P. ADSOFT – PRÁCTICA 4

Alejandro Hurtado y Camilo Jené

28/04/2021

—

Análisis y Diseño de Software

—

Profesor: Pablo Sánchez

INTRODUCCIÓN

En esta práctica vamos a crear un sistema de carreras al estilo de Mario Kart para comenzar a implementar las interfaces. Por cada apartado, nos introducen unas interfaces mediante las cuales debemos partir para crear la funcionalidad que nos piden.

Algunos aspectos a tener en cuenta son que en la siguiente página obtenemos la información necesaria para modificar los números y que salgan con dos decimales en la pantalla:

https://es.stackoverflow.com/questions/144301/mostrar-número-con-dos-decimales

En los siguientes apartados de la memoria, mencionaremos algunas cosas a tener en cuenta a la hora de realizar los apartados.

APARTADO 1

Este apartado es el comienzo de la práctica. En él nos dan la interfaz **IVehicle** con la cual debemos crear 3 tipos de vehículos: Car, Motorcycle y Truck. A su vez nos piden comenzar la clase **Race**. A partir de ahora todas las clases relacionadas con **Race** las meteremos en el mismo paquete src/race, al igual que las clases relacionadas con vehículos irán en el paquete src/vehículos.

Para la lectura de la carrera, creamos la clase **RaceReader**, que introducimos en el paquete src/race. En un principio lo intentamos realizar con un HashMap para evitar el uso de (string).equals, pero tras realizar el apartado 3 vimos que no puede ser ya que no crea nuevos vehículos, sino que el vehículo que extrae es el mismo siempre, por lo tanto, aunque no es la implementación más correcta, lo hemos realizado con if's. El problema: cuando tengamos que introducir un número elevado de vehículos, nos queda un if demasiado grande.

Otro de los aspectos importantes de este apartado es la creación de una nueva excepción para el tratamiento de errores en el caso de que la longitud de la carrera sea menor a la velocidad máxima de cada vehículo. Esta excepción se ha denominado **RaceException**.

APARTADO 2

A continuación, comenzamos a implementar la simulación de la carrera. Para ello creamos el método **simulate()**, en el cuál vamos a imprimir las posiciones de la carrera en cada turno. Lo más importante de este apartado es que hay que actualizar la posición de cada vehículo en cada turno. Para ello nos muestran unas normas para coches y camiones ya que hay un porcentaje de probabilidad en los que éstos se ven penalizados. Para ello, sobrescribimos el método **actualizarPosicion()** en **Car** y **Truck** cada uno con su norma principal. Un ejemplo en **Car:**

```
      int probabilidad = (int)(Math.random()*10+1);

      super.setActualPosition(probabilidad == 1 ? getActualPosition() + 0.9*super.getMaxS
peed() : getActualPosition() + super.getMaxSpeed());
```

Cabe mencionar el uso de **Math.random()** para generar un numero aleatorio entre 1 y 10 ya que queremos obtener el 10%.

APARTADO 3

En este apartado implementaremos el ataque entre vehículos. Para ello, introduciremos los componentes en el juego y en caso de ataque, serán éstos los que se verán afectados. Uno de los atributos más importantes a tener en cuenta es: **isCritical**, que te indica si en caso de avería el vehículo se puede mover o no.

El mayor problema que nos hemos encontrado aquí ha sido a la hora de implementar de forma correcta la lectura de componentes en **RaceReader.java**. Lo hemos intentado hacer con un **HashMap** pero no era la forma correcta, por ello, al final nos hemos decantado por realizarlo con if's que no es lo idóneo pero sí funcional.

APARTADO 4

En este apartado vamos a introducir los powerUps, un método para dar ventaja a cada vehículo según que power up active. Para ello nos piden realizar un 3er **PowerUp** a parte del **Swap** y del **AttackAll**. Nuestra idea ha sido crear la clase **SpeedUp**, un powerUp que hará que el vehículo que lo active mueva por 2 en ese turno.

