

PROGRAMME GROUP: BSE

UNIT TITLE: Game Development

UNIT CODE: BSD 412

Lesson 1:

Coverage Area

- Introduction to game programming
- The architecture of a game
- Game Engine and History of Game Development
- The 5 Best Game Engines for Beginners in Video Game Development
- Best Programming Languages for Game Development

Introduction

This course requires a lot of patience and dedication

- Game development is the **procedure of creating video games which involves working on some platform/software to develop the game**. The development is undertaken by a **game developer** which can be one person or a team of
 - i. **Game Engineers** (for constructing the overall structure and game play),
 - ii. **Designers** (for designing the objects and animations),
 - iii. **Coders** (for coding and giving life to the game including logics, score and other calculations),
 - iv. **Project Manager** (for managing and promoting the game),
 - v. **Game Tester** (who will test the game after it has been developed to find bugs in the game, if any).

Why Programmers Should Learn Game Development?

objective

- ✚ To get better and coding and architecting programs
- ✚ To become a better programmer and learn key programming principles and skills

1. To learn Data Structure and Algorithms Better

- use a data structure like I didn't know that array can be used to create platform games where the screen is just a 2D array, and different objects are just different integer values.
- If you also want to get better at Data structure and Algorithms, I suggest you write a couple of simple games like Tic-Tac-Toe, and Tetris, you won't regret that decision.
- Algorithms skills you need to name a few.

Arrays, Linked Lists, Trees, Hashtables, Stacks, Queues, Heaps, Sort algorithms and Search algorithms

2. To Learn Concurrency

Benefits of Concurrency

results in sharing of resources result in problems like deadlocks and resources starvation.

It helps in techniques like coordinating execution of processes, memory allocation and execution scheduling for maximizing throughput.

Do video games use concurrency?

Concurrency can speed up our programs significantly by optimizing the use of resources and wait time that would be otherwise wasted. In many cases using concurrency might improve the program, but it is often required for high performance systems like real time apps and video games.

- Games also teaches how to use multi-threading because, for any sort of game, you need two threads, a game thread that updates data, and a thread that handles events and renders screen.
- While creating games there are issue that happens when you try to update the GUI in the same thread and how a long-running network query can freeze your GUI. All these experiences help to write better code which is both concurrent, thread-safe, and responsive.
- AREA of Reading to have a good grasp in concurrency: I highly recommend [Java Multithreading, Concurrency & Performance Optimization](#)

3. Learn about Performance and User Experience (is how a user interacts with and experiences a product, system or service.)

- you learn a lot about performance, responsiveness, and user experience by writing games because they become more obvious and critical of games.
- Well, that was even before Unity 3D was born. Also, though most game development experience is in J2ME, Java Micro edition for developing games for mobile devices, I genuinely understand the value they provide in improving Programming and Coding skill.
- **If you are also interested in improving your coding skills.** I also suggest you learn about SOLID design principles; they will help you to write better and more flexible code.
 - **NB** :Uncle Bob Martin in his classic book Clean Code, one of the **must-reads for any programmer.**

SOLID Principles — explained with examples

- S - Single Responsibility Principle (known as SRP)
- O - Open/Closed Principle.
- L - Liskov's Substitution Principle.
- I - Interface Segregation Principle.
- D - Dependency Inversion Principle.

