uOttawa

Faculté de génie
Faculty of Engineering

Deliverable B

Problem Definition, Concept Development, and Project Plan

Prepared by Group Z11

Full name	Student Number
Hawa Salamata Ndiaye	300092449
Ryan Rambali	300235460
Juan Torres	300198775
Layane El Merini	300288563
Geraud Ndagang	300105965

Presented to

Dr. Emmanuel Bouendeu, Ph.D, P.Eng

as part of the course

GNG 2101Z - Introduction to product development & management for engineers & computer scientists

Faculty of Engineering
Department of Mechanical Engineering

16th May 2023

Table of contents

1. Introduction

As part of our project, a first meeting was organized with our client, Chris Aube, a marketing business owner who is part of the Tetra Society, a non-profit organization that provides volunteer engineers across Canada and the US to design and construct custom assistive devices for people with disabilities. Our client Chris told us that he suffers from tremors, which reduces his ability to play video games because he cannot accurately control the joysticks. Our client is also a non-verbal communicator. During the interview, it was possible to identify his expectations and concerns by asking questions and using the Iterative Engineering Design Process Model (IEDP). As a result of our conversation with him and by using empathy, several problems, limitations, desires and needs were identified to serve as a starting point for developing the problem statement.

2. Identification of client's needs and metrics

From our first client's interview, we were able to draw up a list of statements that were later transformed into needs that we will use to draw up our list of metrics.
Table 2.1 contains a list of all the needs to prioritize in the product the team will create.

Table 2.1

No.	Client needs statements	Priority
1	A remote adapted to any kind of video game	5
2	The remote controller is compatible with the Xbox and PC	5
3	The remote controller is inspired by the Axis controller model	3
4	The remote integrates an interface to configure each button	5
5	The remote software has a macro that allows action to be made by clicking one button	5
6	The remote is easy to use	4
7	Buttons can be spaced in any way	2
8	The remote is simple/doesn't have lot of buttons	3
9	The remote can be made in a short amount of time	4
10	The remote is inexpensive	4

11	The remote is custom made	3
12	The remote is robust/The remote is made of high quality products but as low-cost as possible	5
13	The remote includes a sticky trigger	5
14	The remote's sticky trigger has an on and off toggle	5
15	Remote is comfortable	5
16	The software's user interface is accessible	5
17	The remote's design is ergonomic	5
18	The remote has a reasonable weight	4
19	The controller is portable	1
20	Sensitivity of buttons and joysticks can be configured	3
21	The controller have is adapted to the tremors	2

Ranking system:

5 - critical
4 - highly desirable
3 - good but not necessary
2 - unimportant
1- undesirable

3. Problem Statement

To define the problem to be addressed by the design process and taking into account the client's needs, a problem statement has been drafted. It will be used as a reference throughout the project.
Our problem statement answers these three questions: what the problem is, who has the problem, and the scope of our solution. The answers are outlined below :
What: a programmable remote controller.
Who: disabled gamers.
Scope of product: robust, fully programmable, compatible with Xbox and PC.

" Design a robust, programmable and accessible remote controller that is compatible with both Xbox and PC for disabled gamers. "

4. Metrics and Benchmarking

4.1 List of metrics with appropriate units

Table 4.1

Metric number	Need number referent	Metrics	Units
1	1,2,19	Portability of the remote	Mport(p)
2	3,18	Controller weight	kg
3	15,7	Controller shape/dimensions	cm/in
4	20	Pressure-sensitivity of the buttons	gF
5	6	Usability of the remote	N/A
6	1,2,19	Compatibility of the controller with games devices	N/A
7	11,17	The adaptability of the remote	N/A
8	12,10	Manufacturing cost	dollars
9	7,8	Buttons shape/dimension	cm^2
10	13	Dimension of the sticky trigger	cm^2
11	4,16	Color used on the user interface	list
12	4,16	Image on the user interface	pixels
13	4,16	Font size of the user interface	point(pt)
14	9	Manufacturing time	days
15	5	Modularity of the button implementation	Number of code line
16	14	Modularity of the sticky trigger implementation	Number of code line
17	15	Texture of the material	list
18	12	Longevity	years

