PROJECT SPECIFICATION – "TYPERUNNER"

ELEMENTS:

André Daniel Alves Gomes – up201806224
Diana Cristina Amaral de Freitas – up201806230

DESCRIPTION:

Create a typing tutor/racer game with an animated car (video card) while in the game to track progress, a timer to track performance and the inputs from the keyboard to be able to play.

After the game is created. We will develop a basic menu graphic interface with 2 options : "New Game" and "Exit" that will utilize the mouse's buttons to select one of the two.

To implement mouse's movement we will improve the Menu. Creating a mouse cursor to track mouse movement and to be able to select one of the two options by clicking on top of them.

To implement the RTC we will further improve the Menu by adding a 3º option (while making the previous menu we will leave space for this) "High Scores" where it'll be shown the best scores (ranked by words per minute). With this we'll add a pre-MainMenu windows where the user will insert a name and its date of birth for score identification.

For the Serial Port we don't have definite plans, but it would complement the game if it was multiplayer to "race" versus another user in another pc.

The Game sentences will be stored in a .txt file.

DEVICES:

(1) TIMER/COUNTER, KEYBOARD, VIDEO CARD IN GRAPHICS MODE (70%) (ALPHA STAGE)

|Timer Role: Measure time interval from start to end of "run"

|Timer Functionality: Interrupt

_____

|Keyboard Role: Serve as interface for main part of game, the most direct input from the user: reads each key to check if sentence is being correctly typed.

|Keyboard Functionality : Interrupt

_____

|Video Card Role: Show a progress animation for the current status of the game

|Video Card Functionality: Pixel map movement horizontally

GAME SCREEN:

A sentence is presented on screen to be typed with an animated car on top that moves along the sentence that is typed until the sentence ends OR ESC is pressed.

POST-GAME SCREEN:

When the sentence is completed, it is presented on screen some statistics about the "run": Words per Minute; Success Rate; Time Elapsed

GAME:

After a reverse countdown "3 2 1" the game starts:

A sentence is presented on graphic mode with a stopped car on top of it and a timer counting in real time the elapsed time.

The game's objective is to write the sentence, letter by letter without failing (errors are penalized on the final score) in the shortest time possible.

FUNCTIONS TO IMPLEMENT:

- graphstart() – Starts the graph mode with a countdown

- gamestart() – Starts the Game. Presents on screen the sentence, the car and a place where it'll appear what's been written.

    - game() – while cycle to check the keyboard inputs. If the letter is right, the car moves a bit, the corresponding letter in the sentence turns to Green (or the background) and awaits for the next input, until the end of the sentence. In case the player misses , the car doesn't move, the corresponding letter turns to Red (or the background) and it becomes necessary to press BACKSPACE to return to a state where the player can continue the game. The maximum number of wrong letters is the size of the current word that is being typed, if the entire word becomes wrongly typed, the game presents a pop-up saying it is necessary to press BACKSPACE. It is possible to press ESC at anytime to exit the game.

- results() – Shows elapsed time, percentage of completion, words per minute, success rate and more.

To read make/break codes we will use 2 functions:

- A function that receives the letter and returns the make/break code

- A function that receives the sentence and returns a vector of make, break codes

(OBS: it is possible to enter 2 make codes and 2 break codes, in this succession to type, for example "ab". We will have in mind the time between the make and break codes, so that the previous is possible and it does not repeat the first letter all the time, for example "aaaab").

(2) MOUSE BUTTONS (85% CEIL.) (BETA STAGE)

|Mouse Buttons Role: Select one of 2 options (RMB and LMB) for new game or Exit

|Mouse Functionality: Interrupts and interpret packets to mouse presses

FUNCTIONS TO IMPLEMENT:

-MenuMouseButtons() – Shows a simple Menu in which the player can press LMB to enter the game or RMB to exit the program

(3) MOUSE MOVEMENT (85%) (GAMMA STAGE)

|Mouse Movement Role: Create a cursor that moves with the mouse to select menu boxes

|Mouse Movement Functionality: Interrupts and interpret packets to mouse presses and movement

FUNCTIONS TO IMPLEMENT:

-PointerGraphic() – Starts the program on graphic mode in which it is presented a Menu and a Cursor to select options.

-results() – Changes will be made to this function to implement the cursor.

(4) RTC (90%) (DELTA STAGE)

|RTC Role: Record the date and time of the end of run to keep a LeaderBoard

|RTC Functionality: Not yet studied

FUNCTIONS TO IMPLEMENT:

- Menu() – New Menu with the option “Show High Scores” added and a screen that appears before the main menu asking for a name and date of birth.

- showScores() – shows the best scores with the same information as in results() including the sentence and name of the player;

-results() – Add a functionality to store in a file the “runs” made with date and time that it ended

(5) X SERIAL PORT (100%) (EPSILON STAGE)

We do not intend to implement the Serial Port, but if it is possible, the objective would be to create a 2 player game, with each one writing the same sentence on different computers, winner being the player that finishes first/has better words per minute/better success rate.

(6) GAME COMPLETELY FINISHED (OMEGA STAGE)

Extra - Sentences : http://typeracerdata.com/texts