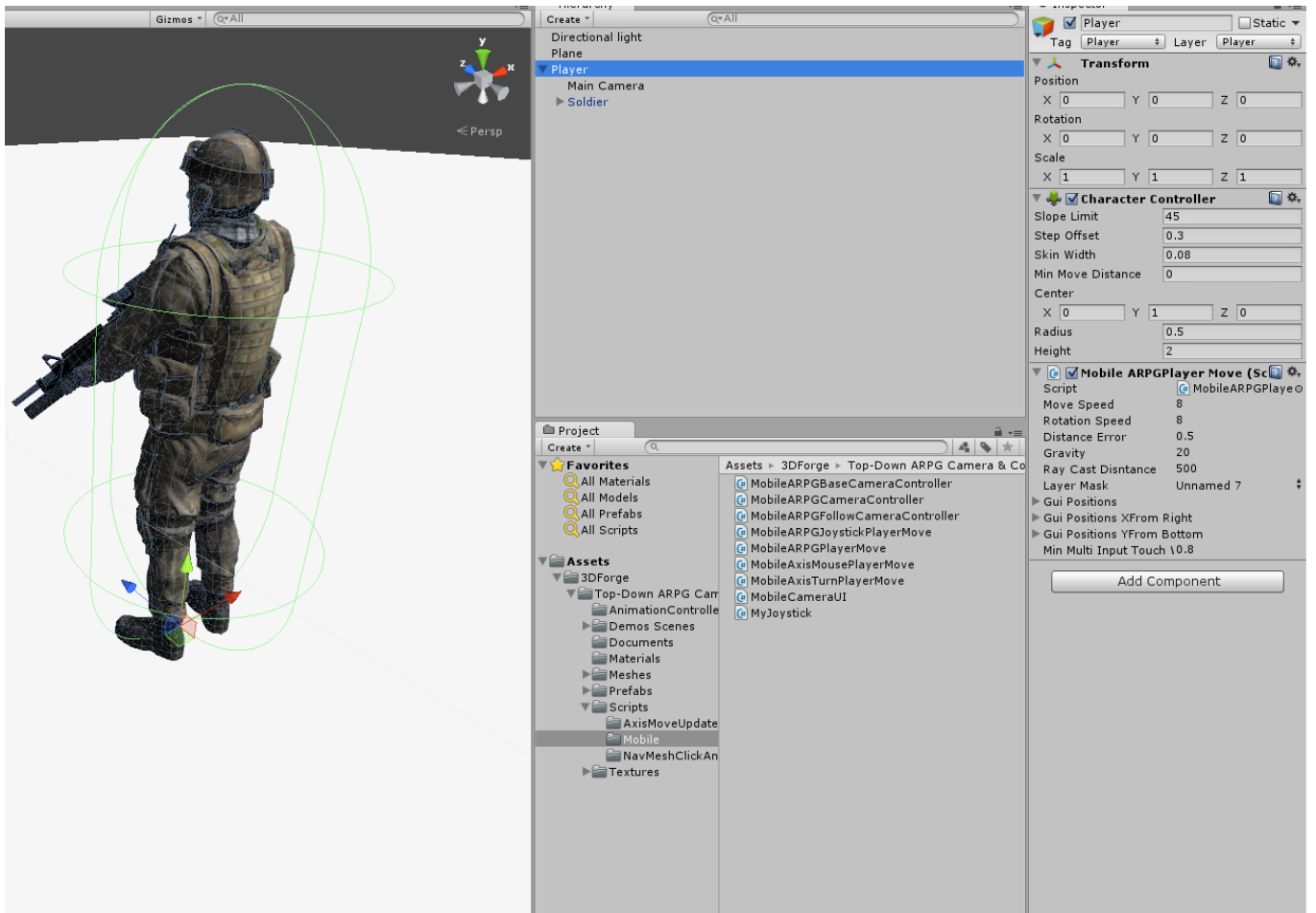
## Setup 1

Setup guide for using touch and move player, with top-down camera.

