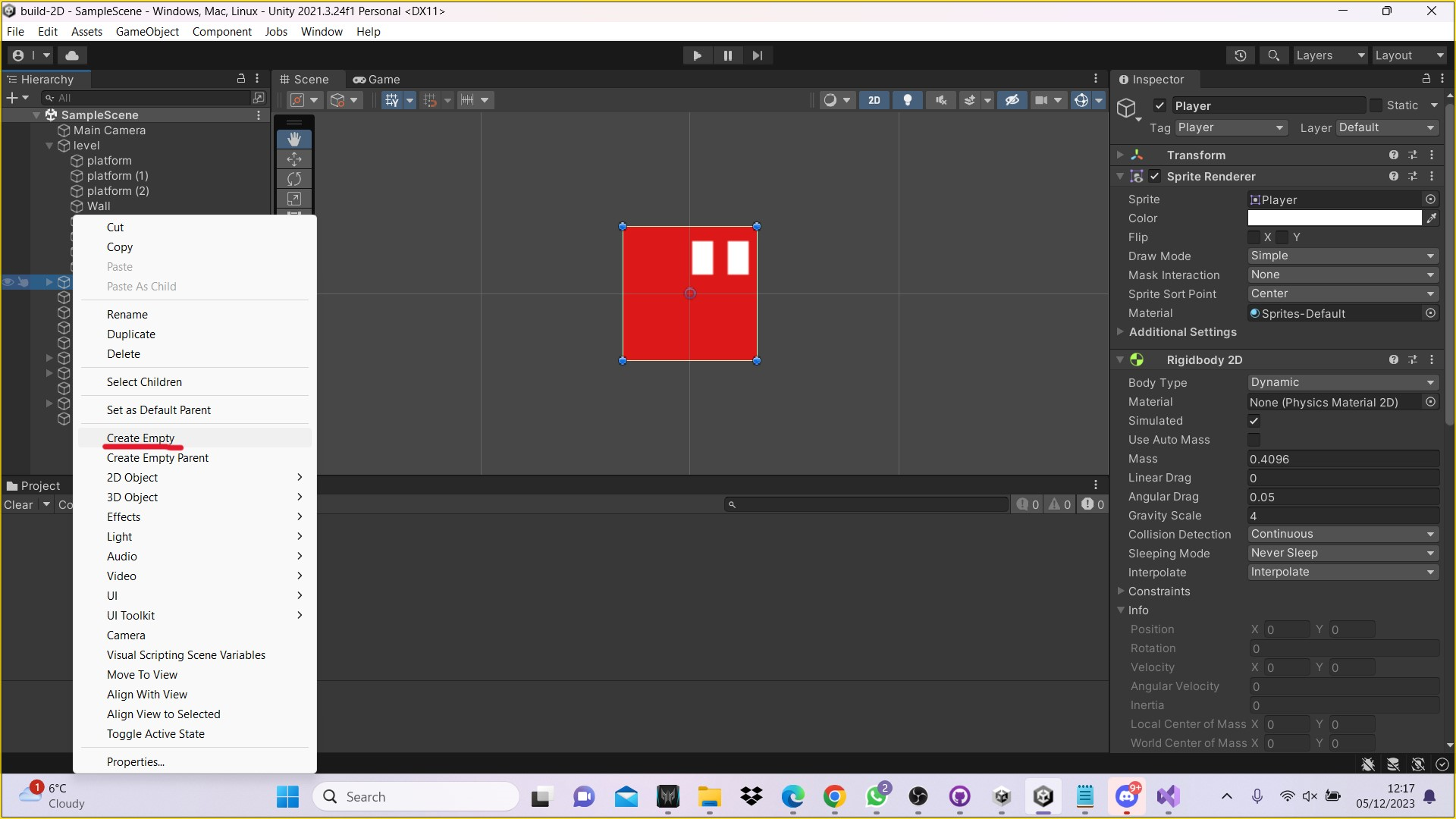
