

LCOM 2018/2019- T3G07

Final Project Specification

Description:

For our final project we plan to develop a game called "Typing Invaders". The game falls in the category of "typing games" and is inspired by the game Typeracer (<https://play.typeracer.com/>).

The goal of the game is to test the player's typing proficiency and mouse accuracy. In it, a player controls a cannon that is located at the bottom of the screen. At the top of the screen some spaceships will start to appear each one containing a random word of a dictionary that can be chosen by the user. When a spaceship appears, the player must type the words on the spaceships, destroying them once the full word has been typed. While the player is typing, the spaceships will gradually descend the screen and if they reach the bottom part they will "damage" the player, ending the game if the player is damaged too many times.

During the game, some care-packages may appear at random points of the screen, for a pre-determined period of time, if "hit" (clicked on by the mouse) while they are visible on the screen, these packages will yield bonus points and/or power-ups.

The game is played by waves, each wave increasing the difficulty by increasing the rate at which the words appear, as well as the overall length of the words (long words are harder to type).

While playing, the game keeps track and displays some stats for the player like: the current "wave" number, the player's current "health", the points obtained so far (points are obtained by destroying spaceships and, as said before, through hitting powerups), the words-per-minute (WPM) count (to access the player's typing speed) and the player's typing accuracy.

Although we've not yet seen how UART serial port works we plan to try to implement a multiplayer option for our game. The multiplayer game will be played by two players and it is a standard typing race. Both users will type a text that appears on their screen, winning the player who finished typing in first.

When the game ends, the player will be shown his run's final stats, where he will be prompted to try again, exit to main menu or save the game's stats (in a leaderboard).

The main menu will consist of, mainly, three options:

- Play: play the game;
- Leaderboards: see the leaderboards for the main typing game;
- (Not set in stone) Multiplayer: option to play a standard typing race against another player through serial port;
- Settings: through this option, the user may select some settings like the dictionary that he chooses to use, the display settings (auto night-mode shift enable/disable, some display options and score options)

Device usage:

For the project we plan to use all the devices used on the lab classes as well as the RTC (real time clock) and the UART serial port. For a detailed description of the usage of each device check bellow:

Keyboard:

- The keyboard will be used for several purposes. The main purpose is the main mechanic of the game which is the typing game. Nonetheless, the keyboard will also be used for the user to be able to input other information to the game, such as the player's name (for leaderboards) and the name of the dictionary used in the typing game.

Mouse:

- The mouse will be used, as described before, in the typing game to obtain some "power-ups". Besides that, the mouse will serve as a navigation tool for the player, that is, he will use it to navigate the menus, clicking the mouse-buttons to push in-application-buttons.

Timer:

- The timer will be essential for timing mechanics used in the game. It will help in tracking the time between appearance of the enemy spaceships, display time of the visual aspects, tracking of the time played, calculation of statistics, controlling the amount of time that the care-packages are displayed.

Video Card:

- The video card will be used as the main bridge of output to the user. The usage of the Video Card is essential to display the sprites and other visual objects associated with the typing game.

RTC:

- We plan to use the real time clock to change the visual aspects of the games as a result periods of the day. This will manifest in an enabling of the "Night-mode" which is a mode that is commonly used in most devices that possess screens as a mean to reduce the eye strain caused by colors of low temperature, and in the changing of the sprites and background pictures of the main game.

UART (serial port):

- As described above, the serial port will be used to implement the multiplayer side of the game: the typing race. To execute a typing race two devices must communicate with each other, one who hosts the game and one who joins. The host must send to the guest the information about the text that will be used in the typing race and the guest must send information the host regarding the end of the game and the stats of the guest player. This information should be sent through the serial port.

Modules:

To implement the game, we plan to split the program into several modules, each one focusing on a different aspect of the final state. For a detailed description of each module check below:

Menu (mouse and video card):

- The menu will be important to the ease-of-use and aesthetics of the game. It is the first contact of the user with the game (first impressions are very important), as such it must have a simple, elegant and functional design. Through the menu, the user will access the settings, leaderboards, single-player and multi-player game and statistics. This module will be a good presentation of the capabilities of the mouse (as they will be present in the “pushing” of the buttons that compose the main menu) and how this device must be in harmony with the video card (displayed image).