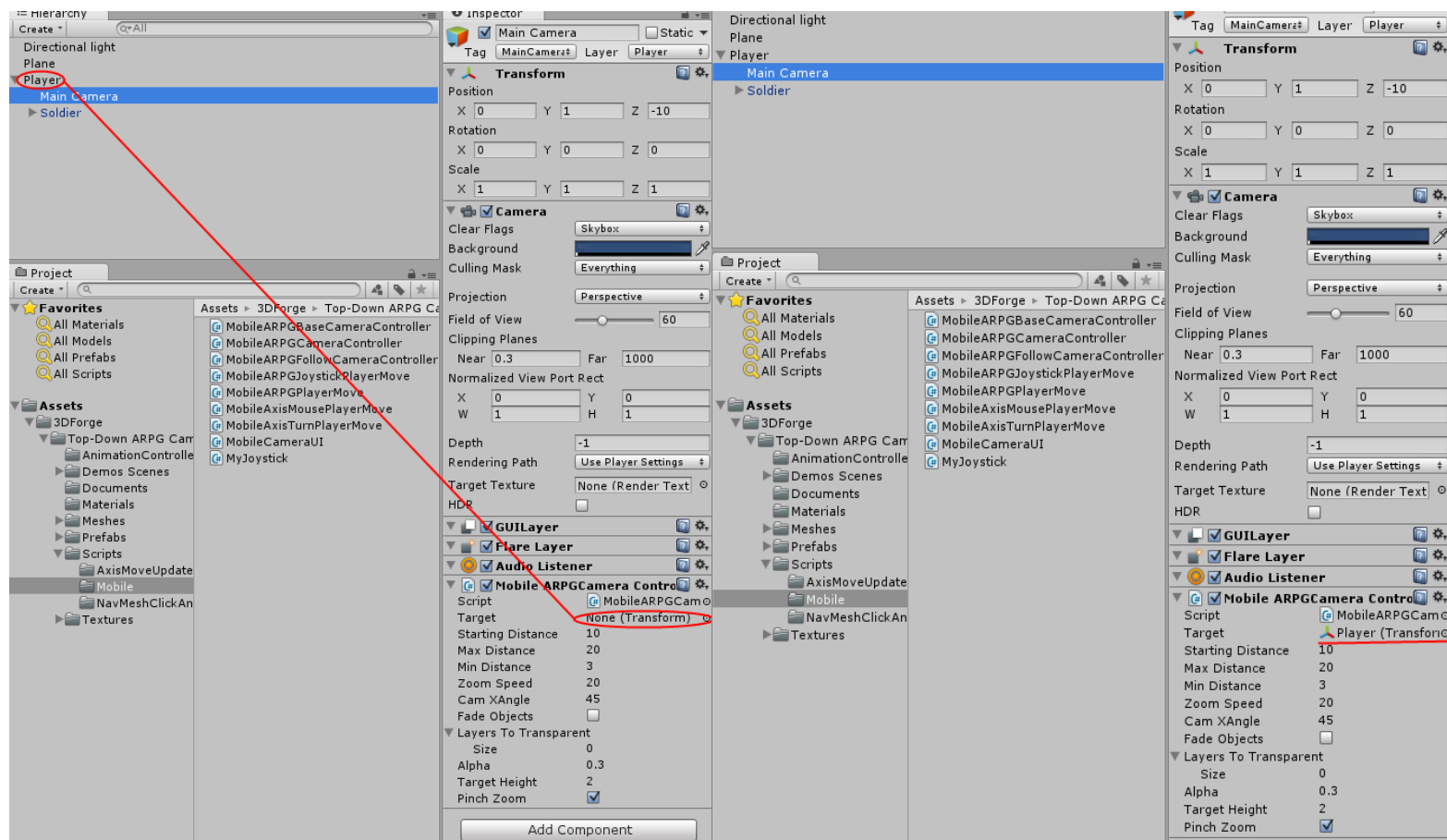
1. Create a new scene and add a plane/terrain to the scene.
2. Create empty GameObject and move its position to (0, 0, 0).
3. Add your toon/character to the scene and move its position to (0, 0, 0).
4. Add your toon/character that is in the scene as a child object of the GameObject we added.
5. Also drag the mainCamera as a child object of the GameObject we added.
6. Now we can rename the GameObject and set its tag and layer (See 11) as player.



7. Now we can add the scripts, let's start with the player move script. You can find the scripts under Assets/3DForge/ Top-Down ARPG Camera & Controller/Scripts/Mobile. We add the MobileARPGPlayerMove.cs to the Player (Parent) object. The Character Controller will be added automatically. Make sure the Character Controller is positioned correctly.

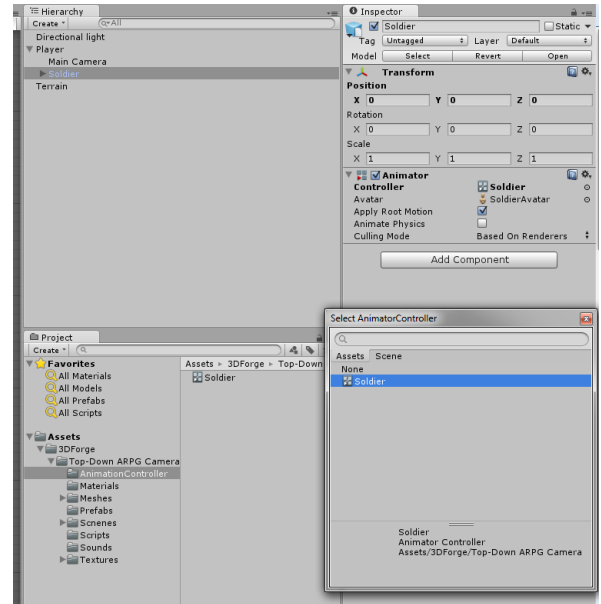
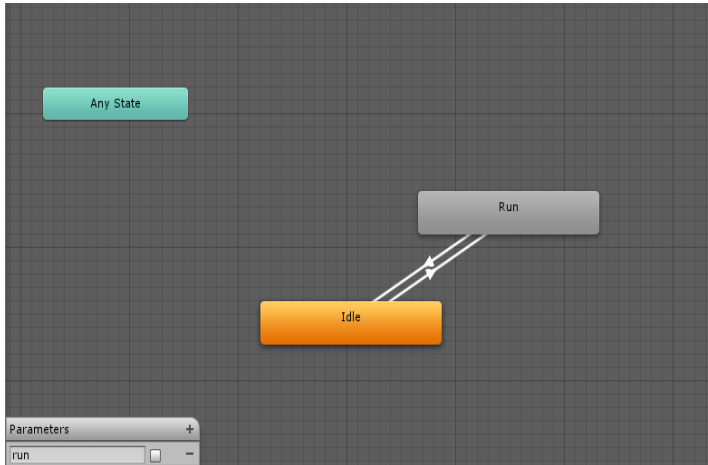


8. We then add the “MobileARPGCameraController.cs” script to the Main Camera as we are building the Normal Top-Down camera. We can then set the target variable to the player game object.