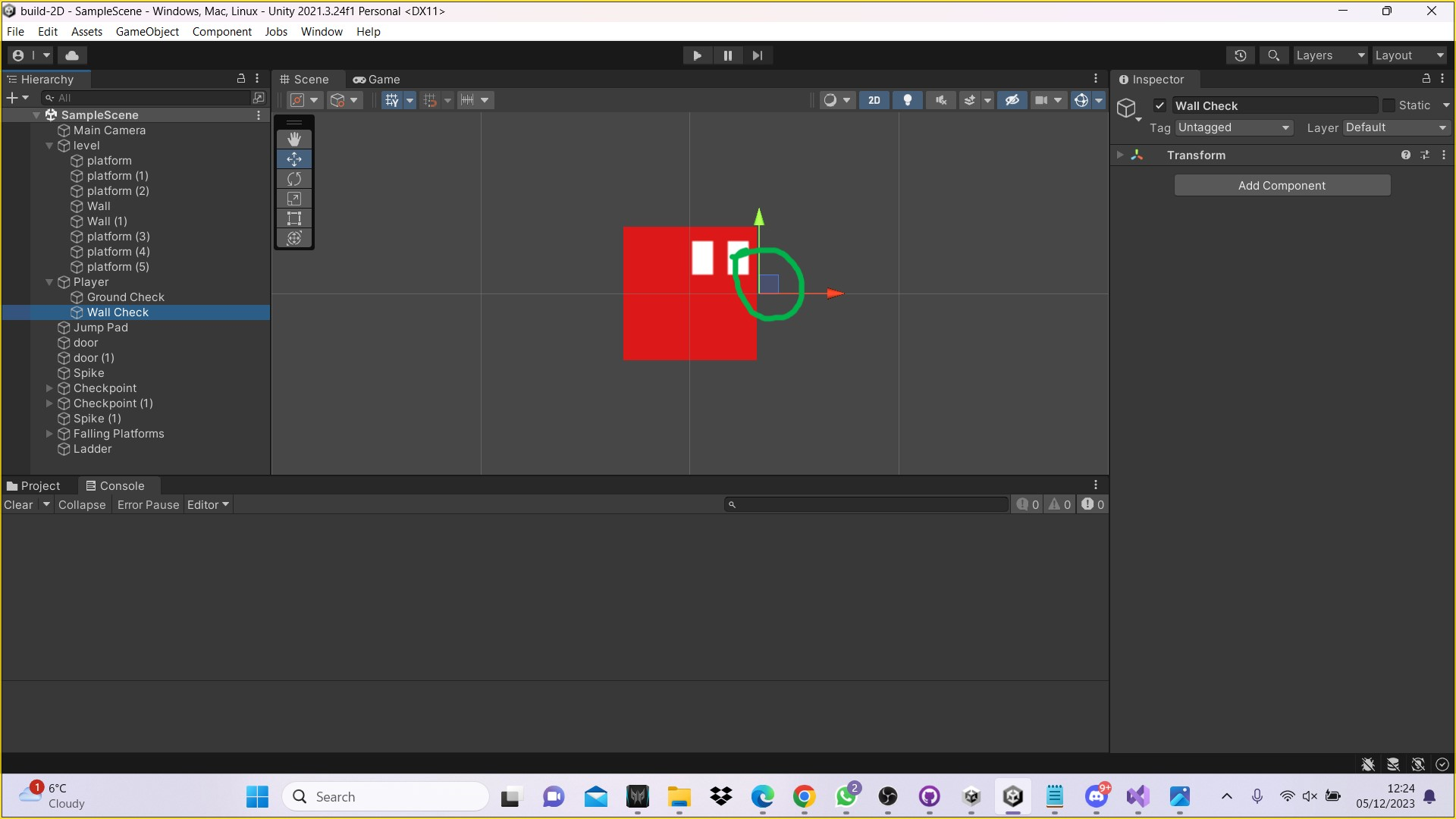
Climb

