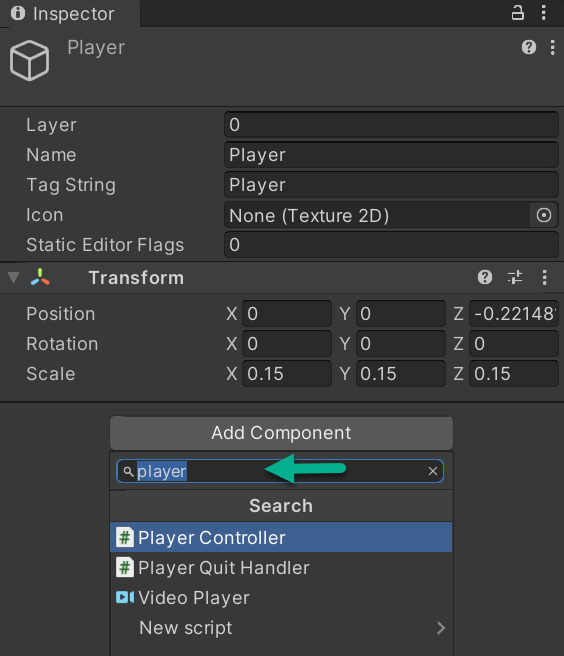
# Milestone 1 – Deliverable 3 Detailed Tutorial

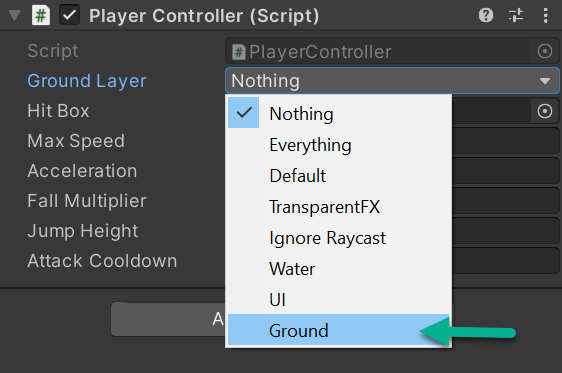
Note\* if you see an error with Assets\Scripts\PlayerController.cs, keep going forward. The code is not yet complete.

## Assembling the Player

In Unity, select the base Player object in Hierarchy. In Inspector > Add Component > search for the PlayerController script and add.



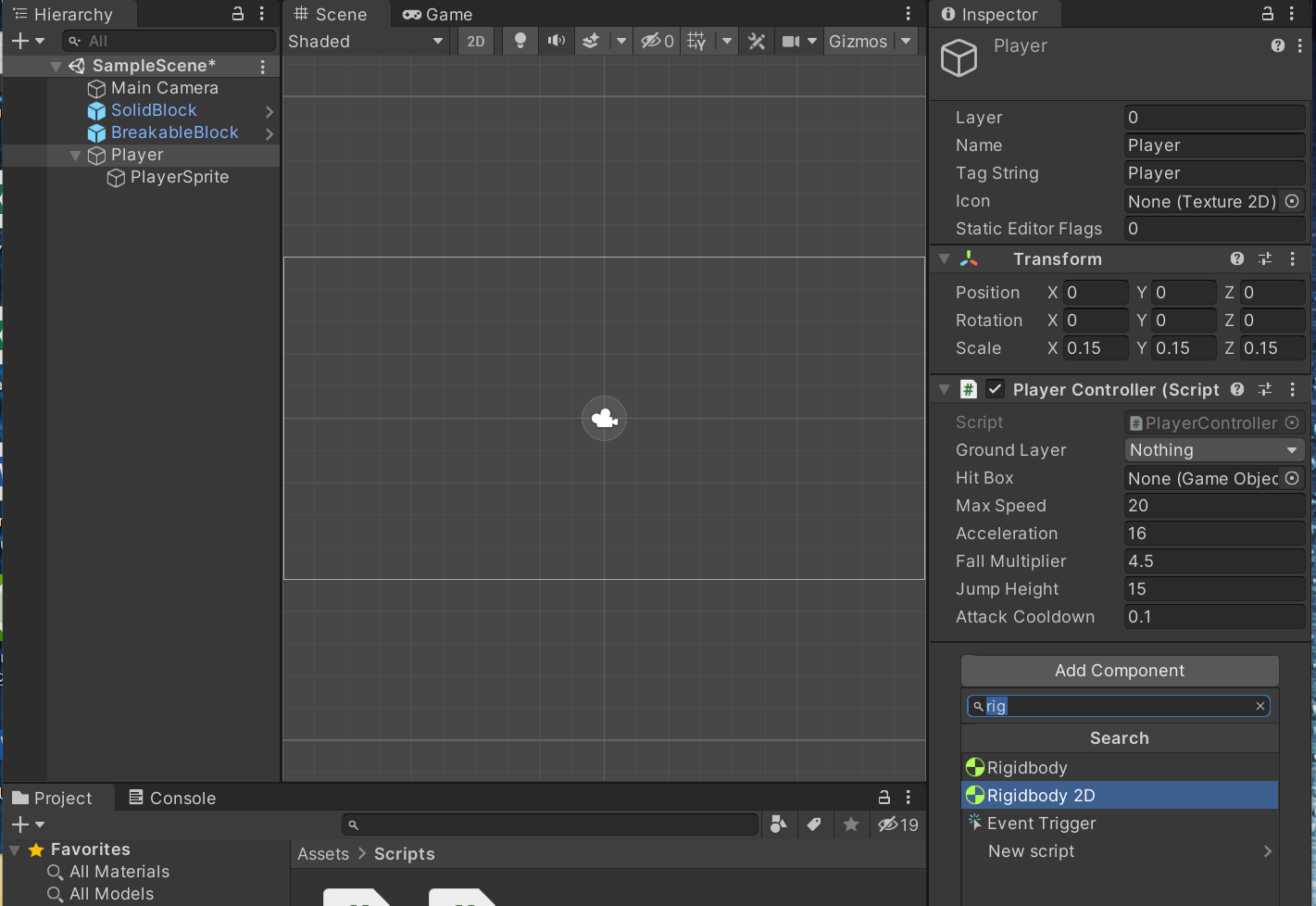
Next, set the value of the groundLayer variable. Click on Nothing and select Ground from the dropdown menu. Save your Project.



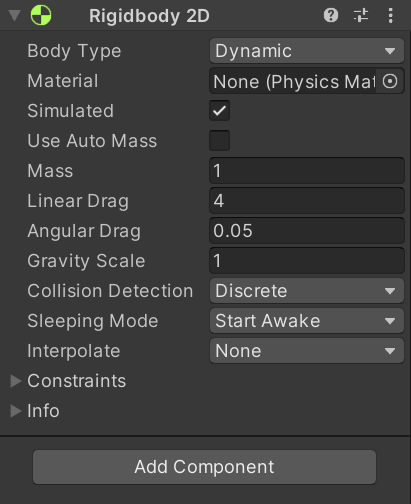
Since we assign values when declaring variables, the first time you add the script to Inspector you will see that all variables that were assigned values in your script (created in Visual Studio) will have populated their fields automatically.

However, after adding the script code, you can't change the public variables values inside the script file. If you want to modify the values so that they affect the player, you need to modify values in the Inspector window.

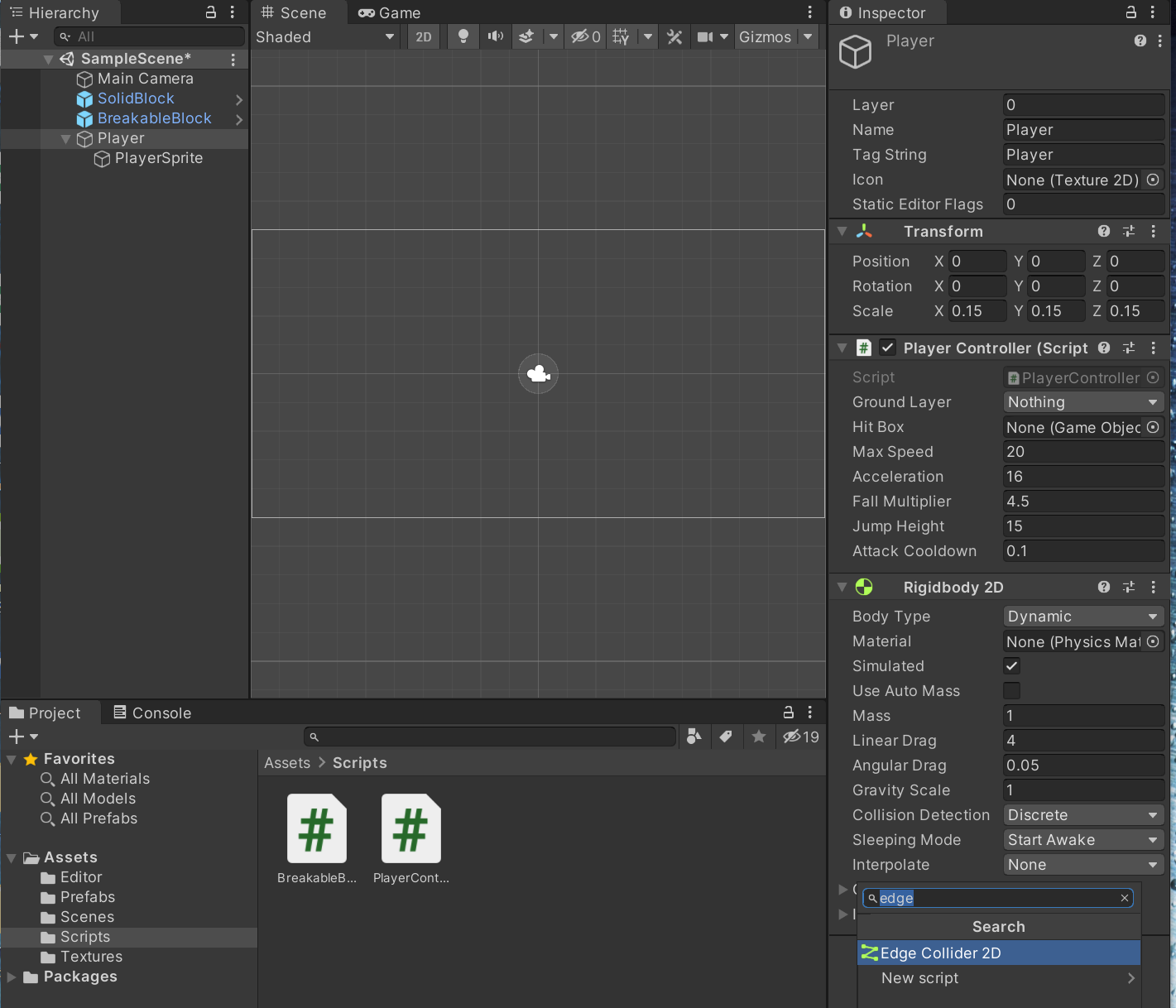
With Player selected in Hierarchy, go to Inspector > Add Component, and add a Rigidbody2D component to the base Player object. Save your Project.



If you want your player to have more friction when they move, change the value of Linear Drag on the Rigidbody2D to 4. You can play around with this value to get movement you like.



Next, add an EdgeCollider2D component. It defines the Physical Shape of the Player object so that the Physics Engine can process it.