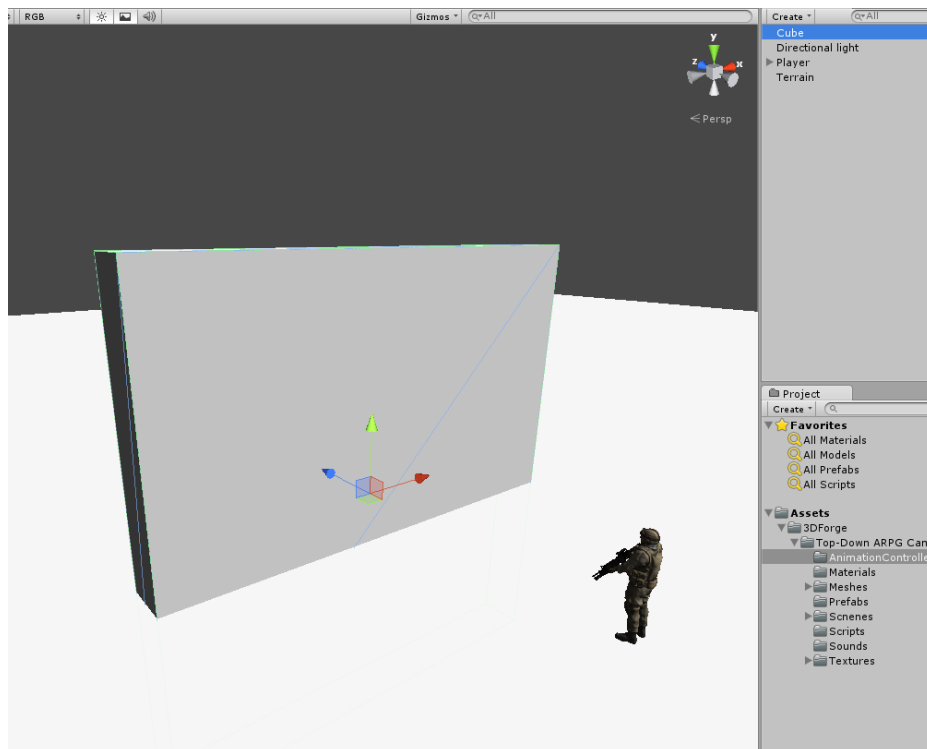




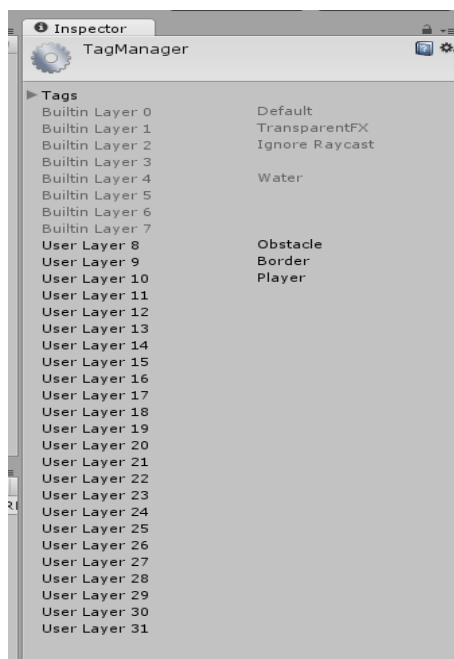
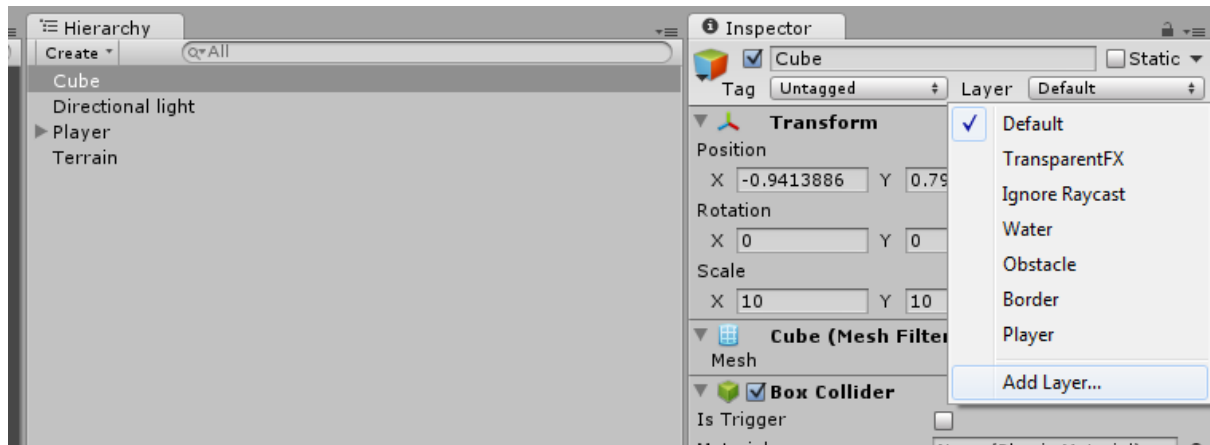
9. Player can move now but we need an AnimatorController for the character/toon to animate. The MobileARPGPlayerMove script only calls one animator variable “run” that is a Boolean. There is already one in the project under Assets/3DForge/ Top-Down ARPG Camera & Controller/AnimationController. Add AnimatorController to character/toon.



