CAB202 - Microprocessors and Digital Systems

Assignment 1

Pedro Alves (n9424342)

April 19, 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

```
// 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*.

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

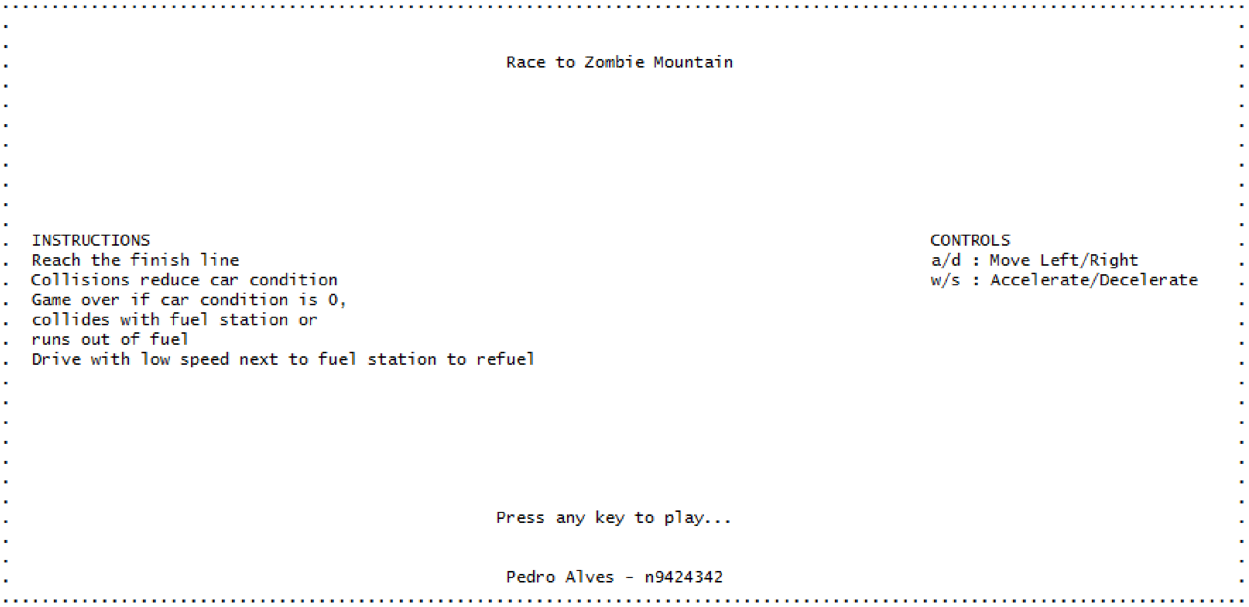
Calculates the x and y coordinates of each string to be shown based on the dimensions of the screen. Will then call *draw_string()* and *draw_center_text()* multiple times to add the strings to the desired location.

```
// 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

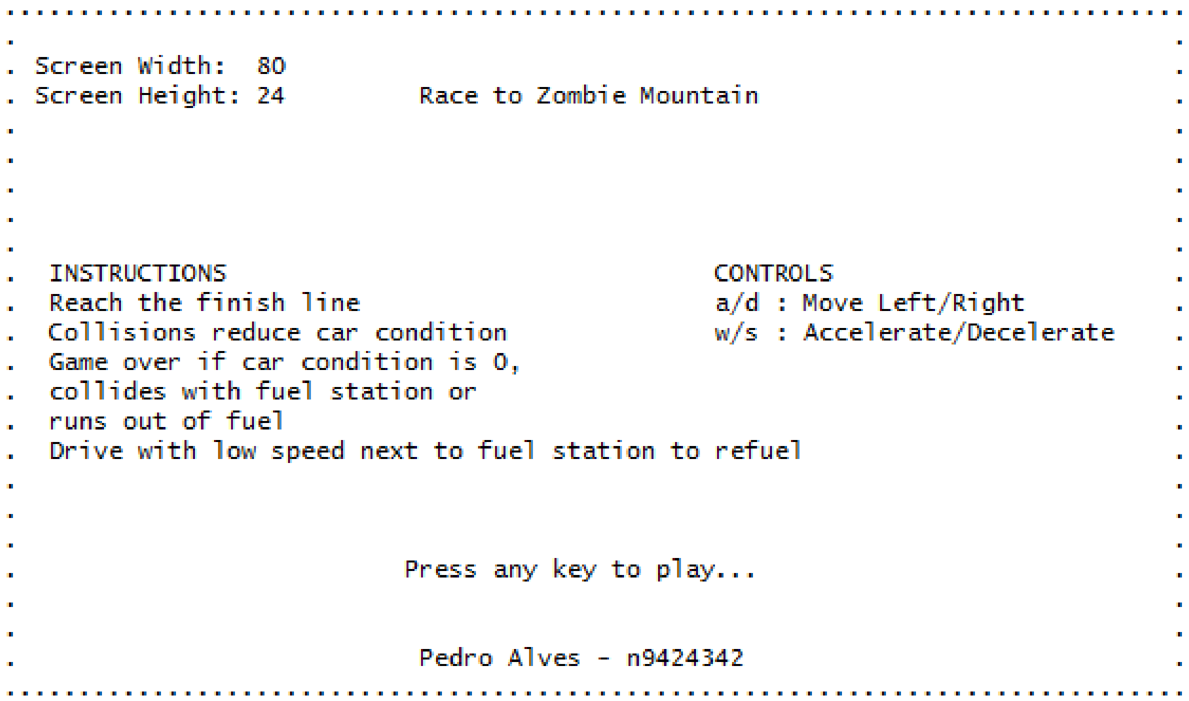
```
// main.c
void draw_borders();
```

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.