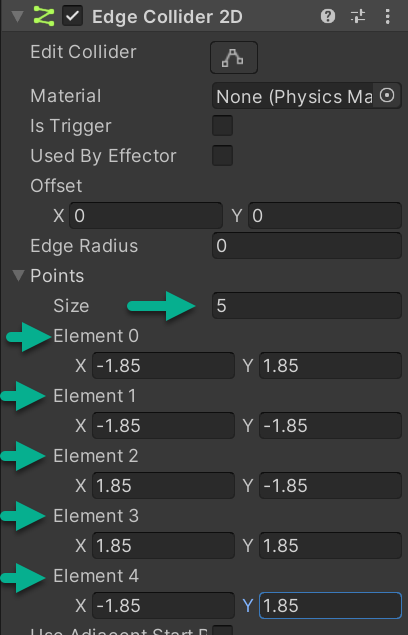


We use EdgeCollider2D instead of a box collider because it has a more adjustable shape and will cause fewer issues when walking on other colliders.

To create the collider, we need to add coordinates.

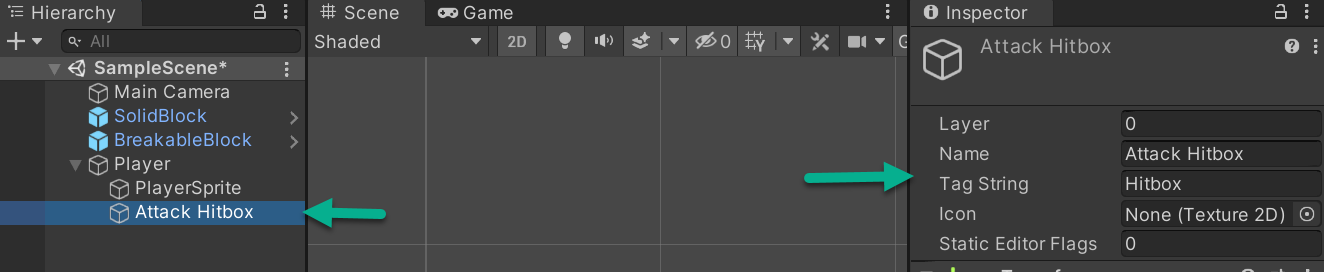
In the Edge Collider 2D dropdown Points field, change the Size field to 5. Then populate the Elements with:

* Element 0. X -1.85, Y 1.85
* Element 1. X -1.85, Y -1.85
* Element 2. X 1.85, Y -1.85
* Element 3. X 1.85, Y 1.85
* Element 4. X -1.85, Y 1.85

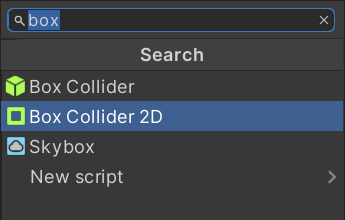


Save your Project.

To add a hitBox/Weapon for the player in Hierarchy, right click on your Player object and create a 2D Object > Sprites > Square. Rename this newly created object Attack Hitbox (includes a space between Attack and Hitbox) and change the Tag string to Hitbox (no space).



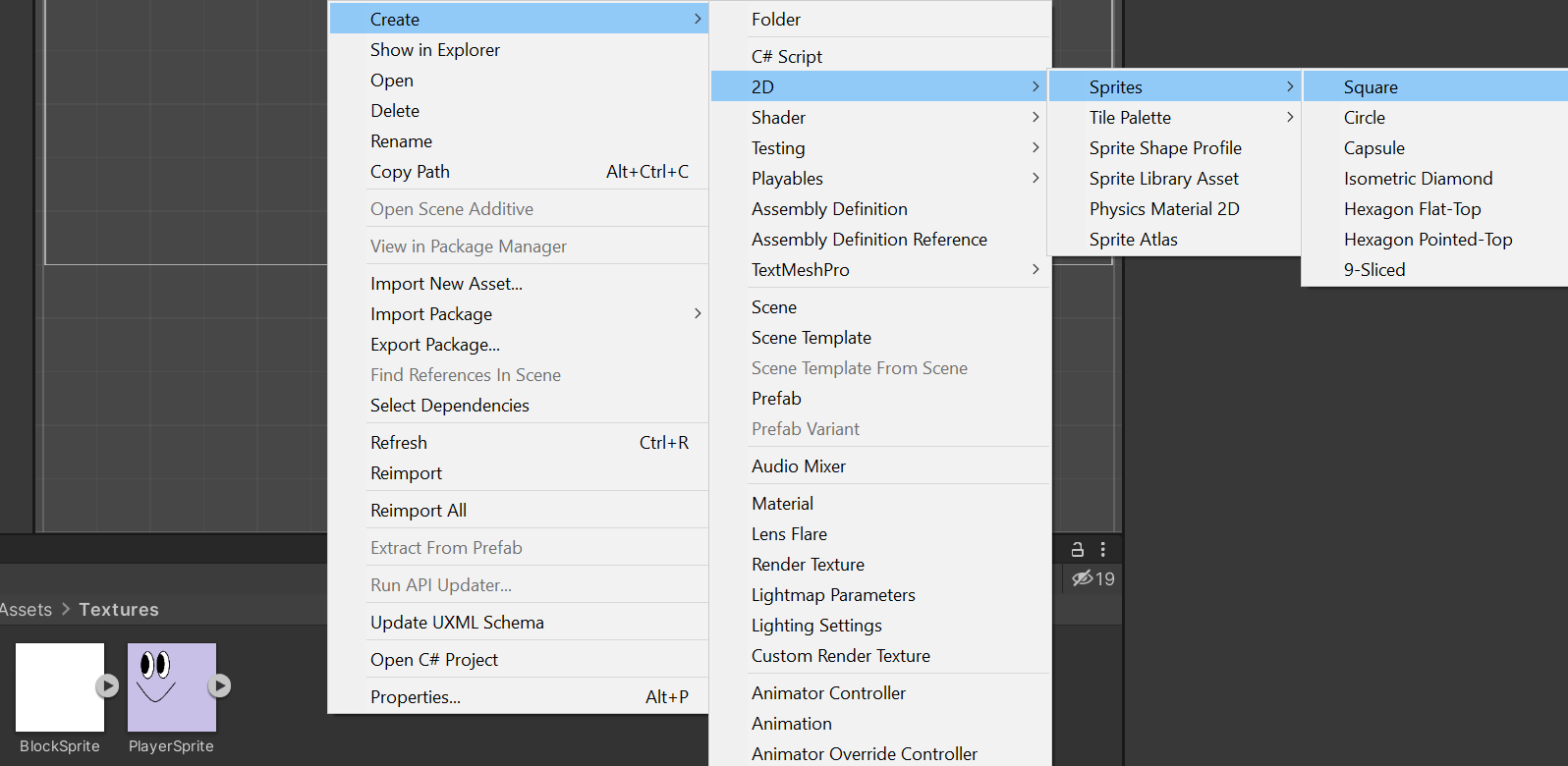
Select Attack Hitbox in Hierarchy and add a BoxCollider2D component. Make sure to check the Is Trigger toggle in Inspector.

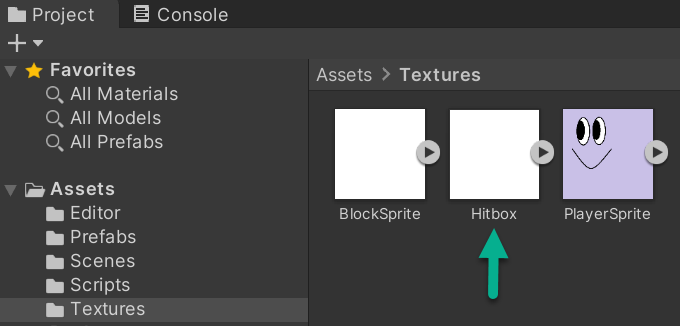




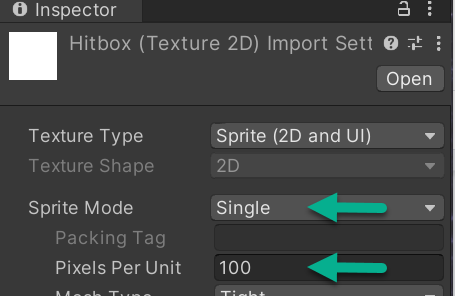
Save your Project.

Inside your Textures folder, create a new sprite (right click and create > 2D > Sprites > Square) and name it Hitbox





Select Hitbox, and in Inspector, change Hitbox Sprite Mode from Multiple to Single and Pixels Per Unit from 256 to 100. Select Apply in the popup box.

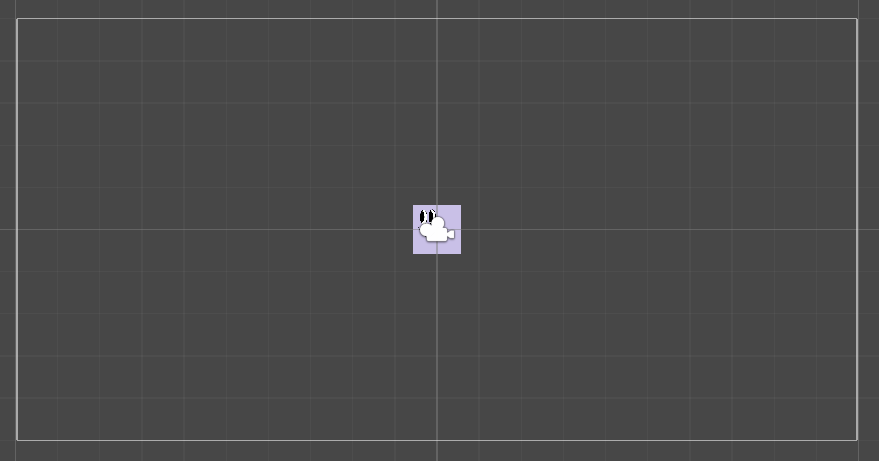


Select Attack Hitbox (child object of Player). With Assets > Textures open, drag and drop your Hitbox sprite to Attack Hitbox Inspector > Sprite Renderer > Sprite > Square.

In the Inspector > Sprite Renderer component, below the sprite field, change the color of the Hitbox object to red. The example shows hexadecimal color FF0000

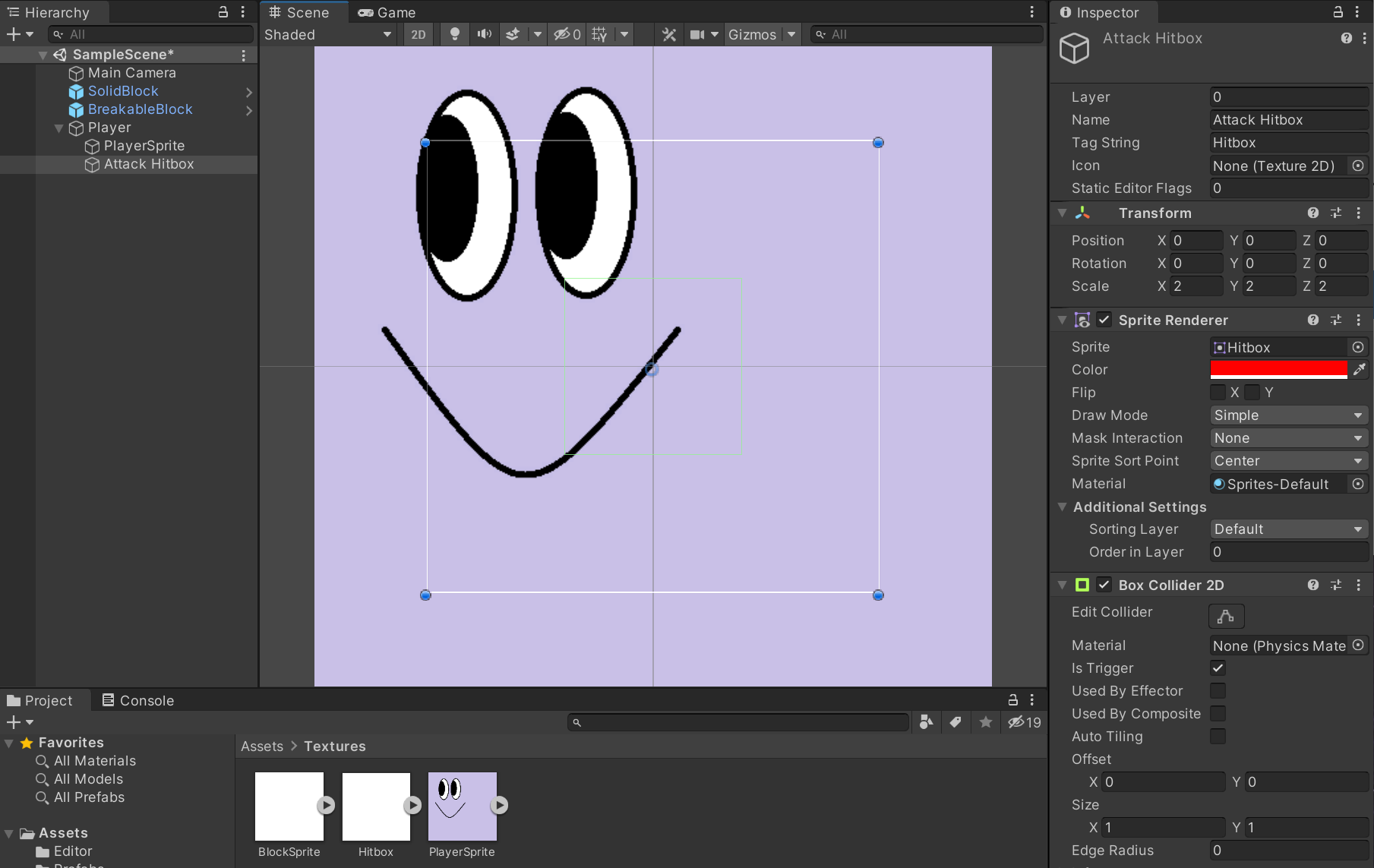
In Inspector > Transform > change Scale to X 2, Y 2, and Z 2

