Tim Rolshud

Patterns Simple Slide Scroller Quiz Study Guide

1. What is a "prefab"?

A prefab is a blueprint for a Unity GameObject.

2. What is a "sprite"?

A Sprite is a visual representation of a GameObject that would appear in Unity.

3. What is the Hierarchy?

The Hierarchy window is the representation of all the GameObjects and their inheritants.

4. What is a Scene?

A scene is one of multiple views a Unity game can have, and a game can switch between different scenes.

5. What is the difference between the Scene Editor and the Game Window?

The Scene editor lets you establish the initial state of the Scene and add all the initial GameObjects, while the Game Window is already during runtime of the game.

6. What is the Project tab for?

The project tab lets you manage the different assets and scripts and lets you refactor them and update the references to them within the Unity Editor.

7. After we save a resource into Unity, can we move that resource around within the project from File Explorer/Finder? How *can* we safely move resources around

By using the project tab as it takes care of proper refactoring, if you move it with just the file explorer, Unity's references will not update it which may cause the project to fail to build

8. Once we drag the Sprite into the game scene, what does it become, and in what important ways is this different from a sprite?

A Spire becomes a GameObject, which is already a physical object that exists within the game as opposed to just a blueprint.

9. What is a Transform?

Transform is one of multiple components that can be attached to Unity GameObjects, with Transform specifically modifying the position, rotation, and scale of whatever GameObject it is modifying.

10. What does it mean for one GameObject to be the child of another GameObject in terms of its Transform?

The child's Transform component will be relative to the parent's Transform component, meaning that if the child's position is (0, 0, 0) and scale is (1,1,1), it will be in the same position and be the same size as its parent.

11. What is Start()?

Unity Method that every GameObject has that gets called **once** upon the creation of the GameObject.

12. What is Update()?

Unity Method that every GameObject has that gets called **every frame** of the game environment until the GameObject is removed from the game environment.

13. What is a RigidBody2D?

Physics component for Unity GameObjects in a 2D game environment

14. What is the difference between a kinematic and a dynamic RigidBody2D?

Kinematic has no physics applied to it except its initial velocity and inertia, while a dynamic RigidBody has all of the forces like gravity, inertia, etc. applied to it.

15. What does GetComponent do, and do you know its syntax?

GetComponent returns the component object that belongs to a GameObject given its type, it uses a templating syntax where the method takes no arguments, but you put the type you are looking for in angle brackets.

16. Is a Vector2 an Object or a Struct?

All Vectors in Unity are Structs, so they are therefore copied/passed by value in C#

17. For that matter, what is the difference between an Object and a Struct?

Objects have the same functionality as they do in Java where they get passed by reference, whereas structs get copied by value

18. What instance variables in our C# scripts appear in the inspector?

Instance variables that have the [SerializeField] flag defined above them

19. What are the benefits of having a variable in the inspector?

Having a variable in the inspector lets you easily debug it and update it on the go.

20. What sorts of methods would we find in the Physics2D class, and why is everything in this class static?

Methods for calculating physics related quantities, they are static because they are meant for general use throughout the codebase and are not bound to any specific GameObject

21. What are layers used for, and how do we create them?

To organize different GameObjects and organize the ways in which they interact with each other. You can define them and then assign GameObjects to specific layers through the Inspector for each GameObject

22. What is a BoxCollider2D?

Acts like a collider only the hitbox is built as a square around the sprite instead of the shape of the sprite itself, allows for more simple but less accurate physics interactions, it might look like something is hitting it without actually hitting it because it hit the box around the GameObject figure

23. When would we make a BoxCollider2D a trigger?

When we want certain events in Unity to happen upon a GameObject colliding with other GameObjects, we would make the GameObject that is being a trigger and handle events that we want to happen upon collision

- For instance, game collectible would have a trigger on it upon colliding with a player it would up their score

24. What does [SerializeField] do?

Exposes the variable that the flag is above to the Unity Inspector and allows you to modify it form within the inspector

25. What is the difference between Awake() and Start()?

Awake also runs once but before start, used to initialize things in the game even before it starts

26. When is OnTriggerEnter2D called, and what is the parameter Collider2D other for?

This method is called when something collides with the Trigger Collider of the GameObject this method is defined under, and the parameter Collider2D is for the collider of the other GameObject that collided with this one

27. Can we call constructors on GameObjects? *What do we do instead?*

No, instead we use the Awake() method to initialize the necessary components and fields of this GameObject before we start using it, and the Start() method to perform something initially after it is first created

28. What is a Canvas?

Canvas is a UI platform within Unity that lets you make UIs

29. Does a Canvas use the same Camera as the standard game?

No, Canvas has a camera pointed directly at it to see the UI

30. How do we write out to the Console?

In C# in general you use the Console.WriteLine() method, but in Unity you use the Debug.Log() method