Fall 2011

CS 529 | Fundamentals of Game Development

Project 2 | Cage

Files (submit folder) due

- Part 1: Friday, September 30, 2011 at 11:55pm
- Part 2: Wednesday, October 5, 2011, at 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 Circle
   b) Static Point to Static Line Segment
   c) Static Point to Static Line Segment
   d) Animated Point to Static Line Segment with Reflection
   e) Animated Circle to Static Line Segment with Reflection
   f) Animated Point to Static Circle with Reflection
   g) 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

- Compress (.zip) the solution folder (Delete the debug/release folders and the .ncb file first), and submit it on distance.digipen.edu.

- Your submitted assignment should use the following naming convention:
  <class>_<student login name>_<assignment#>_<part#>

- Example: John Smith should submit: **CS529_fooboo_ assignment2_Part1.zip**

Description

I. Language: C/C++

II. A template application will be provided

III. The project is divided into 2 parts
   a) Part 1: 2D Vector Library, 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. Implement the Vector2D Library
   a) Function declarations are found in Vector2D.h
   b) Implement the functions in Vector2D.cpp
   c) Detailed explanations are found in the header file

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.cpp
   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.cpp
   c) Detailed explanations are found in the header file

```
bool StaticPointToStaticCircle(Vector2D &P, Vector2D &Center, float
Radius);

float StaticPointToStaticLineSegment(Vector2D &P, LineSegment2D &LS);

float AnimatedPointToStaticLineSegment(Vector2D &Ps, Vector2D &Pe,
LineSegment2D &LS, Vector2D &Ph);

float AnimatedCircleToStaticLineSegment(Vector2D &Ps, Vector2D &Pe, float
Radius, LineSegment2D &LS, Vector2D &Ph);

float ReflectAnimatedPointOnStaticLineSegment(Vector2D &Ps, Vector2D &Pe,
LineSegment2D &LS, Vector2D &Ph, Vector2D &R);

float ReflectAnimatedCircleOnStaticLineSegment(Vector2D &Ps, Vector2D &Pe,
float Radius, LineSegment2D &LS, Vector2D &Ph, Vector2D &R);
```

Testing Part 1

I. Make sure that “TEST_PART_2”, which is found at the top of “LevelCage.cpp” is set to 0

II. If your implementation (Vector2D, LineSegment2D and Math2D) is correct, you should see an orange ball bouncing between 5 line segments (Which constitute the room)

Part 2

I. Implement the following functions:
   a) Declarations are found in Math2D.h
   b) Implement them in Math2D.cpp
   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 “LevelCage.cpp” is set to 1

II. If your implementation (Vector2D, LineSegment2D and Math2D) is correct, you should see an orange ball bouncing off fixed pillars and line segments, in addition to the 5 line segments (which constitute the room) from part 1

Additional Information

I. Use the following drawing functions to draw shapes (Examples can be found in the “GameStateLevelCageDraw” function in “LevelCage.cpp”

II. Declarations are found in Graphics.h

III. Definitions are found in Graphics.cpp

```
void GFX_DrawRectangle(int centerX, int centerY, int height, int width,
unsigned color);

void GFX_DrawEllipse(int centerX, int centerY, int height, int width,
unsigned color);

void GFX_DrawLine(int startX, int startY, int endX, int endY, unsigned
color);

void GFX_DrawText(int startX, int startY, char *buffer, int length,
unsigned color);
```

IV. Setting “DRAW_DEBUG” (Found at the top of “LevelCage.cpp”) to 1 will draw the normal vectors of the line segments, starting from their respective middle points.

Finally, each ".cpp" and ".h" file in your homework should include the following header:

```
/* Start Header -------------------------------------------------------

Copyright (C) 20xx DigiPen Institute of Technology.
Reproduction or disclosure of this file or its contents without the
prior written consent of DigiPen Institute of Technology is prohibited.

File Name:        <put file name here>
Purpose:          <explain the contents of this file>
Language:         <specify language and compiler>
Platform:         <specify compiler version, hardware requirements,
                  operating systems>
Project:          <specify student login, class, and assignment. For
                  example: if foo.boo is in class CS 529 and this file
                  is a part of assignment 3, then write:
                  CS529_fooboo_3>
Author:           <provide your name, student login, and student id>
Creation date:    <date on which you created this file>

- End Header --------------------------------------------------------*/
```