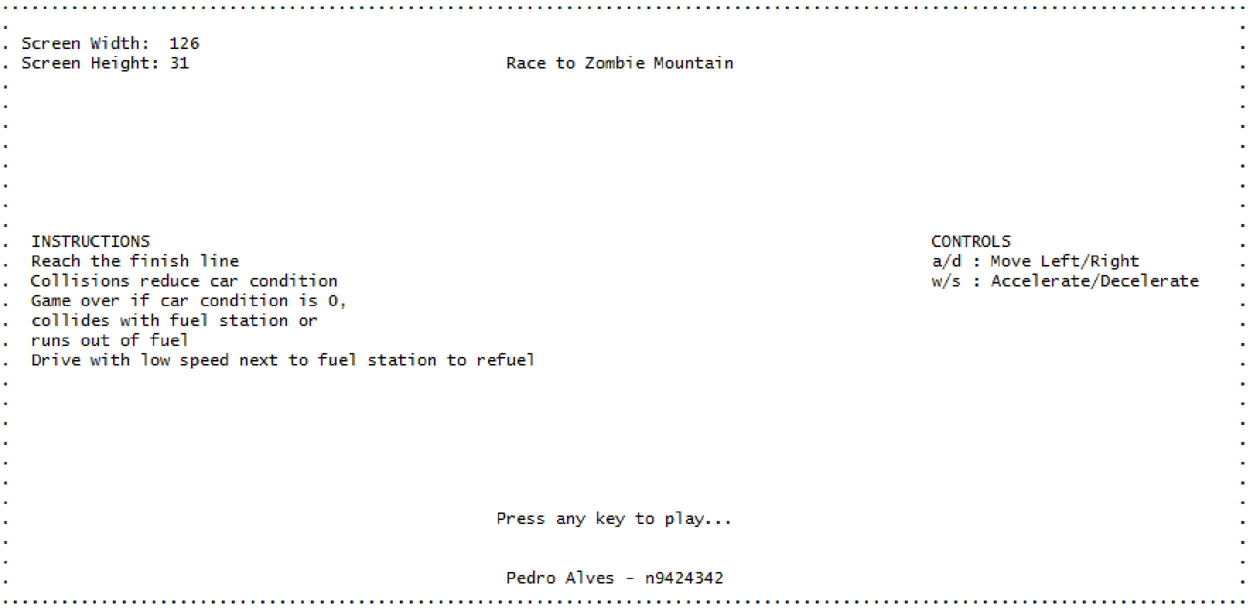
Screen: 80x24



```
................................................................................
.                                                                              .
. 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

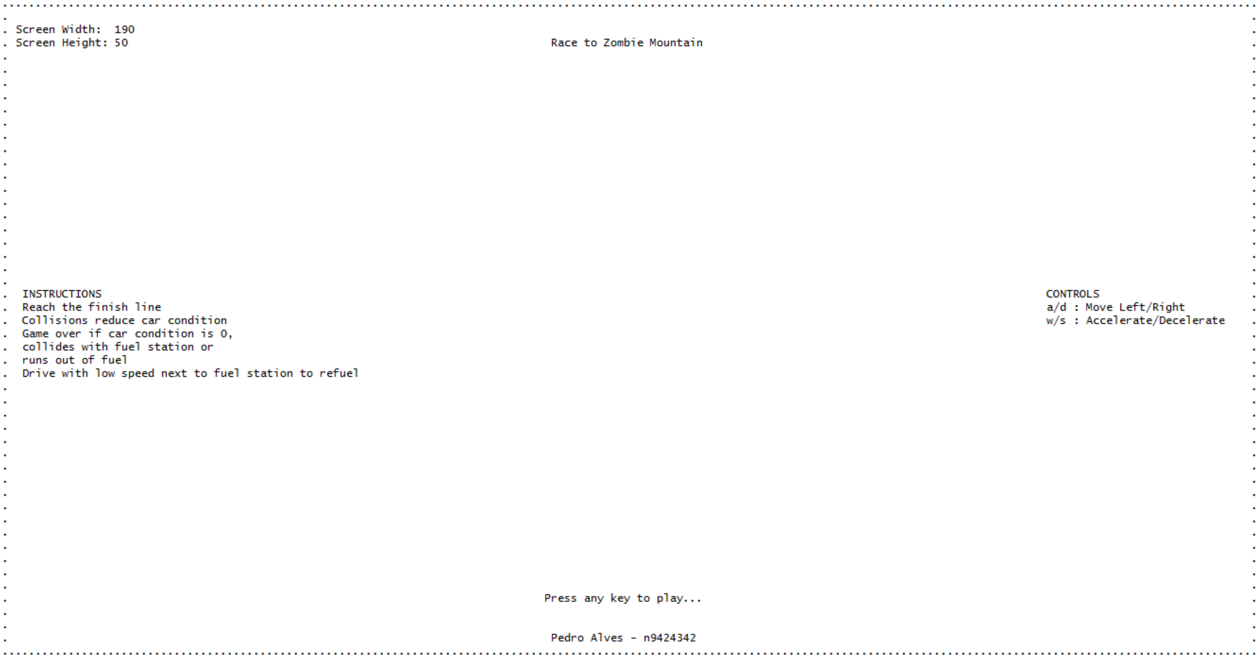
Screen: 126x31



```
 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
// 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.

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

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

```
// 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

Race Car

Horizontal Movement

Acceleration and Speed

Scenery and Obstacles

Fuel Depot

Fuel

Distance Travelled

Collision

Game Over Dialogue

Part B - Highscore Screen

References