Further reading on solid principles

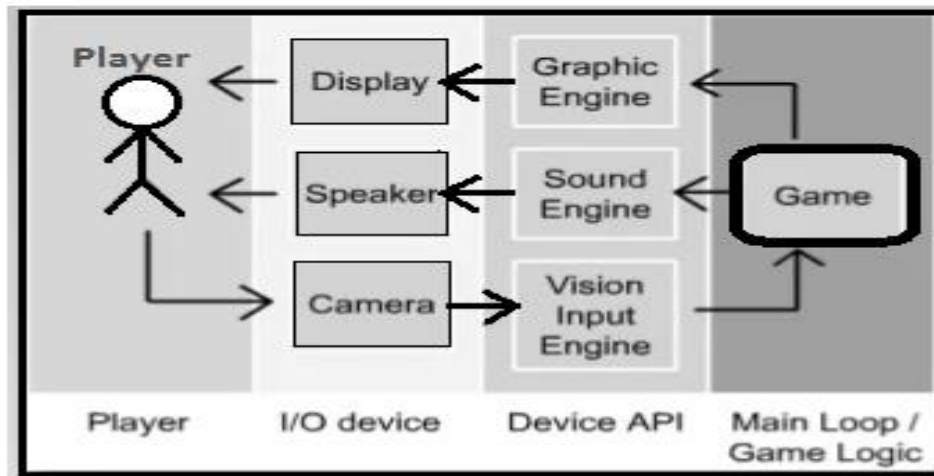
<https://www.baeldung.com/solid-principles>

Game Development Tools

- Game development tools are specialized software solutions which allow and facilitate the development of a video game easier.
- **Visual Studio** or **MonoDevelop** editor to inject specific code to move a game object, all these tools come under Game development tools.

Architecture of a Game (the built in ambiance and environment are known as Architecture of the Game. Or setting/mood created within the game.)

The architecture and structure of a game is similar to that of a software. But it does have some additional components which makes it different from a software. Every game has the following components:



Engines: are the self-processing component(s) of a complete process that are driving automatically behind the original process.

i. Graphics Engine

- A graphic engine is a software which in association with an application program helps to draw graphics on your computer's display device.
- Graphics engine helps to make your game's graphics better by increasing the resolution (Resolution measures the number of pixels in a digital image or display. W x H) and increasing the number of pixels (small dots that makes a n image) per unit area.
- This engine also makes your game's scenes clear and run smooth.

ii. Sound/Audio Engine

- The audio/sound engine is the component that consists of algorithms for dealing with sound and in-built programs are written into it to handle the sound effects embedded in the game.
- It has the capability to perform calculations using the CPU, or on any dedicated ASIC (Application Specific Integrated Circuit). Abstraction

APIs, such as Open-AL, SDL Audio, X-Audio 2, Web Audio, etc. can be available within this engine.

iii. Rendering and Vision-Input Engine

What is rendering? Render term is mainly used in CGI (Computer Generated Imagery). For example any 3D animation or 3D related software save the data in binary or ascii or related language.

- The data conversion of codes to visual images are called rendering.
- So this data conversion from codes to images takes a lot of calculation by rendering softwares.

The rendering engine **along with vision input system produces 3D animated graphics, using different techniques**, like rasterization and ray-tracing. Being programmed and compiled to be executed on any CPU or GPU straightforwardly, majority of rendering engines are developed upon one or more rendering APIs like Direct3D and/or OpenGL that offers a software abstraction layer for the Graphics Processing Unit (GPU).

Game Engine and History of Game Development

Game Engine **is like an integrated development environment, with a readymade suite of visual development tools and reusable software components**. It turns the complex task of game development simple, by providing an abstraction layer,

- **Game Engine** it is a framework that is designed specifically for the construction and development of video games.
- Developers use these game engines for creating games for consoles, mobile devices and personal computers.

Components of Game Engine

1. input

A game is nothing if it cannot be played, the game engine provides with support of array of input devices like mouse, gamepad, touch etc while also providing support for devices like gamepad, joysticks etc.

- two most commonly used are through: **events** and **polling**.
 - **Event** (like right click of mouse, or pressing arrow up key etc)
 - **Polling** (is used to get the position values) for example on which coordinates (x, y) does the mouse pointer is, or tilt angle of Game stick

2. Graphics:

Game engine provides a lot of **features like lighting effects, shadow, bump maps, blending animation** etc to make the imported asset look real.

Physics engines are software which allows performing fairly accurate simulation of most of the real-life physical systems like the movement of rigid body (we will perform that practically in later chapter using Unity 3D), soft body mass and velocity alteration and fluid dynamics, bounciness etc.

