EE2510 – Project A: Sumo-Bot Demolition Derby

3 weeks total – **DEMO required**

Goals:

1. Implement a complete C++ game/simulation program
2. Use inheritance and polymorphism

Assignment Description:

Overview:

Create a simulation of a demolition derby using sumo-bots (Johnson-bots and Widder-bots)

Interface:

The user will be asked for the size of the arena

The user will be asked for the number of each type of sumo-bot to start with.

The user will be asked for the sumo-bot's starting strength

The sumo-bots will be placed in the arena in random locations

When all of one type of bot has been destroyed the game will terminate and a winner will be declared (Johnson-bot!)

Operational requirements:

Each bot will start with a user defined number of strength points

The arena will be surrounded by barriers

Additional barriers can exist randomly throughout the arena

Bots will travel one cell each time step – in a randomly selected direction

When a bot attempts to enter a cell containing a barrier it will (stay in its current cell, reduce its strength points by a barrier member variable defined number of points (1?))

When a bot attempts to enter a cell containing another bot it will (stay in its current cell, reduce the other bots strength by a sumo-bot member variable defined number of points (5?))

When a bot no longer has any strength points it is replaced with a barrier.

Additional functions may be needed or desired

NO global variables

Grading:

Functionality	Structure
Comments – readability	Documentation
Cleanliness (beauty) of the code	On-time

Deliverables:

All code

Description of approach – including use of polymorphism

UML diagram(s)

Eclipse "project explorer" capture showing all files in the project

Video of program run

Hardcopy – via OneNote, no need to put in PowerPoint or pdf

Due: Thursday, 4:00, week 10 – Uploaded to OneNote – No exceptions

OOP Concepts:

- Inheritance (element → Bots, barriers)
- Array of pointers
- Polymorphism (some pointers point to nullptr, barriers, bots)
  (move function, hit function vary with element type)
- Forward declaration
- Simulation (time control)
- Windows display control

Libraries and functions:

- iostream
- windows.h (time, sleep, display functions)
- math.h (srand, rand)
- ctime

Development:

- Create a derby class, element class → barrier class and bot class
  - Derby Class
    - Include a 2-d array of element class pointers
    - Populate the array with null pointers (nullptr)
    - Add the barriers to the array by replacing the null pointers with barrier pointers
    - Add the bots to the array by replacing the null pointers with bot pointers
    - Print the array
  - Element class
    - Pass the array to the other classes to allow movement and collision actions