**Note:** Your Scene images may look different than the image below. When sprites are created and placed at X 0, Y 0, and Z 0 values, they appear stacked. At this point in the tutorial, you are setting up prefabs. It’s fine if sizing and location are not identical from project to project.



You may or may not have the camera icon showing, depending on what's selected on Scene. It’s called a gizmo (more on that later).

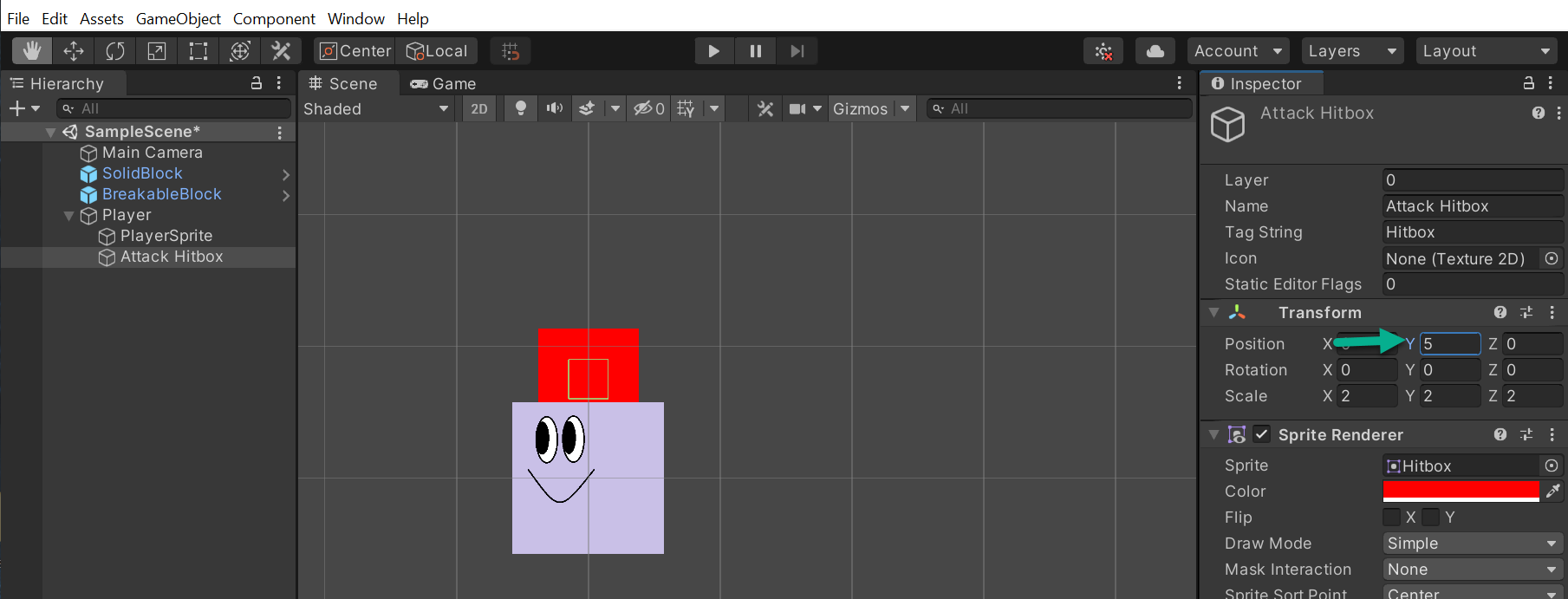
Pressing your F key after selecting a game object in the hierarchy or scene window will focus that object on the center of the scene view. You can use your mouse scroll to zoom in and out too.



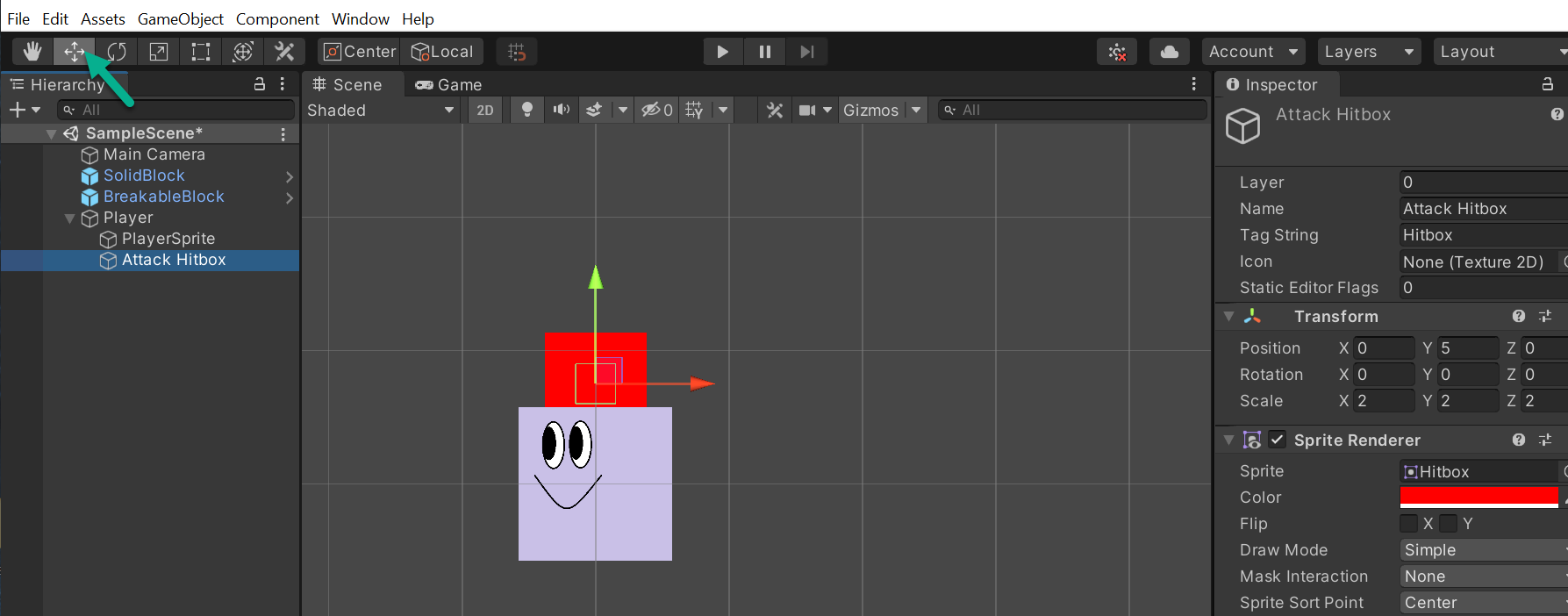
Note: Your mouse placement matters here. Select objects in Hierarchy for the scene window object to focus there. In the image above, Attack Hitbox is selected in Player and you can see the active object is Attack Hitbox in Inspector.

Three ways to position objects in Scene are detailed below. At this point, in Scene are BlockSprite, Hitbox, and PlayerSprite. BlockSprite is white, Hitbox is red, and PlayerSprite is lavender. To undo positioning, in the top menu, select Edit > Undo or Ctrl+Z, or any of the three ways below to adjust or correct your position change.

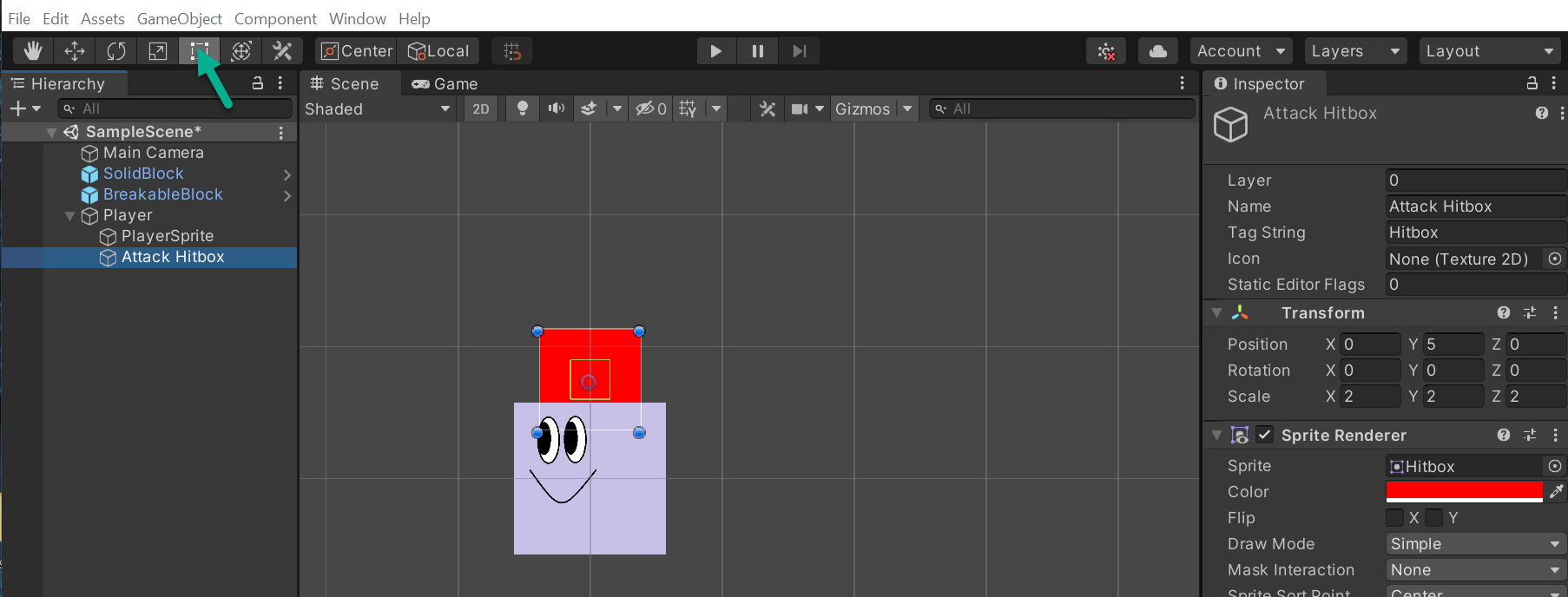
1. The Transform component of the selected object. Open the Player object and select the Attack Hitbox child object in the Hierarchy. In the Inspector tab see Transform component where you can change the x, y, and z position of the selected object. For example, change Y to 5and you’ll see Hitbox change position. After, change Y back to 0.



