Sourabh Mittal

Game Programmer/Software Engineer

A-67 Radhey Shyam Park Extn. Near Parwana Road, Delhi - 110051
+91 9354642395
mittalsourabh@outlook.com

PROJECTS

Unreal Engine Catch The Flag Game— Simple Multiplayer Game

- Players compete in a match to capture the enemy flag and bring it to their base.
- Players have limited health, drop the enemy flag upon dying and then respawn.
- Players return their flag to the base to keep it away from the enemy.
- Players create and join servers via Main Menu.
- Upon creating or joining a server, players travel to lobby until the session is full.
- When the session is full, all the players are transported to the Gameplay where the match begins.
- Set up an Online Subsystem for the management of game sessions.
- Designed for flexibility for more than two players on multiplayer.
- Designed UI for Main Menu and Gameplay.
- Wrote the logic for Multiplayer Gameplay from scratch.

Unreal Engine AI— Vehicle Path following with Traffic System for Games

- Laid down a road track using splines.
- Extensible to work for any shape, size and form of track.
- Implemented Vehicle AI for following spline paths.
- Wrote logic for the Traffic Light system.
- Vehicle can randomly choose any of the three directions ahead at the intersection.
- Implemented UI for fuel system and drifting score.
- Extensible approach where you can add as many tacks and traffic lights you like.

Unreal Engine Game Mechanics— Modified ShooterGame Sample for Multiplayer Environment

- Implemented Several Game Mechanics in Unreal Engine C++ using Advanced Character Movement Component for Massive Networked Environments.
- Tested with Network packet lag of 500ms and variance of 30%.
- Teleport functionality to teleport players 10m ahead.
- Jetpack functionality in game. Jetpack recharges when not in use.
- Time Rewind functionality which rewinds players back in time. Similar to Overwatch and Prince of Persia.
- Shrink Gun shrinks player upon hit. If the player is stomped when shrunk then he dies. After a fixed time, the player automatically unshrinks.
- Freeze Gun freezes players for a fixed amount of time.
- Wall Run functionality allows the player to run on walls. Similar to Overwatch and Prince of Persia.
- Wall Jump functionality allows the player to jump against a wall when flying close to it.
- Drop Weapon where the player drops its weapon upon dying which may either be picked up or disappears after a fixed time.
- All these functionalities have been implemented to work in a massive multiplayer environment using a server-authoritative model.

SKILLS

C++ , Python

Unreal Engine

Artificial Intelligence

Game Mechanics

Data Structures and Algorithms

PORTFOLIO LINK

https://sourabhgo.github.io

GITHUB PROFILE

github.com/sourabhgo

Simple Soccer AI in C++ — *Simplified Soccer Game with AI Bots*

- State Driven Agent Design using Finite State Machines with Messaging capabilities.
- Steering Behaviors such as seek, flee, arrive, pursuit, evade, wander, obstacle avoidance, wall avoidance, interpose, hide, path following, offset pursuit.
- Group Behaviors such as separation, alignment, cohesion and flocking.
- Combining steering behaviors using various techniques, ensuring non-penetration, implementing spatial partitioning for managing crowds and smoothing to reduce jitter.
- Implemented key sports game techniques using strategic decision making at agent level and team level.

Raven Shooter Game — *Full-fledged Shooter with AI Bots*

- Graph algorithms such as depth-first search, breadth-first search, dijkstra's algorithm and A-star algorithm for navigating game maps.
- Fully functional navigation system using map loading.
- Practical path planning for navigating game maps.
- Weapon system with selection using fuzzy logic.
- Goal driven agent behavior.
- Used Lua scripts to load settings.

SpaceWar — *2D Game written in C++ using DirectX*

- Wrote a complete 2D Game Engine from scratch.
- Included Input support for keyboard and Xbox game controllers.
- Built 2D Graphics Engine for rendering Sprites with Textures.
- Incorporated 2D Physics Engine for collision detection among entities in game.
- Added Audio functionality using XACT.
- Programmed Text functionality with support for both sprite based text as well as directx font based text.
- Implemented console functionality within the game.
- Extended the SpaceWar game with networking functionalities in Client and Server side versions of the game.

EDUCATION

University School of Information and Communication Technology, Delhi — *Bachelor of Technology in Information Technology*

AUG 2016 - AUG 2020