Amit Kumar Mehar

www.github.com/amutbkt

+91-9116488276
amit.k.mehar@gmail.com

Education

Indian Institute of Technology, Roorkee **2012-2016**

- **Major:** Electrical Engineering, B-Tech
- **Programming Coursework:** Data Structures
- **EE Coursework:** Network Theory, Control Systems, Microprocessor and Peripheral Devices
- **Projects:** Flappy Bird Automation using Reinforcement Learning to play automatically using servo motor

Employment

Programmer **SUMO Digital Pune** **July 2016, Present**

Snake Pass (C++, UE4, Xbox One, PS4, Steam)

- Implemented shader based sphere+plane interaction system to simulate collision between snake and grass
- Developed Maya scripts to easily create rigs for physics simulation
- Worked with Xbox One and PS4 APIs
- Utilized: UE4, C++, Python, Maya

RAID WW2 (C++, LUA, Profiling, Xbox One, PS4)

- Optimized asset cooker to reduce vertex buffer memory by ~30%
- Reduced GPU memory utilization by adding runtime texture coordinate compression
- Added improvements/bug fixes to shader effects pipeline
- Implemented search and filter functionality for XboxOne session management
- Utilized: C++, LUA, PIX Profiler, Razor GPU

Team Sonic Racing (C++, Graphics, Core Tech and Tools, Xbox One, PS4)

- Optimized GPU performance on Xbox One
- Added new rendering features (IBL for forward pass, BC5 normal texture compression)
- Extensively worked on Xbox One to add various platform features (multiplayer & matchmaking, media player, save game, achievements, leaderboards).
- Worked on AI, UI and new Editor features.
- Utilized: C++, PIX Profiler, Xbox One tools, Maya

Personal Projects (https://amutbkt.github.io/)

Physics Editor is a lightweight, browser-based Box2d powered physics editor and simulator

- Developed an easy to use user interface to create and visualize Box2d world
- Implemented concave mesh decomposition algorithm to triangulate complex shapes
- Added support to export scenes to Unity, SpriteKit(iOS), LibGdx, Cocos2d..)
- Utilized: JavaScript, CSS, C++, C#, Java, ObjectiveC

2d OpenGL Game Engine

- Developed native C++ engine for Android
- Implemented rendering pipeline to support custom GLSL shaders and post process effects
- Integrated OpenAL to add audio effects to the game
- Added support for simple AABB collision detection
- Utilized: C++, Java

3d OpenGL Game Engine

- Implemented Deferred+Forward rendering pipelines to draw opaque and transparent object
- Added HDR support using exposure tone mapping technique
- Order independent Transparency to render transparent objects using Weight Blended OIT
- Added support for importing FBX file for rigged and animated models.
- Integrated NVidia PhysX
- Utilized: C++

Skills

(proficient) C++, UE4, Graphics Programming
(familiar) Java, Javascript, ObjectiveC, Maya