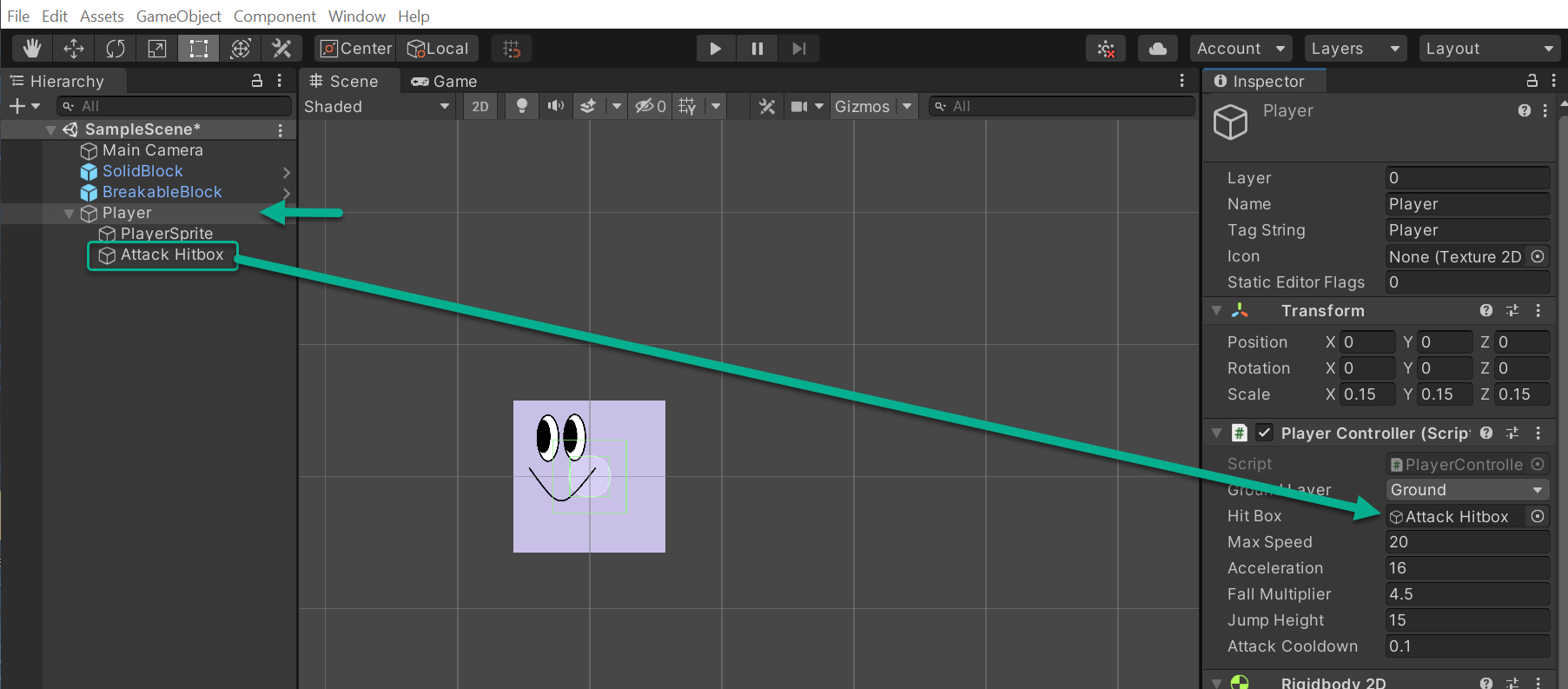
1. Position with the Move tool. Select Move as shown in the image below then select and drag Attack Hitbox in the scene. The Green Arrow inside the highlighted box moves the object vertically. The Red Arrow moves the object horizontally. For example, drag the green arrow up, and Attack Hitbox moves up in Scene.



3. To position or change scale with the Rect Tool, select the Rect icon in the top menu and drag the blue corners to scale and select the Attack Hitbox to move to position.

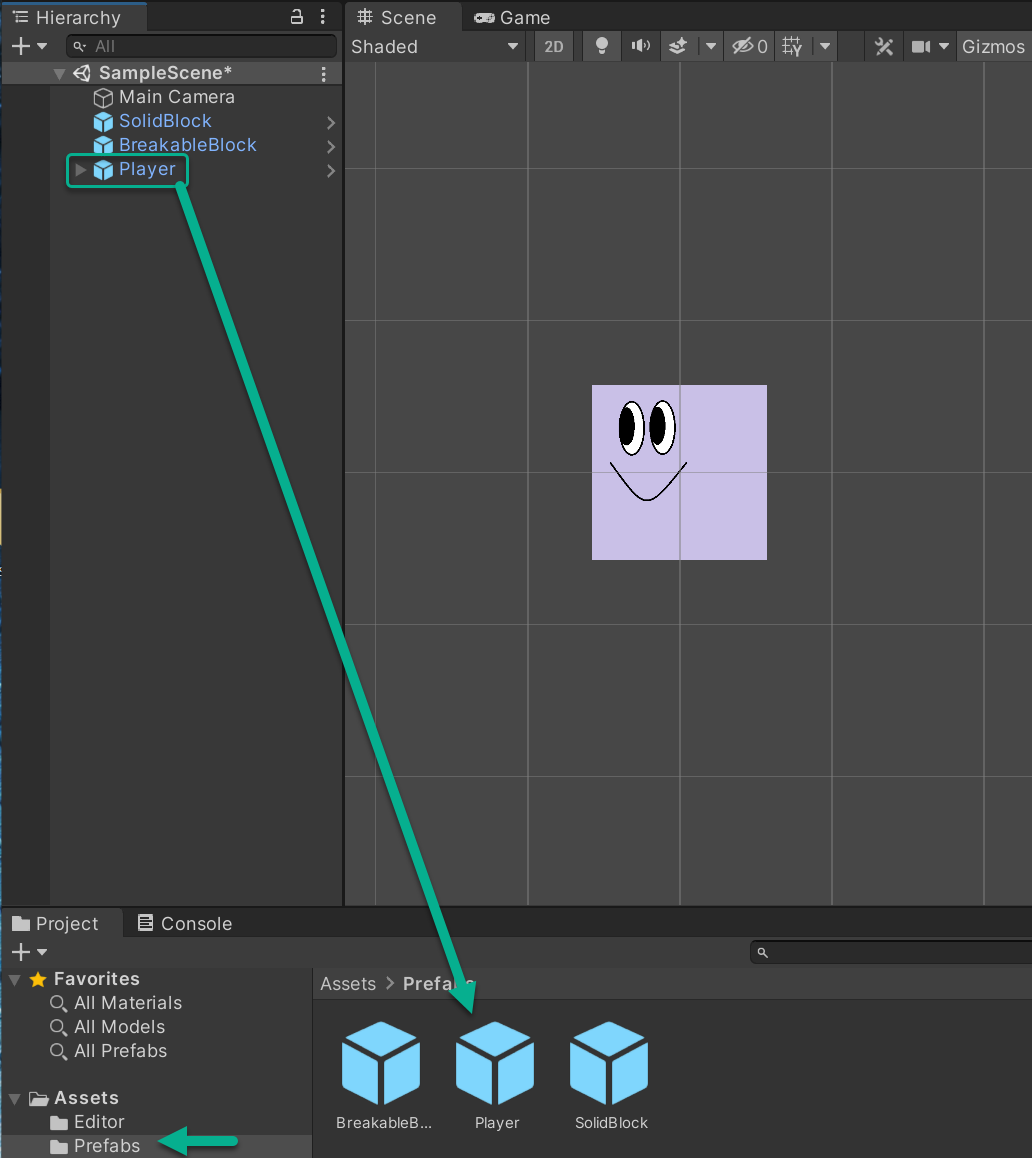


Next, set the reference of the hitBox object in your Player Controller. Select the Player parent object. (In Inspector, make sure the Player Controller section is open and you can view the fields.) With the Player parent object still selected, drag the Attack Hitbox child object from Hierarchy to the Inspector Hit Box field where it says, None (Game Object). After placing it in the field, Hit Box changes to Attack Hitbox. Save your Project.

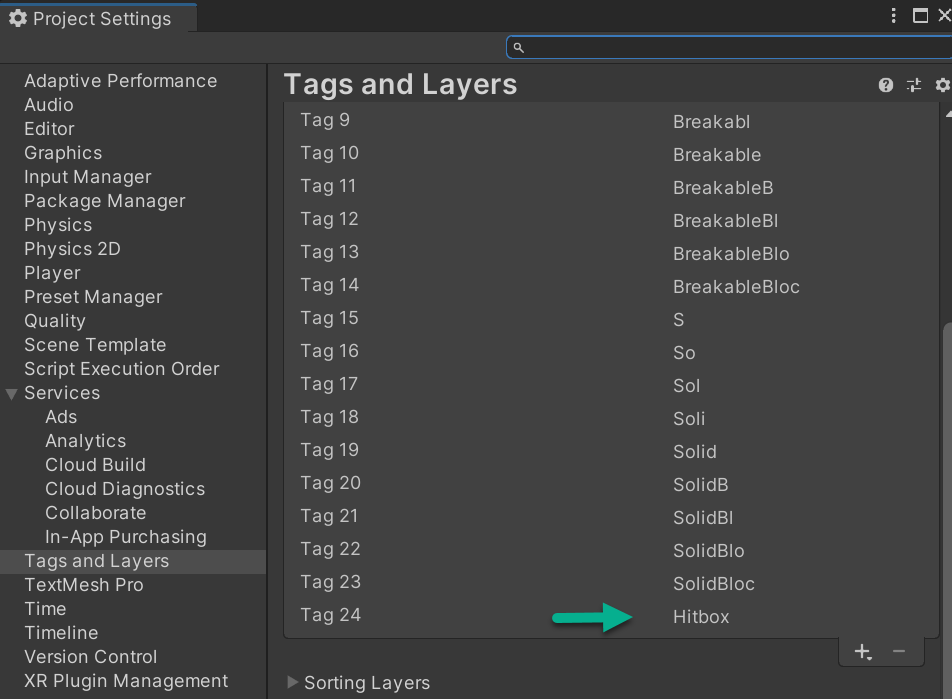




Select Prefabs in the Project Tab. Select Player, in Hierarchy, and drag it into Prefabs to create a Player Prefab. Save your Project.

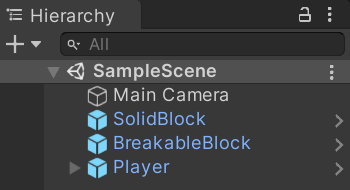


Go to Edit > Project Settings and select Tags and Layers. Open the Tags menu. Scroll down to the + to add a new tag named Hitbox and Save. Close the Tags and Layers window and save your Project.



## Designing the Level

Prefabs are discussed in Milestone 1 Deliverable 1 in Section 3 titled Create Solid Block and Breakable Block Prefabs. Before you go to your new Scene, make sure you made prefabs out of the game objects you created. In Hierarchy you see objects SolidBlock, BreakableBlock, and Player. They match up with Prefabs you have in Assets, BreakableBlock, Player, and SolidBlock.

