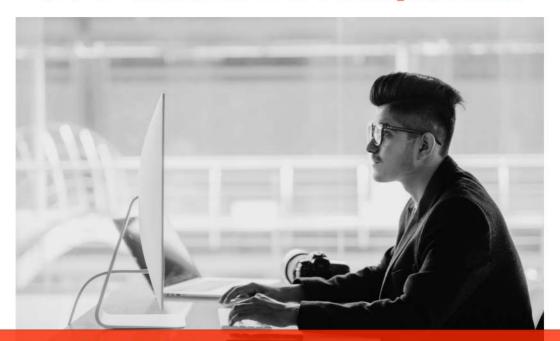


# C++ Game Development



# E-LEARNING COURSE

### ZETLAN TECHNOLOGIES

### **COURSE MODULES**

#### Module 1: Introduction to C++ for Game Development

- Overview of C++ in game development
- Setting up a development environment (Visual Studio, VS Code)
- Understanding game engines (Unreal Engine, SFML, SDL, OpenGL)
- Basics of C++ (variables, data types, operators, control structures)

#### Module 2: Object-Oriented Programming in C++

- · Classes and objects
- · Inheritance and polymorphism
- · Encapsulation and abstraction
- Operator overloading
- · Smart pointers and memory management

#### **Module 3: Game Mathematics and Physics**

- Coordinate systems and transformations
- Vector and matrix operations
- Collision detection (AABB, circle, and pixel-perfect collisions)
- Physics simulation basics (velocity, acceleration, forces)

#### Module 4: Graphics Programming with SDL/SFML/OpenGL

- · Rendering images and sprites
- Handling textures and animations
- · Basic shaders and lighting techniques
- Using OpenGL for 2D and 3D rendering

#### Module 5: Input Handling & Game Loop

- Handling keyboard and mouse inputs
- Implementing a real-time game loop
- Frame rate control and delta time
- Event-driven programming

#### **Module 6: Audio and Sound Effects**

- Adding background music and sound effects
- Using libraries like SDL\_Mixer or OpenAL
- Adjusting volume and effects dynamically

#### Module 7: Game Al and Pathfinding

- Finite State Machines (FSM)
- Basic enemy Al behaviors (patrolling, chasing)
- Pathfinding algorithms (A\*, Dijkstra)

#### Module 8: Scripting and Game Logic

- Integrating Lua or Python for scripting
- Using scripts to control game behaviors dynamically
- Event-driven scripting systems

#### Module 9: Multiplayer and Networking

- Introduction to networking in games
  - Using sockets for multiplayer interactions
- Client-server architecture
- Synchronization and latency handling

# Module 10: Building a Simple 2D Game

- Designing game assets and mechanics
- Implementing player controls and interactions
- Adding UI elements (score, health, menus)
- Game state management (pause, resume, restart)

## Module 11: Introduction to 3D Game Development

Module 12: Optimizations and Performance Tuning

Monetization strategies (ads, in-app purchases, premium mode

- Basics of 3D rendering with OpenGL
- Introduction to 3D physics engines (Bullet, PhysX)
- Working with 3D models and animations

Memory management and profiling

#### Multithreading in game development Reducing CPU and GPU bottlenecks

Module 13: Publishing and Deployment

- Packaging games for different platforms
- Debugging and testing best practices

# Final Project: Develop a Complete Game

- Planning the game idea

  - Implementing gameplay mechanics
  - Testing and debugging
- Showcasing the final product