4.2 Benchmarking

Table 4.2

Metric number	Need number referent	Metrics	Units	Axis 2 Pro	Onpoint Precision Joystick
1	1,2,19	Portability of the remote	Mport(p)	Portable	Portable
2	3,18	Controller weight	kg	2.49	-
3	15,7	Controller shape/dimensions	cm/in	Rectangular box; 12 × 8 × 6 in	Rectangular box; Dimensions unknown
4	20	Pressure-sensitivity of the buttons	gF	55	-
5	6	Usability of the remote	N/A	Good	Good
6	1,2,19	Compatibility of the controller with games devices	list	PC, Xbox (One, 360), PS (3,4)	All devices
7	11,17	The adaptability of the remote	N/A	Compatibility with other platforms using Expansion Port	Compatibility with other platforms using Cronus
8	12,10	Manufacturing cost	dollar	-	-
9	7,8	Buttons shape/dimension	cm^2	7.1 (1.5cm radius)	11.4 (1.9 cm in radius)
10	13	Dimension of the sticky trigger	cm^2	N/A	N/A
11	4,16	Color use on the user	list	N/A	N/A

		interface			
12	4,16	Image on the user interface	pixels	N/A	N/A
13	4,16	Font size of the user interface	pt	N/A	N/A
14	9	Manufacturing time	days	-	-
15	5	Modularity of the button implementation	Number of code line	-	-
16	14	Modularity of the sticky trigger implementation	Number of code line	N/A	N/A
17	15	Texture of the material	list	smooth	Smooth with “sticky” joystick
18	12	Longevity	years	-	-

5. Target Specifications

Table 5.1

Metric number	Need number referent	Metrics	Units	Marginal Values	Ideal Values
1	1,2,19	Portability of the remote	Mport(p)	Semi-portable	Portable
2	3,18	Controller weight	kg	5>	3>
3	15,7	Controller shape/dimensions	in	Rectangular box; Length:11-14 Width: 7-9.5 Height:3.5-6.5	Rectangular box; 12 × 8 × 6
4	20	Pressure-sensitivity of the buttons	g	70 >	55

5	6	Usability of the remote	N/A	-	Good
6	1,2,19	Compatibility of the controller with games devices	N/A	Only works on PC	Works on PC and Xbox
7	11,17	The adaptability of the remote	N/A	Barely adaptable	Adaptable
8	12,10	Manufacturing cost	dollars	100	100 >
9	7,8	Buttons shape/dimension	cm^2	7 <	12
10	13	Dimension of the sticky trigger	cm^2	7 <	12
11	4,16	Color of the user interface	list	-	Light and dark mode
12	4,16	Image on the user interface	pixels	400 < for a large image 100 < for a thumbnail	500 for a large image 150 for thumbnail
13	4,16	Font size of the user interface	pt	16	Customizable
14	9	Manufacturing time	days	14	7
15	5	Modularity of the button implementation	Number of code line	One module	Divided into independent and reusable module
16	14	Modularity of the sticky trigger implementation	Number of code lines	One module	Divided into independent and reusable module
17	15	Texture of the material	list	-	Smooth, sturdy
18	12	Longevity	years	1-2	5

6. Final prototype concepts

As part of our project we were able to come up with 4 solutions for our design which are presented below :

Table 6.1

	Idea 1	Idea 2	Idea 3	Idea 4
Entire system	Take apart a USB Xbox controller, and attach wires to the potentiometer and button contacts, to control the output of the controller using an Arduino.	Use an Arduino micro, to send out control signals using the input protocol. When running on Xbox, use an x-input to GIP converter.	Reverse engineer the GIP protocol, and use an Xbox controller to bypass the cryptographic handshake, to send sniffed USB GIP signals.	Use an external open source platform (Alt Controller) that maps computer inputs to configure keys as we want and create macros, and connect it to a remote that we will build ourselves.
Subsystem 1 (software)	C# WPF GUI app, to change settings of Arduino. Including macros and sticky tricky triggers. Arduino code, to send electrical signals to the controller based on inputs.	C# WPF GUI app, to change settings of Arduino. Including macros and sticky tricky triggers. Arduino code, to send data signals to the device based on inputs.	C# WPF GUI app, to change settings of Arduino. Including macros and sticky tricky triggers. Arduino code, to send electrical signals based on inputs. Arduino code to bypass the cryptographic handshake, to send sniffed USB GIP signals.	Alt Controller to create macros and change controller settings Arduino code, to send electrical signals based on inputs.
Subsystem 2	USB Xbox	The Arduino is	USB Xbox	3D printed box

