

Design Brief: Group 2

Aidan Cauchi, Matthew Vella, Neil Vassallo, Peter Galea St. John

February 13, 2019, Document v.2489

Executive Summary

Everyone likes to have fun! And everyone remembers a certain game in particular. Reminiscing about the time we were kids, arcade games gave us the perfect opportunity to have fun whilst competing with our friends. Given the opportunity to work on this project, as well as all the good memories the team made playing such games, they have decided to make an arcade classic that they all loved, the basketball hoop game. The player to score the most wins!

1 Introduction

The team members have chosen this game as they all love games and it would be interesting to create their own game, with the ARM MDK. Furthermore, while the project does seem challenging since the team is new to the concepts mentioned, it also seems manageable enough such that the members don't end up feeling overwhelmed while managing other projects as well.

The ARM MDK would be set up to a wooden architecture, where the components needed would be attached to it. The player, after interacting with the menu system, would try to shoot the ball in the hoop, and gain points if succeeding in doing so. The ball would return back to the player and they will try to get more points until they run out of time. Once this happens, the final score will then be saved onto persistent storage to keep track of "highscores".

This document is divided into 3 other sections. Firstly, there is the System Design, where the descriptions of logic, architecture, tools, and explanation of the actual implementation of the project are found. Secondly, the Management section involves how the team will work together, and the development methodology they will adopt. Finally, the Closure section contains a small recap and members' expectations with the project going forward.

2 System Design

This section contains the implementation details of the project.

2.1 List of components

The team will be making use of the following components:

- Joystick
- Proximity Sensor
- LCD
- 7-segment display
- SD Card
- Speaker
- LEDs

2.2 Functional Requirements

The following table shows how the team intends to meet the project requirements:

Requirement	How it's addressed
(F.1) – Real time System	Interrupt based system reacting to the ball going through hoop.
(F.2) – Digital Input	Joystick, Proximity Sensor.
(F.3) – Digital Output	LCD, 7-Segment Display.
(F.4) – External Peripheral	LCD, Speaker, 7-Segment Display.
(F.5) – Well defined states	To be tackled with software.
(F.6) – Persistent Storage	Using the SD card interface to store high scores.
(F.7) – Error reporting	LCD for error reporting.
(F.8) – Robust System	Tacked with software.

2.3 Joystick

This will be used to control the menu outputted on the LCD screen. The user will be able to switch through

the menu options and press the joystick when s/he wants to make a choice.

2.4 Proximity Sensor

This sensor will be used to count the amount of times the user manages to score a point. It will do this by telling the board to increment the user's points each time the distance being sensed is shorted by the ball going in.

2.5 Seven-Segment Display

The seven-segment will be used to display the count-down timer.

2.6 LED Lights

These will switch on and flash when a game is started or when a goal is scored.

2.7 Speaker

The Speaker will play a tune when the game is started or when a goal is scored.

2.8 LCD Screen

The LCD screen will be used to display the menu. In it the user can chose to start a new game, view high scores and change game settings.

2.9 SD Card

The SD card is essential as it will be used to store the highscores of the game.

2.10 Gameplay loop

Figure 1 shows the flowchart for the main gameplay loop, where the majority of the logic of the game resides.

2.11 Physical Design

Figures 2 and 3 contain the diagrams related to the physical model. The model will be made of plywood (thick but lightweight) and will be 70cm by 55cm and the back court will be 55cm by 50cm. Finally, the board will be fitted underneath the base.

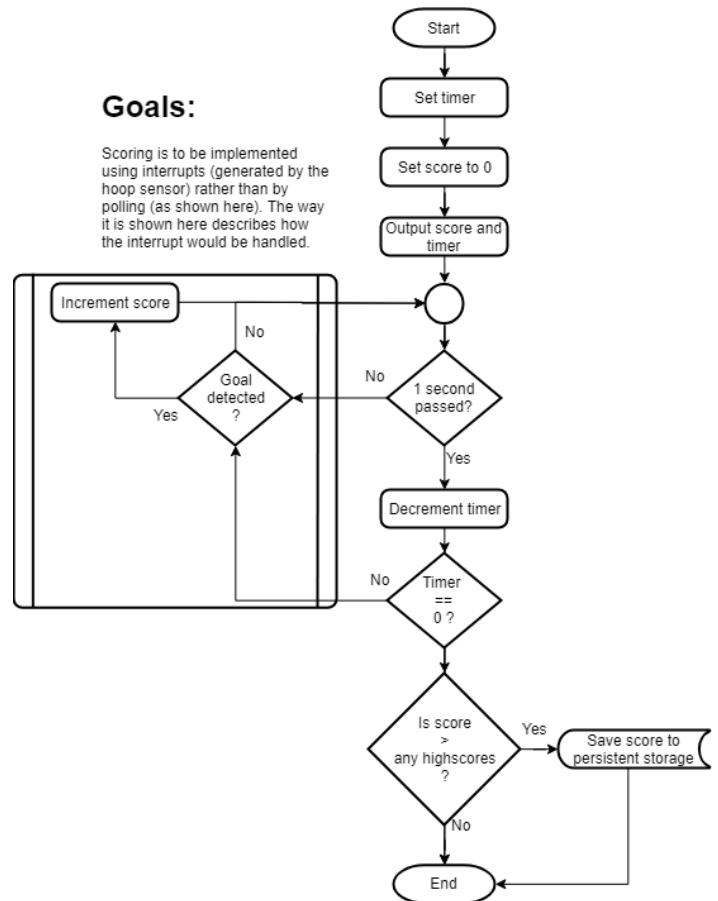


Figure 1: Flowchart of the main gameplay loop.



