Homework DEV2

The DEV team

January 11, 2016

1 Homework 1 - make a car move

Console version

- Build a `Car` class;
- Add the attribute `Position`, which will be a simple integer;
- Add the `Move` method that increments the position by one;
- Make a test program that initialises the car and moves it ten times; print the position of the car on the console at every step.

Pygame version

- Use the `Car` class to draw a pygame screen where the car moves from the left to the right of the screen.

2 Homework 2 - make a list of cars move

Console version

- Build the `Node` and `Empty` classes;
- Add the usual attributes `IsEmpty`, `Head`, and `Tail` to the classes;
- Make a test program that initialises a list of cars, and moves each of them ten times; print the position of each car on the console at every step.

Pygame version

- Add a `VerticalPosition` attribute to the car, so that each car has a different vertical position to distinguish it on the screen;
- Use the list you just implemented to draw a pygame screen where various cars move from the left to the right of the screen.

3 Homework 3 - moving along checkpoints

Console version

- Make a `Checkpoint` class, which contains only a `Position` attribute;
- Make a list of checkpoints;
- In the `Car`, the `Position` will now be a reference to a node in the list of checkpoints;
- In the `Car`, the `Move` method changes position to the `Tail`, which is the next checkpoint;
- Make a test program that initialises a list of cars, and moves them until they all reach the final checkpoint; print the position of each car (which is now a checkpoint) on the console at every step.

Pygame version

- Draw a pygame screen with the checkpoints and the cars;
- The various cars move from one checkpoint to the other (like the metro along the various stations).

4 Homework 4 - crossings

Console version

- Make a `Node2D` class, which contains attributes `TailLeft`, `TailRight`, `TailUp`, `TailDown`, and `Final`; this is effectively the same as a list but with four possible choices for the `Tail` (we call this a **matrix**);
- Make a series of checkpoints and put them into `Node2D`'s;

- In the `Car`, the `Position` will now be a reference to a `Node2D` in the matrix of checkpoints;
- In the `Car`, the `Move` method changes position to one of the `Tails`, which is the next chosen checkpoint; the choice can be random;
- Make a test program that initialises a list of cars, and moves them until they all reach a specific checkpoint with `Final == True`; print the position of each car (which is now a checkpoint) on the console at every step.

Pygame version

- Draw a pygame screen with the checkpoints and the cars;
- The various cars move from one checkpoint to the other (like the cars in the city assignment).

5 Homework 5 - bikes

Console version

- Make a `Bike` class that has the `Move` method just like the car;
- `Bike`'s are fast, so the bike moves by two tiles at a time;
- Add a `PrintPosition` method to the `Car` and the `Bike`, which prints where the vehicle is;
- Make a test program that initialises a list contains a mixture of cars and bikes, and moves them until they all reach a specific checkpoint with `Final == True`; print the position of each car or bike (which is now a checkpoint) on the console at every step.

Pygame version

- Add a `Draw` method to the `Car` and the `Bike`, which draws where the vehicle is with the proper texture; the texture is also added as an attribute of both `Car` and `Bike`;
- Draw a pygame screen with the checkpoints, the bikes and the cars;
- The various cars and bikes move from one checkpoint to the other (like the cars and boats in the city assignment).