(Hardware)	controller connected to the device. An Arduino connected to the inside of the controller, and our control layout of joysticks and buttons	connected to our control layout of joysticks and buttons. It is also connected to the device	controller connected to an Arduino. The Arduino is connected to our control layout of joysticks and buttons. The Arduino is also connected to the device	with an arduino circuit. The joysticks are ergonomically designed.

6.1 Evaluation of the Target specifications

Table 6.2

Selection Criteria	**Idea 1 (reference)**	**Idea 2**	**Idea 3**	**Idea 4**
Portability of the remote	0	+	-	+
Modularity of the button implementation	0	0	-	+
Modularity of the sticky trigger implementation	0	+	-	-
Manufacturing cost	0	-	-	-
Controller shape/dimensions	0	+	-	+
Accessibility of the user interface	0	0	0	+
Compatibility with other devices	0	+	0	-
# of +	0	4	0	4

Selection Criteria	Idea 1 (reference)	Idea 2	Idea 3	Idea 4
# of 0	7	2	2	0
# of -	0	1	-5	3
Total	0	3	-5	1

6.2 Justification of the process and methods used

We used an unweighted decision matrix, as it seemed that most of our chosen criteria were relatively equal in importance. Using this decision matrix allowed us to get an overall view of the pros and cons of the alternative solutions in general, and allowed us to make a more informed decision. It also lets us compare each solution with a reference, in order to properly calculate the positives and negatives of the individual solutions.
Based on our decision matrix and our reference idea, we decided to choose idea 2 as a promising solution. Which is because from our selection criteria, it's the one who, in general, satisfied mostly our critical client needs statements. Although it is more costly, it is the most flexible option that meets the most requirements. Additionally, if the cost goes over budget, we can decide to easily drop a part and the price considerably goes down with relatively little functionality being lost.

6.3 Global design concept

Based on the promising solution chosen, our global design concept consists of two main subsystems. The software component, which consists of an application on the computer that allows the user to remap and customize the buttons on the controller however they want, along with capabilities to create macros - buttons that run button combinations as assigned. This software component is constituted of the code used to run the arduino, and the code used to make the application, which will be made with C# WPF. The hardware subsystem consists of all the joysticks and buttons connected to the arduino board, the controller case/box, and the GIP converter to be able to connect to an xbox. The buttons and joysticks will send signals to the arduino which will then compute the action with the software and output the corresponding x-input signals. If connected to an Xbox, the user will need to use the GIP converter so that the console can receive the signals properly. We will have 12 configurable buttons and 3 preset buttons, along with 3 joysticks.

6.4 Visual concept of our design

Here, we will present our expected prototype.

