Mini-Golf Game (Project 2)

# Goal

Demonstrate your understanding of how to implement forces and collisions between two circles, between a circle and a wall, and between a circle and a convex polygon of infinite mass.

# Game Description

A ball is aimed and set in motion. It rolls with constant rolling friction to a stop before it can be hit again. The course has walls/rectangles on the outside and perhaps interior obstacles. It should also have a circular hole.

You win when the ball falls in the hole. Make the hole’s physics feel somewhat realistic, so that balls overlapping the hole are pulled in. Use a PairForce for the interaction of the ball with the hole (but give the hole infinite mass!). Perhaps also, balls traveling too quickly will pass over the hole with some decrease in speed, rather than falling in.

I also want to see interior changes in turf or elevation. Both should be accomplished with a SingleForce that depends on the position of the ball. You could have rougher turf, which would have a higher friction coefficient. I’d like to see hills of some sort, which provide a constant acceleration in the downhill direction when the ball is on the incline. Indicate hills in some way, such as by a gradient in the shade of green of the turf. Hills that are sloped in more than one direction are trickier, and are optional. Derive the force field as the negative gradient of potential energy.

You can have multiple holes (levels) or just one. If just one, all the above features must be in that hole.

You can have single-player or capability for two players. In the case of two players, perhaps a player could use his ball to bump the other player’s ball.

# Requirements/Rubric

For full credit, your game must have all these features:

(15) Proper setup and rendering of the scene and all objects.

0 = unplayable due missing stuff  
5 = outside walls, ball, and hole  
10 = also includes interior polygonal obstacles  
15 = also all turf/inclines indicated

(10) A way to launch the ball with desired direction and initial speed. Can’t launch the ball when it’s in motion.

0 = ball cannot be launched, unplayable  
 7 = ball launches but is not very intuitive or is too easy to game  
 10 = launching of ball is intuitive and both fun and challenging

(20) Realistic collision physics of ball with walls and polygonal obstacles.

0 = no collisions implemented

10 = collisions with infinite walls only  
15 = collisions with polygons, but a bit buggy  
20 = collisions with polygons done perfectly

(10) Realistic force-based physics of ball with different kinds of turf. Use SingleForce.

0 = no forces implemented  
 5 = only constant friction, no different kinds of turf  
 10 = different kinds of turf

(10) Realistic force-based physics of ball with inclines. Use SingleForce.

0 = no forces implemented  
 5 = force-field-based inclines, but a bit buggy, doesn’t feel right  
 10 = force-field-based inclines perfectly done

(15) Realistic-looking behavior of ball near hole. Win condition feels right. Use PairForce.

0 = no win condition  
 5 = win condition but ball doesn’t interact with hole  
 10 = ball interacts with hole, but it looks hokey  
 15 = ball physics near hole looks right

(10) Creativity, fun, and finesse.

0 = no evidence of effort in creativity  
 5 = some evidence of creativity  
 10 = impressive creativity

(15) Code readability and organization.

0 = can’t make sense of your code, painful to read and no comments  
5 = some comments but still hard to understand  
10 = organization and comments are ok, room for improvement in style  
15 = easy to read and understand at a glance

# Presentations

About half the groups will present the current state of their game on Friday, March 8. Use what you learn from other presentations, or feedback on your own, to improve your game before final submission.

# Due Date

The final version of the game will be due on Wednesday, March 13.