Michael Choquette, mchoquet

The file GSclasses contains the classes Ball, Tile, Goal, Button, and Obstacle, as well as some collision detection functions. Here are some of the relevant methods:

Ball:

location:

returns the current location of the ball

draw:

draws the ball

move:

moves the ball by dx,dy

drawPath:

draws a line from the last location of the ball to its current location

Tile:

setForce:

calculates the force the tile imparts on the ball

draw:

draws the tile and the two lines forming the chevron

getChevCoords:

finds the coordinaes of the lines in the chevron

moveTo:

moves the tile to a certain point

animate:

moves the chevron forward until the end of the tile. If it moves off, move it back

to the beginning. The bounding box of the chevron is recorded in chevBox.

resetChev:

moves the chevron back to the center of the tile

Goal:

achieve:

sets the goal to achieved

unachieve:

resets the goal

intersection:

tests whether or not a given ball, moving along a given line segment, will intersect with the goal

Button:

setText:

sets the text displayed on the button

draw:

draws the button, choosing the fill color based on what type of button it is

Obstacle:

firstIntersection:

finds the first intersection of a ball moving along a line with the

obstacle. It does this by finding all of the intersections of the ball with any part of the obstacle (all sides and all corners) along any part of the line of which x,y to endX,endY is a segment. It then returns the first collision on that segment.

checkEdges:

finds the equation of the line and uded algebra to find intersections with the walls

checkCorners:

finds the equation of the line and uses the quadratic formula to find the

intersections of the line and the corner.

chooseFirstByTime:

it receives a list of collisons, removes any that occur outside of the line or at the start point (to prevent double-counting), and returns the first collision that occurs. When finding the collisions I calculated the time (from 0 to 1) during the update at which the collision occurred, so I simply remove everything with t<=0 and t>1, then choose the minimum t that remains.