CAB202 - Microprocessors and Digital Systems

Assignment 1

Pedro Alves (n9424342)

April 20, 2018

Contents

Executive Summary

Program Overview

Things to talk about
change_state

Splash Screen

The splash screen is the first screen the player sees when they start the game. It provides basic information about the game and will change to the main game screen when the player presses any key.

Functions

```c
// main.c
void update_start_screen();
```

Called every tick of the main game loop. Will change the game's state to *GAME_SCREEN* if there is any key in the input buffer. Since the *change_state()* function already purges the input buffer, we do not have to worry about the game skipping straight to the *GAME_SCREEN*.

```c
// main.c
void draw_start_screen();
```

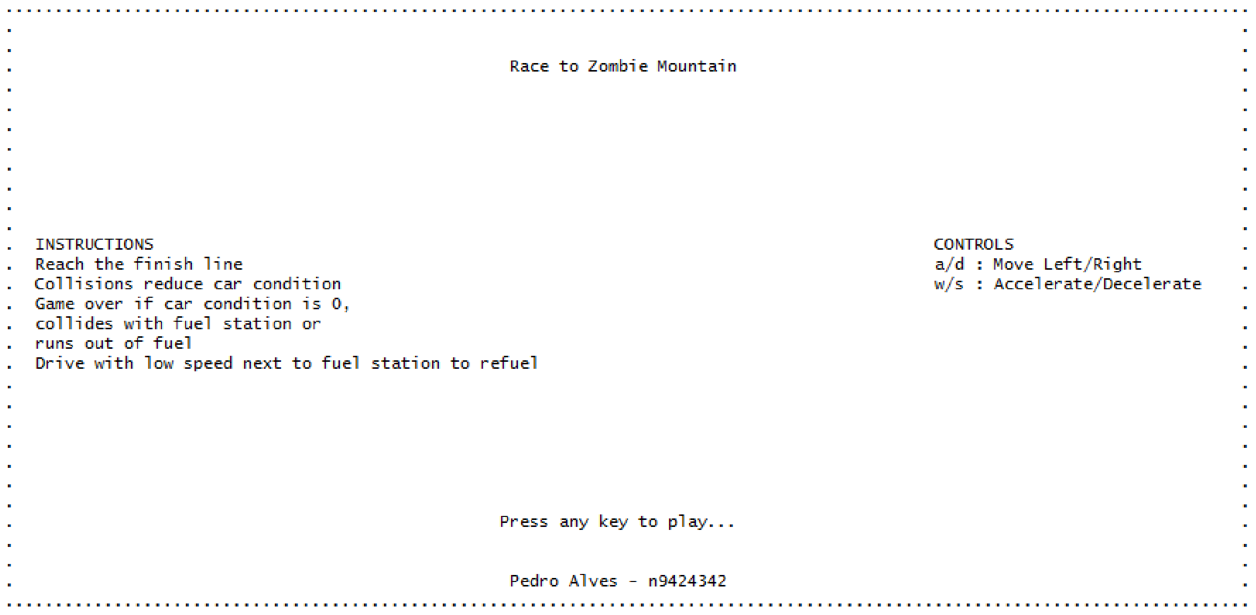
Calculates the x and y coordinates of each string to be shown based on the dimensions of the screen. Will then call *draw_string()* and *draw_center_text()* multiple times to add the strings to the desired location.

```c
// main.c
void draw_center_text(char * text, int y);
```

Calculates what x coordinate is required in order to have the text appear at the middle of the screen. Then calls *draw_string()* to print the text.

Testing

Testing that the splash screen shows up when the game is started.



```
...............................................................................................................................
.                                                                                                                             .
.                                                                                                                             .
.                                                  Race to Zombie Mountain                                                    .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.  INSTRUCTIONS                                                                              CONTROLS                         .
.  Reach the finish line                                                                     a/d : Move Left/Right            .
.  Collisions reduce car condition                                                           w/s : Accelerate/Decelerate      .
.  Game over if car condition is 0,                                                                                           .
.  collides with fuel station or                                                                                              .
.  runs out of fuel                                                                                                           .
.  Drive with low speed next to fuel station to refuel                                                                        .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                                                                                             .
.                                                 Press any key to play...                                                    .
.                                                                                                                             .
.                                                                                                                             .
.                                                  Pedro Alves - n9424342                                                     .
...............................................................................................................................
```

Figure 1: The splash screen when the player starts the game

Border

The border is simply a rectangle that is drawn on the edge of the terminal. It supports every terminal size. The *draw_borders()* functon is the last one called before *show_screen()* in the draw step of the game loop. This ensures that no other graphics ever block the border.

Globals

```
// zombiemountain.h
#define BORDER_CHAR 46
```

The character that will be used to represent the border. The number 46 represents the ASCII character "." (full stop).