Figure 2: Approximation of how the model will look [1]

3 Management

Being four people in the group, it was seen ideal to split the group up; Matthew and Peter will work mostly on hardware while Aidan and Neil will work mostly on software. However, this will not be strictly enforced so that all members will get a chance to work on all aspects of the project.

After the initial meeting, the team agreed to proceed with the Agile Software Development approach. This approach will allow them to work in a flexible manner. As the technology is new to all members, it

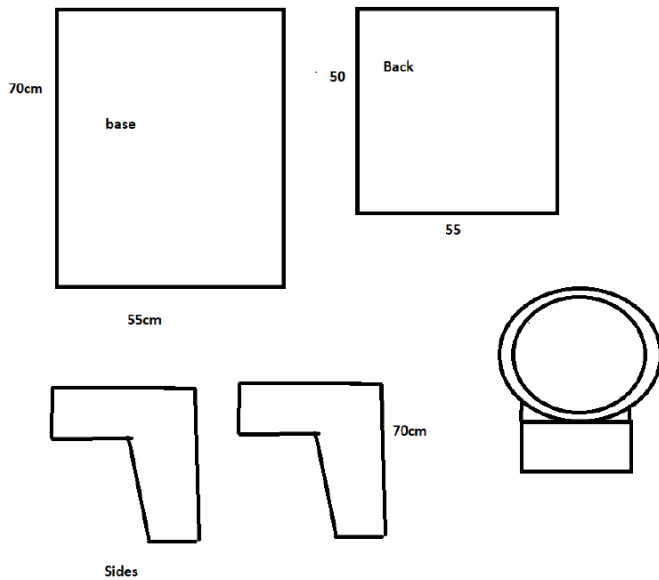


Figure 3: Top-Down Designs

is essential that there is some space to allow requirements to change as necessary without severely halting the progress of the project [2]. However, some key requirements were still noted:

- Interfacing with LCD/7-segment display.
- Wiring/integrating components.
- Testing.
- Lighting LEDs.
- Joystick programming.
- Playing sound through speaker.
- Main menu programming.
- In-game programming.
- High score persistence.

The prioritisation of these requirements will be discussed during the meetings as the team starts to get a better understanding of the technology.

As a result of choosing this methodology, some ground rules needed to be set: The team is to meet every Thursday at 12/12:30. An agenda detailing the points of discussion for the next meeting will be uploaded by a member on SVN before the next meeting. On the agenda, a member is chosen to be the minute-taker for the next meeting. After a meeting has concluded, the minute-taker will compile the minutes using the template provided on SVN and then upload said minutes. The minutes are to contain what was said in the meeting as well as the work assigned to each member for the next meeting. Once uploaded, the minutes and agendas shouldn't be

changed, however some exceptions will be permitted if some information was overlooked, especially in the beginning few weeks as the team gets used to the system. The members taking the minutes and agendas will be rotated every week so that everyone will get a chance to contribute from a managerial perspective. Finally, all the minutes and agendas will be written using Latex and the provided templates.

With respect to the project itself, detailed goals will be listed during each meeting. Since most members of the group have no experience using this technology, it was found to be better to plan specific requirements as the project develops and as the team learns more during the lectures. The Keil uVision IDE will be used to code the implementation. Alongside this, the team will make use of Doxygen to ensure proper and consistent documentation of code. Moreover, SVN will be used to track changes to the project and share files between the team members. Facebook's Messenger will be used for informal communication between the members and for providing online help. Finally, the University email will be used to formally remind each member before each meeting.

4 Closure

To recap, all the work was divided between all members of the group. Minute-taking and the writing of agendas are to be done by weekly-rotations between group members and detailed goals are listed during each meeting.

By evaluating the group performance weekly, the team feels that they're on the right track and moving forward as a group. The project seems very interesting and all members are keen to learn and challenge themselves.

By the end of this project, they hope to have a much greater understanding of coding low level microcontrollers, make better use of SVN and enhance their groupwork skills, especially through the Agile Development Cycle.

References

- [1] "GearBox Creative." <https://www.youtube.com/user/firstsight04/featured>. Accessed: 01-03-2019.
- [2] "Atlassian agile coach." <https://www.atlassian.com/agile>. Accessed: 04-03-2019.