

Roulette

Roulette is a famous casino game, where the croupier first spin a roulette wheel and then throws a small ball on top of it. The ball settles in one of the numbered sections on the roulette wheel. The player has to place a bet on the section of the wheel, the ball will fall in. In today's lab, you guys are going to design a game of roulette. The game will have different objects:

- croupier
- Roulette_Wheel
- Player

Each of these objects will have certain properties, which will be described in the next section of your handout. You will design the game using these objects. We will be making some assumptions to simplify some stuff. These assumptions will be stated alongside everything. So read the handout very carefully and have fun :)

Croupier:

Here comes the guy no one can cheat. Now the croupier's job is to start the whole thing, and give you the result. Your croupier object will contain a Roulette_Wheel object. Which a croupier will spin. Each croupier has some different strength so the initial velocity with the Roulette_Wheel spins is going to be different each time (in simple words, set up a max and min limit and roulette initial speed is that). Now the croupier throws the ball. We will assume that ball has equal chance of landing at every section on the roulette wheel. The croupier also has to announce the outcome of a player's bet (this will be input to the spin function of croupier, we will discuss the player in a bit). For this the roulette wheel has to stop sometime. This is going to be a property of Roulette_Wheel and we will assume that roulette wheel is slowing down at a constant speed. Now the ball travels different sections of settling down. But since this is the ideal world we are going to assume that our ball has a velocity which is the same as wheel at every instance in time, but is opposite in direction. Another simplifying assumption we will make is that the ball settles down only when the wheel stops moving. So if we know that our wheel moved 2.5 cycles that means our ball moved the same just in opposite direction.

One more thing to remember that a croupier, is assigned to only one roulette wheel but he/she may be assigned to a different roulette wheel at a later stage in the game and that the croupier can move the wheel in either direction. You need to take care of these things too.

The croupier just needs to have one public function which takes a player as input, and returns true if a player wins the bet and false if the player loses. When calculating the win or loss of the player, you need to assume the whole situation is as explained above.

Roulette_Wheel:

Here comes the wheel of death (I am just joking). Now the wheel needs to have certain properties. First one is the number of sections the wheel is divided in. Make that a runtime parameter for the roulette wheel. Also assume that width of each section is exact same. Next is the radius of the roulette wheel and then comes the velocity at which the roulette wheel is stopping over time or the deceleration of the wheel. We will assume this to be a constant and a runtime parameter. For example, if a roulette wheel has a deceleration of 10 m/s^2 and an initial velocity of 100 m/s (a very powerful croupier indeed) the roulette wheel will come to stop in about 10 seconds.

That would mean that a point on the circumference of the roulette covered a distance of about $ut + \frac{1}{2}at^2$. Where u is the initial velocity of the roulette and a is the deceleration of the roulette wheel. Using this distance and the radius of the roulette wheel, we can easily find the number of cycles roulette wheel had to move before coming to a stop. And this would tell us the position the ball would be in since we know each section on the wheel is of equal width and the starting point of the ball.

Remember that roulette wheel has numbers increasing in counterclockwise order. And that croupier could spin it in either direction depending on his mood (again something random).

Player:

Now this is the most miserable guy. He can only see whether his wish comes true or not. So we won't complex this one and just have a single attribute associated with this one his bet. Each player will only have a bet as his member.

HAVE A NICE LAB :)