Functions

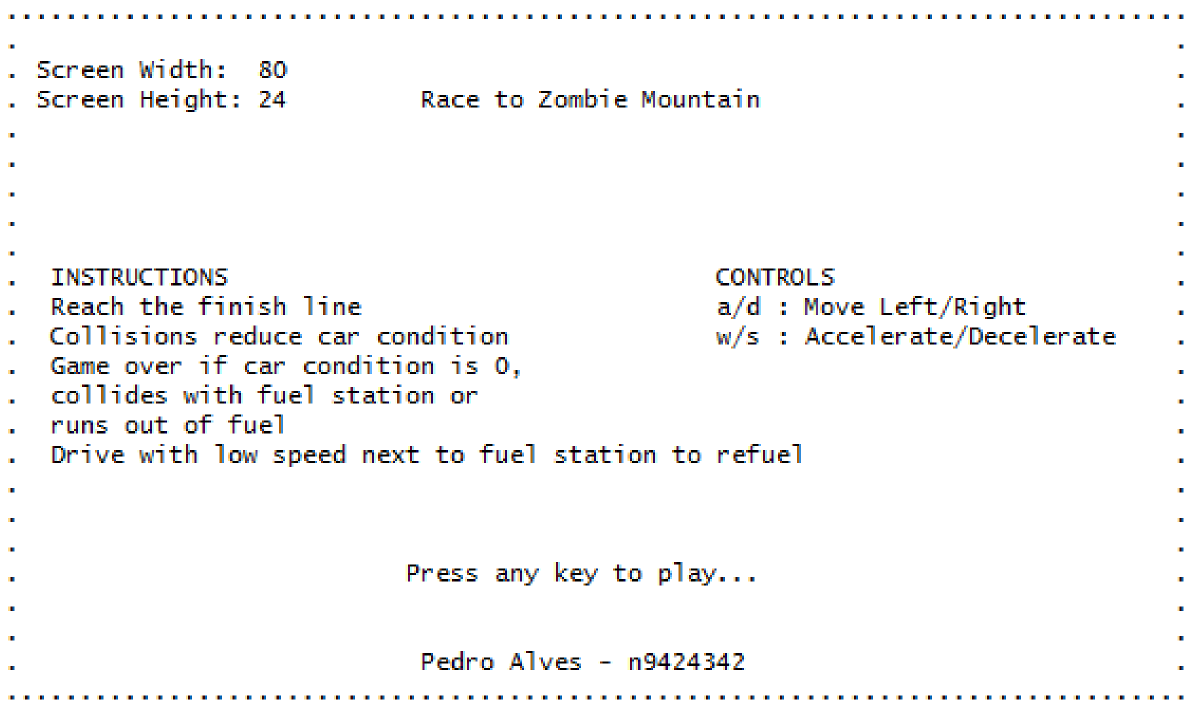
```
// main.c
void draw_borders();
```

Draws 4 lines that form a rectangle on the edge of the screen. The length of these lines are calculated by using the screen width and height in order to make the borders work on every screen size.

Testing

The game is started in different sized terminals and the borders are verified to have been drawn correctly.

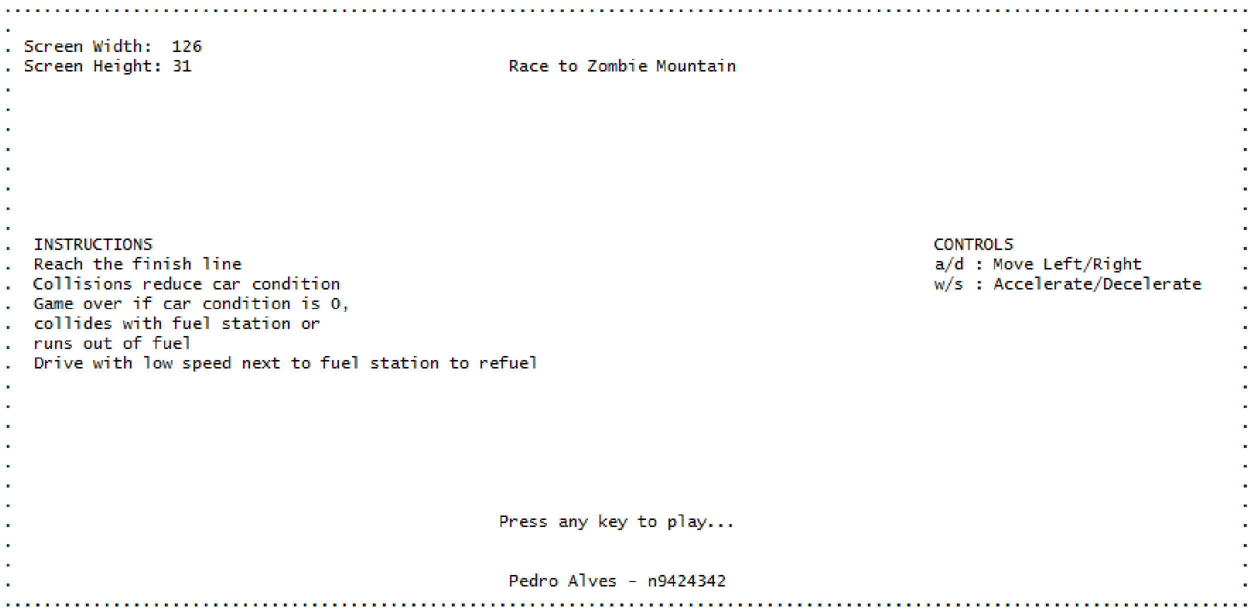
Screen: 80x24



```
................................................................................
.                                                                              .
. Screen Width:   80                                                           .
. Screen Height: 24         Race to Zombie Mountain                            .
.                                                                              .
.                                                                              .
.                                                                              .
.                                                                              .
.                                                                              .
.  INSTRUCTIONS                                 CONTROLS                       .
.  Reach the finish line                        a/d : Move Left/Right          .
.  Collisions reduce car condition              w/s : Accelerate/Decelerate    .
.  Game over if car condition is 0,                                            .
.  collides with fuel station or                                               .
.  runs out of fuel                                                            .
.  Drive with low speed next to fuel station to refuel                         .
.                                                                              .
.                                                                              .
.                                                                              .
.                          Press any key to play...                            .
.                                                                              .
.                                                                              .
.                           Pedro Alves - n9424342                             .
................................................................................
```

