

S. Changfang

Senior Gameplay Programmer

SuphawichCBusiness@gmail.com

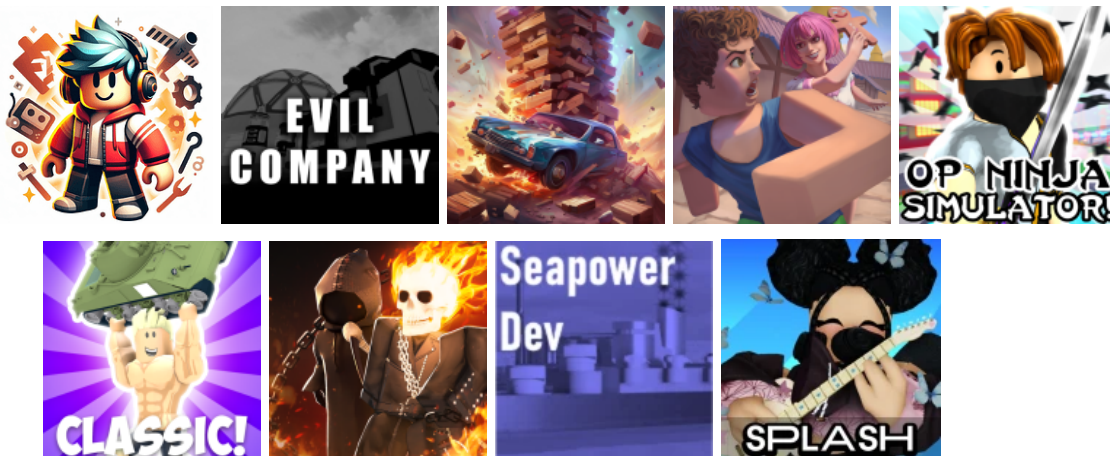
+1 (951) 366-9909

Menifee, CA, United States

About Me

Specializing in Roblox game development and programming, as a software engineer with almost 9 years of experience on the platform, I started in 2016, with a **jack-of-all-trades** approach while focused on programming. Throughout my career, I've contributed to many popular Roblox games, collaborating within diverse teams to ensure polished, user-friendly, engaging, and most importantly, fun experiences. Committed to my personal growth, I am constantly seeking to expand my skills and knowledge. I thrive in collaborative environments where mutual learning, support, and teamwork are valued, prioritizing staying up-to-date with new technologies. I write my own custom tools to speed up and automate my workflow. I love learning more about computers and diving deep to strengthen my understanding of computers. **My ambition is to have a stable career that I love while working on amazing games that brighten our players' days.**

Experience



[2024 JUN - 2025 JUN Gameplay Programmer at Splash \(Contract\)](#)

<https://www.roblox.com/games/4936591712/Splash-WORLDS-DJ-Build-Skate>

In *Splash*, players are tasked with Skating, Mixing music, and Building their own club

My most notable accomplishments:

- Run-time Package importing system
- KaiMix Dialogue and UI
- Tag system for Builders to attach behavior to 3D objects (Hover, Rotate, RGB, etc)
- Skate Gym (Tutorial and Mini-game)

2024 JUN - 2024 SEP Gameplay Programmer at Sea Power (Contract)

In *Sea Power*, players are tasked with capturing control points around the map in Battleships

My most notable accomplishments:

- Round System; starts the round and ends the round with conditions based on players left, ran out of time or Team HP reached 0
- Capture Point System
- Collision and Hitbox filtering (Friendly fire enabled, does more damage to enemies)

[2023 NOV - 2024 JAN Gameplay Programmer at Evil Company \(Contract\)](#)

<https://www.roblox.com/games/14034950224/Evil-Company>

In *Evil Company*, players are tasked with resource scavenging, survival, and successful expedition completion on an abandoned moon base.

My most notable accomplishments:

- Daily Rewards UI providing rewards for daily check-ins after set intervals
- Scanner UI highlights valuable scraps within a range, with information of their name and worth

- Custom Inventory / Hotbar system allowing players to seamlessly pick up and drop items

[2023 JUN -2023 OCT Programmer at Block Buster \(Contract\)](#)

<https://www.roblox.com/games/13744954060/Block-Buster>

In *Block Buster*, a player is chosen at random to become the Destroyer, using a slingshot to launch various projectiles at a tower of Jenga-style stacking blocks, while the Survivors must skillfully run, dodge and hide to survive each round.

My most notable accomplishments:

- A precise slingshot aiming and power adjustment mechanism for Mouse, Keyboard, and **Touch input**
- Designing a round system that manages intermission, round start logic, round end logic, and a countdown timer
- Establishing a system for collecting player feedback and channeling it to Discord

[2020 MAR - 2023 JAN Programmer at Time Attack!](#)

<https://www.roblox.com/games/11364672248/Time-Attack>

In *Time Attack!*, hidere use perks to outwit seekers, annoy them with emotes and survive as long as they can. While seekers can use perks to help them catch hidere, stun them with their selected weapon of choice and finish them off with satisfying animations.

My accomplishments:

- all user interface, such as the HUD, inventory, store, settings, etc
- all gameplay, such as hit direction, finishers, perks, emotes, etc
- all sound and visual effects
- all of the server-side systems, such as saving via data stores, handling robux purchases, the main round loop, etc
- all of the internal utilities, such as our networking module, character spawner, player join and leaver wrappers, etc
- **placing and implement all of the assets from artists, modelers, and designers into the game**
- **polishing and tweaking all aspects of the game to make it feel great**
- Sound Sprite System: one audio file with a collection of sound effects
- UI input detection library which handles mouse and touch events more elegantly than the built-in events do

[2019 NOV -2020 JAN Programmer at OP Ninja Simulator \(111 million visits\)](#)

<https://www.roblox.com/games/4225025295/OP-NINJA-SIMULATOR>

In *OP Ninja Simulator*, players train to gain stats and reach milestones to progress further into the game and become the strongest ninja of either Yin or Yang clan.

