|  |  |
| --- | --- |
| **Shantanu Mane Logo - BW NoName** | **SHANTANU SHRIPAD MANE - GAMEPLAY PROGRAMMER**  **Phone No.:** +1-385-202-9752 | **Email:** [shantanu.m934@gmail.com](mailto:shantanu.m934@gmail.com)  **Portfolio:** [shantanumane.com](https://shantanumane.com/) | [linkedin.com/in/shantanusmane](https://www.linkedin.com/in/shantanusmane/) |

**EDUCATION**

**University of Utah** -*Expected Graduation - May 2019*

Pursuing a Masters in Entertainment Arts & Engineering - Game Engineering Track

**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  **Game Engines -** Unreal Engine 4, Unity  **IDEs -** Visual Studio 2015, Visual Studio 2017 | **Version Control** - Perforce, Git  **Animation -** Maya, Flash  **Software Documentation -** UML, Dia |

**GAME PROJECTS**

**2D Game Engine & Collision System** -*Gameplay Tech Programmer - C++* - Feb ’18 to May ’18 - PortfolioPageLink

* Created the framework & gameplay supporting features for a 2D Game Engine and implemented Pong using it.
* Implemented a collision system using the Swept Separating Axis Test for checks and two types of responses.
* 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 taking advantage of the game world being sparse.
* Created libraries of 4x4 Matrix & Vector4 operations for transformations used primarily by collision system.
* Created own messaging system using delegates and own implementation of Smart/Shared Pointers.

**Memory Manager** -*System Programmer - C++* - Oct ’18 to Dec ’18 - PortfolioPageLink

* Created a memory manager in C++ that passes a robust unit test.
* Created a Dynamic Size Heap Allocator to allocate memory of requested size from the reserved heap of memory.
* Implemented Fixed Size Allocators for certain allocation sizes that use arrays of bits to track their memory blocks.

**Combat System Project** - *Gameplay Programmer - UE4, C++* - Current Project Portfolio Page

A combat system similar to that of Bayonetta, focusing on player input and combat mechanics.

* Implemented a system for chain attacks/combos using a tree structure and improved responsiveness to input for attacks.
* Worked on an effective system for hit information of and reactions to attacks.

**Project Jericho** - *Gameplay Programmer - UE4, C++, Blueprints* - Current Project Portfolio Page, Project website

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

* Implemented the player character’s ‘Thrusters’ and the mechanics tied to it like a speed boost, its ‘fuel’, and camera work for high-adrenaline action-style gameplay.
* Contributed to player-side design to create a unique character and resonating empowering abilities that make the player feel elegant and fierce.

**MaVRick** - *Gameplay Programmer - UE4, Blueprints* - Published April 2018 on [Play Store](https://play.google.com/store/apps/details?id=com.WildWestWorkshop.MaVRickMobileArena) and [itch.io](https://teameetings.itch.io/mavrick)

An action game where you pinball and charge at enemies with your fists to send them flying out with an explosion.

* Implemented a spawn system allowing to create desired intensity in the game by tuning the difficulty of each set of spawned waves and the threshold to spawn every new wave.
* Worked single-handedly on the ‘Fighter’ enemy AI that blocks attacks from the front, needs to be stunned from behind before being able to take damage and can do a short-range charge at the player.
* Setup complete animation state machines for the ‘Fighter’ and ‘Shotgunner’ enemies.
* Designed player abilities and enemies to create intense and high-octane gameplay.

**Warlocks** - *Gameplay Programmer - Unity, C#* - Current Project Portfolio Page

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

* Worked on MOBA-style controls, unit selection, movement and spell-casting to be used with a mouse and keyboard.
* Implemented Object Pools for creating spells/abilities before the start of the game as opposed to on-demand creation during gameplay to eliminate the associated overhead.
* Implemented a well-rounded spell system with ability-specific interactions, spell-cast types, spell levels, cast times, and cooldowns and a robust damage system to tie into it.