Figure 2: The border with screen dimensions of 80x24

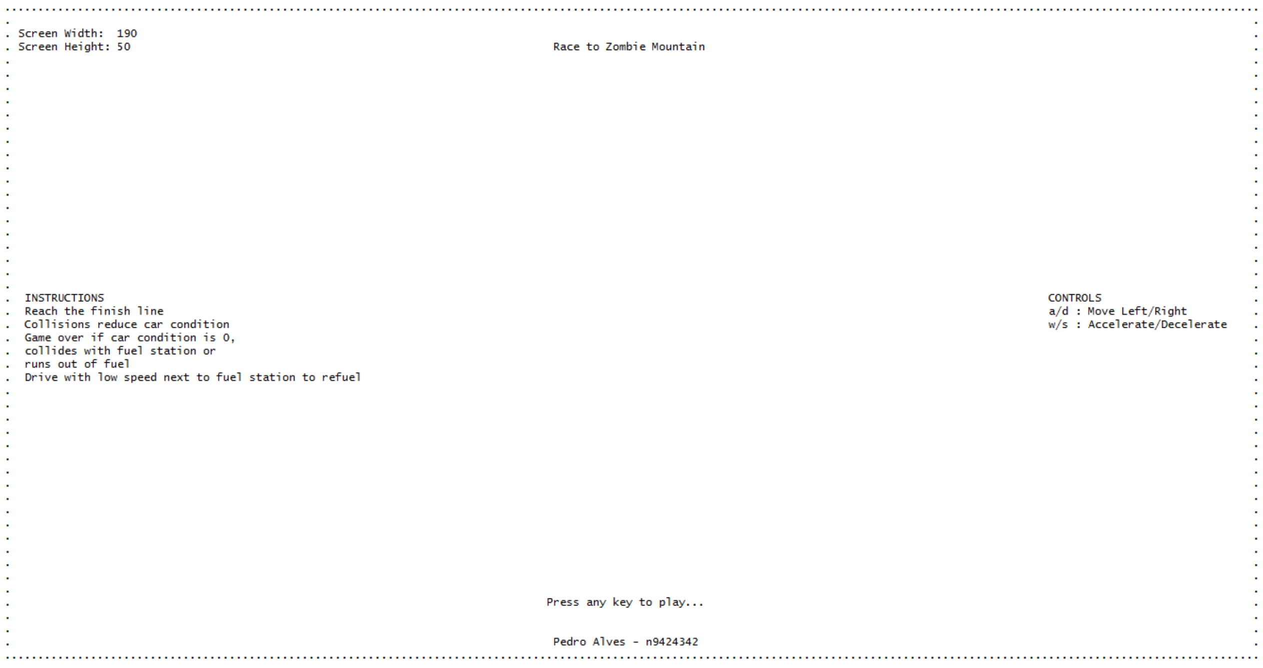
Screen: 126x31



```
. Screen Width:  126
. Screen Height: 31                               Race to Zombie Mountain

.  INSTRUCTIONS                                                                                  CONTROLS
.  Reach the finish line                                                                         a/d : Move Left/Right
.  Collisions reduce car condition                                                               w/s : Accelerate/Decelerate
.  Game over if car condition is 0,
.  collides with fuel station or
.  runs out of fuel
.  Drive with low speed next to fuel station to refuel

                                                 Press any key to play...

                                                  Pedro Alves - n9424342
```

Figure 3: The border with screen dimensions of 126x31

Screen: 190x50



```
. Screen Width:  190
. Screen Height: 50                               Race to Zombie Mountain

.  INSTRUCTIONS                                                                                  CONTROLS
.  Reach the finish line                                                                         a/d : Move Left/Right
.  Collisions reduce car condition                                                               w/s : Accelerate/Decelerate
.  Game over if car condition is 0,
.  collides with fuel station or
.  runs out of fuel
.  Drive with low speed next to fuel station to refuel

                                                 Press any key to play...

                                                  Pedro Alves - n9424342
```

Figure 4: The border with screen dimensions of 190x50

Dashboard

A sub-window in the terminal which displays data regarding the player's car such as condition, speed and fuel as well as displaying stats on the game itself such as time spent and total distance travelled.
Warnings also appear on the dashboard to notify the player that the car is offroad or is refuelling.

Globals

```c
// zombiemountain.h
int dashboard_x;
```

The x-coordinate of the border between the dashboard and the playing area.

```c
// obstacles.h
#define DASHBOARD_SIZE   20
```

The width of the dashboard.

```c
// zombiemountain.h
#define DASHBOARD_BORDER_CHAR 47
```

The ASCII character that will represent the border that separates the playing area and the dashboard.

```c
// zombiemountain.h
int speed;
```

The current speed of the player.

```c
// zombiemountain.h
int fuel;
```

The current fuel available to the player.

```c
// obstacles.h
int car_condition;
```

The condition of the car as a percentage.

```c
// hscore.h
int score;
```

The current score of the player.

```c
// zombiemountain.h
int distance_travelled;
```

The distance travelled since the start of the game.

```c
// zombiemountain.h
double game_start_time;
```

The time in milliseconds that the game started.

```c
// zombiemountain.h
timer_id refuel_timer;
```

A timer that is set when the car starts refuelling.

Functions

```c
// main.c
void draw_dashboard();
```

Draws a border between the playing area and the dashboard area. Calls *draw_string()* and *draw_int()* multiple times to print the relevant globals and their captions.
If the car is offroad or refuelling, a relevant warning will also be drawn. Additionaly for refuelling, will display how long until it is finished.

```c
// main.c
bool car_offroad();
```

Checks if any portion of the car is outside the road boundaries and return true if so.

