

User instructions

- What is our project about:

The player controls a long, thin creature, resembling a snake, which roams around on a bordered plane, picking up food (or some other item), trying to avoid hitting its tail or the edges of the playing area. Each time the snake eats a piece of food, its tail grows longer, making the game increasingly difficult.

- How to use our project:

The first screen that will appear is the main menu. There are two buttons, the play button, and the exit button. At first, the play button is selected (being red), to choose the exit option, you need to click on the "W" or "S" key, and press enter to advance. When you choose the play button, you will go to the play section and to start playing you must press "Enter". To move the snake, use the "W A S D" keys. To stop the game, press "Enter" and to resume press "Enter" again. If you want to increase the speed of the snake, press the left button of the mouse, and to decrease, press the right one. When you lose, you will go to the end game Menu. There you will see your game points, the game time duration, the exit button, and then the play again button. To choose one, press the "A" and "D" buttons and enter to advance.

Project Status

- What functionality did you really implement?

The snake can not touch itself or the walls, if they do it is game over.

The timer is used to control the flow of the game, updating the game with every interruption.

There, in the menu's sections, we call the function to draw them. And in the game stage, we update the movement of the snake, check if there were any collisions and if the snake hit the apple we make the snake bigger and create another apple.

The keyboard is used to control the menu navigation and in the game is used to change the direction of the snake.

The mouse is used only in the game stage. When we left-click the speed of the snake increases and when we right-click the speed decreases. We wanted to draw a cursor in the menus, however, we had problems with it because it crashed and the fps went really low.

The video card is used to show the application menus and the screen display.

Timer	Menu / Changes in the game	Implemented
KBD	Menu navigation / Changes in the game	Implemented
Mouse	Changes in the game	Implemented
Video Card	Application menus and screen display	Implemented

- Graphics card should mention/indicate use of:
 - video mode (with resolution, color mode and number of colors)
 - Moving objects (collision detection, animated sprites)
 - VBE functions
 - Main Game Loop
 - Drawing our objects, buttons, background(draw_snake(), draw_apple(), draw_menu(), ...)
- Keyboard used for:
 - Application control (kbd_logic(kbd_event *p))
 - Moving the selected button
- Mouse use of:
 - In the game section, we use the mouse left and right buttons to increase and decrease the speed of the snake.(mouse_logic(struct mouse_ev *ev)

Code Organization/Structure

- **apple.c**

Used to create, draw and destroy the apple (snake food)

All done by Pedro, Guilherme, and Bianca

- **cursor.c**

Used to create, draw, delete and move cursor that later will be used in the game

All done by Pedro and Guilherme

- **game.c**

Used to control most of the aspects of the game such as creating the game, deleting the game, resetting the game to the original form, creating the pause function on the game, updating the game while the game is played, drawing the static of the game meaning all the objects that are static and do not move such as the screen itself, draw the point, the time.

All done by Pedro and Guilherme

- **kbd.c**

Basic kbd functions created in the labs but updated so that can fit better in our game

All done by Pedro, Guilherme, and Bianca

- **logic.c**

Used to create the logic in the program, mainly used to set variables to new values when the game is changed.

All done by Pedro and Guilherme

- **loop.c**

Used to decide when the game start, when it should be in a loop, meaning it has started but it has not ended yet, and used to end the game

All done by Pedro, Guilherme, and Bianca

- **menu.c**

Used to define the menu of the game. Such as create/draw menu, buttons, and end game menu among other functionalities.

All done by Pedro and Guilherme

- **mouse.c**

Basic mouse functions created in the labs but updated so that can fit better in our game, also function to read/check status and buffer, and function to recognize mouse movements.

All done by Pedro, Guilherme and Bianca

- **proj.c**

Main file used to call the function to initialize the game.

All done by Pedro and Guilherme

- **snake.c**

Used to create the snake, delete the snake, change directions, move snake, grow snake, check for collisions and control the snake speed.

All done by Pedro and Guilherme

- **timer.c**

Basic timer functions created in the labs but updated so that can fit better in our game also functions to get and display timer configuration

All done by Pedro, Guilherme and Bianca

- **utils.c**

Functions used to complement the rest of the code.

All done by Pedro, Guilherme and Bianca

- **vbe.c**

Functions created in the labs used to draw our objects, buttons, and background.

All done by Pedro, Guilherme, and Bianca

The function that calls driver receive is our `game_loop()`. It handles the interruptions of each device and calls the logic functions of each device, `timer_logic`, `kbd_logic`, `mouse_logic` that handles the event that is occurring, in the kbd and mouse (the one with specific events).

Implementation Details

We created several state machines for the game state, if it is in the menu, in the game itself, or at the end of the game when you lose, for keyboard events.

Objection creation: we created the apple, snake, and game class in the form of a struct that contains its parameters.

Conclusions

- Problems: While drawing the cursor the graphic would not work correctly and fps would go down a lot. Also, while the snake velocity is getting bigger, the snake gets partially cut
- Main Achievements: The game is all implements all we had thought in the beginning and we were also able to implement the mouse during the game.