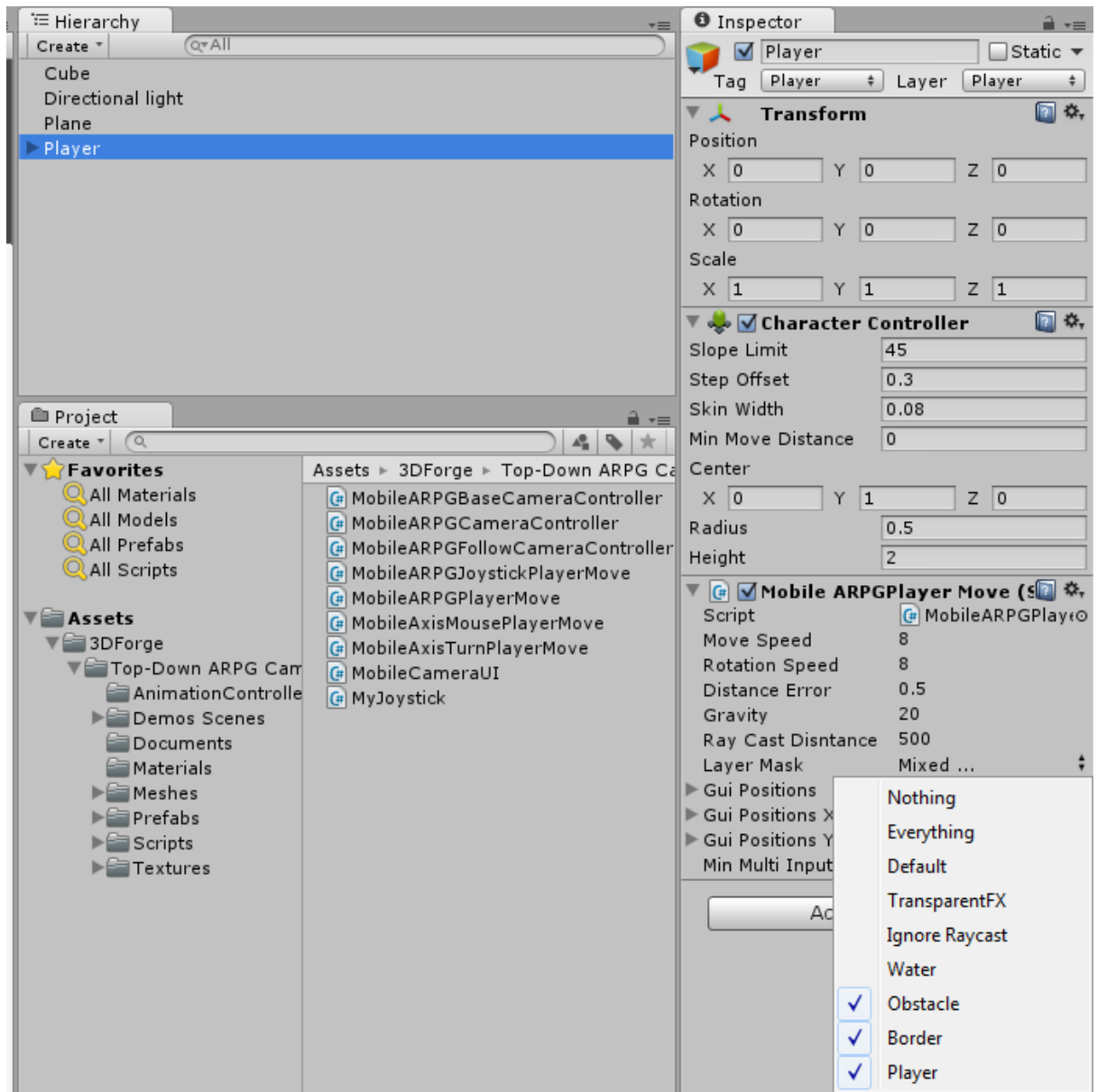
10. Now that we have the player running around looking awesome we can add some obstacles. Add a normal Cube to the scene and change its scale to (10, 10, 1).



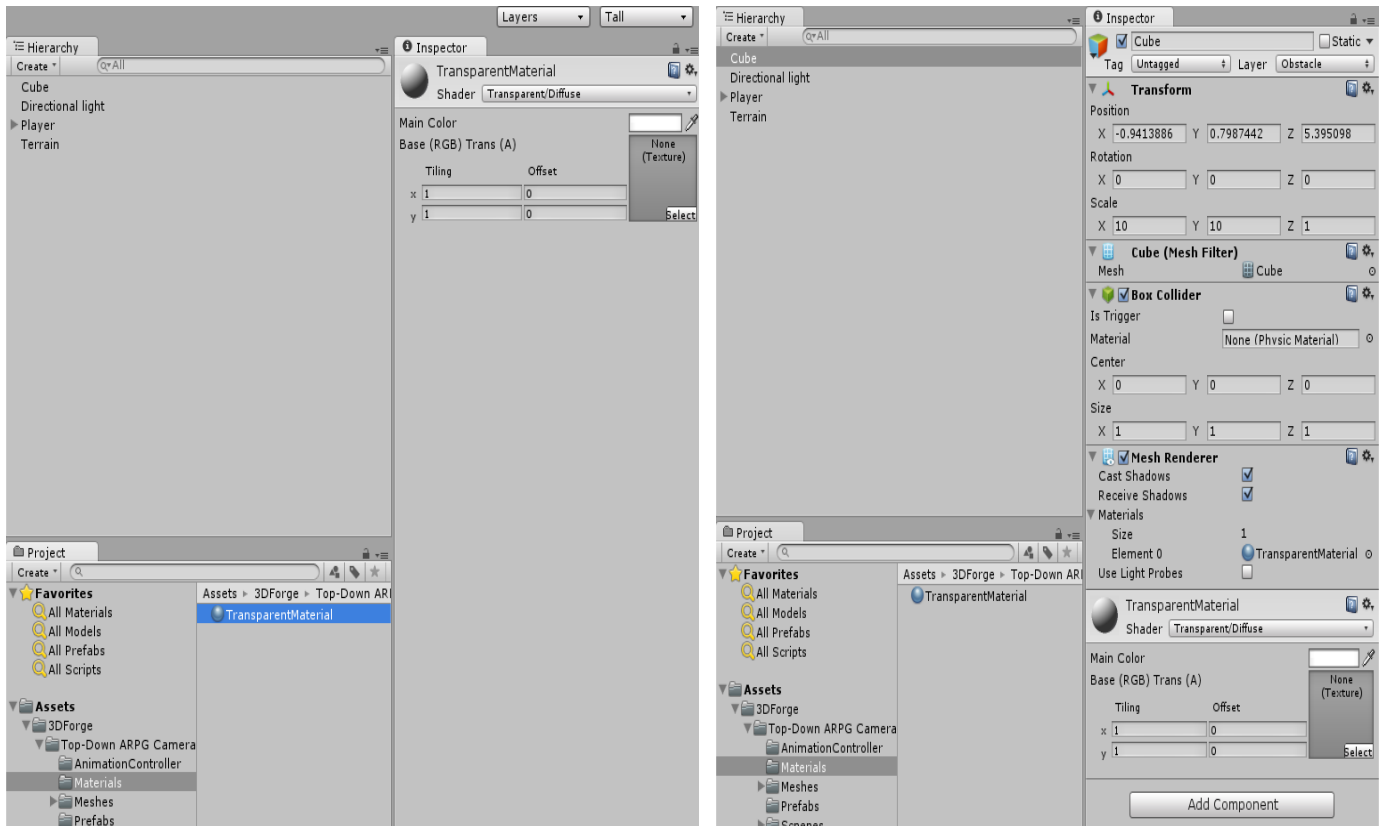
11. Add layers to project. I added Obstacle, Border and Player.