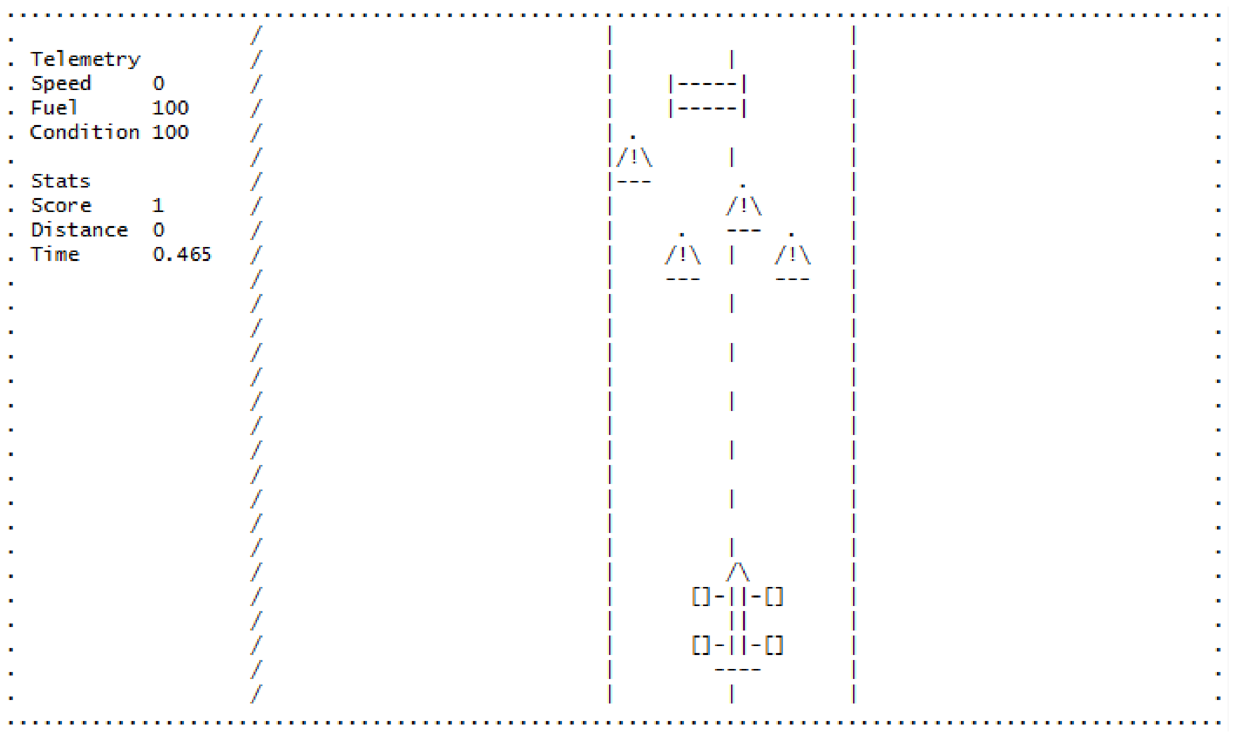
```c
// main.c
double refuel_time_left();
```

Calculates how much time is left to finish refuelling. This is done by calculating the difference between the current time and the time the *refuel_timer* is meant to reset, this is then subtracted from *3.0*.

Testing

Stats are modified with gameplay

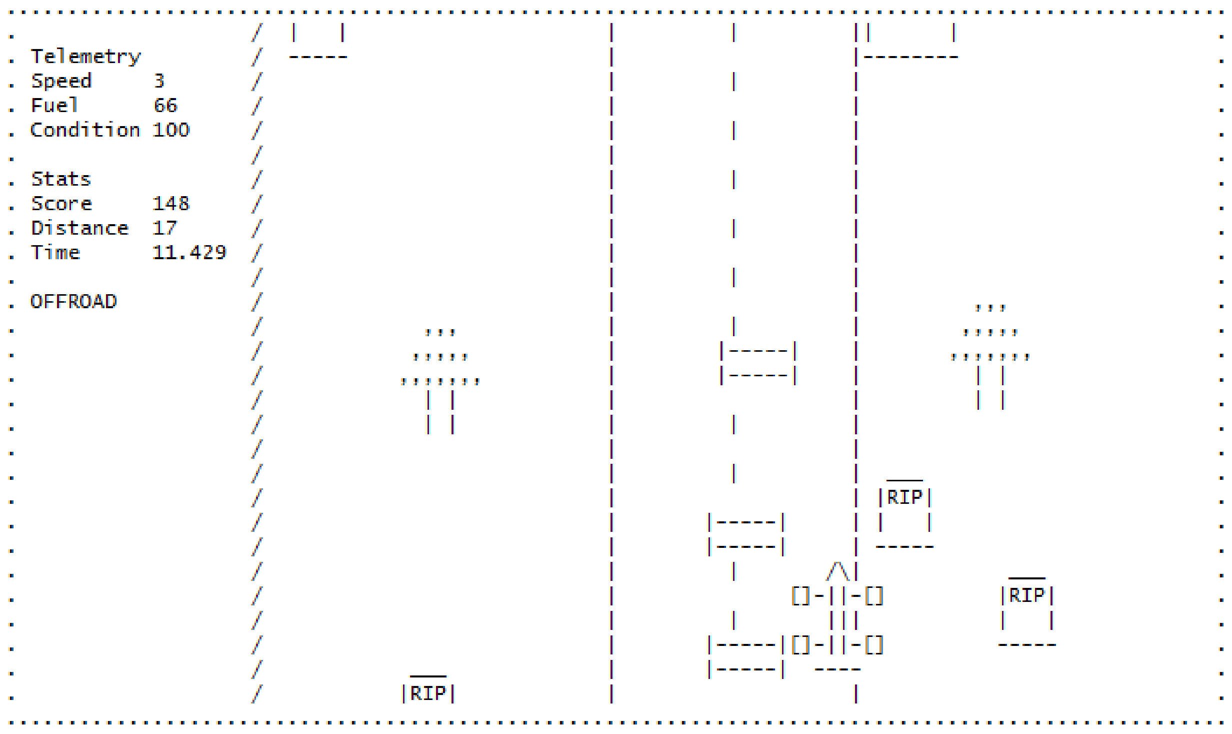
Two screenshots of the same gameplay are taken where the fuel is checked if it is decreased and the time,distance and score are increased.



```
Telemetry
Speed     0
Fuel      100
Condition 100

Stats
Score     1
Distance  0
Time      0.465
```

