AKSHAY R

GAME DEVELOPER

Mob: +91-7012847510 Email: akshay.renjith.1@gmail.com

Portfolio
LinkedIn
GitHub

SKILLS

Programming/ Language : C++, C#
Game Engine : Unity
Version Control : Git

Data Structures and Algorithms (Fundamentals)
Design Patterns: MVC, Singleton, Observer Pattern, State-Machine, Object Pooling, Service Locator

EDUCATION

Bachelor of Technology (CSE) YOP - 2021
APJ Abdul Kalam Technological University (KTU), College of Engineering, Kalloopara

Class 12 : Science (Computer) YOP - 2016
DHSE Kerala Govt., Nair Samajam Higher Secondary School (NSHSS), Mannar

EXPERIENCE

Full Stack Game Developer – Apprentice (Outscal Pvt Ltd.) Apr 2023 - Present

- Learned and worked on different projects and assignments using Unity, C++, Data Structures etc.

PROJECTS

➢ **SNAKE CLASSIC 2D (Unity, C#)** | GitHub Video Play

- A classic snake 2D game with mode selections of Single Player and Co-Op Mode using Unity and C#
- Implemented programming concepts such as Singleton, OOPS, Coroutine, etc.
- Learned and implemented tools like Cinemachine, Post-Processing components, etc.
- Snake growth, shrink, movement, wall-warping, food and power-up spawning, etc were implemented.
- Main lobby, Mode selection, Game over screen changes and Score UI were implemented.

➢ **MINESWEEPER-GAME-CLONE (C++)** | GitHub Play

- A clone console version of the classic Minesweeper game using C++ and Data Structure concepts.
- Code optimization and structuring used to create a smooth flow of commands and its execution.
- Use of OOPs concepts as well as Data Structure concepts such as queues, 2D array, vector, etc.
- It has a winning condition & losing condition, as well as difficulty levels of different board sizes.

➢ **HUNTER ASSASSIN CLONE (Unity, C#)** | GitHub Play

- A 2D stealth action game with 4 levels, each level having different enemies with different abilities/states.
- Scriptable Objects to store object data for bullet, player, enemy, levels, etc.
- Implemented MVC, Observer Pattern and Service Locator design patterns to optimize, decouple and structure the code architecture.
- State Machine pattern used for different enemy states and the proper switching between them.
- Use of proper enum scripts, interface scripts, etc according to the game's need.

➢ **INVENTORY SHOP SYSTEM (Unity, C#)** | GitHub Play

- An Inventory - Shop system simulation to buy/sell/gather resources or items.
- Used Scriptable Objects, enum scripts, interface scripts, etc to structure/ organize data.
- Implemented Service Locator and Dependency Injection to decouple, centralize and manage services.
- Implemented Observer Pattern to observe, invoke and manage events efficiently.

➢ **CHEST SYSTEM (Unity, C#)** | GitHub Play

- A Chest System simulation to spawn, queue and unlock different types of chests, to collect different rewards.
- Implemented Observer Pattern to observe, invoke and manage events efficiently.
- Implemented MVC architecture and Object Pooling to manage and reuse chest types efficiently.
- State Machine pattern used for switching between Locked, Queued, Unlocking and Unlocked state of chests.
- Implemented Scriptable objects for managing chest data, Service scripts for chest queuing, UI, Sound, etc and proper use of enum scripts also.