**CSEE 4840 Embedded System Design**
**Project Proposal**

# Networked Worms like Video Game

*Group Name: WormCraft*

| | |
|---|---|
| *Ning Li* | *(nl2443)* |
| *Tianyi Zhang* | *(tz2210)* |
| *Yuxuan Zhang* | *(yz2580)* |
| *Ziwei Zhang* | *(zz2282)* |

## Overview

WormCraft demo figure

Our project will implement a 2D *Worms®* like video game for two players following the physics rule of gravity and bouncing. The player will throw two kinds of bombs toward each other until one of the player runs out of life. The game will be implemented with System Verilog and C language. The game will accept keyboard input from user and display the battle scene on a VGA output. Multiple types of landforms will be created. It can be played by two people or in two different devices in Ethernet network

## Input And Output

**Inputs**

Standard Qwerty Keyboard

**Outputs**

VGA 2D graphic display using RGB color

Multiple sound effects for worm moving, bouncing, getting injured, bomb emission, explosion and background music.

## Implementation Description

WormCraft will simulate a physics model

A model and algorithm for bomb track and explosion animation effect

A simple algorithm for selecting the direction of emitting bomb

A algorithm for worm moving and bouncing on uneven surface

A simple algorithm for dynamic background and worms

Multiple game dynamic scenes and multiple autonomous actions of worms

Multiplayer mode based on Ethernet protocol (optional)

## Algorithm Description

The time holding the "power" key will determine the initial velocity of the bomb. After the "power" key is released, a trajectory of the bomb will be calculated based on the initial speed, toss angle and gravity factor. If bouncing occurs, such as the grenade hits the wall, the bouncing trajectory will be calculated by the same algorithm will the initial velocity vector determined by the momentum theory. If possible, building new blocks as barrier will be allowed.

## Hardware and Software Split

We plan to implement the main algorithm in software while using hardware to accelerate the computations and physical models such as reserving vector and the degree of bomb destruction. Keyboard input and background music will be handled by hardware.

## Timeline

**Milestone 1:**

Configure Keyboard and VGA interface
Implement the physics engine on software
Design several landforms

**Milestone 2:**

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

**Milestone 3:**

Add audio output into the memory of the board
Test and improve the game performance