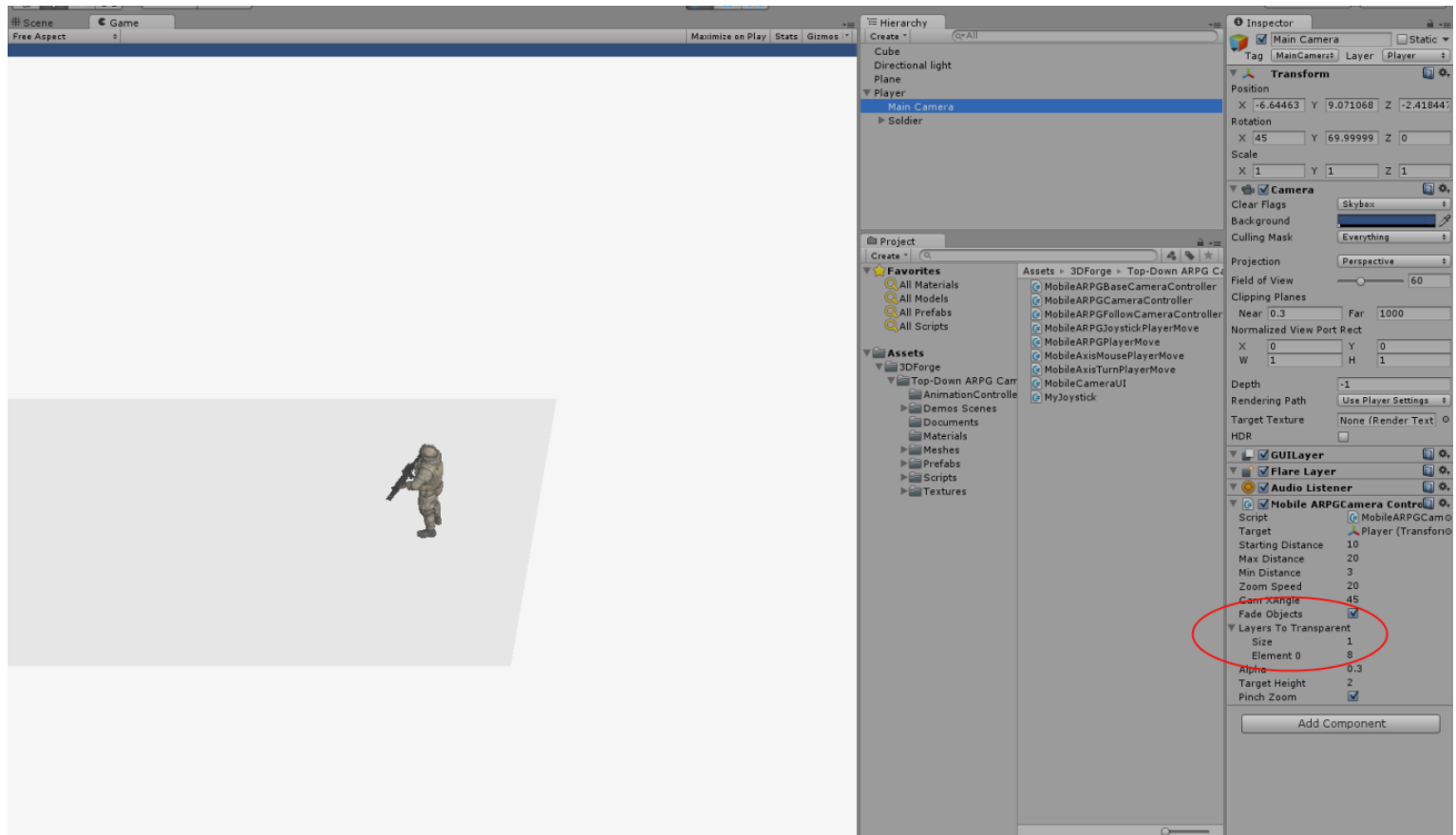
12. Now set the cube's layer to Obstacle.
13. On the player look at the MobileARPGPlayerMove script then on the LayerMask variable check the layers you wish the touch event need to ignore.



