|  |  |
| --- | --- |
| Test Case 1 |  |
| Line Error | *FindObjectOfType<GameSession>().ResetGame();* |
| Error Explanation | The Game cannot find the *GameSession* in the Scene |
| Error Correction | *GameSession* prefab was moved in the Hierarchy |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 2 |  |
| Line Error | Serialize Field int damageWaveBullets = 1; |
| Error Explanation | *The type or namespace ‘Serialize’ could not be found* |
| Error Correction | Write SerializeField as one word with capital S and F and put it in square brackets. |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 3 |  |
| Line Error | var obstacle = Instantiate(waveConfig.GetObstaclePrefab(), waveConfig.GetWayPoints()[0].transform.position, Quaternion.identity); |
| Error Explanation | An obstacle is not assigned in Wave 1 in the Obstacle Prefab field, and so the code from the ObstacleSpawner.cs can’t get and return the obstacle prefab |
| Error Correction | Click in the Waves folder, click on Wave1 and in the Obstacle Prefab field choose rockObstacle |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 4 |  |
| Line Error | for (int obstacleCount = 1; obstacleCount <= WaveConfig.GetNumberOfObstacles(); obstacleCount++) |
| Error Explanation | The variable WaveConfig does not exists in the current context of the coroutine in the parameter |
| Error Correction | The w of the waveConfig which is found in the parameter of the for loop need to be written in a lower case. |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 5 |  |
| Line Error | // Update is called once per frame  void Update()  {  ObstacleMove();  } |
| Error Explanation | The method named ObstacleMove() does not exists in the ObstaclePathing script |
| Error Correction | In the void method instead of writing ObstacleMove(); write ObstacleMoving(); |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 6 |  |
| Line Error | yield return Coroutine(SpawningAllTheWaves()); |
| Error Explanation | The Coroutine can not be used as a method that can start SpawingAllTheWaves |
| Error Correction | Before Coroutine the word Start need to be typed down else it won’t start SpawningAllTheWaves. This needs to be typed as yield return StartCoroutine(SpawningAllTheWaves()); |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 7 |  |
| Line Error | IEnumerator Start( |
| Error Explanation | The method IEnumerator Start does not finish with two round brackets, and so a bracket ) is expected to be at the end of this line |
| Error Correction | After IEnumerator Start add another bracket IEnumerator Start() |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 8 |  |
| Line Error | [SerializeField] float backgroundScrollingSpeed = 1.0; |
| Error Explanation | Trying to assign and convert a value of type double to a type of float |
| Error Correction | After the value 1.0, a small f should be added to this 1.0f |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 9 |  |
| Line Error | Vector23 offSet; |
| Error Explanation | The type Vector23 could not be found because this game is a 2D game and so it will only work with Vector2 because it contains the points of X and Y and so Vector23 does not exists |
| Error Correction | Instead of writing Vector 23 write Vetor2 offSet; |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 10 |  |
| Line Error | myMaterialBackground.mainTextureOffset += offSet \* Time.Time; |
| Error Explanation | Time.Time will not be able to move the background GameObject in the y axis which means that the background can’t scroll upwards |
| Error Correction | After Time. I need to add delta like this Time.deltaTime; |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 11 |  |
| Line Error | Update() |
| Error Explanation | The method update must have a return type |
| Error Correction | Before the word update write void in small letters |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 12 |  |
| Line Error | AudioSource.PlayClipAPoint(obstacleSound, Camera.main.transform.position, obstacleSoundEffect); |
| Error Explanation | *AudioSource dos not contain a definition of PlayClipAPoint* |
| Error Correction | Instead of writing AudioSource.PlayClipAPoint(obstacleSound, Camera.main.transform.position, obstacleSoundEffect); I need to write AudioSource.PlayClipAtPoint(obstacleSound, Camera.main.transform.position, obstacleSoundEffect);  After PlayClipA I need to add a small t after A which is like this: PlayClipAt |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 13 |  |
| Line Error | waypointElement+; |
| Error Explanation | Currently the waypointElement is equal to 0 and so the obstacle is on waypoint0, due to waypointElement+; is like that, the obstacle can’t move to the next waypoint |
| Error Correction | After the first plus add another plus waypointElement++; so the obstacle can be able to move on all the waypoints |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 14 |  |
| Line Error | obstacle.GetComponent<>().SetWaveConfig(waveConfig); |
| Error Explanation | I am not specifying which component or script I want to get from the GameObject. The GetComponent function doesn’t know which type of component I want to get because there are many components. |
| Error Correction | In the triangular brackets write <ObstaclePathing>, so you can be able to gather all the information of the obstcales which is saved in ObstaclePathing and later on, the obstacles can spawn continuously in different waves |
| Error Correction ScreenShot |  |

|  |  |
| --- | --- |
| Test Case 15 |  |
| Line Error | if(FindObjectsOfType<SessionPlay>(). > 1) |
| Error Explanation | It is not accessing a class or a variable or a method on what I am providing. The problem is that SessionPlay is not accessing the array Length. Error identifier expected means that it needs to access something within something else. |
| Error Correction | Write Length after the dot(.) |
| Error Correction ScreenShot |  |