

- Activating and deactivating objects
- Tags and layers
- Collision and trigger events
- Raycasting
- Coroutines
- Object references
- Basic game-play systems through code

15. Input Systems

- Keyboard input
- Mouse input
- Touch input basics
- Old Input Manager
- New Input System
- Player movement
- Interaction controls
- Input-based gameplay actions

16. 2D Game Development

- Sprites and sprite types
- Sprite Editor
- Slicing spritesheets
- 2D physics
- Rigidbody2D and Collider2D
- 2D character movement
- 2D animation
- Tilemap basics
- 2D level setup

FLAPPY BIRD CLONE

18. Animation System

- Animation window
- Animator component
- Animator Controller
- Parameters and transitions
- Playing animations
- Triggering animations from script
- 2D and 3D animation workflow
- Basic character animation setup

19. Audio System

- Importing audio files
- Audio Source and Audio Listener
- Background music

- Sound effects
- Playing audio through script
- Volume control
- Looping audio
- Audio management basics
- Practical use in gameplay and UI

20. Particle System and Visual Effects

- Introduction to particle system
- Creating effects like smoke, fire, sparkle
- Particle properties
- Triggering effects in gameplay
- Combining particles with animation and audio

21. Lighting and Materials

- Directional light
- Point light
- Spot light
- Materials basics
- Shadows
- Scene lighting setup
- Basic visual polish
- Lighting for 2D and 3D scenes

22. Scriptable Objects

- What are ScriptableObjects
- Why use ScriptableObjects
- Storing reusable game data
- Creating custom data assets
- Using ScriptableObjects for configuration
- Practical examples: weapon data, player stats, audio data, level data

Angry Birds Style Physics Game for 2D Development

23. 3D Game Development

- 3D objects and primitives
- Transform in 3D
- Mesh Renderer and Mesh Filter
- Rigidbody and Collider in 3D
- 3D camera basics
- 3D player control
- 3D environment setup
- Basic interaction in 3D scene

3D First Person Shooter (FPS) Game

ADVANCED

Photon PUN Multiplayer Module

- Server connection and Photon setup
- Lobby and room management
- Creating and joining multiplayer rooms
- Network player spawning
- Synchronizing players and game objects
- PhotonView and RPC communication
- Room and player custom properties
- Multiplayer UI and gameplay flow
- Scene syncing and match handling

2D Platformer Multiplayer Game using Photon PUN

22. AR Foundation

- Introduction to Augmented Reality in Unity
- Setting up AR Foundation
- AR Session and AR Session Origin / XR Origin
- Plane detection
- Object placement in real world

- Raycast in AR
- Image tracking basics
- Face tracking basics
- Working with AR camera

AR Basketball Project using AR Foundation

23. VR Development / XR Interaction Toolkit

- Introduction to VR in Unity
- OpenXR basics
- XR Interaction Toolkit
- XR rig setup
- Interactors and interactables
- Grab, select, and UI interaction in VR
- VR object interaction
- Teleportation basics
- VR scene setup
- Building immersive applications

VR Interaction Project using XR Interaction Toolkit

UNITY GAME DEVELOPMENT PROGRAM

Learn
2D, 3D,
AR, VR &
Multiplayer
Development

Duration
4
Months

Learn how modern games and interactive applications are built using Unity. This program covers 2D and 3D game development along with AR, VR, and multiplayer systems through hands-on projects.



No prior game development experience required. Basic programming knowledge is helpful but not mandatory.

BUILD 6 REAL PROJECTS DURING THE COURSE



PROJECTS YOU WILL BUILD

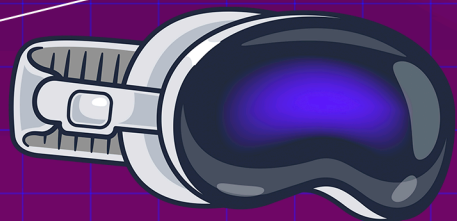
- Flappy Bird Clone
- Angry Birds Style Physics Game
- 3D First Person Shooter Game
- Multiplayer Game
- AR Basketball Application
- VR Interaction Project

WHAT YOU WILL LEARN

- Unity Fundamentals
- C# Programming for Games
- 2D Game Development
- 3D Game Development
- Multiplayer Game Development
- Augmented Reality (AR)
- Virtual Reality (VR)

Who Should join this Program

- Students interested in learning game development
- Students who want to explore AR and VR technologies
- Students interested in interactive application development
- Students who enjoy coding and creating projects
- Beginners who want to learn Unity development from the basics



BASIC COURSE

1. Unity + First Script

- Introduction to Unity Hub
- Unity Editor installation
- Introduction to Unity scripting
- What is a GameObject and Component
- Creating a C# script
- Attaching script to an object
- Understanding Start() and Update()
- Using Debug.Log()
- First practical script in Unity

2. Variables and Data Types

- What are variables
- Declaring and using variables
- Integer, float, string, boolean
- Public and private variables
- Taking values from the Inspector
- Real examples: score, speed, health, player name

3. Casting and Conversions

- Why type conversion is needed
- Converting float to int
- Converting int to float
- String to number conversion
- Using Parse(), ToString(), and Convert
- Practical use in Unity UI and gameplay

4. Operators

- Arithmetic operators
- Assignment operators
- Increment and decrement
- Comparison operators
- Logical operators
- Using operators in gameplay conditions

5. Conditions

- if statement
- if-else
- else-if
- Nested conditions
- switch statement
- Real examples: game over, level unlock, door open system

6. Loops

- Introduction to loops
- for loop
- while loop
- do while loop
- foreach loop
- Using loops for repeated tasks
- Practical examples: spawning enemies, printing scores, repeating actions

7. Functions

- What is a function
- Why functions are important
- Creating and calling functions
- Parameters and arguments
- Return values
- Function overloading
- Reusing code for actions like jump, attack, reset

8. Arrays

- What is an array

- Storing multiple values
- Accessing array elements
- Looping through arrays
- One-dimensional arrays
- Introduction to multi-dimensional arrays
- Real examples: enemy list, quiz questions, level names, spawn points

9. Math for Gameplay

- Using math in games
- Math basics
- Movement calculations
- Speed and direction
- Using Time.deltaTime
- Distance and angle calculations
- Random values in gameplay
- Practical examples: movement, jump, timer, object rotation

10. Classes and Objects

- Introduction to object-oriented programming
- What is a class
- What is an object
- Fields and methods
- Creating custom classes
- Organizing game logic with classes
- Real examples: Player, Enemy, Weapon, Bullet

11. Inheritance / Abstraction / Interfaces

- Understanding inheritance
- Base class and derived class
- Code re-usability using inheritance
- What is abstraction
- Abstract classes and abstract methods
- What is an interface
- Implementing interfaces in Unity
- Real examples: enemy types, damage system, collectible system

12. Accessing Components and Object Properties

- Understanding Unity components
- Accessing components in code
- Accessing transform
- Changing Position
- Changing Rotation
- Changing Scale
- Reading and modifying component properties
- Practical examples using Transform, SpriteRenderer, and Collider etc

13. Unity UI / Canvas System

- Canvas basics
- Text, Image, Button, Panel, Slider
- RectTransform basics
- Anchors and responsive UI
- Button click events
- Updating UI with scripts
- Health bar, score, timer, menus
- UI organisation for games

14. Object Handling and Game-play Coding

- Instantiating objects with Instantiate()
- Destroying objects