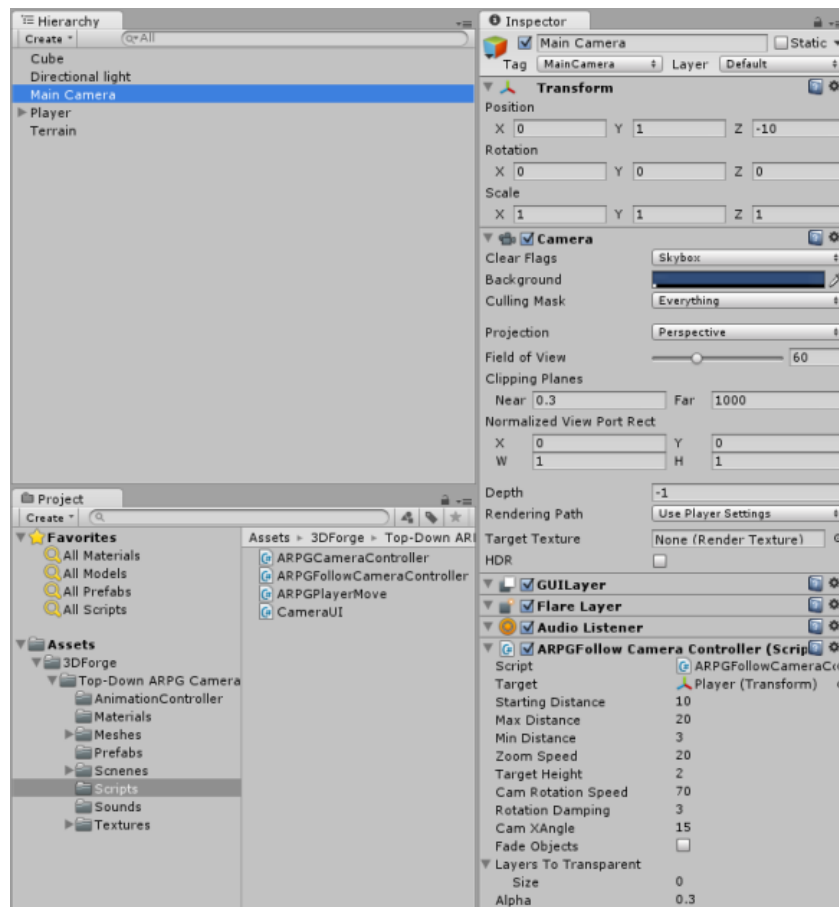
14. We can also set certain objects to go transparent using the layers we created, first we need to change the cubes materials to a transparent diffuse, create a transparent material then set the cubes material to the transparent one.



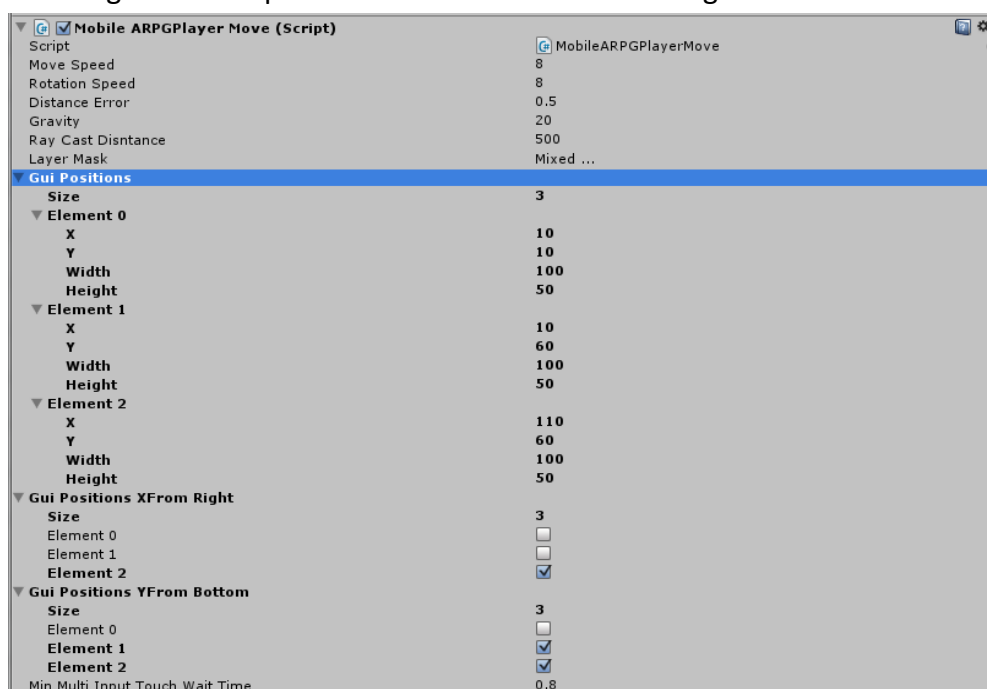
15. Now that the cube is setup we need to set the camera layers to test on, click on the Main Camera in the scene then on the script, check the Fade Objects variable. On the Layers to Transparent variable change the size to 1 and set Element 0's value to 8 (Obstacle layer).



16. This setup was for the normal top-down ARPG camera, if you want to use the follow camera you just keep the Main Camera on the root.