1. With the mouse right click on player then click create 'Empty' option

name it 'Wall Check'.



2. When you see the angle of the blue square on screen this means the player will be able to jump and hold or slide down or collide.

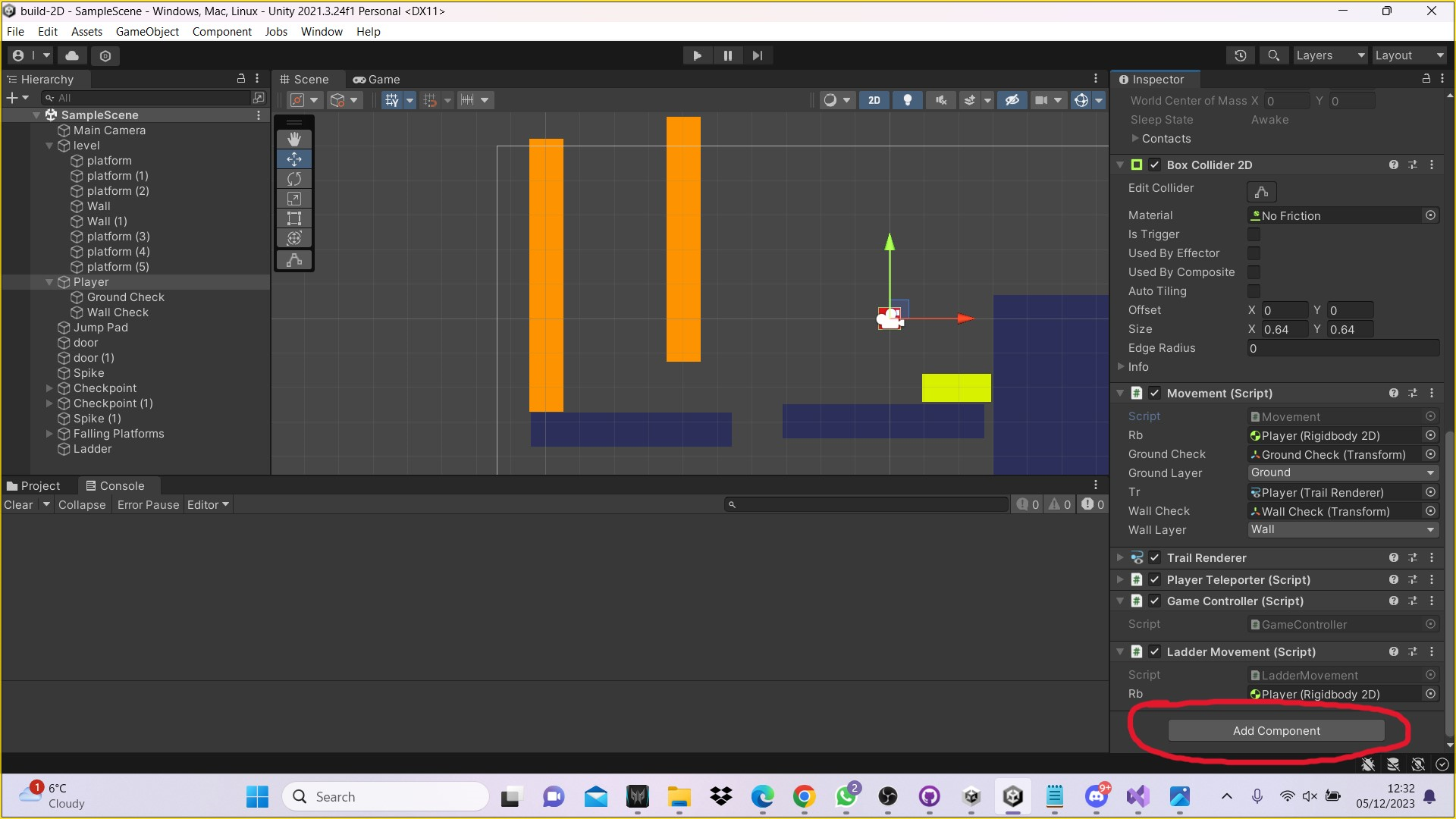


Create the wall

1. To create the wall you need to click on 'Add Component' in the 'Inspector' panel.

2. In the search tab select 'Box Collider 2D'

3. Click on 'Player' in the left-hand side Hierarchy Console panel. In the Inspector panel on the right-hand side double click on 'Script' under Movement (script). This will open up the build 2D Microsoft Visual Studio Script window.