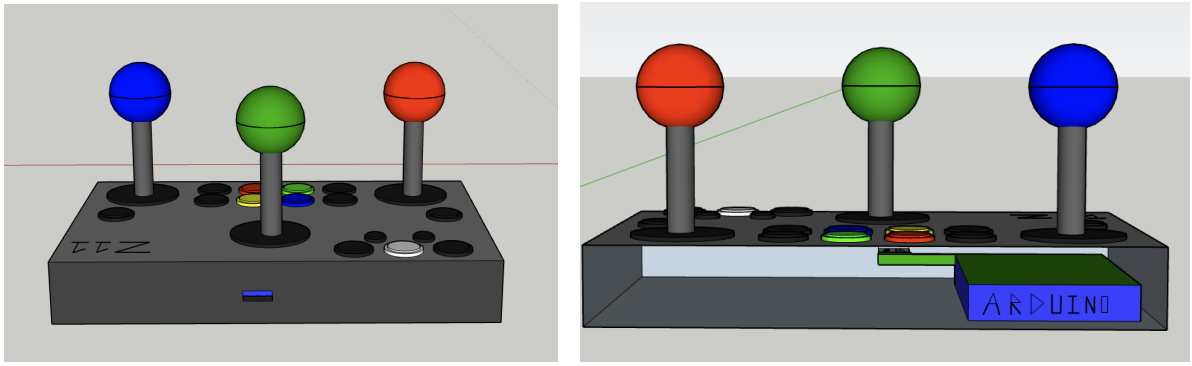


Figure 6.1 - Hardware concept

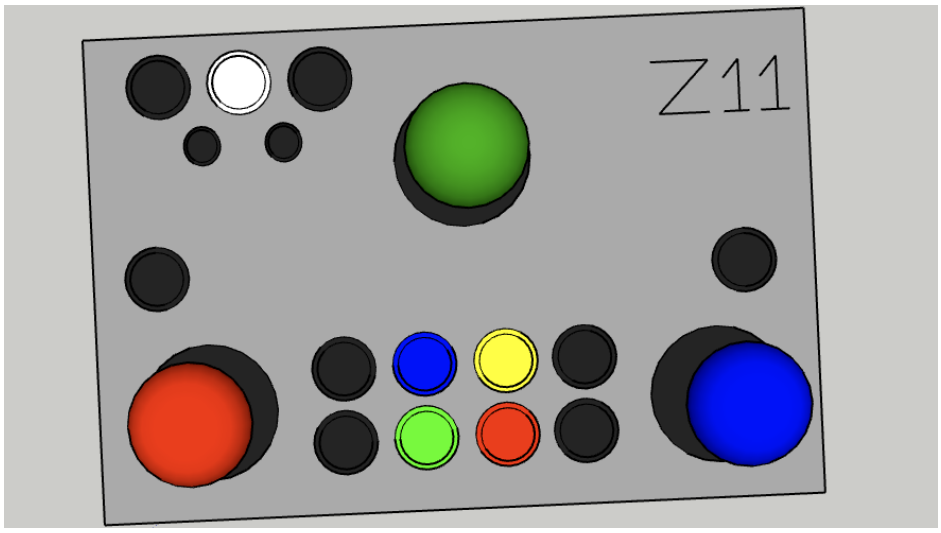


Figure 6.2 - Hardware concept

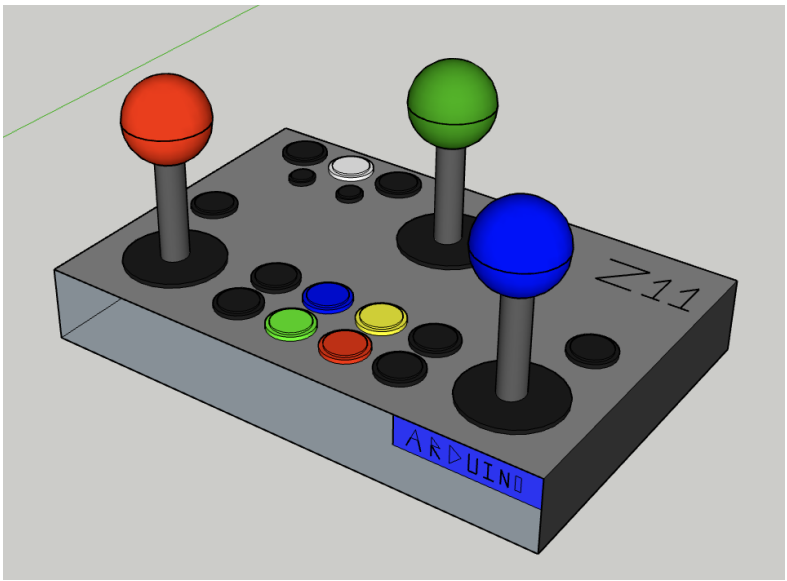


Figure 6.3 - Hardware concept

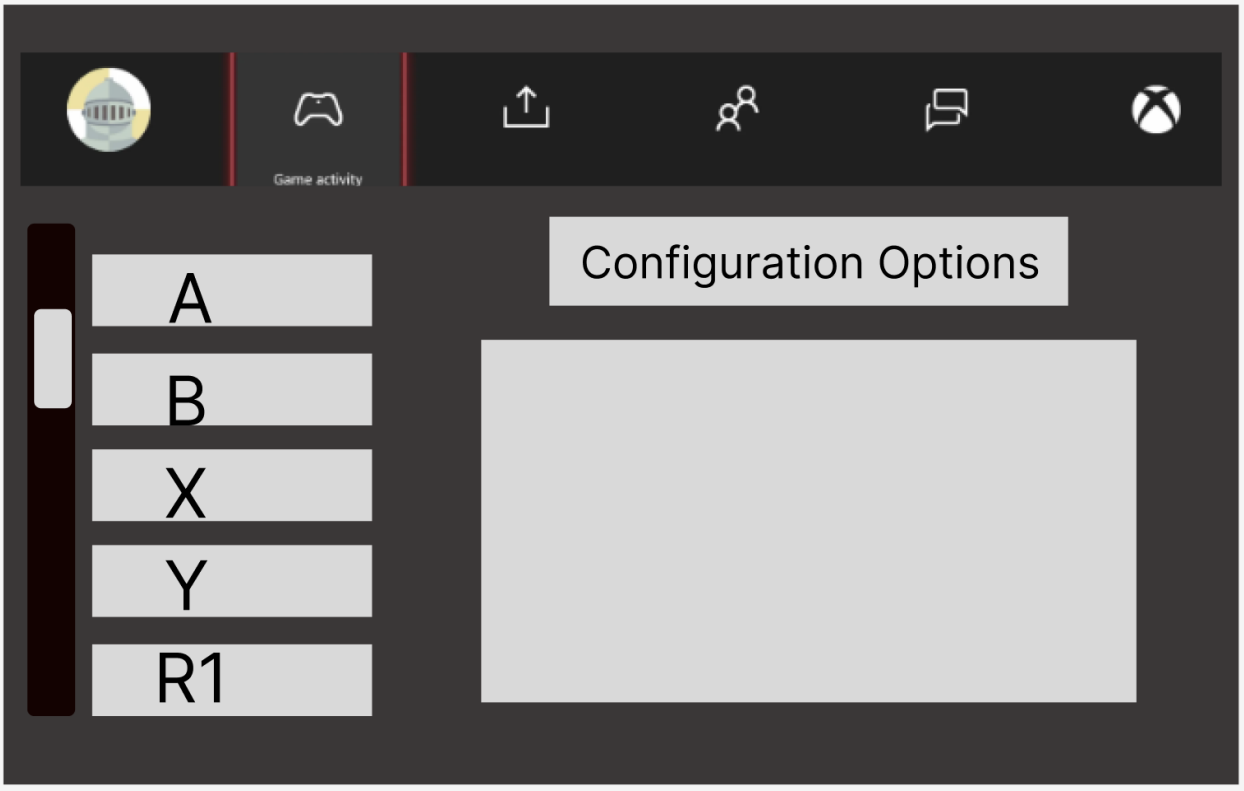


Figure 6.4 - Software concept

The concept has a lot of going for it when we take into account the target specifications. First, let's list off the pros in comparison to the target specifications. The remote is relatively portable, and easy to use. It is compatible with both Xbox and PC. The buttons seem easy to press and are spaced out to the client's desires. The buttons are programmable and have a sticky trigger implementation, meaning the adaptability is high. It also seems like it will last a long time, and the user interface is easy to understand. The cons are few, but should be taken into consideration. It will take a decent amount of time to properly manufacture, and it is fairly expensive. Overall, this seems like it meets all the target specifications for functionality.

