Project description

Multiplayer racing game

Date:

9 May 2017

Made by:

Alexandru Ungureanu (224469)

Dominik Lukac (239837)

Vladimir Rocin (239854)

Supervisors:

Ib Havn

Lars Sørensen

Richard Brooks

Background description

When talking about real-time operating systems, the main notion that arises is the “deadline”. An application is considered to be real-time only in the case that all its tasks can meet the specified deadlines, i.e. that the system produces a response within a predefined amount of time. This topic is of big importance in the realm of embedded programming.

In order to consolidate the knowledge gained during the real-time programming course, this semester project comes to provide a challenge of designing a real-time application based on FreeRTOS. Therefore, it is important to note that this project is of great relevance for the students of embedded engineering.

Using a game console composed of a microcontroller, a joystick, a 14x10 LED display and a USB connection port, the challenge is to create a game that would run simultaneously on both the console and an external PC to which the console is connected. Because this should be a two players game, both the console and the PC application should provide an interface. However, all the computation should take place on the gaming console, so that the PC application implements only the view of the game.

In this context, it is important to take note of the communication between the 2 parts of the system and some kind of protocol is needed. Even though it is required that the communication is performed through the USB, it is still relevant to establish a standard for the data packages, so that it is recognized by both parts.

One more aspect that should be considered, are the error detection and correction methods that should be considered alongside the communication protocol. Some choices could be represented by the implementations of byte stuffing and hamming codes.

Furthermore, it is necessary to implement a game that would fit with the given software and hardware limitations. One of the largest impediments is that the LED display is quite small and that the application is a 2-player game, which makes it relatively hard to design a game suited for these restrictions. One more thing to be considered are the movements of the joystick, which map to 5 different states, which are up, down, left, right and pressed. In this context, it becomes necessary to restrict the controls from the PC part so that they are in concordance with the possibilities of the console.

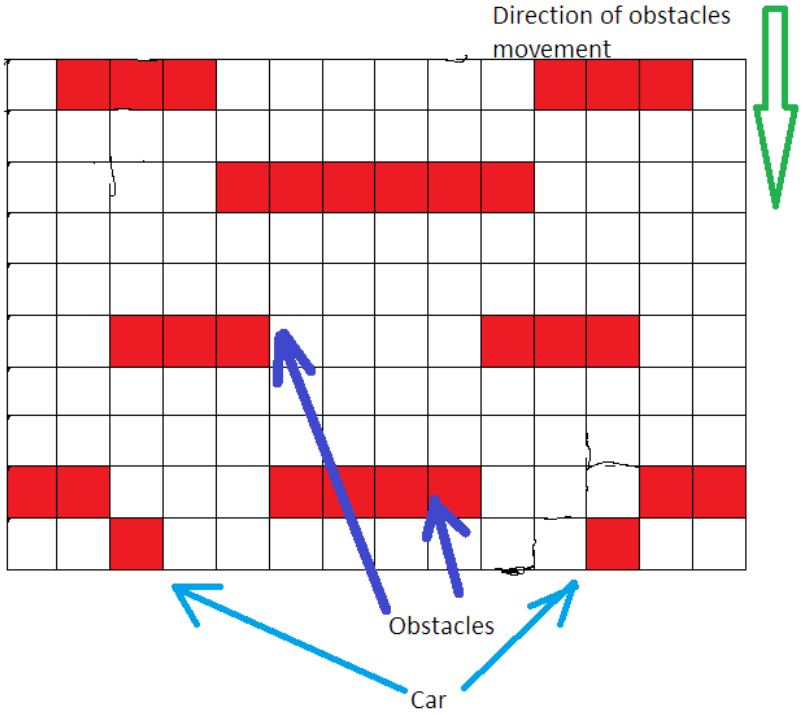
Thus, it is necessary to note that the development of this kind of system represents a very good challenge and opportunity due to being imposed to work with and take in consideration various limitations.

Purpose

The goal of this project is to create a game represented as a real-time application for a simple game console. The game should be for 2 players and the second one would control the game from a computer, which should have the game console connected to it.