Step 4 as shown on screenshot

4. Add code name:

public class Movement : MonoBehaviour

{

private bool isWallSliding;

private float wallSlidingspeed = 2f;

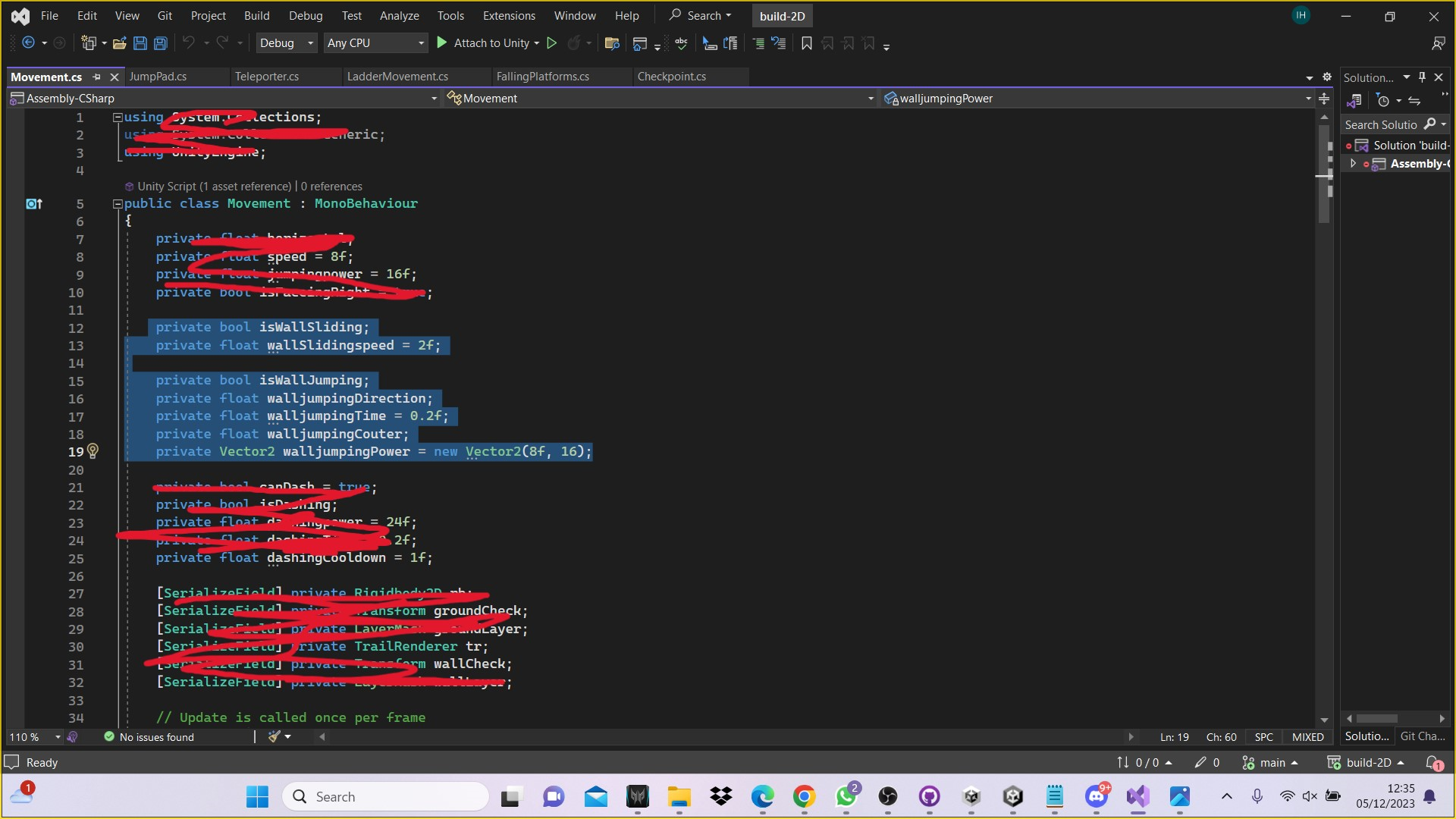
private bool isWallJumping;

