

Yogesh Krishnan S

[Portfolio](#)
[LinkedIn](#)
[Github](#)

Mob: +91-8921430367

Email: yogeshsuresh28@gmail.com

SKILLS

Programming : C++, C#, Python

Data Structures and Algorithms

Game Engine : Unity

Design Patterns: Model-View-Controller, Singleton, Observer Pattern,

Version Control : Git

State-Machine Pattern, Object Pooling

ACADEMIC QUALIFICATION

Bachelor of Technology (Mechanical Engineering) [CGPA: 9.69]

Aug 2017 - Jul 2021

APJ Abdul Kalam Technological University (KTU), M.A College of Engineering

Class 12: Science (Computer) [98 %]

Jul 2015 - May 2017

DHSE Kerala Govt., St. Michael's HSS

EXPERIENCE

Full Stack Game Developer – Apprentice (Outscal Pvt Ltd.)

Sep 2022 - Present

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

Product and Operations Intern (Outscal Pvt Ltd.)

Mar 2023 – Jun 2023

- Worked on implementing improvements on Outscal's learning materials. [HTML, CSS]
- Performed testing and maintaining automations and features. [Airtable, Zapier]

Assistant System Engineer (Tata Consultancy Services Ltd.)

Aug 2021 – Jan 2023

- Worked in IT application support in service activities which help leverage the business activities of client.
- Incident, Change and Problem ticket management and Daily monitoring and reporting activities.
- Handled critical incidents efficiently involving different stakeholders and third-party vendors.

Applications handled:

- Kenexa Recruitment Systems: an application tracking system which helps manage hiring process.
- Salesforce: worked in Salesforce Administrator role involving user management, debugging existing code, data security and managing different orgs and sandboxes.

PROJECTS

➤ Robo Brawl (Unity, C#) | [github](#)

- A single-player 3D shooter game where player needs to kill a boss enemy with limited time to win.
- Model-View-Controller(MVC-S) architecture used for player and enemies.
- Enemy AI using unity's Nav Mesh and State Machine pattern for Patrol, Chase and Attack states.
- Object Pooling pattern used for bullets during shooting.
- Observer Pattern for events like game start, game over etc.

➤ Make New Way (Unity, C#) | [github](#)

- 3D puzzle game with 10 levels.
- Model-View-Controller(MVC-S) architecture used for levels.
- Undo functionality implemented using stack.
- Locked levels and unlocking system.

➤ Chest System (Unity, C#) | [github](#)

- A chest unlocking system where different chests are placed in available slots for unlocking.
- Scriptable Objects to implement different type of chest and associated rarity with different rewards.
- Model-View-Controller(MVC-S) architecture used for chests.
- State Machine pattern used for Locked, Unlocking and Unlocked state of chests.
- A centralized Event Service with Event Controller classes used for Observer Pattern.

➤ Cowardly Hunter (Unity, C#) | [github](#)

- A side scrolling infinite runner game, scrolling background implemented with parallax effect.
- Collectibles and obstacles keep spawning in run time.
- Several conditions for win and loss employed.
- Player detection and alert circle around enemy mechanic implemented.