Make sure to save your Project. Loading a new scene will delete any changes if you haven't saved your current scene. You'll also add the prefabs created in this scene to your new scene.

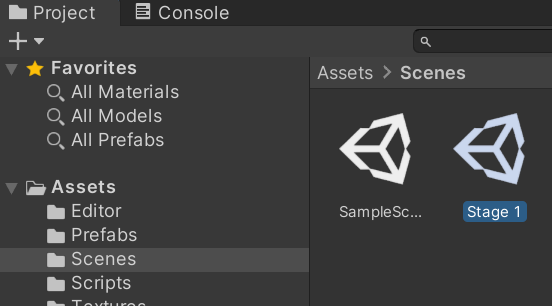
With all the parts of the game now created, it's time to work on the first stage of design.

In Project > Assets, check to see if you have a folder named Scenes. If yes, you will use that folder. If not, create a new folder in Assets called Scenes. You can create and name a folder from the top-level Assets menu or right click on Project > Assets.

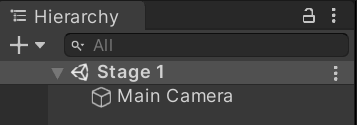
Whether Scenes is created for you or if you create it depends on your Unity version.

Double click to open Scenes. Right click to Create > Scene, and name it Stage 1

You may or may not see another file, SampleScene.unity. You’re fine either way, keep moving through the tutorial.



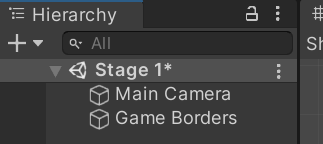
Save your Project and double click on Stage 1. In Hierarchy, you’ll see Stage 1 with default Main Camera and no additional game objects.



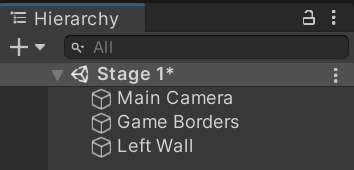
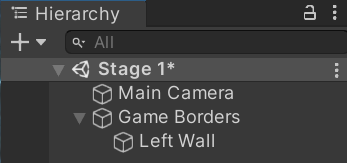
3a. Creating the Border

The first item you’ll create on Stage 1 is a border made of solid blocks. The purpose of the border is to create an enclosed space so your player can't jump out of bounds of the game. You’ll start with one wall and add three more. This is a fun and time-consuming process.

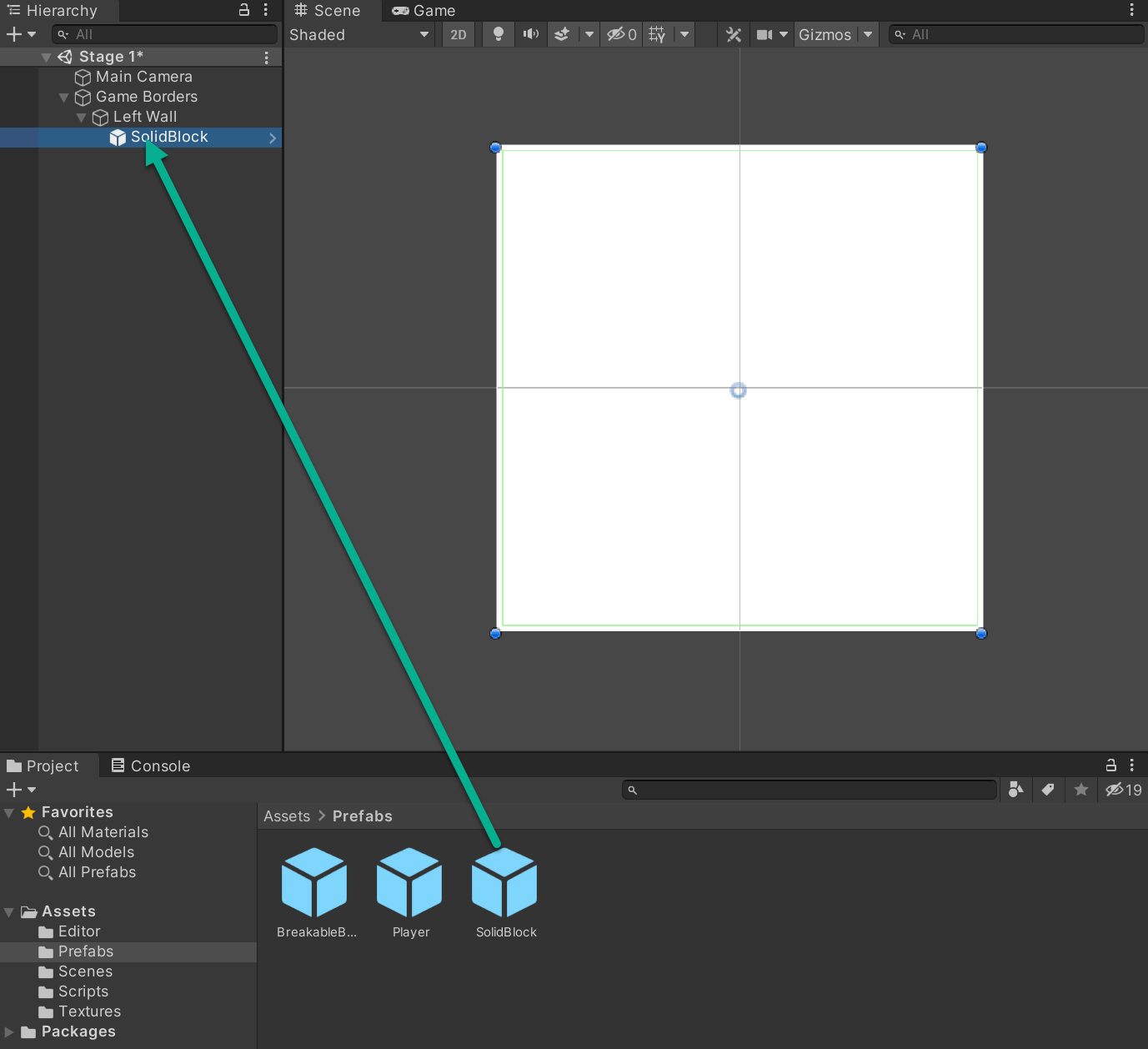
In Hierarchy, right click and Create Empty > GameObject. Rename GameObject Game Borders (yes, there’s a space between Game and Borders).

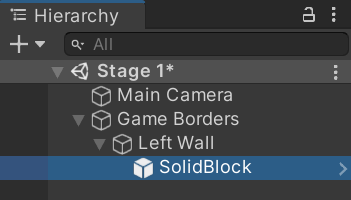
 

Right click to Create Empty and name the GameObject Left Wall then drag Left Wall to be a child of Game Borders.

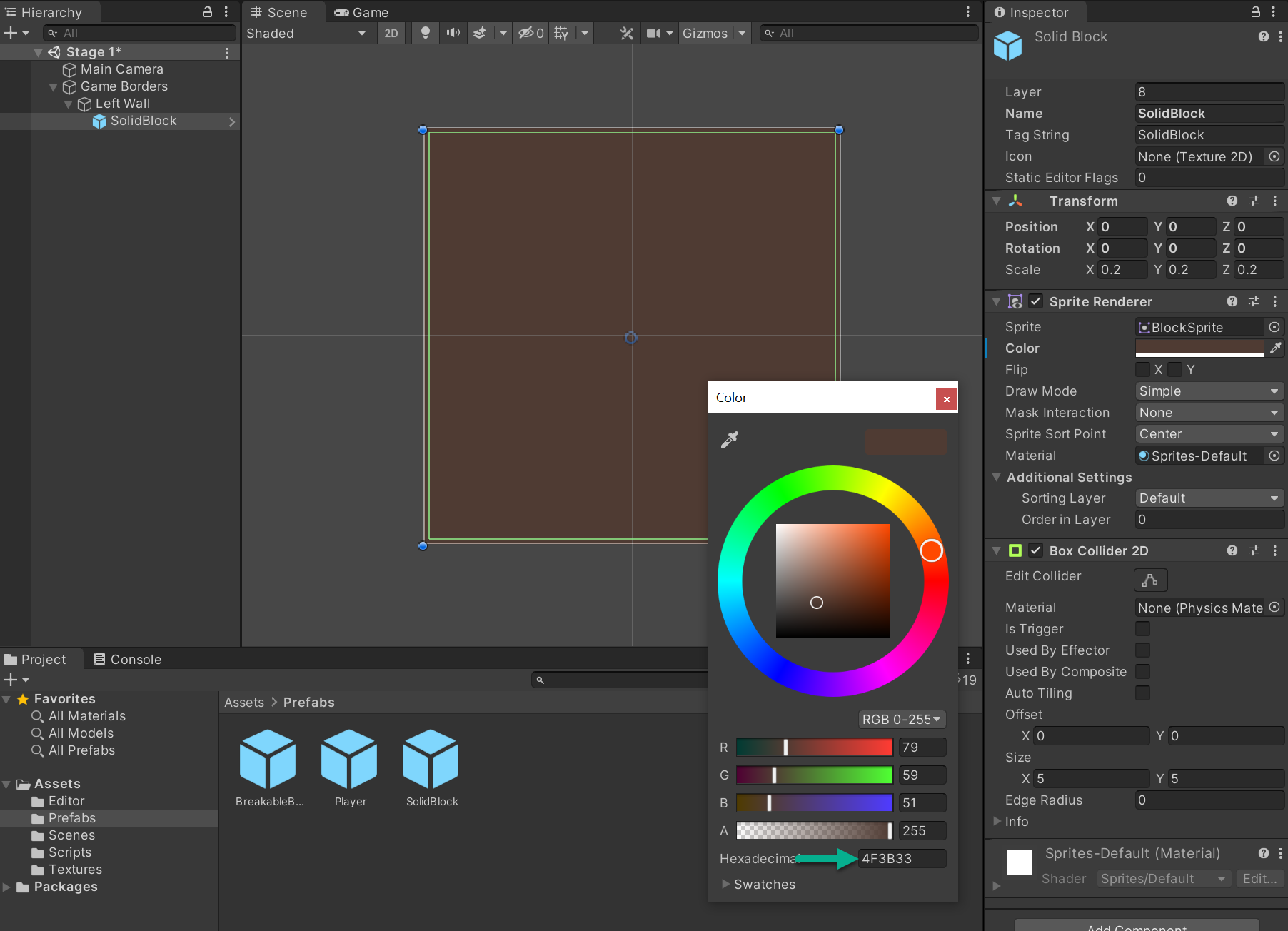
 

In Project, select Assets >Prefabs and drag the SolidBlock prefab inside Hierarchy to be a child object of Left Wall.

