CS4250 Interactive Computer Graphics Fall 2019
Handout #5 9/30/19

Homework Assignment 1 Interactive GAMES!

UN Syrian Food Drop

Due: Mon. 10/7/19 (60 minutes before class)

Write an OpenGL program that let's users play a 2D (or optionally a 3D) bomber game (with a politically correct twist). Your UN food transport (bomber) has the ability to drop food to starving people. The player **must be able to select** which starving people each food drop food will go to. The plane will drop the food and, as long as the target is close enough, the drop will get to the people. You are free to implement what should happen if the starving people aren't close enough. You must **implement a scoring mechanism**, and players should get points for how many people they feed. Players must have a maximum number of drops they can make per round, so the goal is to get food to as many people as possible with as few drops as possible.

Bad guys – **you must implement bad guys** who will try to steal the food, you may choose to have them have additional behaviors as well. If a player incorrectly targets one of these guys they lose points. These bad guys should move towards the food, if they get there first, the food drop fails.

Trees – **you must also implement some sort of trees** to block the food from getting to people (i.e. there must be a clear path for the food to get to the people). Trees should also pose as barriers to the bad guys.

Levels (*optional*). If time permits, you can implement several different levels with different difficulties.

Multiple simultaneous targets (*optional*) one variation is to be able to have the one food drop be able to split and go to several simultaneously targeted people relatively close to each other. This concept is like the MIRV (multiple independently targeted reentry vehicle) concept for nuclear weapons.

It is usually best to make each round last at most a couple of minutes.

The **goal** of this assignment is to **practice selection**, so **you must describe in detail how you implemented the selection mechanism** to select the people to target in the submitted **readme.txt** file. Also desribe how you implemented the movement of the plane/ground objects, and in what way you achieved all the requirements of this project.

(*Optional*) Since the games might be long, optionally allow users to save a game at any point during the game play. Your program will accomplish this by reading and writing game states to/from a file. You are free to implement any file format you wish.

Follow the coding standards from the course web page. Include with your submission a write-up of your implementation decisions in a file called `readme.txt`. Be prepared to explain your program in class to the other students, and to demo the working program. Consider what features to add to make the game more visually exciting. The grading will be 50% correctness, 25% style of coding, 25% coolness factor. Be sure to explain what aspects of your game you think deserve coolness points in your `readme.txt` file.