Christian Rudder
CS225 Sec.012
4/18/2022

Proxy Implementation in RaceTrack

I think adding a proxy design pattern to the project would be a nice addition for security reasons. Say if the program was expanded, and there were administrators who controlled the simulation and there were spectators who wanted to view the car data. It would be a bad idea to give them the race track manager outright, as they would have too much access to information and start and stop the race. To fix this, a proxy class is added so that spectators can interact with an interface that gives them access to car data, but not so much so that they tamper with the race's integrity itself. This in another way makes it easier to get information from the race easily without having to deal with the clutter of the race track manager.

```
Car
─────────────────────────────────────────────
- rand : Random
- transition : PathTransition
- path : ArrayList<Double>
- speed : int
- color : String
─────────────────────────────────────────────
+ carPrinterTester() : void
+ randomizeSpeed() : void
+ equals(obj : Object) : boolean
+ toString() : String
+ getSpeed() : int
+ getColor() : String
+ getpath() : ArrayList<Double>
+ startEngine() : void
~ Car(speed : int, carColor : Color, path : ArrayList<Double>, colorName : String)
~ Car()
```

```
RaceTrack
─────────────────────────────────────────────
~ timer : Timer
~ rand : Random
- lowSpeed : int
- highSpeed : Car
- cars : ArrayList<Car>
- carPath : ArrayList<Double>
- instance : RaceTrack
─────────────────────────────────────────────
+ getWinner() : Car
+ toString() : String
+ getCars() : ArrayList<Car>
+ getTrack() : ArrayList<Double>
+ startRace() : void
+ getRaceTrack() : AnchorPane
+ generateRandomSpeedList() : ArrayList<Integer>
- shiftCheckpointList(list : ArrayList<Double>) : ArrayList<Double>
+ RaceTrack()
+ getInstance() : RaceTrack
+ newInstance() : RaceTrack
```

```
SceneController
─────────────────────────────────────────────
- instance : SceneController
- startButton : Button
- raceTrackContainer : BorderPane
─────────────────────────────────────────────
+ winAlert(winner : Car, lowSpeed : int) : void
+ startRaceButton(event : ActionEvent) : void
+ initialize(arg0 : URL, arg1 : ResourceBundle) : void
+ getInstance() : SceneController
+ SceneController()
```

```
Tester
─────────────────────────────────────────────
─────────────────────────────────────────────
+ main(args : String[]) : void
+ start(stage : Stage) : void
```

```
RaceSpectator
─────────────────────────────────────────────
─────────────────────────────────────────────
+getCarsInfo() : ArrayList<String>
+getRaceStatus() : String
+getTrackDetails() : String
```