Akshay Patel
Blockchain Developer | Game Programmer

Canada | (438) 389-8043 | <u>Email</u> | <u>LinkedIn</u> | <u>Portfolio</u> | <u>GitHub</u>

Profile

I am a blockchain developer with professional experience in front-end and back-end development, as well as designing, testing, deploying of smart contracts, NFTs, cryptocurrencies, and DeFi apps. I also have experience with the Unity game engine and I am knowledgeable in different parts of game development, algorithms, architecture, and game engine tool creation. I've worked on multiplayer, hyper-casual, action, and arcade games in the past.

Blockchain Developer

Plutonium, Montreal, Qc **/** Nov 2021 **–** Present

· Worked on crypto trading website to which allows users to trade their cryptocurrency with each other.

· Designed and developed smart contracts using the Solidity and implemented test scripts on the smart contracts using the truffle framework.

· Optimized the smart contract code and increased the efficiency by 15%, using the library functionalities of the solidity.

· Successfully reached the phase-2 of the Project.

Skills

· **Programming Languages:** Solidity, C#, C++, java, JavaScript, TypeScript, HTML, CSS, JSON.

· **JavaScript frameworks:** React, Node, Chai, Mocha.

· **Ethereum Tools:** Truffle, Hardhat, Ganache CLI, Ethereum Virtual Machine, MetaMask.

· **Technologies:** Web3, Ethereum, Solana, Smart contract, NFT, Crypto Token, DeFi.

· **Database:** MySQL

· **Unity:** Editor Variables/Menus, UI, Physics, Animations, Sound, Particle Systems, Controller Support, Nav Mesh, Materials and Lighting, Shader Development, Rewired, Unity Input System, Local Multiplayer Support, Pro Builder.

· **IDE and Other tools:** Visual Studio, VS Code, Eclipse, Unity Engine, IntelliJ Idea, PyCharm, GitHub desktop, Davinci Studio, Photoshop.

· **Project Management:** Agile Scrum, Jira, Trello, Git, GitHub.

Projects

Earth ERC20 Token (Solidity, Ethereum Blockchain, Ganache, Truffle, React.js)

· Earth Token is a social trading token that can be used to make transactions with an Ethereum crypto wallet like **MetaMask**.

· I have used **Ganache-cli, Truffle Framework**, and **Solidity** to develop, deploy, and test my smart contracts.

· I've also built a Clint side website using **React Js** so user can quickly purchase and sell crypto currency.

· *Source Code*

Red Moon Token (RMT) (Go, Solana Blockchain, Smart Contracts)

· The Red Moon Token (RMT) is a trading token based on the **Solana blockchain** with a supply limit of 100,000,000 RMT.

· Using a crypto wallet like **Phantom Wallet**, you can transfer and receive RMT tokens.

· To create this, I have forked an official Solana-labs repository from **GitHub** and then deployed RMT tokens on the Solana blockchain using smart contracts.

· *Source Code*

Crypto Women NFT (JavaScript, Ethereum Blockchain, Smart Contracts)

· The Crypto Women is my personal project that I created using JavaScript and Smart Contracts.

· To make each **NFT unique**, I have used **10 different properties** with varying rarities.

· In the future, I intend to integrate these NFT in a game.

· *Source Code*

Plutonium Exchange (Smart Contracts, Solidity, Hardhat, Waffle, Chai, React, Tailwind CSS)

· Plutonium Exchange is a **crypto trading platform** where user can search about latest crypto trends and **trade** various crypto currency with each other.

· To create back-end we have used Solidity to write smart contracts and used **Hardhat** and **Waffle framework** to test and deploy smart contracts.

· To create front-end we have used HTML, React Js and **Tailwind CSS**.

· *Source Code*

Arrow Maniac | PC (Unity, Shaders, Rewired, Unity Particle System, Singleton Pattern)

· Arrow Maniac is a fast-paced local multiplayer **2D platformer game**. In this game, you fight with a bow and arrow, and your goal is to eliminate the other players before they eliminates you.

· In this game I worked as Lead Level Designer and **Gameplay programmer** to create various features using Unity game engine.

· I used **Shaders**, Unity particle system, Tile map, Time Manager, 2D Animation, and other tools to construct all of the game's maps, abilities, and game UI.

· *Source Code*

Space Bomber | PC (Unity, Unity Particle System, Singleton Pattern, Mixamo)
· Space Bomber is a **3D top-down** wave survival game.
· In this game, the player must survive against hordes of enemies while killing them as quickly as possible in order to obtain the highest possible score.
· My responsibilities in this game were to develop the **particle effects**, levels, the ammo system, the **wave system**, the **game UI**, and three different types of bombs.
· *Source Code*

Flocking AI | PC (Unity, 3D, Shaders, Flocking AI, C#)
· A **3D unity AI project** to demonstrate flocking AI behavior in fishes.
· To achieve realistic looking flock behavior, I created a **Predator and Prey relationship** between all types of fishes.
· Big Fishes (**Predator**) will try to hunt down small fishes (**Prey**) and small fishes will try to avoid them while staying as a group and searching for fish food.
· Flocking AI is built on a **highly scalable architecture** that allows for an endless number of flocks to be supported. In addition, each flock is maintained by nine additional factors.
· *Source Code*

Unity Editor Tools | PC (Unity Editor Scripting, C#, Scriptable Assets, Generic Class Generation)
· The **Color Editor tool** that helps to color the texture and then later on that texture can be applied on different 3D objects.
· I also created one **Object control tool** using which you can add, remove, duplicate and change position of the objects in array directly from unity inspector.
· *Source Code*

Education

Video Game Programming | Jan 2020 – Nov 2021
ISI, L'institut Supérieur d'Informatique, Montreal, QC

Bachelors in Information Technology | Aug 2015 – Jun
2019 Gujarat Technological University, Gujarat, India