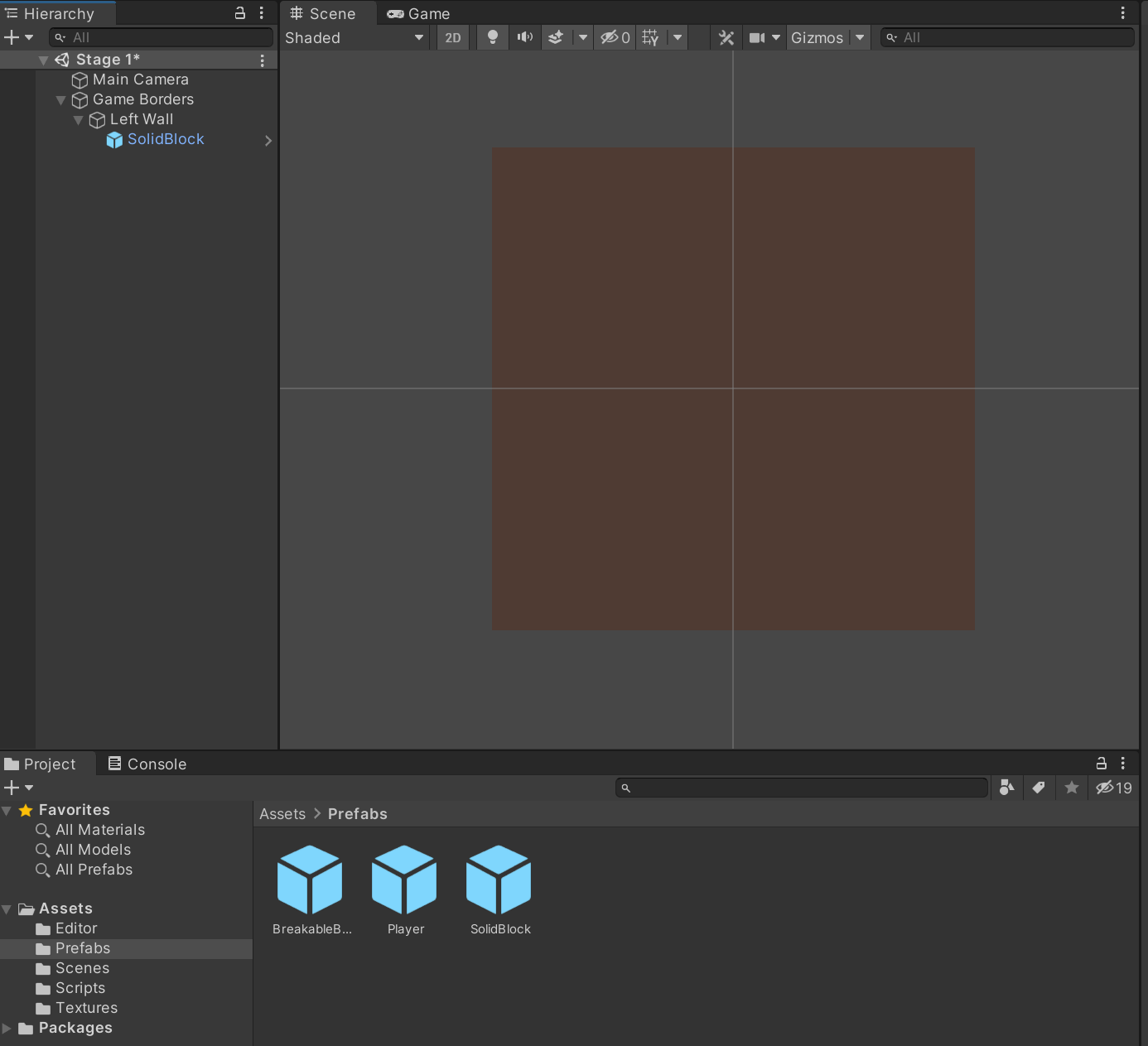




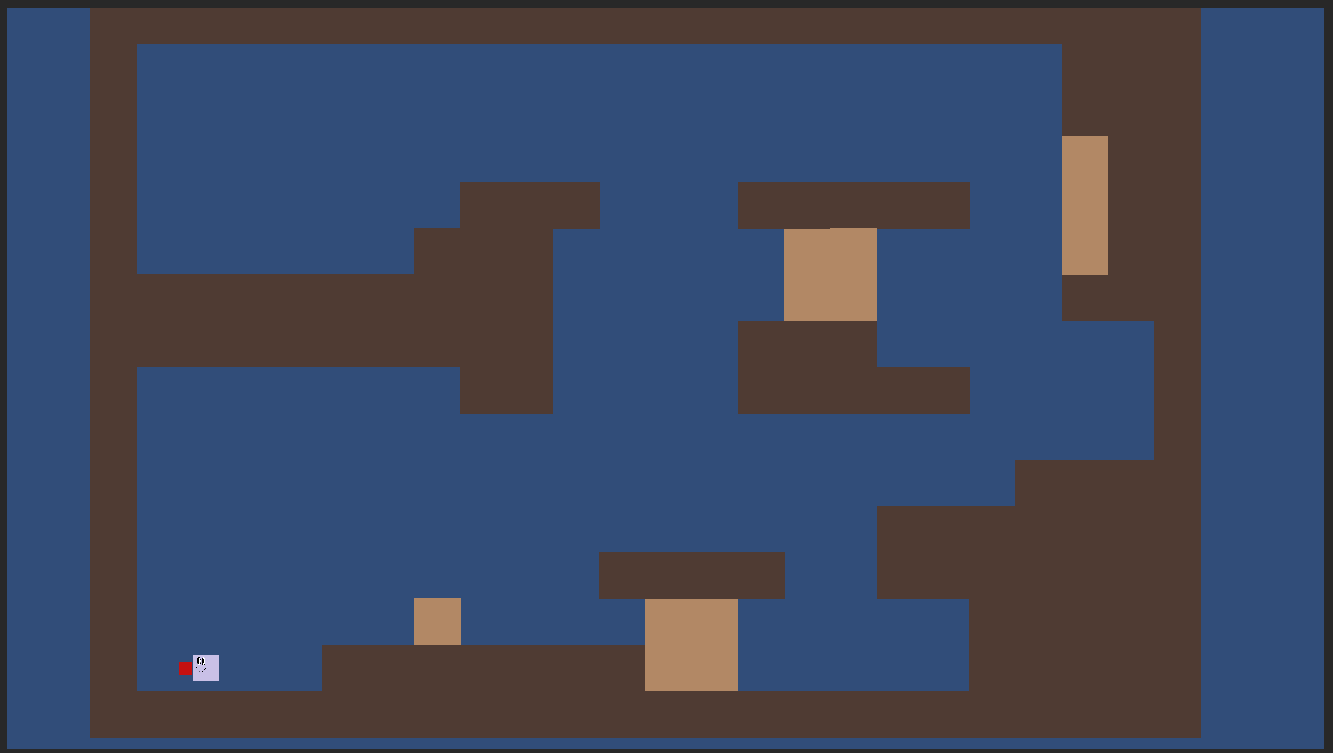
Your first goal is to build a 15-block vertical wall that serves as your left border. Select Left Wall > SolidBlock. In Sprite Renderer, change Color to hexadecimal 4F3B33



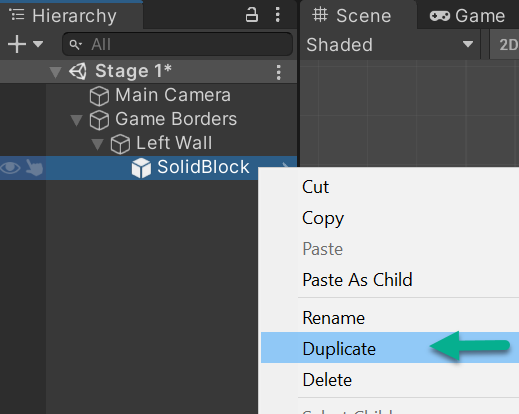
Resulting in:

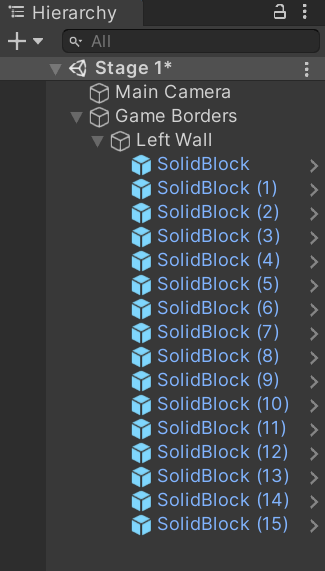


You'll start with Left Wall and follow the tutorial to build your version of the example below. This tutorial will mirror this image, you may elect to design a different wall in shape and/or color.

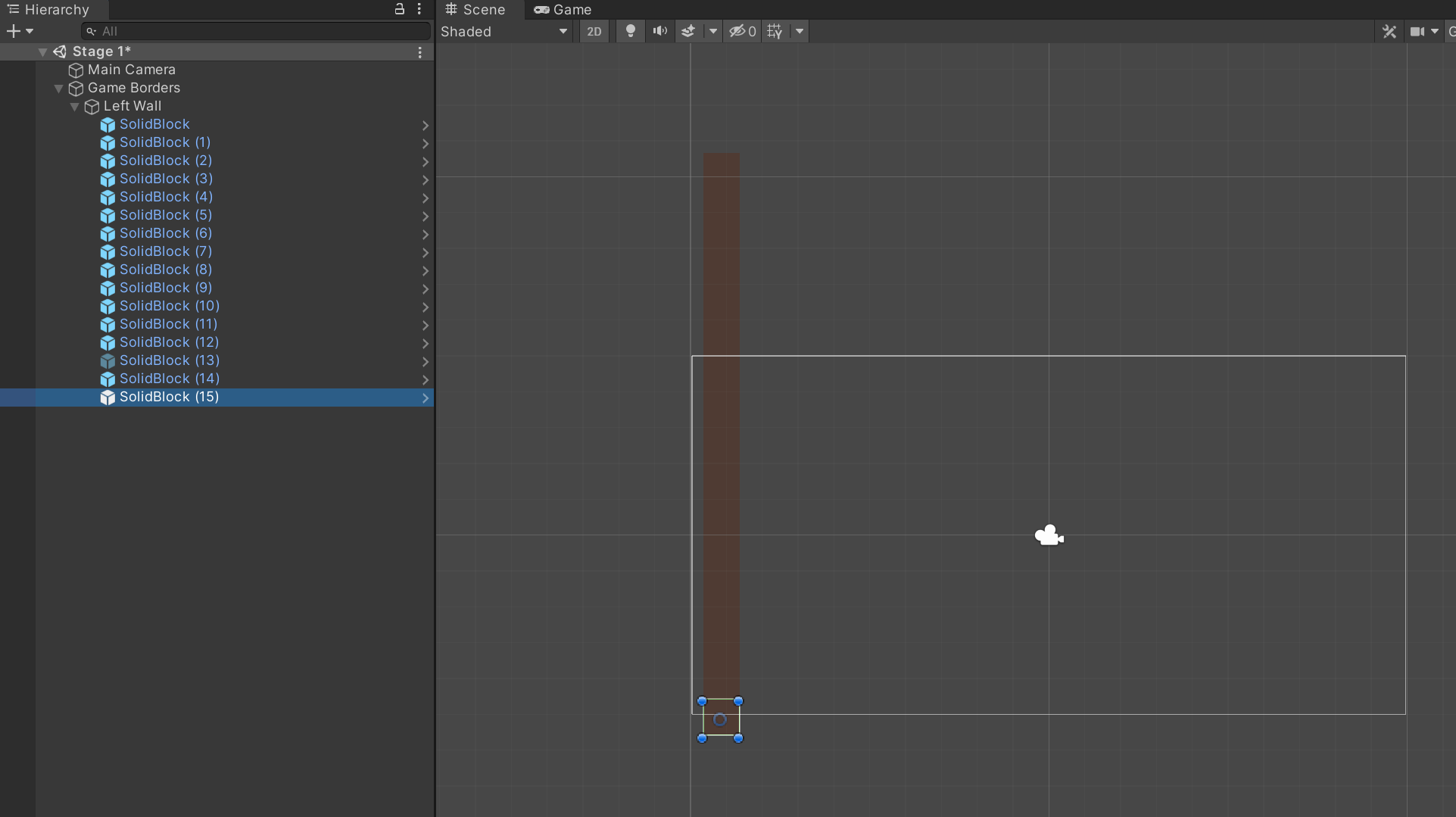


Select SolidBlock and key Control + D to Duplicate SolidBlock 15 times. Or right click > Duplicate to duplicate one at a time.