Cabe mencionar que para el correcto funcionamiento hemos creado un método abstracto en **PowerUp** denominado **realizarAccion()** en el que cada powerUp realiza la acción correspondiente. Por ejemplo, en el caso de la clase creada por nosotros realiza lo siguiente:

```
	@Override
	public void realizarAccion(List<Vehiculo> vehiculos, IVehicle v) {
		v.setActualPosition(v.getActualPosition() + v.getMaxSpeed());
	}
```

La salida de este apartado es la siguiente:

```
--------------------
Starting turn:1
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 0.0.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 0,00
 Car(1) distance to Motorcycle(3): 0,00
Truck(2). Speed 6.0. Actual position 0.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 0,00
 Truck(2) distance to Motorcycle(3): 0,00
Motorcycle(3). Speed 9.0. Actual position 0.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: false. Is critical: true.
 Motorcycle(3) distance to Car(1): 0,00
 Motorcycle(3) distance to Truck(2): 0,00

>>> Not attacking turn.
```

Turn with no powerups

Ending Turn: 1

Starting turn:2
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 7.0.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
Car(1) distance to Truck(2): 1,00
Car(1) distance to Motorcycle(3): 2,00
Truck(2). Speed 6.0. Actual position 6.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
Truck(2) distance to Car(1): 1,00
Truck(2) distance to Motorcycle(3): 3,00
Motorcycle(3). Speed 9.0. Actual position 9.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: false. Is critical: true.
Motorcycle(3) distance to Car(1): 2,00
Motorcycle(3) distance to Truck(2): 3,00

>>> Not attacking turn.

Turn with no powerups

Ending Turn: 2

Starting turn:3
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 14.0.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
Car(1) distance to Truck(2): 2,00
Car(1) distance to Motorcycle(3): 4,00
Truck(2). Speed 6.0. Actual position 12.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
Truck(2) distance to Car(1): 2,00
Truck(2) distance to Motorcycle(3): 6,00
Motorcycle(3). Speed 9.0. Actual position 18.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: false. Is critical: true.
Motorcycle(3) distance to Car(1): 4,00
Motorcycle(3) distance to Truck(2): 6,00