3. Physics (**real-time and real-life simulation must be portrayed.**)

Physics engines are software which allows performing **fairly accurate simulation of most of the real-life physical systems like the movement of rigid body** (we will perform that practically in later chapter using Unity 3D), soft body mass and velocity alteration and fluid dynamics, bounciness etc

4. Artificial Intelligence:

The implementation of AI in games is usually done using readymade scripts that are designed and written by software engineers who are specialized in AI. **For example:** How our character reacts on hitting a wall, or seeing an animal etc can be done easily by **building a tree of behaviour nodes, rather than writing complex code.**

5. Sound:

They provide a software abstraction of GPU using the multi-rendering API's like Direct3D

6. Networking:

Since a decade now, games support online multiplayer and social gaming, which connects your gaming adventures with your friends. Most of the gaming engines, provide complete support and scripts for such requirements, so you do not have to worry about TCP/UDP traffic, social API integrations etc

Brief History of Game Programming.

Game Programming Algorithms and Techniques: Overview

<https://www.informit.com/articles/article.aspx?p=2167437>

The five Best Game Engines for Beginners in Video Game Development

- A game engine is a framework that facilitates game development, using a range of tools designed for game development.
- Some game engines may call these tools nodes, others may call them APIs, but they all work together to enhance your game development experience.

1. Unity

- According to Unity Technologies, 71% of all mobile games use its engine, creating a total of five billion downloads per month of apps built with Unity.
- This game engine is perfect for beginners as it has a learning resource center that's equipped with tutorials, projects, courses, and live training workshops to get you from beginner to pro in no time.
- Unity provides 2D & 3D game development that's done in real-time. Several subscription options are available, but Unity Personal is the only free one. This

subscription type provides access to all the engine features, along with the cloud-based service ones.

- ✚ **The main drawback** to Unity Personal is that you can't remove Unity's splash logo from your game until you upgrade to one of the paid versions.

2. Unreal Engine

- Unreal Engine is a development engine **that prides itself on being one of the more advanced real-time 3D creation tools**. Yet it's great for beginners and you'll even find a learning resource section on its website.
- Unreal Engine **also creates other real-time applications such as architectural visualization, automotive visualization, and linear films**, among others
- **Unreal provides full access to its C++ source code**, giving game developers unfettered access to customize or even expand the Unreal Engine code.
- **It has an Unreal Editor tool**, which is a development environment that allows collaboration of all members in a game development team. **It does this through multi-user editing**. Like all the top game engines, Unreal Engine provides some high-quality world-building, animation, rendering, and gameplay features.

3. Godot

- **Godot is a free-to-use open-source game engine**. Godot's main selling point is that it uses common tools, which makes it ideal for beginners.
- **This engine has a visual editor** that allows you to create both 3D and 2D UIs and enables game developers to do live editing.

GDScript (which is much like python)

C#

C++

Visual script

Rust

Nim

D

4. Defold

- the ultimate game engine for web and mobile
- It has a Visual editor and code editor that allows game developers to write their game logic in the Lua scripting language.

Why is Unity a good platform and most preferred?

1. **Unity is a multi-platform environment** and works great for creating cross-platform AR/VR applications, mobile games, console games, due to the possibility to create 3D graphics with a relatively small amount of work in a very easy way without extensive programming knowledge needed...good web plug in
2. **2D, 3D, VR and AR**
 - Unity holds the supremacy for AR and VR as well, with over 60% of the developed content.
3. **Less coding:**
 - The fact that you don't need to have a great deal of coding knowledge or experience, **Unity uses C#, JavaScript and Boo**, but you can also develop a game from start to finish without writing a single line of code.
4. **Strong community:**
 - Having such a large community, Unity developers can ask questions and find solutions to their issues by asking other members, all this in a friendly and supportive environment.

Best Programming Languages for Game Development

What is the best programming language for game development?

