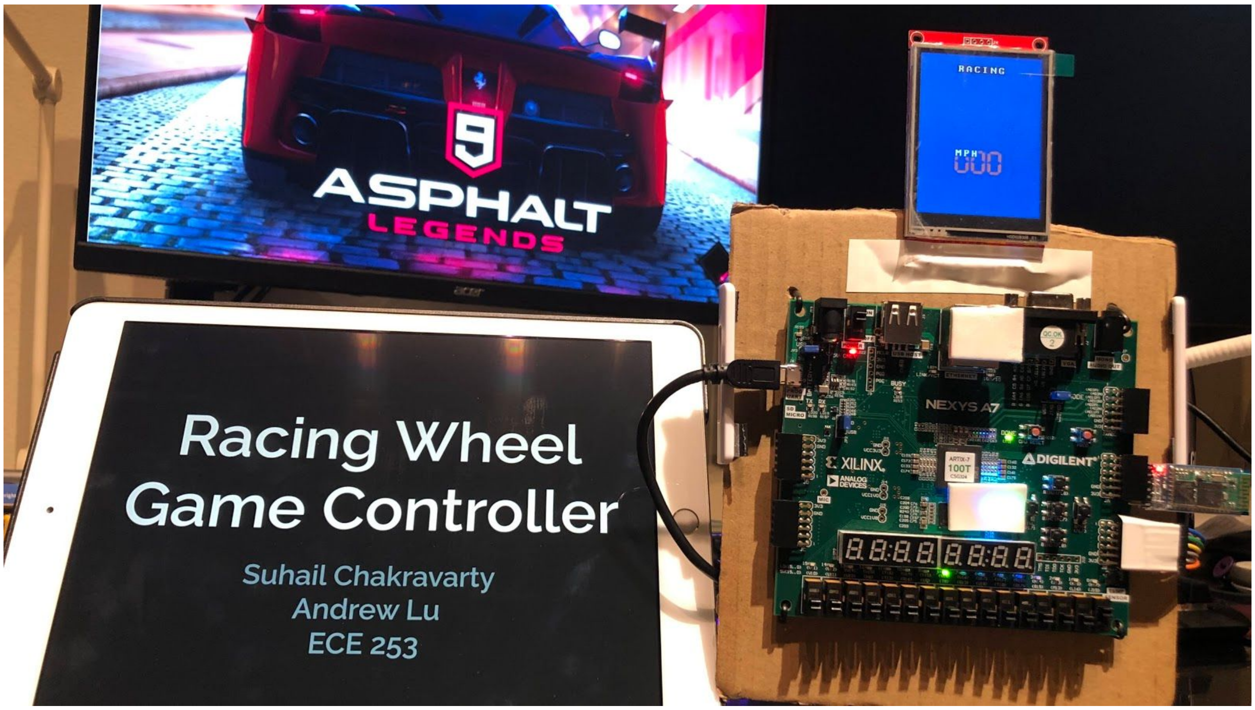
ECE 253 Final Project: Racing Wheel

by Andrew Lu and Suhail Chakravarty



Purpose and Goals

The purpose of this project is to design a Racing Wheel that can be used to play racing games using an FPGA, an accelerometer peripheral, and an LCD display. For this project, it is important that the game is reasonably playable, the board is easily configurable, and the experience is comparable to a real-world racing wheel controller or even a real-world car.

Methodology

Design Requirements

In order to develop this project the main features that are important will be the following:

- Playability - This one is a bit obvious as it is the core functionality of our project and thus is the most important. This involves being able to complete a race without significant delays and bugs in the controls, as well as being able to use the controller in other games.
- Configurability - While many games have a sensitivity setting in game, we thought that a way to configure the controller directly from the racing wheel itself would have a positive impact on the project. The settings we settled on were game modes, sensitivity, speedometer toggle, and communication method.
- Usability (UI)- Tightly paired with the configurability is the User Interface. This would be based on the ability to switch between different modes of controller quickly and easily, as well as changing the configuration without too much confusion.