>>> Starting attack phase

```
Car(1) fails attack.
Truck(2) attacks Car(1) Window.
Motorcycle(3) can not attack.

Vehicle: Car(1) applying power-up: SpeedUpPowerUp
Vehicle: Truck(2) applying power-up: AttackAllPowerUp
Vehicle: Motorcycle(3) applying power-up: SwapPowerUp
Car(1) Window is being repaired 1/2.
Motorcycle(3) Wheels is being repaired 1/3.

Ending Turn: 3
--------------------

--------------------
Starting turn:4
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 25.0.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: true. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 7,00
 Car(1) distance to Motorcycle(3): 4,00
Truck(2). Speed 6.0. Actual position 18.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 7,00
 Truck(2) distance to Motorcycle(3): 3,00
Motorcycle(3). Speed 9.0. Actual position 21.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: true. Is critical: true.
 Motorcycle(3) distance to Car(1): 4,00
 Motorcycle(3) distance to Truck(2): 3,00

>>> Not attacking turn.

Turn with no powerups
Car(1) Window is being repaired 2/2.
Motorcycle(3) Wheels is being repaired 2/3.

Ending Turn: 4
--------------------

--------------------
Starting turn:5
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 32.0.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 8,00
 Car(1) distance to Motorcycle(3): 11,00
Truck(2). Speed 6.0. Actual position 24.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
```

```
 Truck(2) distance to Car(1): 8,00
 Truck(2) distance to Motorcycle(3): 3,00
Motorcycle(3). Speed 9.0. Actual position 21.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: true. Is critical: true.
 Motorcycle(3) distance to Car(1): 11,00
 Motorcycle(3) distance to Truck(2): 3,00

>>> Not attacking turn.

Turn with no powerups
Motorcycle(3) Wheels is being repaired 3/3.

Ending Turn: 5
--------------------

--------------------
Starting turn:6
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 39.0.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 9,00
 Car(1) distance to Motorcycle(3): 9,00
Truck(2). Speed 6.0. Actual position 30.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 9,00
 Truck(2) distance to Motorcycle(3): 0,00
Motorcycle(3). Speed 9.0. Actual position 30.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: false. Is critical: true.
 Motorcycle(3) distance to Car(1): 9,00
 Motorcycle(3) distance to Truck(2): 0,00

>>> Starting attack phase
Car(1) fails attack.
Truck(2) attacks Motorcycle(3) Wheels.
Motorcycle(3) can not attack.

Vehicle: Car(1) applying power-up: SwapPowerUp
Vehicle: Truck(2) applying power-up: SwapPowerUp
Vehicle: Motorcycle(3) applying power-up: SwapPowerUp
Motorcycle(3) Wheels is being repaired 1/3.

Ending Turn: 6
--------------------

--------------------
Starting turn:7
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 37.0.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
```

```
 Car(1) distance to Truck(2): 1,00
 Car(1) distance to Motorcycle(3): 2,00
Truck(2). Speed 6.0. Actual position 36.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 1,00
 Truck(2) distance to Motorcycle(3): 3,00
Motorcycle(3). Speed 9.0. Actual position 39.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: true. Is critical: true.
 Motorcycle(3) distance to Car(1): 2,00
 Motorcycle(3) distance to Truck(2): 3,00

>>> Not attacking turn.

Turn with no powerups
Motorcycle(3) Wheels is being repaired 2/3.

Ending Turn: 7
--------------------

--------------------
Starting turn:8
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 43.3.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 1,30
 Car(1) distance to Motorcycle(3): 4,30
Truck(2). Speed 6.0. Actual position 42.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 1,30
 Truck(2) distance to Motorcycle(3): 3,00
Motorcycle(3). Speed 9.0. Actual position 39.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: true. Is critical: true.
 Motorcycle(3) distance to Car(1): 4,30
 Motorcycle(3) distance to Truck(2): 3,00

>>> Not attacking turn.

Turn with no powerups
Motorcycle(3) Wheels is being repaired 3/3.

Ending Turn: 8
--------------------

--------------------
Starting turn:9
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 50.3.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
```

```
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 2,30
 Car(1) distance to Motorcycle(3): 2,30
Truck(2). Speed 6.0. Actual position 48.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 2,30
 Truck(2) distance to Motorcycle(3): 0,00
Motorcycle(3). Speed 9.0. Actual position 48.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: false. Is critical: true.
 Motorcycle(3) distance to Car(1): 2,30
 Motorcycle(3) distance to Truck(2): 0,00

>>> Starting attack phase
Car(1) fails attack.
Truck(2) attacks Motorcycle(3) Wheels.
Motorcycle(3) can not attack.

Vehicle: Car(1) applying power-up: SwapPowerUp
Vehicle: Truck(2) applying power-up: SpeedUpPowerUp
Vehicle: Motorcycle(3) applying power-up: SpeedUpPowerUp
Motorcycle(3) Wheels is being repaired 1/3.

Ending Turn: 9
--------------------

--------------------
Starting turn:10
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 57.3.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 2,70
 Car(1) distance to Motorcycle(3): 0,30
Truck(2). Speed 6.0. Actual position 60.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 2,70
 Truck(2) distance to Motorcycle(3): 3,00
Motorcycle(3). Speed 9.0. Actual position 57.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: true. Is critical: true.
 Motorcycle(3) distance to Car(1): 0,30
 Motorcycle(3) distance to Truck(2): 3,00

>>> Not attacking turn.

Turn with no powerups
Motorcycle(3) Wheels is being repaired 2/3.

Ending Turn: 10
--------------------
```

```
--------------------
Starting turn:11
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 63.599999999999994.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 2,40
 Car(1) distance to Motorcycle(3): 6,60
Truck(2). Speed 6.0. Actual position 66.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 2,40
 Truck(2) distance to Motorcycle(3): 9,00
Motorcycle(3). Speed 9.0. Actual position 57.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: true. Is critical: true.
 Motorcycle(3) distance to Car(1): 6,60
 Motorcycle(3) distance to Truck(2): 9,00

>>> Not attacking turn.

Turn with no powerups
Motorcycle(3) Wheels is being repaired 3/3.

Ending Turn: 11
--------------------

--------------------
Starting turn:12
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 70.6.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 1,40
 Car(1) distance to Motorcycle(3): 4,60
Truck(2). Speed 6.0. Actual position 72.0.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 1,40
 Truck(2) distance to Motorcycle(3): 6,00
Motorcycle(3). Speed 9.0. Actual position 66.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: false. Is critical: true.
 Motorcycle(3) distance to Car(1): 4,60
 Motorcycle(3) distance to Truck(2): 6,00

>>> Starting attack phase
Car(1) fails attack.
Truck(2) fails attack.
Motorcycle(3) can not attack.
```

```
Vehicle: Car(1) applying power-up: SwapPowerUp
Vehicle: Truck(2) applying power-up: SpeedUpPowerUp
Vehicle: Motorcycle(3) applying power-up: AttackAllPowerUp
Car(1) Wheels is being repaired 1/3.
Truck(2) Engine is being repaired 1/3.

Ending Turn: 12
--------------------

--------------------
Starting turn:13
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 72.0.
-> Wheels. Is damaged: true. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 4,60
 Car(1) distance to Motorcycle(3): 3,00
Truck(2). Speed 6.0. Actual position 76.6.
-> Engine. Is damaged: true. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 4,60
 Truck(2) distance to Motorcycle(3): 1,60
Motorcycle(3). Speed 9.0. Actual position 75.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: false. Is critical: true.
 Motorcycle(3) distance to Car(1): 3,00
 Motorcycle(3) distance to Truck(2): 1,60

>>> Not attacking turn.

Turn with no powerups
Car(1) Wheels is being repaired 2/3.
Truck(2) Engine is being repaired 2/3.

Ending Turn: 13
--------------------

--------------------
Starting turn:14
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 72.0.
-> Wheels. Is damaged: true. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
 Car(1) distance to Truck(2): 4,60
 Car(1) distance to Motorcycle(3): 12,00
Truck(2). Speed 6.0. Actual position 76.6.
-> Engine. Is damaged: true. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
 Truck(2) distance to Car(1): 4,60
 Truck(2) distance to Motorcycle(3): 7,40
Motorcycle(3). Speed 9.0. Actual position 84.0.
-> Engine. Is damaged: false. Is critical: true.
```