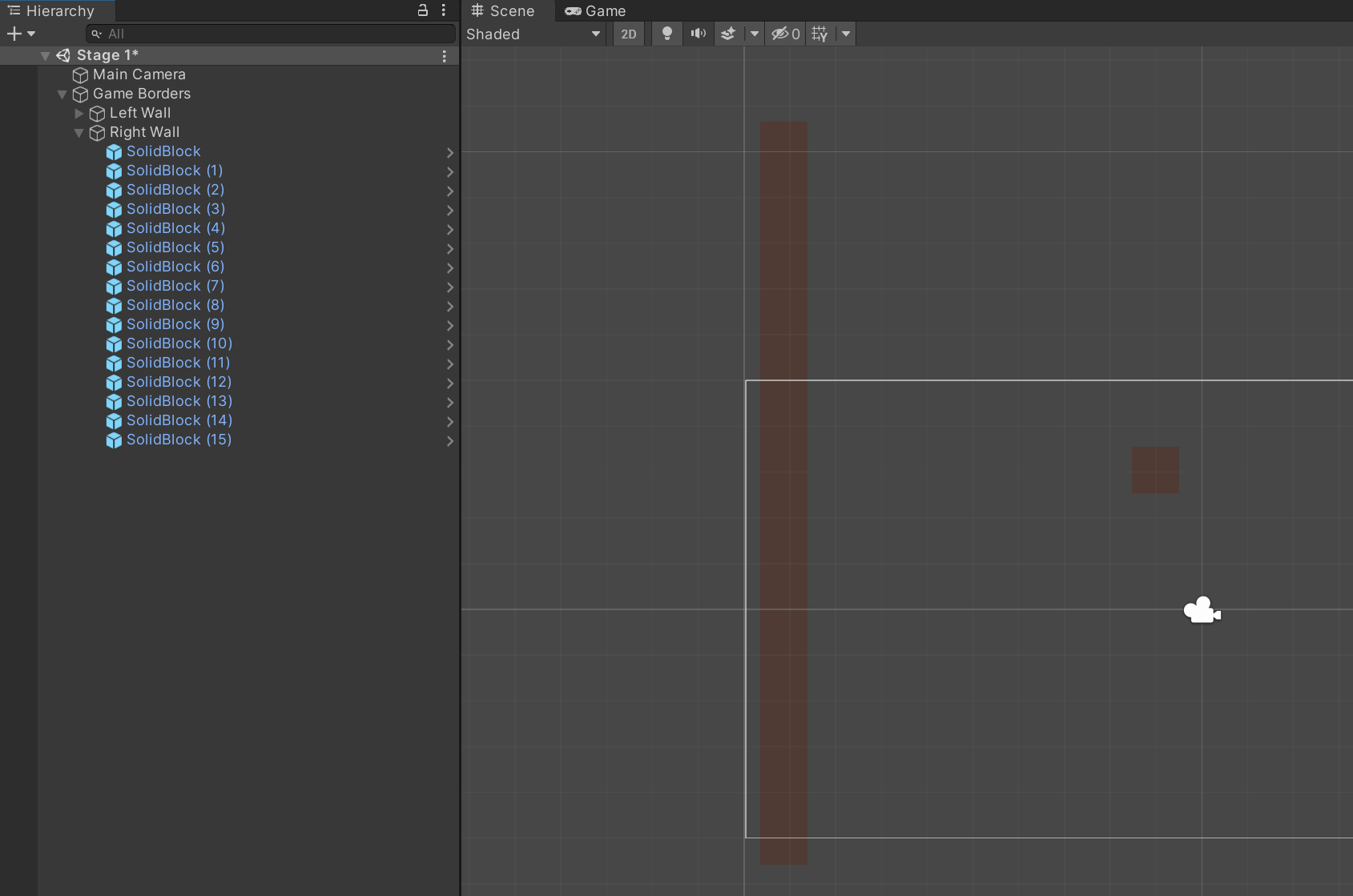
Next, position every block to build the left wall in the scene using the Rect Tool. Select each of the 16 (SolidBlock through SolidBlock (15)) one at a time and move them to build a left wall. Don't be concerned about aligning to the Camera icon area because it's there to let you know where Main Camera is focused at this moment.



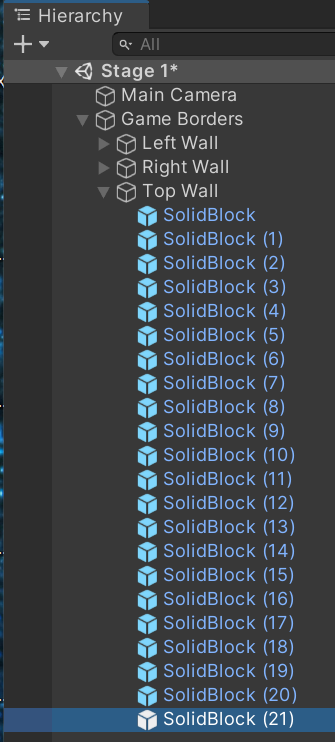
Repeat for all SolidBlock objects. See the bullets below. It’s good practice to get them as close as possible by hand. You can also use Inspector > Transform > Position:

* Use your mouse roller to zoom in and out for space to place your SolidBlocks (click into Scene first)
* To focus on any SolidBlock, double click on that block
* Up and down arrow (keyboard) lets you traverse up and down the left border in Scene (click into Scene first)
* Make sure each SolidBlock lines up straight with no spaces between blocks
* Select any block on the wall and which SolidBlock it is will highlight in Hierarchy
* Sometimes it’s easier to set up all blocks and then take time to align them straight and tight together (zoom in)
* If you grab and resize a block, go to the top menu Edit > Undo Resize (or Ctrl+z) to reverse the size change.

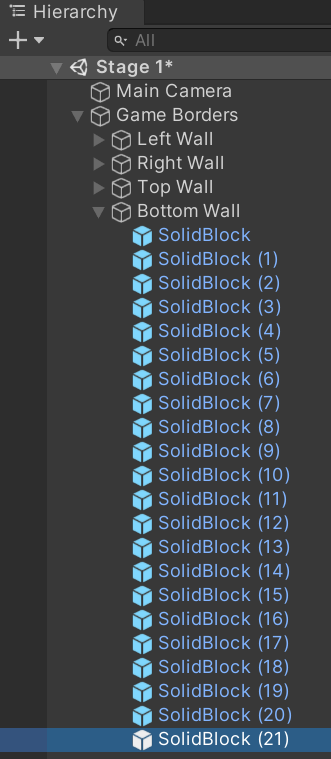
Next, follow the same process you developed to create Left Wall for Right Wall. Right click in Hierarchy to Create Empty > Right Wall. Drag Right Wall to be a child of Game Borders (not a child of Left Wall). Drag and drop Prefabs > SolidBlock up to Right Wall. Change the color of SolidBlock to hexadecimal 4F3B33. Duplicate SolidBlock 15 times. Right Wall duplicates to match Left Wall. Save your Project. Rather than lay out Right Wall now, you will first create Top Wall and Bottom Wall to save time in counting out blocks to ensure alignment with the two side walls.



Right click in Hierarchy to Create Empty > Top Wall. Drag Top Wall to be a child of Game Borders (not a child of Right Wall). Drag and drop Prefabs > SolidBlock up to Top Wall. Change the color of SolidBlock to hexadecimal 4F3B33. Duplicate SolidBlock 21 times. Top Wall is wider than Right or Left Wall are high. Save your Project.

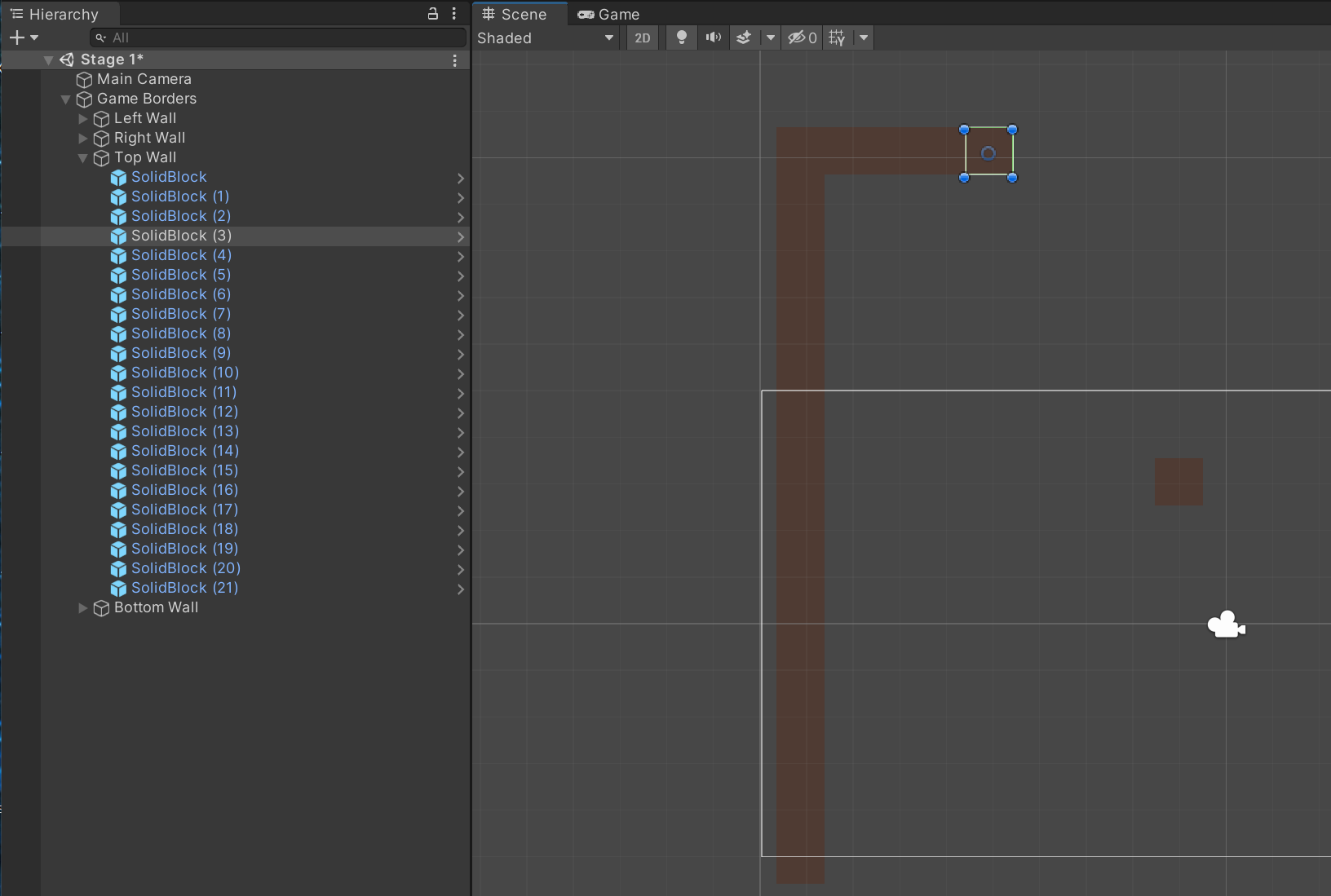


Right click in Hierarchy to Create Empty > Bottom Wall. Drag Bottom Wall to be a child of Game Borders (not a child of Top Wall). Drag and drop Prefabs > SolidBlock up to Bottom Wall. Change the color of SolidBlock to hexadecimal 4F3B33. Duplicate SolidBlock 21 times. Save your Project.