Figure 5: A screenshot of the game as it starts

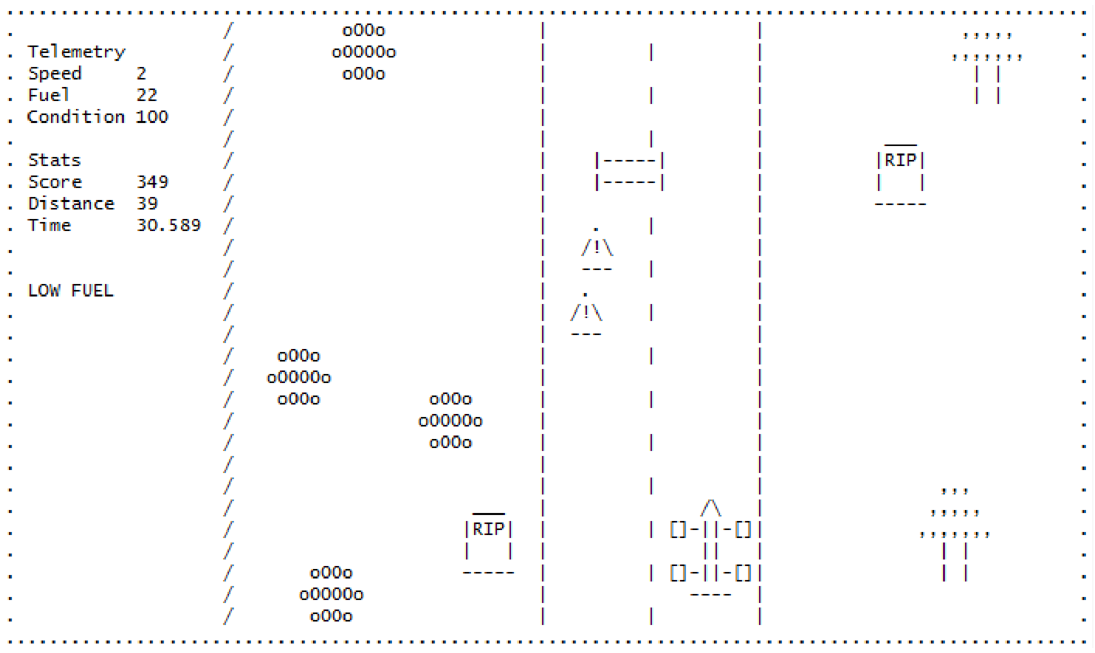
Figure 5 shows the dashboard on the left side of the terminal with a border clearly separating it from the playing area. To achieve the second part of the test, the 'w' key was pressed 3 times to test if the speed is increased and the car was moved horizontally where necessary to avoid obstacles.
The condition stat will be tested in the *Collision* section and the refuelling warning and timer will be tested in the *Fuel* section.



```
Telemetry
Speed     3
Fuel      66
Condition 100

Stats
Score     148
Distance  17
Time      11.429

OFFROAD
```

Figure 6: A screenshot of the same game sessions as the figure above but 11 seconds into gameplay

Figures 5 and 6 show that the stats are represented in the dashboard and change when supposed to. The *OFFROAD* warning also appears when the car moves beyond the border of the road.
Figure 7 shows the low fuel warning appears when fuel falls below 25%.



```
Telemetry
Speed     2
Fuel      22
Condition 100

Stats
Score     349
Distance  39
Time      30.589


LOW FUEL
```

Figure 7: Low Fuel warning appearing when fuel is below 1/4 the maximum

Race Car and Horizontal Movement

The race car is a sprite 8 units wide and 5 units tall. This sprite is always stuck in the same position with the illusion of movement given by the obstacles being moved downwards. The speed at which the obstacles move is proportional to the speed setting.

Globals

```
// imagemngr.h
#define PLAYER_WIDTH   8
```

The width of the car sprite.

```
// imagemngr.h
#define PLAYER_HEIGHT  5
```

The height of the car sprite

```
// zombiemountain.h
#define INPUT_MOVE_LEFT     'a'
```

The keyboard input that will make the car turn left.

```
// zombiemountain.h
#define INPUT_MOVE_RIGHT   'd'
```

The keyboard input that will make the car turn right

```
// zombiemountain.h
sprite_id player;
```

The car sprite which the player controls.

Functions

```
// main.c
void setup_player_car();
```

Place the car sprite in the middle of the road, 2 units above the bottom of the screen. Also sets the car condition to 100% and fuel to max.

```
// main.c
void handle_input();
```

Get the next character from the input buffer. If it is a valid key, call the specific input handler.

```
// main.c
void handle_movement_input(int key);
```

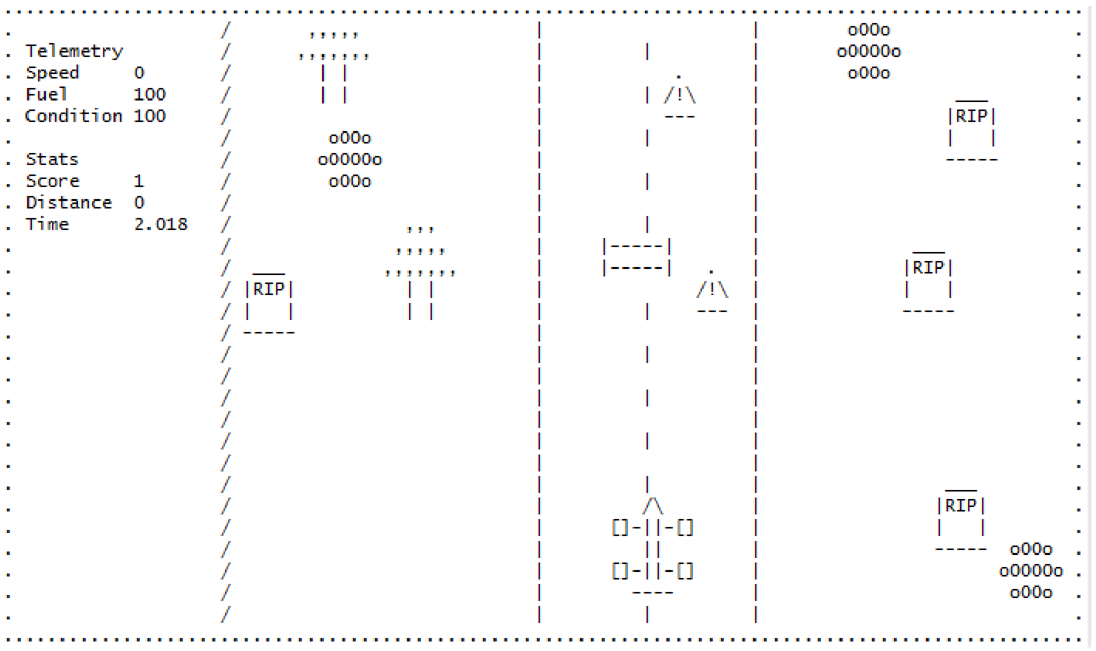
Checks if the *key* variable wants the car to turn left or right. Will then check if the car will be in the bounds of the playing area, if it'll collide laterally with any obstacle and if the speed is above zero. If all three checks pass, then the *sprite_move()* function is called.

```
// main.c
bool in_bounds(int x, int y)
```

Checks if the *(x,y)* coordinate is in bounds of the playing area, returns true if so.