private float walljumpingDirection;

private float walljumpingTime = 0.2f;

private float walljumpingCouter;

private Vector2 walljumpingPower = new Vector2(8f, 16);



Step 5 as shown on screenshot

5. Add check object in code:

[SerializeField] private Rigidbody2D rb;

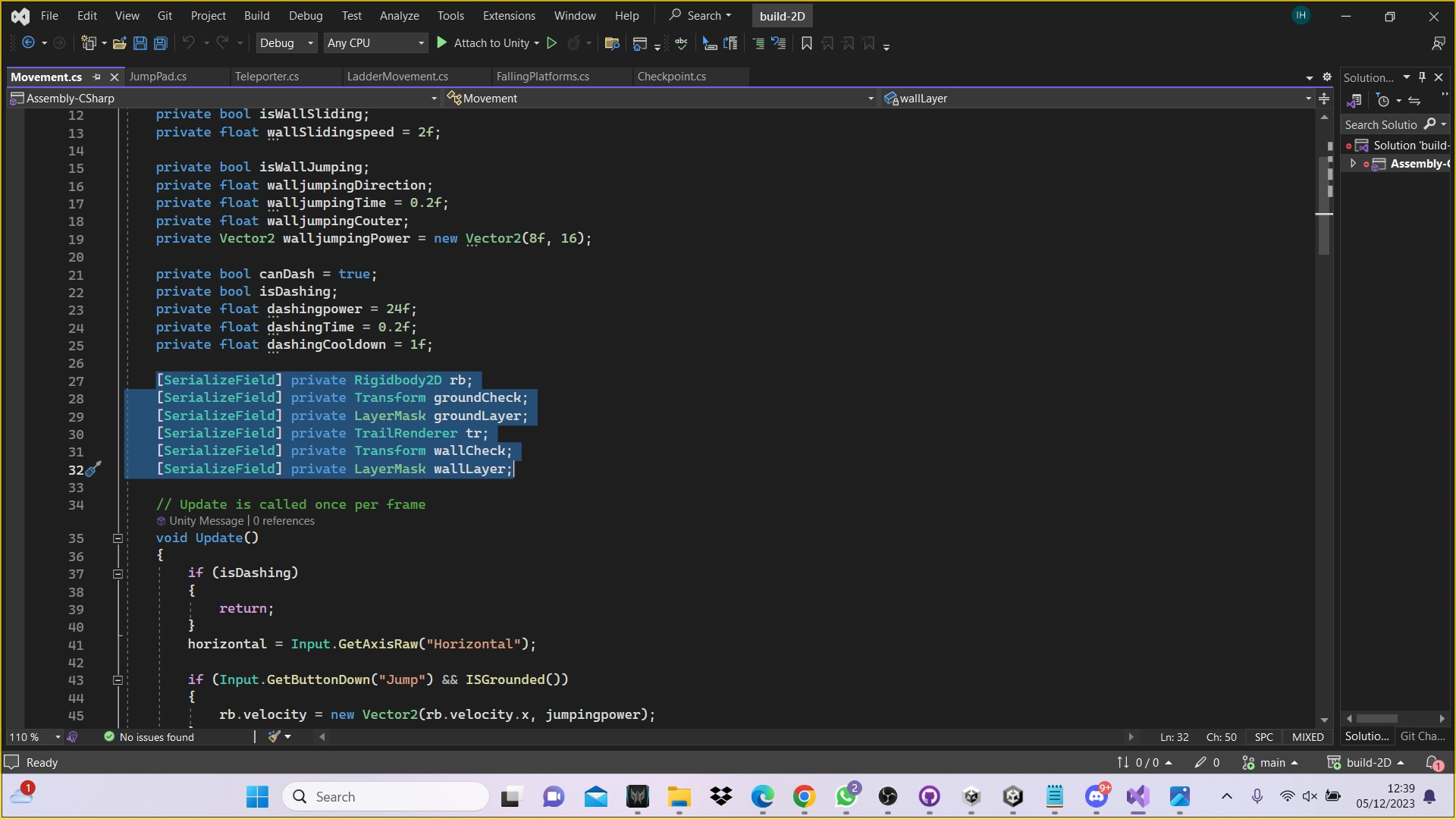
[SerializeField] private Transform groundCheck;

