**Unity - Learn with Code - Challenge 3 - Balloons and Booleans**

A screenshot of a computer

Description automatically generated

A screenshot of me opening Challenge 3 of Unity – Learn with Code, which is a game where you control a balloon and need to navigate it through a city, collecting dollar signs and avoiding bombs. Unfortunately, when I hit the play button, the game is plagued with problems, with the main being that the balloon just randomly exploded when I hit the space button. Other problems include the background still moving when the balloon popped, none of the objects are appearing on screen, the background is repeating in the wrong way and fireworks are appearing to the side of the balloon. This is where I needed to start fixing the problems of this floating balloon game.

A screenshot of a computer program

Description automatically generated

A screenshot of me fixing the fixing first problem of the game, which was the player not having any way to control the balloon at all, so I added this into void Start() in the PlayerControllerX.script: playerRb = GetComponent<Rigidbody>();, to which I can now control the balloon by pressing the space button.

A screenshot of a computer program

Description automatically generated

A screenshot of me fixing the second problem of the game, where the background is only moving when the game is over, to which this was fixed adding “== false” on the void Update() if (playerControllerScript gameOver) on MoveLeftX.cs

A screenshot of a computer program

Description automatically generated

A screenshot of me fixing the third problem of the game, where none of the objects are spawning into the game, to which the problem was in SpawnManager.cs where the term, “PrawnsObject” was used, which needed to be replaced with “SpawnObjects”.

A screenshot of a computer

Description automatically generated

A screenshot of me fixing the fourth problem of the game where the fireworks are appear on the side of the balloon instead of being on it, so I used the moving tool to move the fireworks onto the balloon, so it doesn’t feel odd-looking to the player.

A screenshot of a computer program

Description automatically generated

A screenshot of me fixing the fifth problem where the background was not repeating properly, to which I fixed this preplacing the size.y with a size.x on the repeatWidth on the RepeatBackground.cs.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

Two screenshots completing the first bonus problem in the game, where the balloon is floating way too high, to the point of leaving the game completely, so we fixed this by going into the PlayerController.cs script, by adding in “public bool isLowEnough;” into the public class, and then going into void Update() to add a if(transform position.y > 13).

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

Three screenshots of me fixing the second bonus problem of the game, which is not letting the balloon hit the floor, to which we did by creating a Ground tag for the Ground object, created an else if on the private void OnCollisionEnter for the newly created Ground tag in PlayerControllerX.cs and then creating a public AudioClip bounceSound and the playerAudio.PlayOneShot(bounceSound, 1.5f);