CSEE 4840 Embedded System Design
Project Proposal

CUDoom

A Raycasting Video Game about Getting to Class on Time

Alden Goldstein (ag3287)
Edward Garcia (ewg2115)
Minyun Gu (mg3295)
Wei-Hao Yuan (wy2211)
Yiming Xu (yx2213)

OVERVIEW

Our project will implement a video game using raycasting techniques. The game will accept keyboard input from users and display a 3D like world on a VGA output. The goal of the game will be to reach a destination on another side of the map. Multiple levels will be created and user will have the option to select between them.

BROAD DESCRIPTION

Our project will use a basic 2D world map, and give a pseudo 3D depiction of the world by using a ray casting algorithm as described below. Every time the player moves or rotates, we will recalculate the "3D" image using the algorithm, and then refresh the screen accordingly. The 2D map will be stored in memory, and the raycasting algorithm will be able to access the map and calculate the pseudo 3D output matrix based on the position and orientation of the user. A game logic module will interface with the keyboard, and communicate with the algorithm block and the VGA controller, and let both know when an update needs to be registered. The game logic module will also update the position on the map.

INPUT AND OUTPUT

Inputs

- PS/2 Keyboard for input

Outputs

- VGA output of raycasted maze
- Music for world and sound effect of different actions

ALGORITHM DESCRIPTION

Raycasting creates a pseudo 3D world from a 2D map. Examples of games that have implemented this algorithm are Doom and Wolfenstein 3D. At the expense of fixing the camera rotation around the player's vertical axis, computation is greatly decreased while maintaining the illusion of a 3D world. For our implementation, all the walls will have the same height and will be orthogonal squares on a 2D grid.

HARDWARE AND SOFTWARE SPLIT

We plan to implement the main algorithm in software while using hardware to accelerate costly functions like multiplication and division. Keyboard input will be handled by hardware and audio will be stored and played in hardware.

MILESTONES

Milestone 1

- Implement raycasting algorithm on software
- Design several mazes

Milestone 2

- Integrate the algorithm with FPGA
- Realize hardware acceleration for the algorithm
- Display the world properly on screen

Milestone 3

- Add audio output to the game
- Complete game features, i.e. player movement, interface