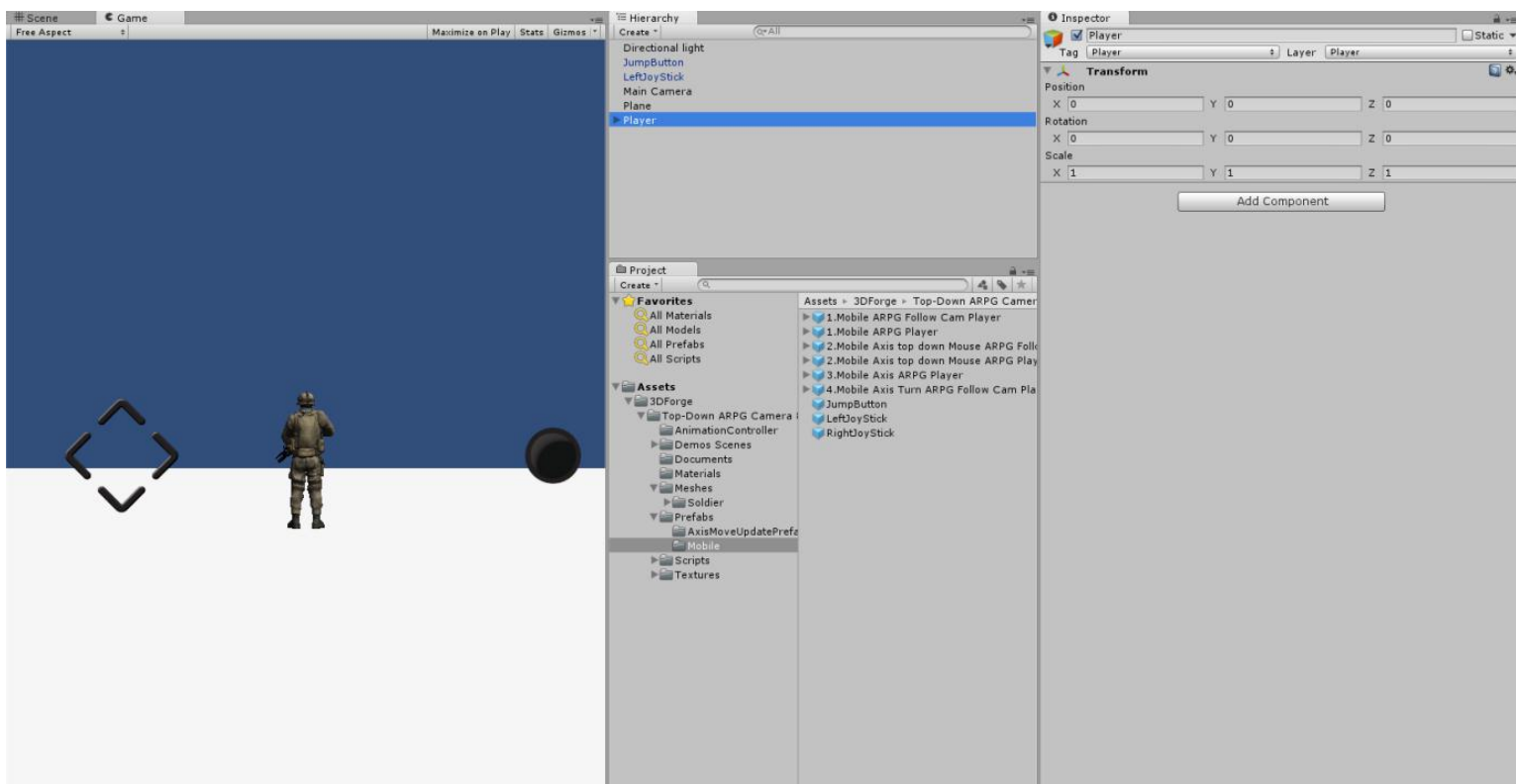
17. If you have GUI with interaction then to make sure the character doesn't move when you interact with the GUI we set their positions on the "MobileARPGPlayerMove.cs". Let's say you have 3 GUI positions A = (10, 10, 100, 50), B = (10, Screen.height-60, 100, 50) and C = (Screen.width-110, Screen.height-60, 100, 50). On the player object containing the script we set the values for parameters GUI Positions, Gui Positions X From Right and Gui positions Y From Bottom. See image!



