TOMMY TANG

He/Him | Redmond WA, 98052 | (+1) 425-614-9579| Portfolio | tommy.tang@digipen.edu | Linkedin

Skill

- Languages and Engines: C/C++, C#, UNITY, UNREAL, OpenGL
- Tools: Git, SVN, Perforce, CI/CD, ImGui, WSL, Slack, RenderDoc, Visual Studio, Visual Studio Code.

Education

DIGIPEN INSTITUTE OF TECHNOLOGY **2020/9 - 2024/4**

BS in Computer Science in Real-Time Interactive Simulation *Redmond, WA*

Work Experience

TEACHING ASSISTANT **2022/9 – 2022/12**

DigiPen Institute of Technology

- Assisted students in answering questions about Algorithm/Data Structure assignments.
- Assisted students in doing labs of C/C++.

QUALITY ASSURANCE ANALYST **2019/10 - 2020/5**

Rayark Inc.

- Worked on a multi-region published game: Soul of Eden, that has 1m+ downloads on IOS/Android platforms.
- Implemented an automation tool to test daily quests and player tutorials, saving QA one hour of manual testing per day.

UNITY SOFTWARE ENGINEER **2018/10 - 2019/5**

So-cayenne Entertainment

- Implemented a time zone system in Unity for RENKA, a mobile game published in multiple regions, which allows in-house designers to effortlessly schedule and publish game events across different time zones.
- Implemented a CI (Continuous Integration) environment on Gitlab to help the team check daily build stability.

Projects

GAME PROGRAMMER, UNITY

Sumo Spinning Top, solo project.

- Implemented whole gameplay with Nintendo Switch Unity Dev tools on Nintendo Switch
- Supports four-player gameplay and motion control with Joy-Con.

AI AND SOUND PROGRAMMER, UNITY

Hidden World, 3-people team project.

- Developed a 3D tech demo in Unity for procedural content generation. Utilized the Backtracking technique to generate a maze procedurally.
- Conducted personal research to evaluate the applicability of Wave Function Collapse to maze generation.
- Constricted a tool that selectively integrates appropriate sound effects to enhance natural auditory experience.

ENGINE PROGRAMMER, C++ CUSTOM ENGINE AND UNITY

Game AI Projects, solo project.

- Implemented an advanced behavior tree with decision-making via a utility system.
- Implemented A* Pathfinding: using smoothing and rubber banding algorithm to make the path more natural.
- Implemented Terrain Analysis, Occupancy Map, Influence Map, Visibility Map, Search, and Propagation Function which can be widely applied to various 2D top-down games.

PHYSICS AND GAMEPLAY PROGRAMMER, C++ CUSTOM ENGINE

Split Spirit, 11-people team project.

- Used the simple Euler method and Newton's law to simulate real-world physics.
- Implemented 2D Circle, ABBB collision detection, and resolution to simulate collision in real world.
- Used Vector, Linear Algebra, and Physics about elasticity to simulate spring beds, and implemented elastic mushroom beds according to designers' needs to improve the gameplay.
- Implemented player controller and cooperated with the designer to adjust the feel.