-> Wheels. Is damaged: false. Is critical: true.
Motorcycle(3) distance to Car(1): 12,00
Motorcycle(3) distance to Truck(2): 7,40

>>> Not attacking turn.

Turn with no powerups
Car(1) Wheels is being repaired 3/3.
Truck(2) Engine is being repaired 3/3.

Ending Turn: 14

Starting turn:15
Race with maximum length: 100
Car(1). Speed 7.0. Actual position 79.0.
-> Wheels. Is damaged: false. Is critical: true.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
Car(1) distance to Truck(2): 3,60
Car(1) distance to Motorcycle(3): 14,00
Truck(2). Speed 6.0. Actual position 82.6.
-> Engine. Is damaged: false. Is critical: true.
-> Window. Is damaged: false. Is critical: false.
-> BananaDispenser. Is damaged: false. Is critical: false.
-> Wheels. Is damaged: false. Is critical: true.
Truck(2) distance to Car(1): 3,60
Truck(2) distance to Motorcycle(3): 10,40
Motorcycle(3). Speed 9.0. Actual position 93.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: false. Is critical: true.
Motorcycle(3) distance to Car(1): 14,00
Motorcycle(3) distance to Truck(2): 10,40

>>> Starting attack phase
Car(1) attacks Truck(2) Window.
Truck(2) attacks Motorcycle(3) Wheels.
Motorcycle(3) can not attack.

Vehicle: Car(1) applying power-up: AttackAllPowerUp
Vehicle: Truck(2) applying power-up: SpeedUpPowerUp
Vehicle: Motorcycle(3) applying power-up: SpeedUpPowerUp
Truck(2) Window is being repaired 1/2.
Truck(2) Wheels is being repaired 1/3.
Motorcycle(3) Wheels is being repaired 1/3.

Ending Turn: 15

Motorcycle(3). Speed 9.0. Actual position 102.0.
-> Engine. Is damaged: false. Is critical: true.
-> Wheels. Is damaged: true. Is critical: true.
has won the race.