Testing

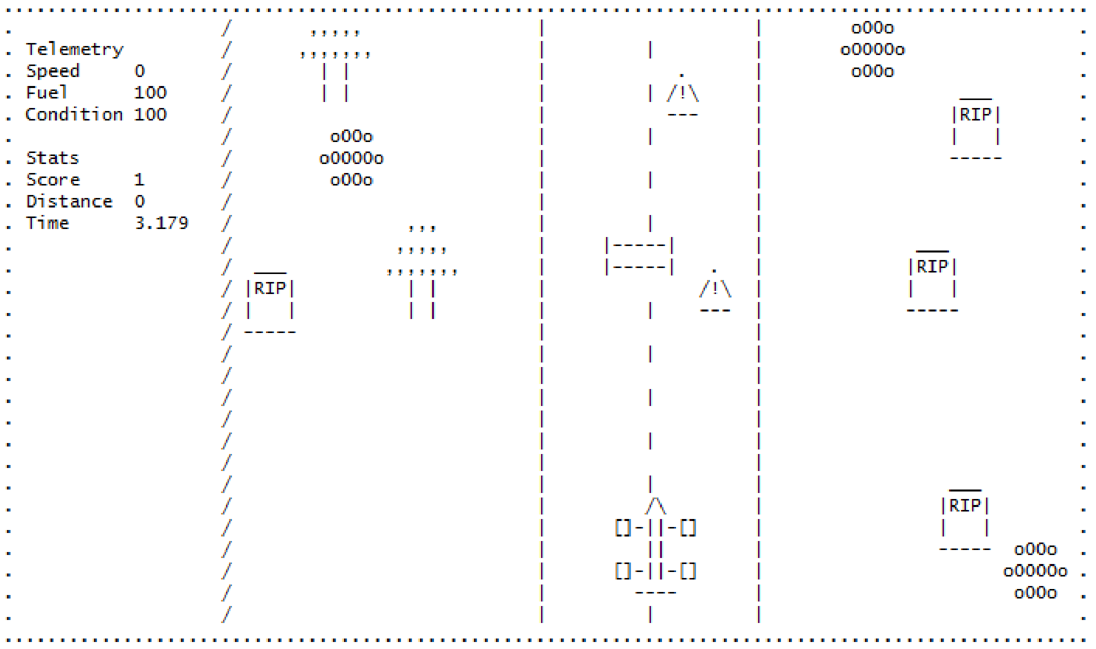
Car doesn't move when speed is 0



```
. Telemetry
. Speed     0
. Fuel      100
. Condition 100
.
. Stats
. Score     1
. Distance  0
. Time      2.018
```

Figure 8: Car in the middle of the road with speed equal zero

The following inputs were pressed and the result shown in Figure 8: "a,d,a,d"

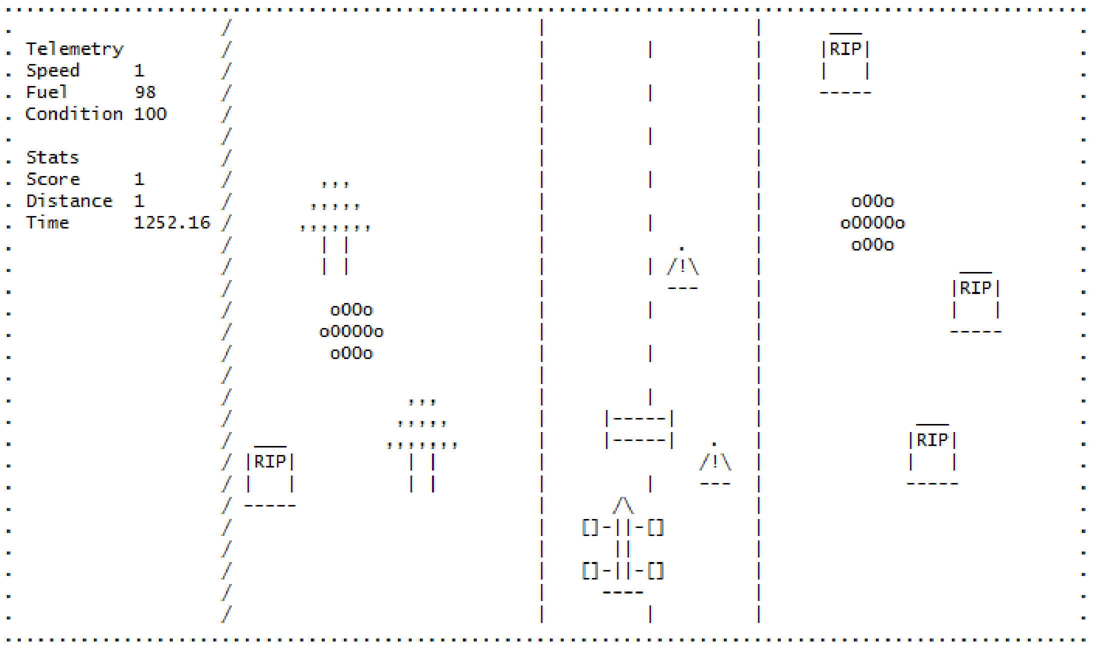


```
. Telemetry
. Speed     0
. Fuel      100
. Condition 100
.
. Stats
. Score     1
. Distance  0
. Time      3.179
```

