Project 2

Faysal Khatri
CSC240
9/24/2017

Implementation Notes

1. I struggled most with converting between the object type enum value and string. First, I assumed input to be in upper case, but your input file uses lower case. So I found `strupr()` on the Internet – but this method isn't part of the ANSI standard. I wrote my own `toUpper()`, but then noticed that you required a `const char*` in the `string_to_type()` signature. So I left it and checked for lower and upper cases, but not a combination.
2. In your `object_from_string()`, you use `atof()` for direction, but direction is an int. I guess the compiler figures out how to cast it. I used `atoi()` for direction in `get_object()`.
3. I also struggled with the map. I read `print_map()` to find out that it expects -1 when nothing exists.
4. Another issue is the map does not indicate when >1 object exists in a location.
5. Printing the user-entered objects is another issue. `fgets()` saves a new line at the end of each input string. This new line gets printed out when I print the label, making it a little less tidy.

mapobject.h

```
/*Faysal Khatri*/
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

#define LABEL_LEN 10

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

typedef struct {
	char label[LABEL_LEN];
	int xloc;
	int yloc;
	float speed;
	int direction;
	object_type_t type;
} object_t;

void get_object(object_t * obj);
void print_object(const object_t * obj);
char get_map_representation(object_type_t type);
object_type_t string_to_type(const char * type_string);
char* stringFromType(object_type_t type);
```

get_object()

```
void get_object(object_t * obj) {
      char label[LABEL_LEN];  /*buffers for user input strings*/
      char xloc[5], yloc[5], speed[5], direction[5];
      char type[8]; /*OBSTACLE is the longest type*/

      printf("Enter name of object: ");  fgets(label, LABEL_LEN, stdin);
      printf("Enter x location: ");  fgets(xloc, 15, stdin);
      printf("Enter y location: ");  fgets(yloc, 15, stdin);
      printf("Enter speed: ");            fgets(speed, 15, stdin);
      printf("Enter direction: "); fgets(direction, 15, stdin);
      printf("Enter type: ");       fgets(type, 9, stdin);

      strncpy(obj->label, label, LABEL_LEN);
      obj->xloc=atoi(xloc);
      obj->yloc=atoi(yloc);
      obj->speed=atof(speed);
      obj->direction=atoi(direction);
      obj->type=string_to_type(type);

    print_object(obj);
}
```

find_all_of_type()

```
void find_all_of_type(const node_t * start, object_type_t target) {
      const node_t * curr;
      char *strTarget = stringFromType(target);
      object_t *currObj;
      curr = start;

    while (curr != NULL)  {
            currObj = (object_t *) curr->data;
            if (strcmp(strTarget, stringFromType(currObj->type)) == 0) {
                  print_object(currObj);
            }
            curr = curr->next;
    }
}
```

Sample Output

```
$ make
gcc  -Wall -pedantic -ansi -c mapobject.c
gcc  -Wall -pedantic -ansi project2.c mapobject.o basiclist.o map.o -o project2
$ ./project2
Student: Faysal Khatri
ENTRY : RedCar  Location: (10, 12)  Speed: 65.00  Direction: 0  Type: CAR
ENTRY : BlueCar  Location: (14, 5)  Speed: 45.00  Direction: 270  Type: CAR
ENTRY : Bus  Location: (8, 10)  Speed: 55.00  Direction: 90  Type: CAR
ENTRY : Police1  Location: (18, 10)  Speed: 55.00  Direction: 180  Type: POLICE
ENTRY : Tree1  Location: (22, 2)  Speed: 0.00  Direction: 180  Type: TREE
ENTRY : Tree2  Location: (4, 8)  Speed: 0.00  Direction: 180  Type: TREE
ENTRY : Tree3  Location: (12, 12)  Speed: 0.00  Direction: 180  Type: TREE
ENTRY : Tree4  Location: (14, 8)  Speed: 0.00  Direction: 180  Type: TREE
ENTRY : Mattress1  Location: (20, 12)  Speed: 0.00  Direction: 180  Type: OBSTACLE
Enter name of object: Fcar
Enter x location: 19
Enter y location: 10
Enter speed: 44.56
Enter direction: 30
Enter type: car
Fcar
  Location: (19, 10)  Speed: 44.56  Direction: 30  Type: CAR
Enter name of object: Ftree
Enter x location: 20
Enter y location: 10
Enter speed: 32.33
Enter direction: 30
Enter type: tree
Ftree
  Location: (20, 10)  Speed: 32.33  Direction: 30  Type: TREE
Printing map:
+----+----+----+----+----+----+
|    |    |    |    |    |    |
|    |    |    |    | *  |    |
|    |    |    |    |    |    |
|    |    |    |    |    |    |
+----+----+---O+----+----+----+
|    |    |    |    |    |    |
|    |    |    |    |    |    |
|   *|    |   *|    |    |    |
|    |    |    |    |    |    |
+----+--O-+----+--#O*----+----+
|    |    |    |    |    |    |
|    |    O *  |    X    |    |
|    |    |    |    |    |    |
|    |    |    |    |    |    |
+----+----+----+----+----+----+

print_objects():
Ftree
  Location: (20, 10)  Speed: 32.33  Direction: 30  Type: TREE
Fcar
  Location: (19, 10)  Speed: 44.56  Direction: 30  Type: CAR
Mattress1  Location: (20, 12)  Speed: 0.00  Direction: 180  Type: OBSTACLE
Tree4  Location: (14, 8)  Speed: 0.00  Direction: 180  Type: TREE
Tree3  Location: (12, 12)  Speed: 0.00  Direction: 180  Type: TREE
Tree2  Location: (4, 8)  Speed: 0.00  Direction: 180  Type: TREE
Tree1  Location: (22, 2)  Speed: 0.00  Direction: 180  Type: TREE
Police1  Location: (18, 10)  Speed: 55.00  Direction: 180  Type: POLICE
Bus  Location: (8, 10)  Speed: 55.00  Direction: 90  Type: CAR
BlueCar  Location: (14, 5)  Speed: 45.00  Direction: 270  Type: CAR
RedCar  Location: (10, 12)  Speed: 65.00  Direction: 0  Type: CAR

find_all_of_type(CAR):
Fcar
  Location: (19, 10)  Speed: 44.56  Direction: 30  Type: CAR
Bus  Location: (8, 10)  Speed: 55.00  Direction: 90  Type: CAR
BlueCar  Location: (14, 5)  Speed: 45.00  Direction: 270  Type: CAR
RedCar  Location: (10, 12)  Speed: 65.00  Direction: 0  Type: CAR

find_object(RedCar):
RedCar  Location: (10, 12)  Speed: 65.00  Direction: 0  Type: CAR
Student: Faysal Khatri
```