[SerializeField] private LayerMask groundLayer;

[SerializeField] private TrailRenderer tr;

[SerializeField] private Transform wallCheck;

[SerializeField] private LayerMask wallLayer;



Step 6 as shown on screenshot

6. Add a Fixed Update code if need. if it is already there then leave it.

private void FixedUpdate()

{

private void WallSlide()

{

if (ISWalled() && !ISGrounded() && horizontal != 0f)

{

isWallSliding = true;

rb.velocity = new Vector2(rb.velocity.x, Mathf.Clamp(rb.velocity.y, -wallSlidingspeed, float.MaxValue));

}

else

{

isWallSliding = false;

}

}

7. see code name: void Update() if you find void update then you need to add in this WallSlide(); because then if there is a wall then the box can hold on and slide down.

8. add code:

WallSlide();

9. Save this code using CTRL + S. Now go back into Unity, if it still is not working in Unity then you save the code in Unity as well.

10. Once saved, go on to Unity and click on 'Wall' in the left-hand side panel and then on the 'inspector' panel on the right-hand side click on 'layer' select 'Add Layer.

11. On the same panel Create Wall which option 7. Where Wall is explained in the code it needs to match with the 'Tags & Layers' panel.

12. Click on 'Player' in the left-hand side Hierarchy panel. On the right-side panel go down to where it states Movement (Script).

13. Drag and drop the Wall Check from the left-hand side Hierarchy panel to the Wall Check option on the right-hand side 'Inspector Movement (Script) panel.

14. Under Wall Layer tab on the Movement (Script) panel this will give you option to select

- Wall or Ground

- You can select either just wall or wall and ground together.

15. The next line private void WallSlide(): When climbing in 2D you can change your direction of the jump. This is shown in the code below.

private void WallJump()

{

if (isWallSliding)

{

isWallSliding = false;

walljumpingDirection = -transform.localScale.x;

walljumpingCouter = walljumpingTime;

CancelInvoke(nameof(StopWallJumping));

}

else

{

walljumpingCouter -= Time.deltaTime;

}

if (Input.GetButtonDown("Jump") && walljumpingCouter > 0f)

{

isWallJumping = true;

rb.velocity = new Vector2(walljumpingDirection \* walljumpingPower.x, walljumpingPower.y);

walljumpingCouter = 0f;

if (transform.localScale.x != walljumpingDirection)

{

isFaceingRight = !isFaceingRight;

Vector3 localscale = transform.localScale;

localscale.x \*= -1f;

transform.localScale = localscale;

}

Invoke(nameof(StopWallJumping), walljumpingDirection);

}

}

Wall jumping

16. wallslide() name the next line wall slide: the player would jump onto the wall and can flip from one wall to the other by changing direction.

WallJump();

if (!isWallJumping)

{

Flip();

}

}

For your wall jump

17. The next step is jumping with speed and extra height. As the character flips across the wall they get higher up the wall until they reach the top.

{

if (isWallJumping)

{

rb.velocity = new Vector2(horizontal \* speed, rb.velocity.y);

}

18. Final step is to check the game is working.