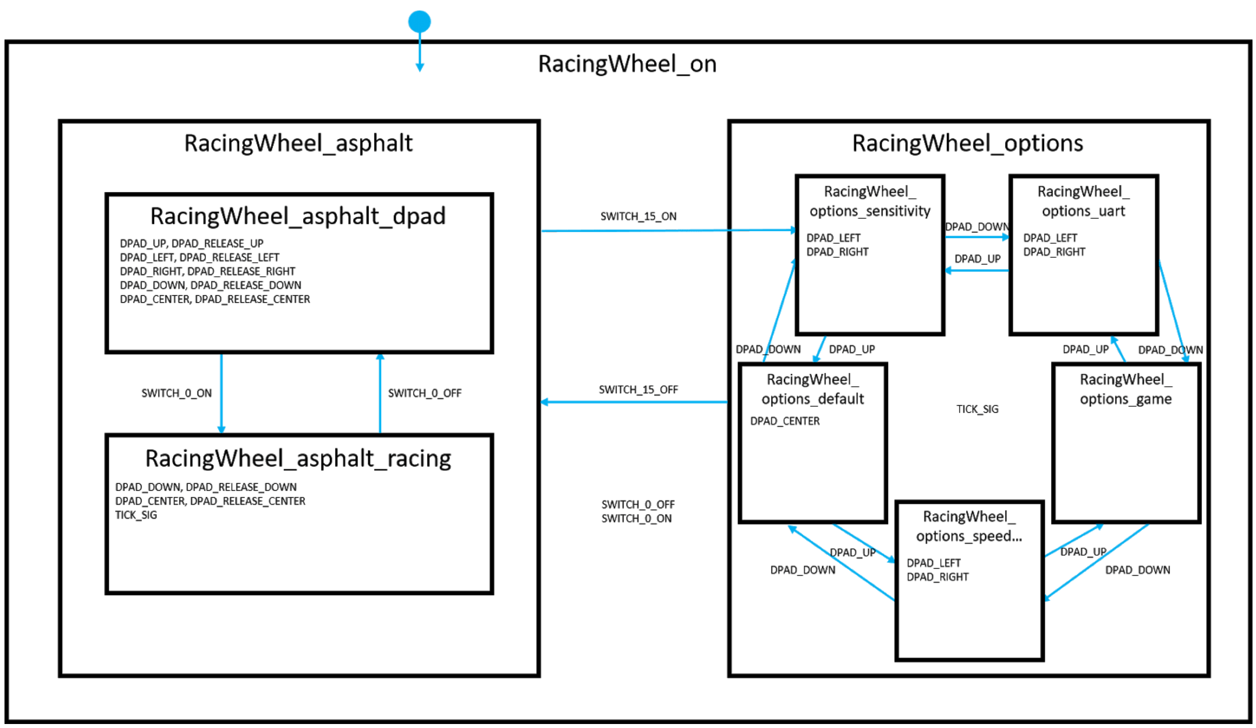
By meeting these requirements, the racing wheel will have a good user experience and provide the fun that it is meant for.

Testing

Unfortunately for this project, it is difficult to develop a set of quantitative metrics for performance since UI and Playability are rather subjective requirements to meet, input delays are a challenge to measure, and our racing skills are very inconsistent. However, for testing purposes, we will both be using the game “Asphalt 9: Legends'' where multiple players are able to compete together remotely and in real time. By doing this, we can test it in a real-world gaming scenario and subjectively evaluate the performance of the project. If the game is unplayable, has a very large input latency, or is impossible to navigate, then we know that the project needs to be adjusted. On the other hand, if we feel that the controller’s latency is not too high and the general experience with the controller is good, allowing the game to be played without having to switch back to a keyboard and mouse, then the project is a success.

Hierarchical Design

This project uses the QP-Nano framework for managing states. Currently, there are two primary states: one for the Asphalt 9 game and one for the options menu. Each state has multiple sub-states. In the Asphalt state, the substates represent the Navigation (D-Pad) mode, where no accelerometer data is sent and the D-Pad buttons are mapped to arrow and select keys in the game, and Racing mode, which sends accelerometer data, as well as drift/boost commands. In the Options state, each sub-state represents one option in the menu.



