Fall 2016
CS 529 | Fundamentals Of Game Development
Project 3 | Cage

Files (submit folder) due
- Part 1 & Part 2: Tuesday October 18th 2017
- 11.55pm

Topics
The assignment will cover the following topics

I. Implementing Cage, which is a compilation of different object intersection and reflection techniques:
   a) Static Point to Static Line Segment
   b) Static Circle to Static Line Segment
   c) Animated Point to Static Line Segment with Reflection
   d) Animated Circle to Static Line Segment with Reflection
   e) Animated Point to Static Circle with Reflection
   f) Animated Circle to Static Circle with Reflection

Goal
The goal of this assignment is to have a ball bouncing in a room that contains walls (Static line segments) and pillars (Static circles). Accurate time based collision will be used to determine the exact time and position of the intersection, which will then be used to reflect the main ball.

Assignment Submission
- Check the course syllabus regarding the naming and submission convention.

Description

I. Language: C/C++

II. A template application will be provided

III. The project is divided into 2 parts
   a) Part 1: 2D Line Segment Library, Ball Intersection and Reflection with the room's walls
   b) Part 2: Ball intersection with pillars and line segments (from both sides)

Part 1

I. Copy your Math2D, Matrix2D and Vector2D functions to the provided template.

II. Implement the LineSegment2D Library
   a) A single function that builds the data of a 2D line segment
   b) Function declaration is found in LineSegment2D.h
   c) Implement it in LineSegment2D.c
   d) Detailed explanations are found in the header file

III. Implement the following functions:
   a) Declarations are found in Math2D.h
   b) Implement them in Math2D.c
   c) Detailed explanations are found in the header file

```
float StaticPointToStaticLineSegment(Vector2D *P, LineSegment2D *LS);

float AnimatedPointToStaticLineSegment(Vector2D *Ps, Vector2D *Pe,
LineSegment2D *LS, Vector2D *Pi);

float AnimatedCircleToStaticLineSegment(Vector2D *Ps, Vector2D *Pe, float
Radius, LineSegment2D *LS, Vector2D *Pi);

float ReflectAnimatedPointOnStaticLineSegment(Vector2D *Ps, Vector2D *Pe,
LineSegment2D *LS, Vector2D *Pi, Vector2D *R);

float ReflectAnimatedCircleOnStaticLineSegment(Vector2D *Ps, Vector2D *Pe,
float Radius, LineSegment2D *LS, Vector2D *Pi, Vector2D *R);
```

Testing Part 1

I. Make sure that "TEST_PART_2", which is found at the top of "GameState_Play.c" is set to 0
II. If your implementation (Vector2D, LineSegment2D and Math2D) is correct, you should see a yellow ball bouncing between 5 line segments (Which constitute the room).

Grading Part 1

I. Ball goes through walls/incorrect intersection: -100%
II. Missing/Incorrect reflection off walls: -50%
III. Crash/Freeze/Doesn't compile: -100%
IV. No/Wrong submission: -100%
V. Other possible mistake: Up to -100%

Part 2

I. Implement the following functions:
   a) Declarations are found in Math2D.h
   b) Implement them in Math2D.c
   c) Detailed explanations are found in the header file

```
float AnimatedPointToStaticCircle(Vector2D *Ps, Vector2D *Pe, Vector2D
*Center, float Radius, Vector2D *Pi);

float ReflectAnimatedPointOnStaticCircle(Vector2D *Ps, Vector2D *Pe,
Vector2D *Center, float Radius, Vector2D *Pi, Vector2D *R);

float AnimatedCircleToStaticCircle(Vector2D *Center0s, Vector2D *Center0e,
float Radius0, Vector2D *Center1, float Radius1, Vector2D *Pi);

float ReflectAnimatedCircleOnStaticCircle(Vector2D *Center0s, Vector2D
*Center0e, float Radius0, Vector2D *Center1, float Radius1, Vector2D *Pi,
Vector2D *R);
```

Testing Part 2

I. Make sure that "TEST_PART_2", which is found at the top of "GameState_Play.c" is set to 1
II. If your implementation (Vector2D, LineSegment2D and Math2D) is correct, you should see a yellow ball bouncing off fixed pillars and line segments, in addition to the 5 line segments (which constitute the room) from part 1

Grading Part 2

I. Ball goes through inner walls/incorrect intersection: -100%
II. Ball goes through circles/incorrect intersection: -100%

III. Missing/Incorrect reflection off circles: -50%
IV. Crash/Freeze/Doesn't compile: -100%
V. No/Wrong submission: -100%
VI. Other possible mistake: Up to -100%

Debug Information

I. Setting "DRAW_DEBUG" (Found at the top of "GameState_Play.c") to 1 will draw the normal vectors of the line segments, starting from their respective middle points.

General Testing Input

I. Press 'S' to **S**top the simulation.
II. When stopped:
    a) Trigger 'R' to let the simulation **R**un again.
    or
    b) Trigger 'S' to let the simulation run for 1 **S**tep, where the frame rate will be set to 0.016 seconds.
    or
    c) Press 'G' to keep the animation **G**oing for as long as 'G' is pressed, where the frame rate will be set to 0.016 seconds.

Grading

Your project will be graded with an Alpha Engine build that DOESN'T include any of the functionalities that you are required to implement. Make sure to replace variable types and function names with your own.

Example:

- **Replace each AEVec2 with Vector2D**
- **Replace AEBuildLineSegment2D with BuildLineSegment2D**
- **Etc…**

Finally, each ".c" and ".h" file in your homework should include a header, which is included in the course syllabus.