Figure 9: Results after attempting to move horizontally with speed equal zero

Car moves left and right

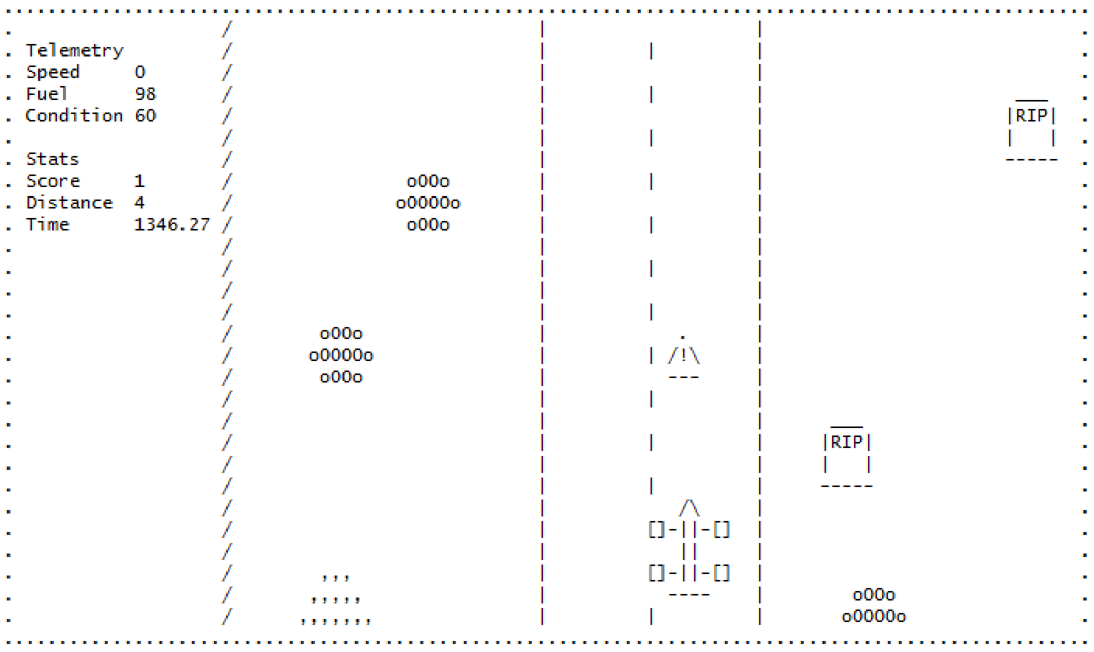
From the position in Figure 9, the following inputs were pressed ”w,a,a,a”.



```
Telemetry
Speed     1
Fuel      98
Condition 100

Stats
Score     1
Distance  1
Time      1252.16
```

Figure 10: Moving the car left with speed equal 1

After the car is reset due to the inetivable collision in Figure 10, the following inputs were pressed ”w,d,d,d”.

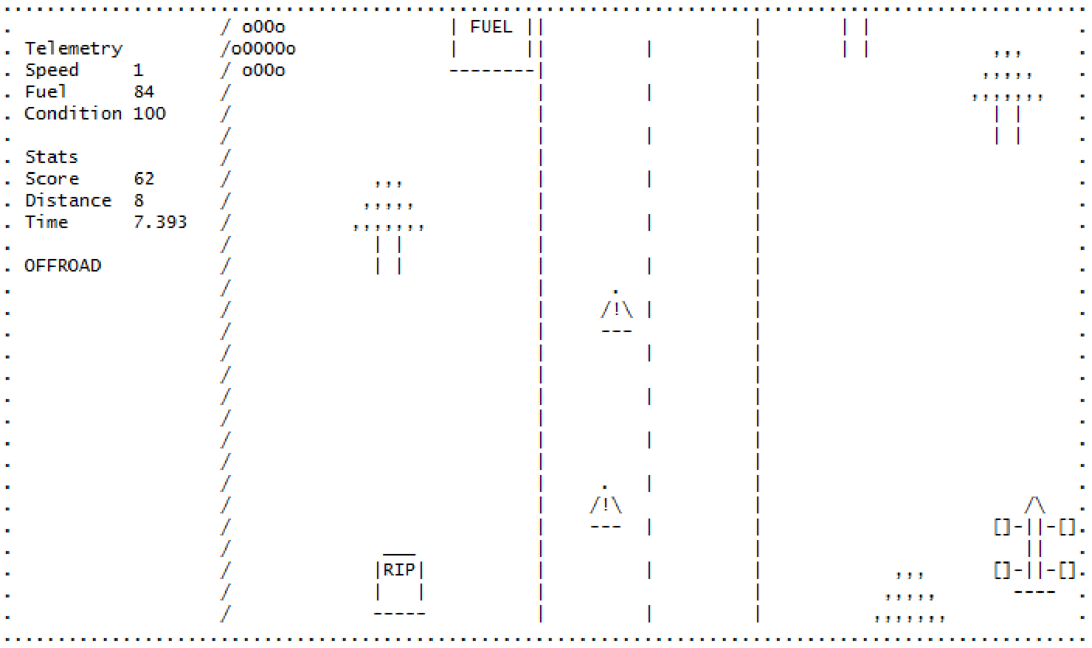


```
Telemetry
Speed     0
Fuel      98
Condition 60

Stats
Score     1
Distance  4
Time      1346.27
```

Figure 11: Moving the car right with speed equal 1 (car was stopped to grab screenshot)