Graphics (video card):

- The graphics are one of the most important features of the game because, without them, the project will be rendered rather dull as the user won't have nothing interesting to look at. So, the graphics must be aesthetically pleasing to make the user enjoy the game. The graphics also serve as a bridge between the game mechanics and the player, as the results of these mechanics are shown to the player through the screen. In this module we must implement the sprites that compose the buttons (of the menu), spaceships, cannons, missiles, care-packages, the backgrounds for the menu and the games, and the font of the alphabetic characters used on the game and the menus.

File handling:

- File handling will be needed to include persistency in the game, that is, keep track of the leaderboards, statistics, between sessions. It will also be needed to import the words and texts use both in the single-player and multi-player mode.

Single player game mechanics:

- This module is composed of several sub-modules essential to implement the main mode of the game:

- . **Enemy movement:** spawning and moving the enemies through the screen as well as updating the sprites;
- . **Word selection:** selecting the words that will be used next in the spaceship;
- . **Typing tracking(keyboard):** track the writing of word that is on a space-ship;
- . **Timing(timer):** these module will make use of the timer to time important events(wave control, enemy movement, sprite animations,...) that are relevant and will aid in the calculation of statistics.
- . **Data presentation (video card, timer):** track in real time the information that the player “produces” that is relevant for the statistics and output them in the screen;
- . **Sprite handling (video card):** show on the screen the sprites of the spaceships, cannons, projectiles, background;
- . **Animations (timer, video card):** animate the sprites. This module must work in harmony with the sprite handling and the enemy movement;
- . **Wave control (timer):** this sub-module dictates and controls the duration of the waves as well as assuring the increasing difficulty of each wave;
- . **Stat-saving:** the goal of this module is to implement the persistency of the game’s stats. It will save the information about sessions in files so that it can be consulted in later sessions of the game.

Multi player game mechanics:

- This mode will implement the secondary(multiplayer) mode of the game. It also consists of various submodules, many of which can be “recycled” from the single-player mechanics module:

- . **Text selection:** selecting the text that will be used next in the game;
- . **Typing tracking:** track the writing of words that are on the text (sequentially);
- . **Data presentation:** track in real time the information that the player “produces” that is relevant for the statistics and output them in the screen
- . **Stat-saving:** the goal of this module is to implement the persistency of the game’s stats. It will save the information about sessions in files so that it can be consulted in later sessions of the game.

Network communication (UART):

- The goal of this module is to do the communication between to devices (use of the UART serial port) to transmit the necessary information for the multiplayer mode. Basically, it

must establish the connection, transmit the text, signal the start of the game, signal the end and save the stats of both players.

“Actual time” related events (RTC):

- This is the module that makes use of the real time clock. It is used to get information from the RTC and to use it to automatically change settings of the game like the “Night mode” and background.

Program flow module:

- This will be one of the most important modules, as it will be the one connecting the functionality of each of the other modules, in a way that they all complete each other so the game works as desired, by calling the right functions at the right time, depending on the state of the program.

Dispatcher:

- This module will be responsible for handling interrupts and will bridge the program itself and the IO devices.

Plan:

- Taking in account that we have four weeks to prepare for the demo lab class and that in that lab class we must present a version that makes use of the keyboard, the timer and the video card, our plan starts by implementing the graphics and the single player mechanics module since they make use of all required devices and serve the basis of the project.

- That being said, an overall goal for our first four weeks are:

- . Week 1: add the graphics module and basic animations;
- . Week 2: add the main mechanics of the game (keyboard and timer);
- . Week 3: add the mouse mechanics and secondary features of the main game;
- . Week 4: add the menu and other tabs (leaderboards, settings, ...);

- On the remaining weeks we will try to implement the remaining features of the project, that should be the use of the RTC and the UART serial port (multiplayer game). During this time we will also try to perfect the previous modules by improving the graphics, the animations and the optimization of the game.

Module assignments:

Menu: David

Graphics: Luis

File handling: Luis

Single player game mechanics:

- . Enemy movement: David
- . Word selection: Luis
- . Typing tracking: David
- . Timing: Luis
- . Data presentation: David
- . Sprite handling: David
- . Animations: David
- . Wave control: Luis
- . Stat-saving: Luis

Multi player game mechanics:

- . Text selection: Luis
- . Typing tracking: David
- . Data presentation: David
- . Stat-saving: Luis

Network communication: David

“Actual time” related events: Luis

Program flow module: David

Dispatcher: Luis

David Luis Dias da Silva

201705373

Luís Pedro Pereira Lopes Mascarenhas Cunha

201706736