Syllabus for Unity Game Development Club

Club description: This club is for students who want to learn how to create video games with Unity and C#. Unity is one of the most popular game engines out there and has been used to make several high-quality games, but it still manages to be very beginner friendly. In this club, we will explore Unity from the ground up, looking at the Unity UI system, scripting, 2D and 3D game development, physics, animation, sound, and more. By the end of this club, students will have the skills and confidence to make their own games with Unity.

Objectives: Students will be able to:

- Use and understand the Unity editor and its components.
- Write basic C# scripts to control game logic and behavior.
- Create user interfaces with text, images, buttons, sliders, etc.
- Make 2D and 3D games with sprites, models, cameras, lights, etc.
- Apply physics and collisions to their game objects.
- Animate their characters and objects with keyframes and curves.
- Add sound effects and music to their games.
- Build and deploy their games to different platforms.

Prerequisites: No prior experience with Unity or C# is required. However, some basic knowledge of programming concepts such as variables, functions, loops, and conditions would be helpful. They will also need a computer that can run Unity and a stable internet connection.

Club materials: Students will need to download and install the instructed version of Unity. Students will also need a code editor such as Visual Studio or Visual Studio Code. They can find some free assets for their games from the Unity Asset Store or other online sources.

Club outline: The club is covering the following topics throughout the semester:

1. **Introduction to the environment**
    - Learn how to download and install Unity
    - Explore the Unity editor and its windows
    - Create their first Unity projects and scene
    - Learn how to use the hierarchy, inspector, project, console, and game windows
    - Learn how to create and manipulate game objects and components
    - Learn how to save and load scenes

2. **Unity Basics**
    a. Learn how to use the scene view and the game view
    b. Learn how to use the transform tool, the rotate tool, the scale tool, and the rect tool
    c. Learn how to use prefabs and instantiate them in their scenes
    d. Learn how to use tags and layers to organize their game objects
    e. Learn how to use cameras and camera settings
    f. Learn how to use lights and lighting settings
3. **Introduction to the game engine**
    a. Learn how to use the play mode and the pause mode
    b. Learn how to use the time settings and the time class
    c. Learn how to use the input settings and the input class
    d. Learn how to use the physics settings and the physics class
    e. Learn how to use the quality settings and the quality class
    f. Learn how to use the graphics settings and the graphics class
4. **Unity and the different platforms**
    a. Learn how to use the build settings and the player settings
    b. Learn how to build their games for different platforms such as Windows, Mac, Linux, Android, iOS, etc.
    c. Learn how to test their games on different devices or simulators
    d. Learn how to optimize their games for performance and compatibility
    e. Learn how to troubleshoot common errors and issues
5. **Scripting**
    a. Learn how to create C# scripts in Unity
    b. Learn how to attach scripts to game objects or components
    c. Learn how to use variables, constants, data types, operators, expressions, etc.
    d. Learn how to use functions, parameters, return values, scope, etc.
    e. Learn how to use classes, objects, constructors, inheritance, polymorphism, etc.
    f. Learn how to use interfaces, abstract classes,
6. **Unity in action**
    a. Learn how to use events such as Awake(), Start(), Update(), FixedUpdate(), LateUpdate(), etc.
    b. Learn how to use coroutines for asynchronous tasks
    c. Learn how to use delegates, events, actions, lambdas, etc. for callbacks
    d. Learn how to use collections such as arrays, lists, dictionaries, etc. for storing data

7. **Creating a simple game**
    a. Apply what they have learned so far to create a simple game of their choice
    b. Follow the steps of game development such as planning, designing, prototyping, testing, debugging, etc.
    c. Use the Unity UI system to create menus, buttons, text, etc.
    d. Use the Unity animation system to create animations for their characters and objects
    e. Use the Unity audio system to add sound effects and music to their games
    f. Add some polish and flair to r game with particles, shaders, post-processing, etc.

Evaluation: The club will not have any formal exams or grades. However, students will be expected to complete small assignments and one big project. The assignments will be based on the topics covered in each session and will require students to apply what they have learned. The project will be based on the students' own game ideas and will require students to create a playable prototype of their game. The students will showcase their games to the rest of the school at the end of the semester. Hopefully, this showcase will peak other students' interests in game development as well.

Expectations: The club expects students to follow these rules and guidelines:

- **Be respectful** of each other's ideas, opinions, work, and feedback.
- **Be responsible** for your own learning and progress.
- **Be cooperative** and **collaborative** with your teammates and peers.
- **Be creative** and **curious** about game design and development.
- **Be passionate** and **enthusiastic** about making games.