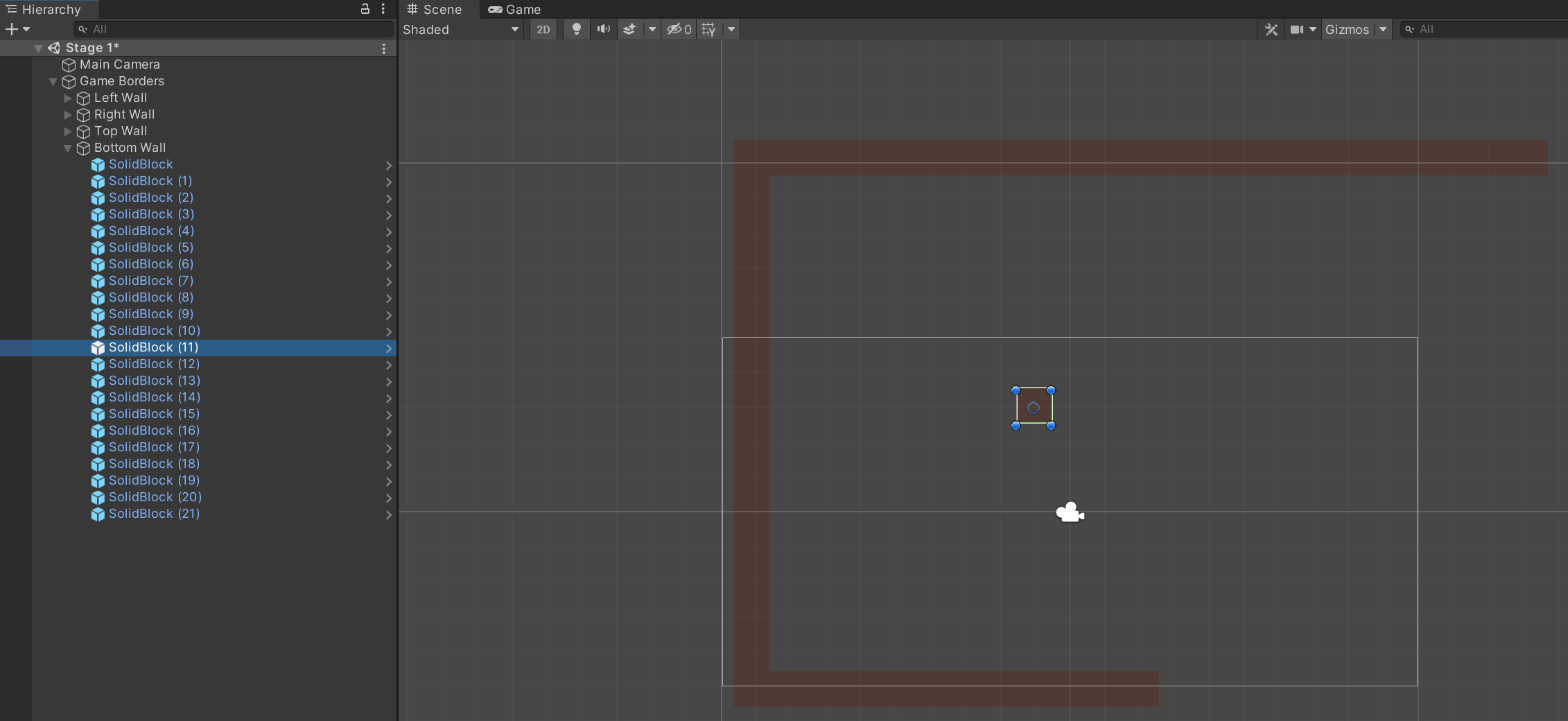


Next, create your top wall with your Top Wall SolidBlock objects. Your bottom wall with your Bottom Wall SolidBlock objects. And your right wall with your Right Wall SolidBlock objects. This rectangle results in your game's border. You might start with Top Wall by selecting each block in Hierarchy and positioning along the top, starting next to the top block in Left Wall.

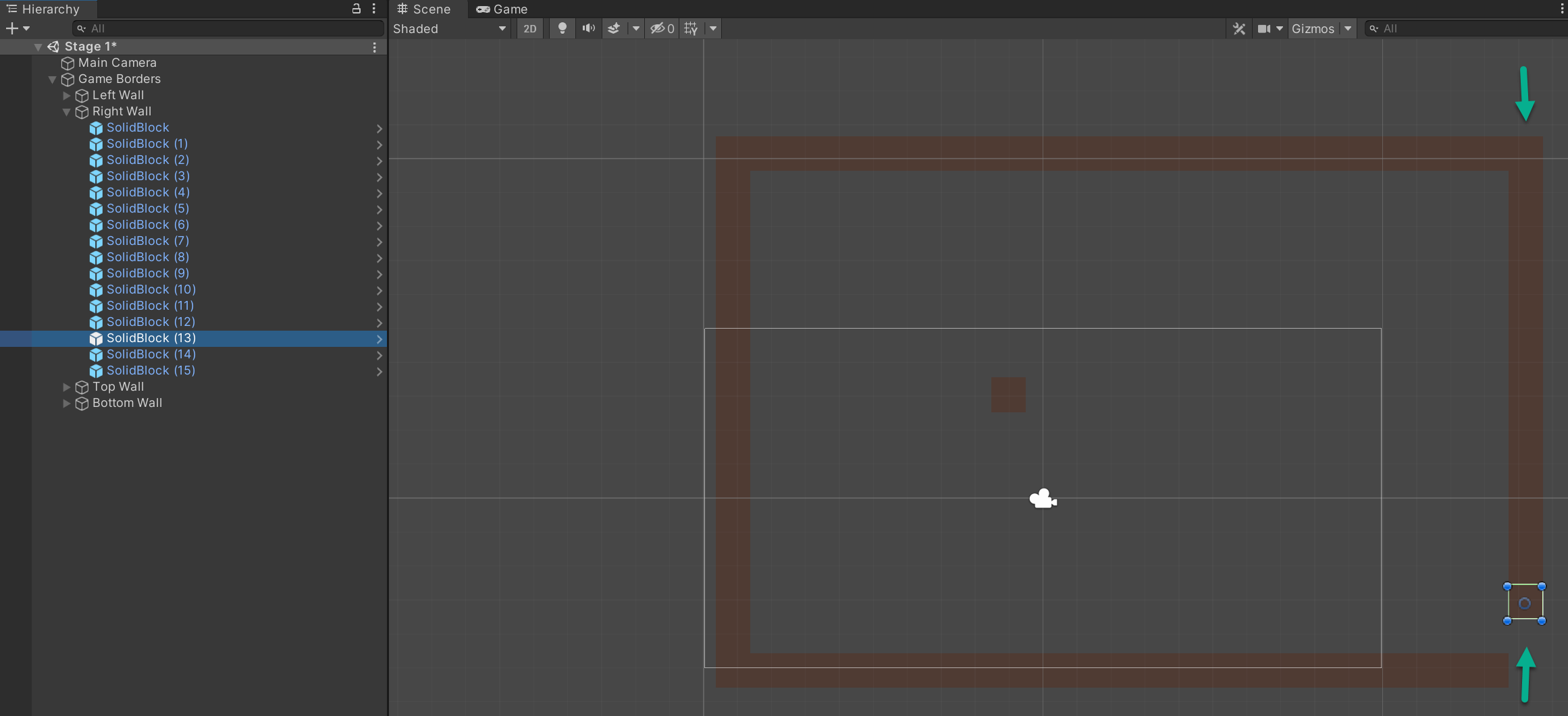
Reminder\* if moving a block distorts its size or shape, use Ctrl+Z to return to original size and shape. If you lose track of which block, select the block in Scene to view in Hierarchy or select the block in Hierarchy to view in Scene.



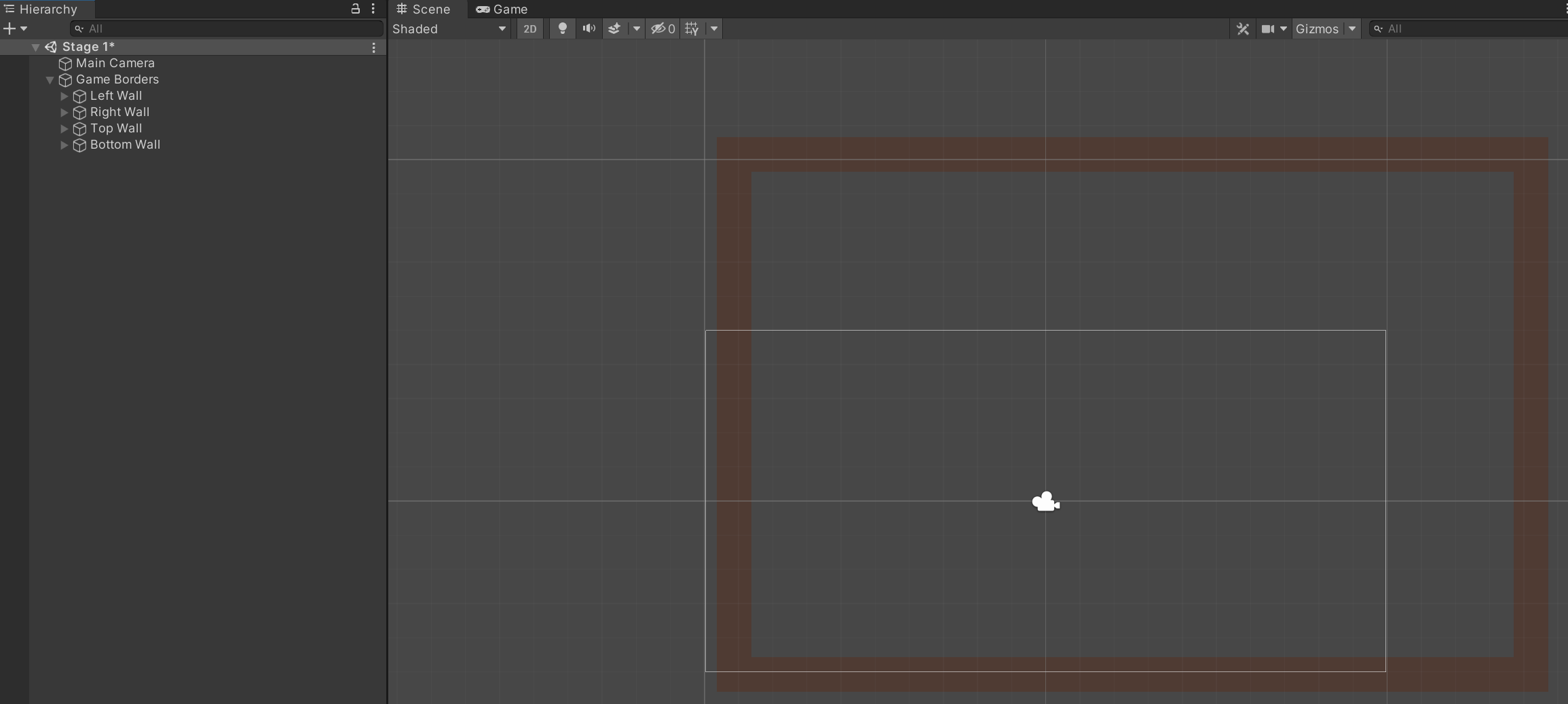
Next, repeat for Bottom Wall.



And to close your border, repeat for Right Wall. Start the top of Right Wall **after** the last block of Top Wall, not under Top Wall and not between the last block in Top and Bottom Wall. As shown in the image below, Right Wall aligns to the right of the last blocks in Top and Bottom Wall. If you misplace blocks, you can easily move them without causing any issues with your code.



Your complete border appears as:



Congratulations on creating your game border!

This ends Milestone 1 Deliverable 3 of Downey Platformer. Proceed to: milestone1\_deliverable4.pdf where you will stack SolidBlocks and BreakableBlocks within your border to interact with the Player and build the first stage of your game.

Please link to:

<https://www.oer4cte.org/downey_platformer_tutorials/milestone1_deliverable4.pdf>

and right click to  
Save as…  
and download the tutorial to your computer.