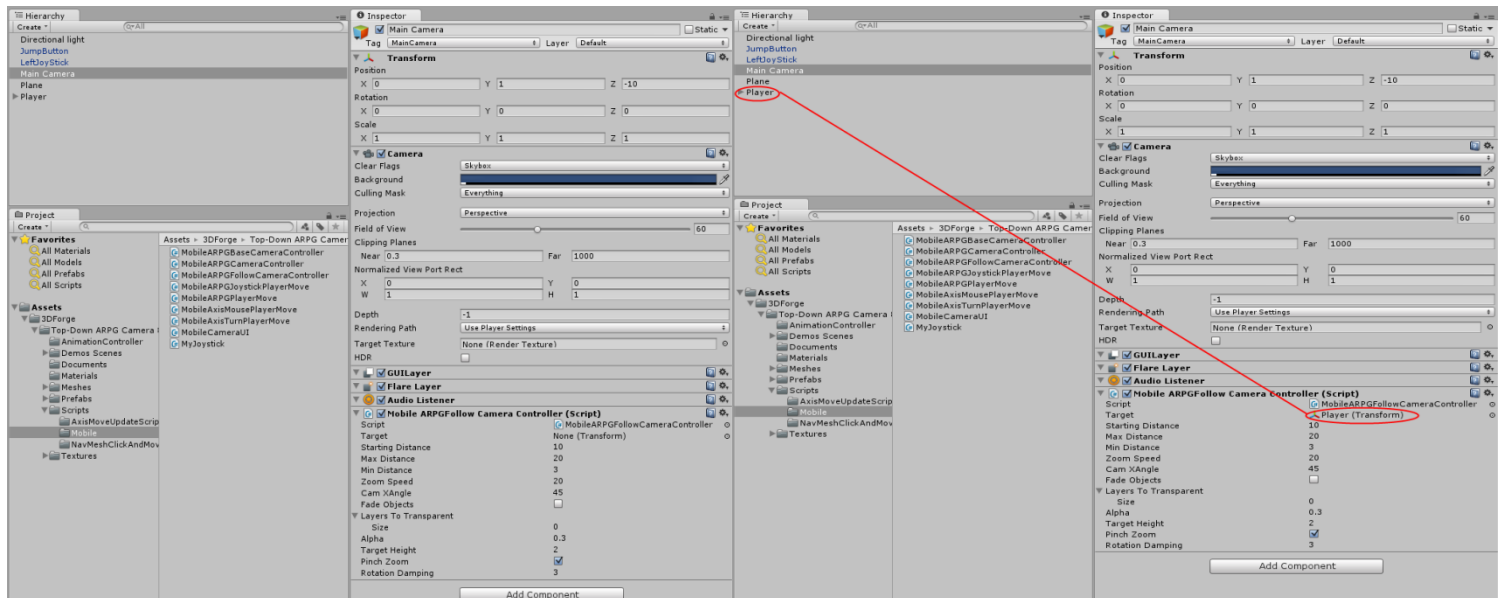
## Setup 2

Setup guide for joystick move and turn with the follow camera.

1. Create a new scene and add a plane/terrain to the scene.
2. Create empty GameObject and move its position to (0, 0, 0).
3. Add your toon/character to the scene and move its position to (0, 0, 0).
4. Add your toon/character that is in the scene as a child object of the GameObject we added.
5. Keep the mainCamera outside in root hierachy .
6. Now we can rename the GameObject and set its tag and layer (See Setup 1.11) as player.
7. Add the left joystick prefab and the jumpButton prefab (Optional don't need jump button). Prefabs can be found in Assets/3dForge/Top-Down ARPG Camera Controller/Prefabs/Mobile

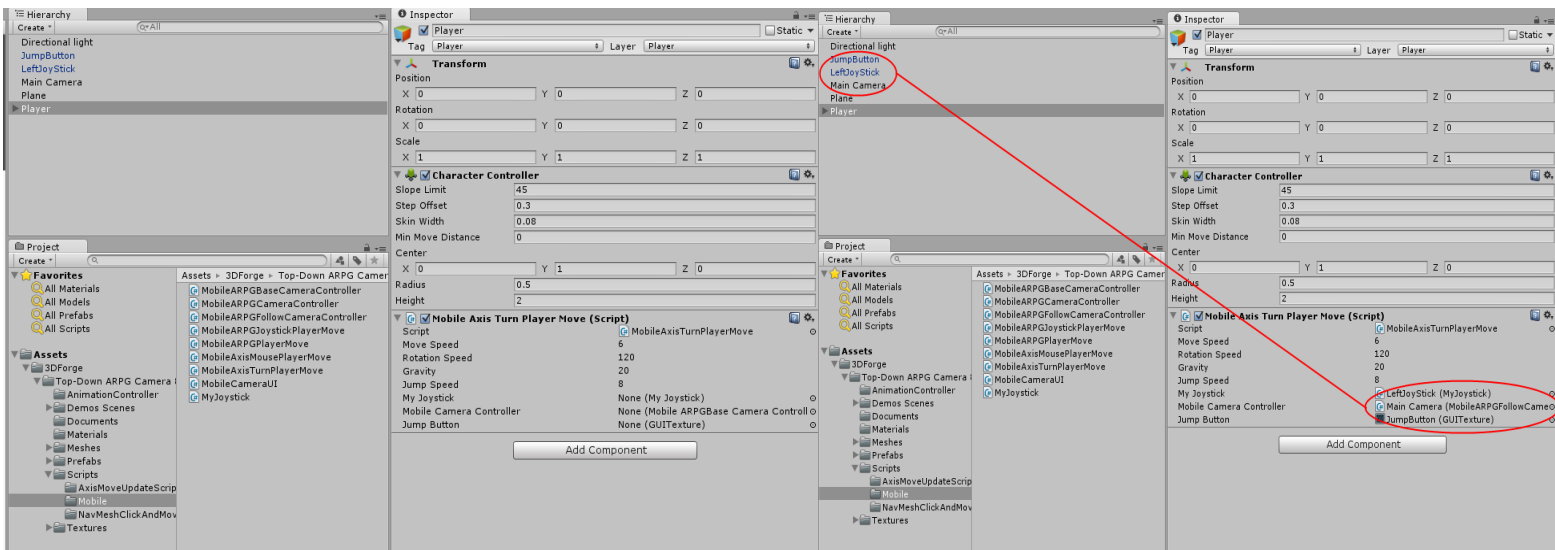


8. Now we can add the scripts, let's start with the player move script. You can find the scripts under Assets/3DForge/ Top-Down ARPG Camera & Controller/Scripts/Mobile. We add the MobileAxisTurnPlayerMove.cs to the player (Parent) object. The Character Controller will be added automatically. Make sure the Character Controller is positioned correctly.
9. We then add the "MobileARPGFollowCameraController.cs" to the Main Camera as we are building follow camera. We can then set the target variable to the player game object.





10. Now we add the mainCamera, jumpButton and LeftJoystick to the player move script.



11. See Setup 1 9-16 for the rest.

The rest of the controllers are similar to Setup 2.