Mohamed Kazma

mohamed_kazma@hotmail.com | +16137000828
https://moekaz.github.io/

EXPERIENCE

GEARBOX SOFTWARE | Animation Programmer

April 2023 – Present | Remote

- Worked on different animation tools in UE5 to provide designers with easier workflows to build content
- Redesigned the cloth system from UE5 to support cloth asset sharing and syncing and reworked cloth LODing
- Worked on different optimizations to the cloth asset system which optimize setup time of cloth actors and drastically decrease memory footprint of assets
- Wrote a corpse actor system that allows adding an optimized character corpse to a scene and providing it with a pose to hold
- Fixed different bugs related to animation systems and worked some blueprint related animNodes for animator workflows

ROCKSTAR GAMES | Animation Programmer

March 2020 – April 2023 | GTA, ON

- Worked on Procedural Motion Reaction systems with physics-based animation driven ragdolling
- Worked on animation system state machines for managing AI states with procedural reactions
- Wrote state and animation networks with Blend Trees and Nodes for blending between animations
- Implemented systems and logic for streaming animation clipsets for running behaviors

ROSS VIDEO | Software Developer

Sept 2018 – Dec 2018 | May 2019 - Sept 2019 | Ottawa, ON

- Worked on DashBoard, a program that simplifies connecting and configuring broadcasting devices with scriptable logic and API calls
- Developed a scheduler product that creates a manager for broadcasting devices and send certain API calls either on a scheduled time or instantaneously
- Worked on build pipeline with Maven/Tycho to generate a build script that is used to create an executable build of the product

PROJECTS

MTRX ENGINE | August 2018

- Physics Engine based in C++ using GLM, GLAD, GLFW, and Spdlog
- Implemented Rigidbody dynamics with Newtonian Physics with application and integration of said forces and generating torques to create rotation when necessary as well as basic inertia tensors used for simulating said rotational forces
- Added basic bounding collider volumes(Sphere, Capsule, Box, Convex Shape colliders) that used in basic collision detection algorithms and are also helpful for raycasting query algorithms
- Implemented GJK, an algorithm used for collision detection of any convex shape, for more complex and finer collision detection
- Implemented a BVH(Bounding Volume Hierarchy) which creates hierarchies of bounding volumes(colliders) which optimizes collision detection calculations
- Added force generators that are used to easily add a certain force to a Rigidbody(Gravity, Drag, Buoyancy, Spring Forces, etc...)

LINKS

Github: @moekaz
Linkedin: Mohamed Kazma

EDUCATION

CARLETON UNIVERSITY

BS in Computer Science,
Game Development (Hons.)
Ottawa, ON
Cum. GPA: 3.5/ 4.0
Major GPA: 3.55/ 4.0

SKILLS

LANGUAGES:

C++ • C • C# • Java • Javascript • CSS • HTML • Python

FRAMEWORKS:

Node.js • Pug(Jade) • Express.js • JUnit

TECHNOLOGIES:

Unreal Engine • Unity • MongoDB • Git • SVN • SQL • Perforce

LIBRARIES:

OpenGL • GLSL • SFML • JQuery • Socket.io • Swing