My most notable accomplishments:

- An "honor" stat which would be awarded or deducted according to kills and deaths
- Effects, animations, hit detection, and critical hit calculations for both melee and ranged weapons
- Filters and sanity checks for cheat prevention as part of weapon hit detection

[2019 JUN -2020 JAN Programmer at Lifting Simulator \(553 million visits\)](#)

<https://www.roblox.com/games/3652625463/Lifting-Simulator>

In *Lifting Simulator*, players train to gain muscle mass, sell it and buy upgrades to become the strongest bodybuilder and dominate all players in the game.

My most notable accomplishments:

- **Implemented game-wide leaderboards for the top 1000 players using data stores**
- Tracked player stats as they played to align them with Hero or Villain teams
- Implemented consumable products and timed products
- Designed a system that allows players to locate and connect to another player's server instance when they are listed on the global leaderboard

[2018 SEP -2019 JAN Programmer at Super Power Training Simulator \(247 million visits\)](#)

<https://www.roblox.com/games/2202352383/SPTS>

In *Super Power Training Simulator*, players *increase stats through training* and unlock abilities through quests, with the ultimate goal of becoming the most powerful superhero or supervillain in the game.




My most notable accomplishments:

- Tied into the default chat system to implement private server commands such as banning and toggling PvP

- Handled user controls and camera manipulation to create satisfying special attacks
- Implemented developer product handling as part of game monetization and item gifting

Showcase

Youtube

-  Cellular Automata (Conway's Game of Life) Showcase
- [Whack a mole game - YouTube](#)
- Simon game
- [Dictionary showcase - YouTube](#)
-  Hoverboard and Triangle Terrain Showcase
-  Blackhole showcase

Custom Programs

- Automatic PNG Compressor using TinyPNG api
- Base64 Image to PNG
- Convert RAW image files to JPEG
- Automatic TV Show Downloader

Custom Libraries

- GUI Handler Module
- Custom Networking Module
- Custom Signal Module

Demo

- [Floating Orbs - Live demo](#)
<https://www.roblox.com/games/start?launchData=FloatingOrbs&placeId=15932309201>
- [Shockwaves - Live demo](#)
<https://www.roblox.com/games/start?launchData=Shockwaves&placeId=15932309201>
- [Black hole Simulation using Parallel Lua - Showcased at 2022 RDC](#) (Place file available upon request)

Programming Languages

- Proficient with **Luau**/Lua
- Intermediate with C, GDScript

- Learning Zig, Rust

Skills

- Tool set: [TestEz](#), [Promise](#), Maid / [Janitor](#)
- **Test Driven Development and Quality Assurance**
- Object oriented programming, Functional programming, Procedural programming, Entity Component System, Event driven programming, Modular design
- Game programming patterns, architecture and designs
- Github actions (Cloud deployment and automation)

Roblox Specific Skills

- [Server - Client network model](#), [Network Ownership](#), Server-side authority, user input validation, Unreliable Remotes
- [CFrame math](#), Vector math, [Raycasting](#), Sphere and Block Cast, Shape Cast, CSG, Physics, [Inverse kinematic](#), Aerodynamics, Fluid forces
- User interface, User input, Camera, Character Control, Drag Detector
- **Touch device support and Low end device support** through [StreamingEnabled](#), [Level of detail](#)
- [Parallel Luau](#), [Type checking](#), Asynchronous programming, Deferred Engine, Luau buffer types, Data compression, Data serialization
- Cloud services; [Datastore](#), [Ordered datastore](#) [Memory stores](#), [Cross-Server Messaging](#), Cloud API
- [Mechanical Constraints](#), [Mover Constraints](#)
- Animation, Audio API, Visual effects, EditableMesh, EditableImage
- Voxel Terrain and procedural generation
- Pathfinding, State machine and Decision trees, NPC Vision
- Monetization through Gamepass, Developer product, Subscription, Premium, Immersive Ads
- HttpService, Social Invites, Notification, TeleportService
- Plugin development and developer tools
- [Roblox Cloud API](#)

Software

- Proficient with **Roblox Studio**, VS Code, Rojo, Zune/Lune, Git
- Intermediate with Audacity, Paint.NET, Photopea, Capcut
- Learning Löve, Godot, OpenGL, Raylib

Interests

Computer Science, Software Engineering, Game Development, **Game Design**, Psychology, User Experience, Photography, Cooking, Physics

Favorite Games

- **Steam:** **Oxygen Not Included**, Don't Starve Together, Subnautica, Muse Dash, Animal Crossing, Apex Legends, Potion Craft, Cult of the Lamb, Palworld, Balatro, Schedule 1, Repo, MiSide
- **Roblox:** Lumber Tycoon 2, **Bee Swarm Simulator**, Before the Dawn, Bed Wars, Flee the Facility, Violence District

Professional References

- **Matt Marcin:** RaidTheory, matt@raidtheory.com
- **Jacob Robinson:** Skyreacher, jacob@skyreacher.com
- **Grey:** Stone Acid, greyrushlight@gmail.com
- ★ **Nathaniel:** Splash Music, nathaniel@splashmusic.com

Education

High School Diploma, Yothinburana (2013-2016)