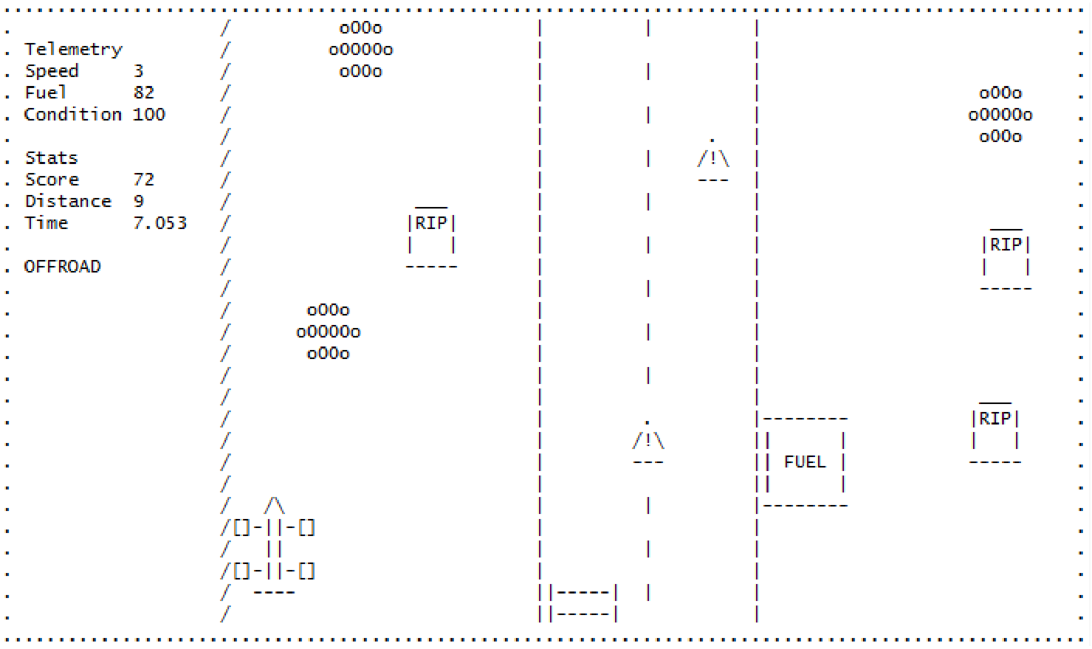
Car stays in bounds

The car was moved to both extremes of the playing area with the lateral movement input held down. Figure 12 shows the result of holding down 'd' and Figure 13 shows the result of holding down 'a'.



```
....................................................................................................
.                   / o0Oo               | FUEL ||                   |       | |                   .
. Telemetry         /o0000o              |      ||         |         |       | |                   .
. Speed     1       / o0Oo               --------|                   |                     ,,,     .
. Fuel      84      /                            |         |         |                    ,,,,,    .
. Condition 100     /                            |                   |                   ,,,,,,,   .
.                   /                            |         |         |                     | |     .
. Stats             /                            |                   |                     | |     .
. Score     62      /             ,,,            |         |         |                             .
. Distance  8       /            ,,,,,           |                   |                             .
. Time      7.393   /           ,,,,,,,          |         |         |                             .
.                   /             | |            |                   |                             .
. OFFROAD           /             | |            |         |         |                             .
.                   /                            |      .            |                             .
.                   /                            |     /!\ |         |                             .
.                   /                            |     ---           |                             .
.                   /                            |         |         |                             .
.                   /                            |                   |                             .
.                   /                            |         |         |                             .
.                   /                            |                   |                             .
.                   /                            |         |         |                             .
.                   /                            |                   |                             .
.                   /                            |     .   |         |                             .
.                   /                            |    /!\            |                        /\   .
.                   /                            |    ---  |         |                     []-||-[].
.                   /              ___           |                   |                        ||   .
.                   /             |RIP|          |         |         |            ,,,      []-||-[].
.                   /             |   |          |                   |           ,,,,,       ----  .
.                   /             -----          |         |         |          ,,,,,,,            .
....................................................................................................
```

Figure 12: Result of holding down 'd' when next to the right border



```
....................................................................................................
.                   /          o0Oo              |         |         |                             .
. Telemetry         /         o0000o             |                   |                             .
. Speed     3       /          o0Oo              |         |         |                             .
. Fuel      82      /                            |                   |                   o0Oo      .
. Condition 100     /                            |         |         |                  o0000o     .
.                   /                            |               .   |                   o0Oo      .
. Stats             /                            |         |    /!\  |                             .
. Score     72      /                            |              ---  |                             .
. Distance  9       /                 ___        |         |         |                             .
. Time      7.053   /                |RIP|       |                   |                    ___      .
.                   /                |   |       |         |         |                   |RIP|     .
. OFFROAD           /                -----       |                   |                   |   |     .
.                   /                            |         |         |                   -----     .
.                   /       o0Oo                 |                   |                             .
.                   /      o0000o                |         |         |                             .
.                   /       o0Oo                 |                   |                             .
.                   /                            |         |         |                             .
.                   /                            |                   |                   ___       .
.                   /                            |         .         |--------          |RIP|      .
.                   /                            |        /!\        ||      |          |   |      .
.                   /                            |        ---        || FUEL |          -----      .
.                   /                            |                   ||      |                     .
.                   /   /\                       |         |         |--------                     .
.                   /[]-||-[]                    |                   |                             .
.                   /   ||                       |         |         |                             .
.                   /[]-||-[]                    |                   |                             .
.                   /  ----                      ||-----|  |         |                             .
.                   /                            ||-----|            |                             .
....................................................................................................
```

Figure 13: Result of holding down 'a' when next to the left border

Acceleration and Speed

The car can accelerates and decelerates with the 'w' and 's' keys. The speed can never go negative or higher than 10. When the car is offroad (indicated by the *OFFROAD* warning in the dashboard) the speed is limited to a maximum of 3.

Globals

```
// zombiemountain.h
#define INPUT_ACCELERATE   'w'
```

The character input to accelerate the car.

```
// zombiemountain.h
#define INPUT_DECELERATE   's'
```

The character input to decelerate the car.

```
// zombiemountain.h
#define MAX_SPEED        10
```

The maximum speed the car can reach.

```
// zombiemountain.h
#define MAX_SPEED_OFFROAD  3
```

The maximum speed the car can reach while offroad.

```
// zombiemountain.h
#define SPEED_INTERVAL   87
```

Used to set the reset time for *speed_timer*.

```
// zombiemountain.h
#define LOOP_INTERVAL  17
```

Used with *SPEED_INTERVAL* and *speed_timer* to decide when to increment *speed_ctr*.

```
// zombiemountain.h
int speed;
```

The speed of the player, this affects how fast the obstacles scroll down.

```
// zombiemountain.h
int speed_ctr;
```

Is compared with the current speed to decide when to update the main game logic (increasing distance, making obstacles scroll, etc.).

```
// zombiemountain.h
timer_id speed_timer;
```

This timer controls when *speed_ctr* is increased.

Scenery and Obstacles

Fuel Depot

Fuel

Distance Travelled

Collision

Game Over Dialogue

Part B - Highscore Screen

References