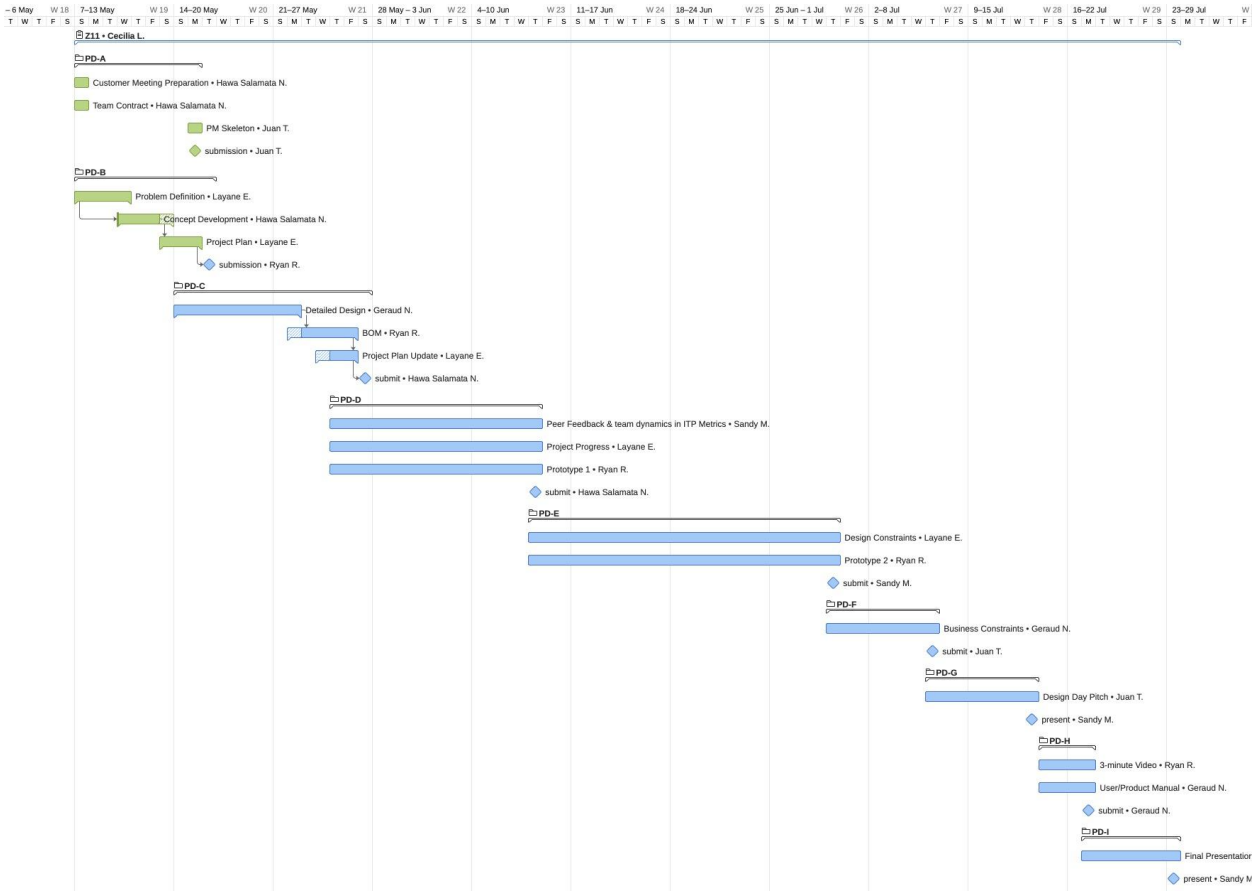
7. Conclusion

After our first client interview, we were able to compile a list of client needs and rank them in order of importance. This list allowed us to develop the problem statement. By showing empathy during the meeting with the client, we were able to accommodate his preferences and articulate specific needs, which helped us determine the metrics. With these information, we also set target specifications which are crucial for us in generating original concepts.
In order to carry out with our project, several concepts related to the subsystems were generated from the problem statement and from the identified design criteria. After a thorough analysis, a final concept was chosen, which will be used in the next steps of our project.

B.1 References

[1] Enhanced Wired controller Xbox (*visited on 10/05/2023*)
[2] How to emulate an Xbox controller Arduino (*visited on 11/05/2023*)
[3] GIP Controller (*visited on 11/05/2023*)

B.2 Project Plan Update



Snapshot :
https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=tCXzWNgW2ApTg6HcC8UczwJIot1PVtbe%7CIE2DSNZVHA2DELSTGIYA