1. **C++**

C++ is a [powerful low-level language](#) built for machine readability. It's a challenging language to learn for beginners compared to languages like Python, but C++ is well worth the trouble for game developers looking to maximize their creativity.

C++ is the most popular language for creating game engines – the development environments where game programmers create and host their interactive worlds.

most popular engines use C++, including:

Unreal Engine

Lumberyard

Godot

Source

Cryengine

Why is C++ used in games?

- C++ is a popular video game programming language because it is fast, powerful, and flexible:
- It's a very fast language.

Gamers expect high performance and minimal to no delay in a quality game.

Unfortunately, video games are prone to high server load, which can lead to latency if the code converts too slowly. C++ converts quickly, even when multiple tasks run at the same time.

- It works in close proximity to hardware.

Video games rely on hardware for control functions. C++ operates closely to that hardware, so engines perform better with less chance of stalling.

- **It works cross-platform.**

C++ lets you release projects on multiple operating systems and gaming platforms. It features high-performance tools. C++ has a rich set of open-source libraries that take some of the legwork out of coding. There are C++ libraries for many different aspects of game development, from graphics design to debugging.

- **It's compatible with other game programming languages.**

C++ gives you the flexibility to incorporate languages like C# or JavaScript to improve particular aspects of a game, like graphics or audio.

2. C#

Where does C# show up in game development?

- C# is the default language for Unity 3D, a popular game engine. Unity was the engine of choice for 61% of developers in 2021, making it the top engine of the year. It's the powerhouse behind many popular mobile games, including Pokémon Go and Temple Run .
- Unity is also a popular game engine for VR games – 90% of all Samsung Gear and 53% of Oculus Rift VR games use Unity. If you want to create games in Unity or for VR, C# should be in your toolbox.
- C# is also a back-end language; in web-development contexts, developers use C# to communicate with the server rather than the user. When a player takes an action in a multiplayer game, C# tells the server how to interpret that action.

Many developers have used C# to create Xbox and other Microsoft games on the XNA Framework. Although Microsoft has retired XNA, C# remains compatible with its successor, MonoGame. It's also the scripting language behind many Playstation console games.

3. JavaScript

A leading front-end web development language, JavaScript is primarily used in computer programming to add interactivity to web pages. If you use a drop-down menu or click to play a video, JavaScript usually powers that action.

How is JavaScript used in game development?

JavaScript-powered HTML5 games run in web browsers and on mobile platforms like iOS and Android. You can host these games directly on a website, so they're easier to share than console games. They're also easy to maintain, thanks to the built-in stability of the JavaScript language.

What game engines use JavaScript?

JavaScript game developers have access to JavaScript-specific and compatible engines, such as:

Three.js

Famo.us

Goo Engine

Allegro

Impact.js

Because JavaScript is such a popular web development language, it has a diverse set of tools and frameworks, including the cross-platform graphics library WebGL and HTML5 game framework Phaser.

4. Java

Java is a versatile language commonly used by small game development companies. It's powered some of the world's top iOS and Android games including Minecraft and Mission Impossible III. Plus, Java is a cross-platform language, so it runs on nearly any system including Microsoft and Linux.

Why learn Java for game development?

Java is a beginner-friendly language that is popular for mobile apps and games.

What are Java's game development features?

- The **programming language supports multithreading**, a feature that lets two or more instruction sets run at the same time. Game developers often prefer to have separate threads for different types of tasks, such as graphics rendering and gameplay logic.
- **Java also offers socket programming**, which powers two-way communications with servers. This makes it easier for developers to create multiplayer games without the need for additional tools.

Summary

If you want to create games from scratch, C++ is the way to go. It offers:

- i. More control over hardware.
- ii. Customization options for rendering graphics.
- iii. Better memory management.

If you want to script games using existing frameworks and engines, go with C#.

You'll get to:

- i. Learn the basics faster and start coding games sooner.
- ii. Code within popular engines like Unity.
- iii. Receive error warnings if something's wrong, so you learn quicker and see more success.