Problem formulation

What is the game going to be like?



We decided to create a multiplayer racing game. Cars must avoid obstacles and each other. If a car hits anything in front of it, it is destroyed and the other player wins. Obstacles will be spawned at one end of the display and move towards the other one to simulate progressing movement of the cars. A car would be represented by two pixels/diodes and would be allowed to move freely around, while obstacles would move towards cars and then out of screen once the cars pass them. The goal of the game is to last longer than the opponent.

The map would be either generated randomly with some restrictions so there is always a path or it would be hard coded. This depends on how quickly it is possible to work out the rest of the project.

This supposes that a suited communication protocol be agreed upon. Furthermore, the interface and flow of the application have to be derived from the above stated conditions.

Delimitation

Software limitations:

- The game is for two players only
- The speed of obstacles movement will increase during the game
- Cars can move only sideways unless they collide with an obstacle or another car
- Real-time application

Hardware limitations:

- Display is 10*14 pixels with a single colour
- The joystick on a console has only 5 inputs – up, down, left, right, enter
- Low memory
- Console has to be connected to a laptop through the USB

Methodology

Objective	Question	Model	Method	Source
Developing the idea	What should the game look like?	Research	Research	Internet
Designing a communication protocol	How will the PC and the game console communicate?	Hamming codes, byte stuffing, TCP	Research	Teachers' slides/Internet
Designing the core of the application	How will the game be displayed, controlled and operate on the console?	RTOS	Research, Analysis	Teachers` slides
Designing the display on the PC	How will the game be presented on the PC?	Object-oriented programming (Java, .NET)	Analysis, Use Case modelling	Internet
Sketching the overall flow of the application	How should the game support the two players?	RTOS, Object-oriented programming	Research, Analysis, Use Case modelling	Internet Teachers` slides

Sources

Literature:

- Andy Wellings and Alan Burns – Real-time systems and programming languages
- Paul Scherz – Practical electronics for inventors

Webpages

- https://www.edx.org/
- https://en.wikipedia.org/
- https://www.youtube.com/
- https://stackoverflow.com/

Authorities on the subject

- Ib Havn
- Lars Sørensen
- Poul Væggemose

Project Schedule

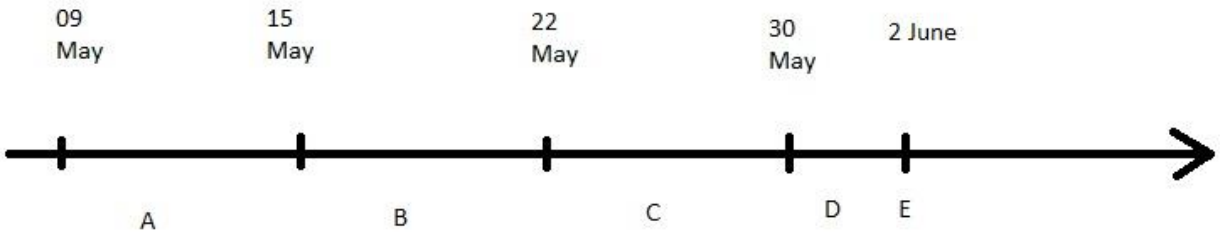
09/05/2017 – 15/05/2017 – **A -** Work on project analysis

16/05/2017 – 22/05/2017 – **B -** Design and gathering of additional knowledge

23/05/2017 – 30/05/2017 – **C -** Implementation and testing of the application

30/05/2017 – 02/06/2017 – **D -** Time reserve and documentation

02/06/2017 – **E -** Submission of the project



The work will include topics as:

- Deciding concrete requirements
- **R**esearch, **D**esign and **I**mplementation of communication protocol
- RDI for interfacing the game on PC
- RDI of the system architecture
- Possibly designing a state machine
- Design a scheduling system which will ensure that processes are started in time to meet their deadlines.
- Research on FreeRTOS
- Design of the racing map
- Testing of the software – probably mostly testing by using the software
- Multiple meetings with supervisors during most of the processes