Project 3

Faysal Khatri
CSC240
10/8/2017

Implementation Notes

1. It took a while to realize that the inputted type strings have newlines at the end. I added `\n` to the compare and got it to work.
2. Also had to look up how to convert strings to ints and floats. `stoi` and `stof` work nicely
3. Also took a bit of playing around to get the right syntax for using `this`.
4. I moved the includes from map.cpp into map.hpp and added a guard to map.hpp.

mapobject.hpp

```
//Faysal Khatri
#ifndef MAP_OBJECT
#define MAP_OBJECT
#include <string>

using namespace std;

typedef enum {CAR, TREE, POLICE, OBSTACLE, EMPTY} object_type_t;

class MapObject {
      private:
      string label;
      int xloc, yloc;
      float speed;
      int direction;
      object_type_t type;

      public:
      MapObject(string l, int x, int y, float s, int d, object_type_t t=EMPTY);
      MapObject();
      void get();
      void print();
      bool collision(const MapObject&);
      string getLabel() const;
      object_type_t getType() const;
      int getX() const;
      int getY() const;


};

object_type_t string_to_type(string type_string);
char get_map_representation(object_type_t type);
string string_from_type(object_type_t);
#endif
```

MapObject::get()

```
void MapObject::get() {
      string label, xloc, yloc, speed, direction, type;

      printf("Enter name of object: ");   getline(cin, label);
      printf("Enter x location: ");       getline(cin, xloc);
      printf("Enter y location: ");       getline(cin, yloc);
      printf("Enter speed: ");            getline(cin, speed);
      printf("Enter direction: ");        getline(cin, direction);
      printf("Enter type: ");             getline(cin, type);

      (*this).label = label;
      (*this).xloc=stoi(xloc);
      (*this).yloc=stoi(yloc);
      (*this).speed=stof(speed);
      (*this).direction=stoi(direction);
      (*this).type=string_to_type(type);

}
```

MapObjectList::get_all_of_type()

```
void MapObjectList::get_all_of_type(object_type_t target_type) {
      for (int i=0; i<objects.size(); i++) {
            if (objects.at(i).getType() == target_type)
                  objects.at(i).print();
      }
}
```

Sample Output

```
$ make
g++  --std=c++11 -c mapobject.cpp
g++  --std=c++11 -c map.cpp
g++  --std=c++11 -c mapobjectlist.cpp
g++  --std=c++11 project3.cpp mapobject.o map.o mapobjectlist.o -o project3
$ ./project3
FAYSAL KHATRI
+----+----+----+----+----+----+
|    |    |    |    |    |    |
|    |    |    |    | *  |    |
|    |    |    |    |    |    |
|    |    |    |    |    |    |
+----+----+---O+----+----+----+
|    |    |    |    |    |    |
|    |    |    |    |    |    |
|   *|    |   *|    |    |    |
|    |    |    |    |    |    |
+----+--O-+----+--#-+----+----+
|    |    |    |    |    |    |
|    |    O *  |    X    |    |
|    |    |    |    |    |    |
|    |    |    |    |    |    |
+----+----+----+----+----+----+

print():
RedCar
  Location: (10, 10)  Speed: 65  Direction: 0  Type: CAR
BlueCar
  Location: (14, 14)  Speed: 45  Direction: 270  Type: CAR
Bus
  Location: (8, 8)  Speed: 55  Direction: 90  Type: CAR
Police1
  Location: (18, 18)  Speed: 55  Direction: 180  Type: POLICE
Tree1
  Location: (22, 22)  Speed: 0  Direction: 180  Type: TREE
Tree2
```

```
  Location: (4, 4)  Speed: 0  Direction: 180  Type: TREE
Tree3
  Location: (12, 12)  Speed: 0  Direction: 180  Type: TREE
Tree4
  Location: (14, 14)  Speed: 0  Direction: 180  Type: TREE
Mattress1
  Location: (20, 20)  Speed: 0  Direction: 180  Type: OBSTACLE

get_all_of_type(CAR):
RedCar
  Location: (10, 10)  Speed: 65  Direction: 0  Type: CAR
BlueCar
  Location: (14, 14)  Speed: 45  Direction: 270  Type: CAR
Bus
  Location: (8, 8)  Speed: 55  Direction: 90  Type: CAR

find("RedCar")
RedCar
  Location: (10, 10)  Speed: 65  Direction: 0  Type: CAR

collision() test:
TEST1: SUCCESS!
TEST2: SUCCESS!
FAYSAL KHATRI
```