



## SHANTANU SHRIPAD MANE - SOFTWARE ENGINEER GAMEPLAY - HEARTHSTONE

Phone No.: +1-385-202-9752 | Email: [shantanu.m934@gmail.com](mailto:shantanu.m934@gmail.com)  
Portfolio: [shantanumane.com](http://shantanumane.com) | [linkedin.com/in/shantanusmane](https://linkedin.com/in/shantanusmane)

### EDUCATION

**University of Utah, Salt Lake City, USA - May 2019**

Secured a Master of Entertainment Arts & Engineering - Game Engineering Track with *GPA 3.88/4.00*

**K.J. Somaiya College of Engineering, Mumbai, India - June 2015**

Secured a Bachelor of Engineering in Computer Engineering with *First Class Honors*

### SKILLS

**Programming Languages** - C++, C#, Blueprints, Lua, GLSL

**Software** - Unreal Engine 4, Unity, OpenGL, Maya, MotionBuilder

**Soft Skills** - Iteration, Collaboration, Creative Problem Solving

**Computer Science** - 3D Math, Data Structures, Animation Programming, Memory & Cache, Algorithms, Code Optimization & Architecture

### WORK EXPERIENCE

**SIE Santa Monica Studio, Los Angeles, USA - Gameplay Engineer Intern - C++, C#, Lua - Jul '19 to Sep '19**

- ◆ Implemented a root-motion related animation tool feature to make viewing animations in game more convenient.
- ◆ Improved a combat collision system to perform more accurate shape intersection tests to better support designers' vision.
- ◆ Optimized a fact-checking system to keep certain types of fact buckets pre-sorted and sort other fact buckets only when necessary which saved 0.2 - 0.3 ms of frame time.

**Actually A Game Company, Salt Lake City, USA - Gameplay Engineer - C++, Blueprints, UE4 - Sep '18 to May '19**

*Hard Light Vector* - [Released on Steam](#) Mar '19 - [Portfolio Page](#), [Project website](#)

An action-adventure FPS game with your fast-paced traversal techniques as tools to conquer giant mechanical monsters.

- ◆ Implemented an action-elements system to control VFX and screen effects for flair & feedback based on player state.
- ◆ Worked on an interaction system to indicate and handle interacting with interactable elements near the player.
- ◆ Implemented the HUD and various UI elements to achieve a sci-fi feel and power fantasy.
- ◆ Implemented & iterated on the player character's 'Thrusters' that give a small upward boost when you are in-air.
- ◆ Contributed to player-side design to create a unique character and resonating abilities that make you feel fast and fierce.

### GAME PROJECTS

**Combo Attacks System Project - Gameplay Animation Engineer - C++, UE4 - Aug '18 to Present - [Portfolio Page](#)**

- ◆ Created a gameplay and animation system for chain attacks/combos based on input timing, animation events & branches which is robust enough to allow adding any number of combat moves by designers and chaining between them.
- ◆ Improved responsiveness by accepting next attack input before an attack finishes and later executing the 'Pending Attack'.

**Warlocks - Gameplay Engineer - C#, Unity - Aug '18 to Dec '18 - [Portfolio Page](#)**

A recreation of a MOBA-esque King-of-the-Hill PvP where you cast spells to fight and defeat other players.

- ◆ Created a controller system to switch input actions (select, move, target, cast) & handle character states for each action.
- ◆ Implemented movement status effects like Stun & Knock-back for spell interactions with characters.
- ◆ Implemented a well-rounded spell system with ability interactions & spell target types, levels, cast times & cooldowns.
- ◆ Created robust Unit Statistics, Damage and Status Effects systems and pipelines.
- ◆ Optimized spells' Game Object creation by instantiating into Object Pools before game start rather than during gameplay.
- ◆ Integrated network functionality for gameplay elements like movement, animation & spells for multiplayer mode.

**2D Collision System - Gameplay Engineer - C++ - Feb '18 to May '18 - [Portfolio Page](#)**

- ◆ Created the Collision & gameplay supporting systems for a 2D Game Engine and implemented Pong using it.
- ◆ Implemented the Swept Separating Axis Test for collision checks, and two types of responses to them - block & overlap.
- ◆ Optimized collision system by updating coordinate transformation matrices only for moveable objects, checking collision of only the ball with other objects & responding to only the earliest collision, capitalizing on the game world being sparse.
- ◆ Created libraries of 4x4 Matrix & Vector4 operations for transformations used primarily by collision system.