CO1301: Games Concepts
Prototypes - Pool

This exercise is the development of a simple pool prototype. You will need to use the collision detection and vector work from the previous labs. You will be given guidance and the appropriate artwork, but you will need to make some of your own decisions in approaching this problem.

Design Decisions
I suggest that you develop this game as a sequence of simple prototypes:

Prototype 1
- Create a pool table, but initially ignore the pockets. The pool table model is called "PoolTable.x".
- Put only one white ball on the table.
- Fix the camera to the ball and allow the player to rotate the camera around and zoom in and out
- Allow the player to hit the ball (with fixed strength) in the direction that the camera is currently looking
- Calculate collisions between the ball and the cushions, and reflect the ball as appropriate

Prototype 2
- Add a second red ball to the table
- Calculate collisions between this second ball and the cushions
- Calculate collisions between the balls, and bounce them as appropriate

Prototype 3
- Update the cushion collision code to allow the balls into the pocket areas
- See if either ball comes close to the pockets and is “potted”
- Create a simple scoring system

And Beyond…
- Improve the UI…
- Create a proper scoring system…
- Support a two player game…
- Put front end menus…
- Add more balls…
- Etc…

Technical Details
The first problem is to rotate the camera around the main ball:
- The easiest way to do this is to make the camera a child of the ball, and then rotate the ball.
- However, we don’t really want the ball to rotate just because the camera is moving.
- Use a dummy object in between the ball and camera to overcome this. See Worksheet 13 for a discussion of dummy objects.

The next problem is to allow the balls to move realistically:
- For each ball you should have an variables for the X and Z velocity
- You can ignore Y values in this program, as the balls will stay on the horizontal.

- Each frame, move the balls by their velocities.
- Add some friction to slow the balls down – simply reduce the velocities a little each time you move them.

The next step is to start the white ball moving:

- When a shot is taken, calculate the direction vector from the camera to the ball
- The X and Z values from this vector can be used as an initial velocity for the ball
- You may want to scale them to make a good initial speed

The collisions with the cushions are fairly straightforward:

- Use the diagram below to get the locations and sizes of the cushions
- You can use the sphere-box collision method detect a collision
- You can simplify the method a little if you think about it
- When the ball hits a cushion, negate its X or Z velocity as appropriate, i.e. reverse the sign of its X or its Z velocity. This has the effect of bouncing the ball off the cushion.

Adding a second ball will first require duplication of the velocity and cushion code. Then use the sphere-sphere collision method to detect if the balls have collided.

After detecting a collision between the balls you will know:

- The X and Z velocities of the two balls (say X1, Z1 and X2, Z2)
- The vector between the centres of the two balls (say, DX and DZ) and the length of this vector (DLen). You will have used these values in the collision calculation.
- The physics is a little tricky so this is what you do if you detect a collision:

```
DX = DX / DLen;
DZ = DZ / DLen;

float momentum = (X1 * DX + Z1 * DZ) - (X2 * DX + Z2 * DZ);
X1 += momentum * DX;
Z1 += momentum * DZ;
X2 -= momentum * DX;
Z2 -= momentum * DZ;
```

- This code block will update the velocities of the two balls to bounce them apart.

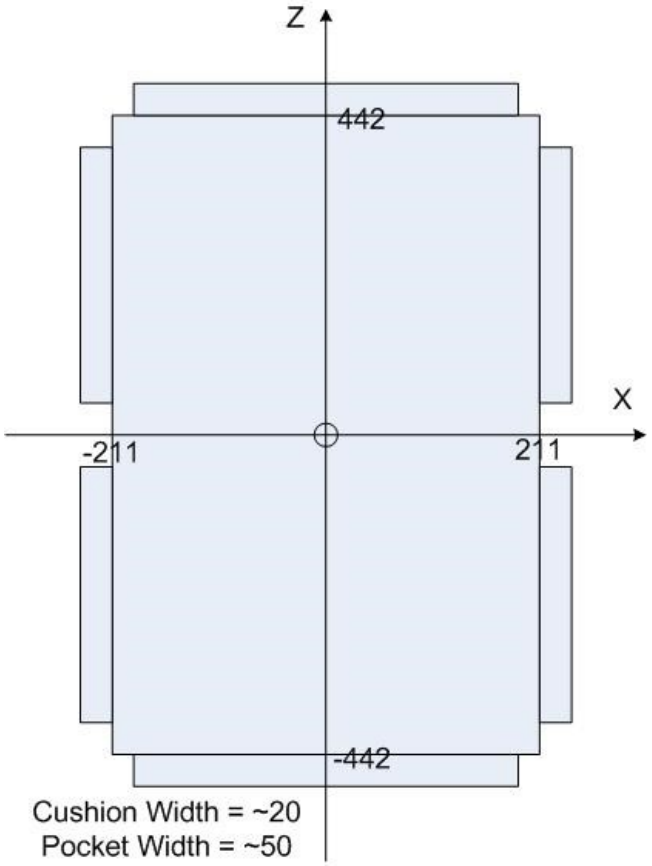
Artwork

All of the models can be found in the TL-Engine media folder. The two models used for this game are “PoolTable.x” and “PoolBall.x”. When you create pool ball models they will initially be white. To change their colour use ‘SetSkin’:

```
ball->SetSkin("RedBall.jpg"); // Set ball model to be red
```

There are four skins “RedBall.jpg”, “YellowBall.jpg”, “BlackBall.jpg” and “WhiteBall.jpg”

The dimensions of the pool table are shown below. Remember to ignore the pockets for your first test prototypes.



The pool table top is at Y=0.
The pool balls have a radius of 20.