Physics Loop Description

Note that this implementation only works if there is only one ball on the board.

For this method to work for more than one ball on the board we would need to find the time until collision for each ball, determine the minimum time until collision and use that value for the conditional statement below (instead of *tuc(Time Until Collision)* we would use *min_tuc*). Considerations would need to be made if multiple collisions where to occur at the same time.

public void moveBall() {

double moveTime = 0.05; 0.05 seconds (20 times per second)

CollisionDetails cd = timeUntilCollision();

double tuc = cd.getTuc();

We now check whether the time until collision is less than or equal to the move time 0.05s.

if (tuc > moveTime) {

No collision, the ball will move for the full 0.05s

ball = movelBallForTime(ball, moveTime);

All other active objects will move for 0.05s

} else {

Collision will occur, we must move the ball and all other objects for tuc

The ball should only move for time equal to the minimum time until collision.

ball = movelBallForTime(ball, tuc);

Resolve the collision of the ball, set new velocity and direction after collision.

ball.setVelo(cd.getVelo());

Check if the gizmo that the ball hit has a trigger attached. If so, set the respective trigger (will not move until next tick)

}

Notify observers and redraw the updated view

this.setChanged();

this.notifyObservers();

Apply Friction and set new vector for the ball

Apply Gravity and set new vector for the ball

}