Jayesh Singh

jayeshsingh62@gmail.com jayeshsingh62.github.io github.com/jayeshsingh62 +1-5852902023

Education

- **Rochester Institute of Technology** — Rochester, New York
  MS - Game Design and Development ; GPA: 4.0 (Current) — *2024 - 2026*
  Courses: *Game Dev Processes, Game Design, AI Programming, Game Graphics Programming, Game Engine Development*
- **Bharati Vidyapeeth's College Of Engineering, GGSIPU** — Delhi, India
  Bachelor of Technology - Information Technology; GPA: 8.37 — *2018 - 2022*

Experience

- **FanClash11 Fantasy Private Limited** — Remote
  Junior Intern Developer — *Oct 2021 - Nov 2021*
  - **Documentation**: Authored a technical design brief for a new mobile title, outlining core mechanics, engine requirements, and performance targets to align the development team.
  - **Menu Prototype**: Engineered a responsive UI and menu system using Unity's Scene Management, optimizing asset loading to achieve sub-10-second level transitions on target mobile devices.
  - **Gameplay**: Developed core gameplay mechanics in C# for player character, profiling and optimizing scripts to maintain a consistent 30 FPS on mobile hardware.
- **A3ED** — Remote
  Freelancer — *Mar 2023 - Jan 2024*
  - **Bulldozer in VR**: Developed an interactive VR training simulation in Unity using the Oculus SDK, allowing users to manipulate, inspect, and disassemble a complex 3D bulldozer model with XR Ray Interactors.
  - **Company Showcase Stand VR**: Built a cross-platform virtual tradeshow booth using Unity and WebXR, enabling users to interact with product showcases and media in a web browser without requiring a native application.
  - **AR Animals**: Created an AR project in Unity powered by Google's ARCore. The users would be able to see animals in AR at selected ares chosen in real world locations with 3D maps with the help of Cesium combined with newly introduced Geospatial from Google.

Projects

- **Project: Brawler (Unreal Engine 5)**:
  - **Locomotion**: Engineered a versatile 3D character controller in C++ with locomotion features including walking, sprinting, crouching, and dodging, integrated with a **Blueprint Animation State Machine** for fluid movement.
  - **Combat System**: Architected a C++ based, data-driven **melee combat system** supporting multi-stage punch and kick combos, handling input buffering and sequence validation to ensure responsive attacks.
  - **Animation System**: Implemented dynamic gameplay animations by triggering **Animation Montages** from C++ for evasive dodges and context-sensitive enemy hit reactions, providing clear and impactful combat feedback.
- **Level**: Made a level to demonstrate different features like animations with the help of Unreal Engine 4's blueprint system using **State Machines** to set up various animations and creating transitions between them. Added **Spatial Audio** which is audible upto 10 metres and implemented **Level Streaming** to **improve performance by 15%.** (Unreal Engine 4)
- **Drag Race & Feed the animals**: Made a **split screen local multiplayer** level in which players get to drive a car, there are oncoming vehicles and obstacles on the road and the camera view can be changed from 3rd person view to driver view. Used **physics** to make the car accelerate and decelerate accurately. In feed the animals, made a top-down view game with animals coming towards the player from the top and shoot steak to feed them. Used **Object Pooling** for managing large number of objects for optimization and **increase the performance by 30%.** (Unity)
- **2D Runner & Knock the balls**: Made a 2D endless runner game with obstacles coming towards the player from the right which are destroyed once they leave the camera view for **performance optimization** and jump over them to keep the game going. Background music, sound effects and particle effects are added for more immersion. In Knock the balls, made a prototype containing a sphere to control and knock enemy spheres off the ground using **physics**. Used Enum for storing different types of powers. (Unity)
- **Bomberman Prototype**: Recreated the classic Bomberman gameplay, implementing key features such as **grid-based movement**, a bomb-laying system with timed explosions and **destructible environments** (Unity)
- **Endless Runner 3D**: Made a 3D Endless Runner having the necessary functions. The scripting is done in **blueprint** and **inheritance** is used for obstacles for easier configuration of individual obstacle while keeping the base properties of their parent. (Unreal Engine 5)
- **Shards of Kalia**: Worked as the **AI programmer** on a **2D local coop** game with magic abilities where I designed and implemented enemy behaviors using **State Machines** with some work on UI and sound effects too. (Unity)

Skills Summary

- **Programming Languages:** C++, C#, C
- **Engines & APIs:** Unreal Engine 5, Unreal Engine 4, Unity Engine, OpenGL
- **Tools & Technologies:** Git, Visual Studio, VR (Oculus SDK, OpenXR), AR (ARCore, Geospatial API)
- **Concepts:** Object-Oriented Programming, Gameplay Systems Design, AI Programming, Performance Optimization, UI/UX Implementation