HSM Diagram of our Game Controller's QP-Nano Implementation

Results and Implementation

User Interface

The three modes of use that we settled on are Navigation, Racing, and Options, in order to cover all features that players would find useful. The Navigation mode was made so that players are able to traverse the menus leading up to the actual race by using the D-Pad buttons on the Nexys board. This allows us to avoid having to switch between the keyboard on the PC and the game board and instead simply stay on the one platform. Next is the Options menu where one can toggle the leftmost switch to access the various settings and enable different configurations of the board including sensitivity, game mode, and speedometer presence, and communication method (for users with a HC-05 Bluetooth module). Finally, we have the Racing mode where the player is able to control the actual game by tilting the board side to side like one would a steering wheel, as well as use the D-Pad buttons to activate Nitro boosts and drifting.

Image of Speedometer of Racing Mode

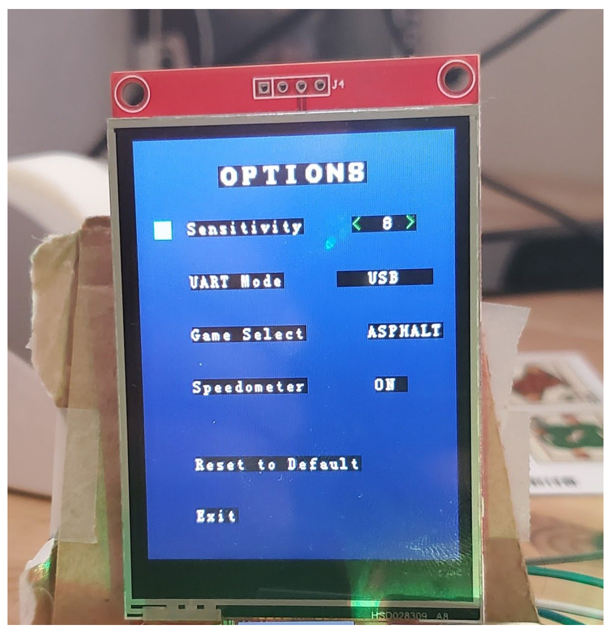


Image of Options Menu

Implementation

We were able to implement the GUI and all the states using the QP-Nano Framework and the state machine shown in the methodology. All of the external signals were generated from peripherals already available on the Nexys board. Specifically, the external inputs that this project uses are an accelerometer using SPI communication, a UART communication to the PC via USB or Bluetooth, a set of D-Pad GPIO buttons, and two GPIO switches, along with various LED indicators and an LCD display being controlled by the outputs of the QP-Nano System.

Roadblocks

Initially for this project, we had aimed to control the PC using the HID hub available on the Nexys board. However, after days of research and effort, it became apparent that the HID port was only compatible as a host and not a client. Additionally, there was an absence of documentation regarding the usage of the USB port in the Xilinx SDK. As a result, the next best solution was to develop a Python script on the PC to receive UART commands from our Nexys board and convert them into keystrokes. This was not ideal since we would have preferred that the board not need external software in order to operate, but we had no other choice to pivot given the lack of support. However, this change did allow us to add support for a speedometer using Tesseract (an OCR library) as well as Bluetooth UART using the HC-05 Bluetooth module.

Final Product Evaluation

Despite the roadblocks, this project has come out to be a very fun and playable experience. In the demo attached, you will see Andrew and Suhail playing Asphalt 9 using this project, in which Suhail used a wired UART communication while Andrew played over Bluetooth communication, both of which proved to be a good experience for the user. While it is difficult to evaluate the board’s success with any quantitative metrics due to many external factors (such as CPU/GPU limitations on the computer, internet latency when playing online, and our inability to race well), a visual demonstration of the product can be subjectively evaluated by the viewer as successful or not.

The demo can be seen here:

https://www.youtube.com/watch?v=vX0i1uezqp4 (6:03)

Issues and Future Work

Despite generally being a success, there are a few portions that could be further developed and improved. First, although beyond the scope of this course, the mechanical construction of the board could be improved, as cardboard / tape enclosures turned out to be clumsy and would fall apart quickly. A 3D-printed enclosure would greatly help in this aspect. In addition, the speedometer readings off the game are sometimes a bit unreliable and sometimes causes the keyboard inputs to have a higher latency than desired. While the latency is not as low as we would have preferred, given that the communication method had to change to UART, it does not seem to affect gameplay too much. Finally, some future work would be to expand the board functionality to support more game modes so the board can be used for more games (and not just racing games).