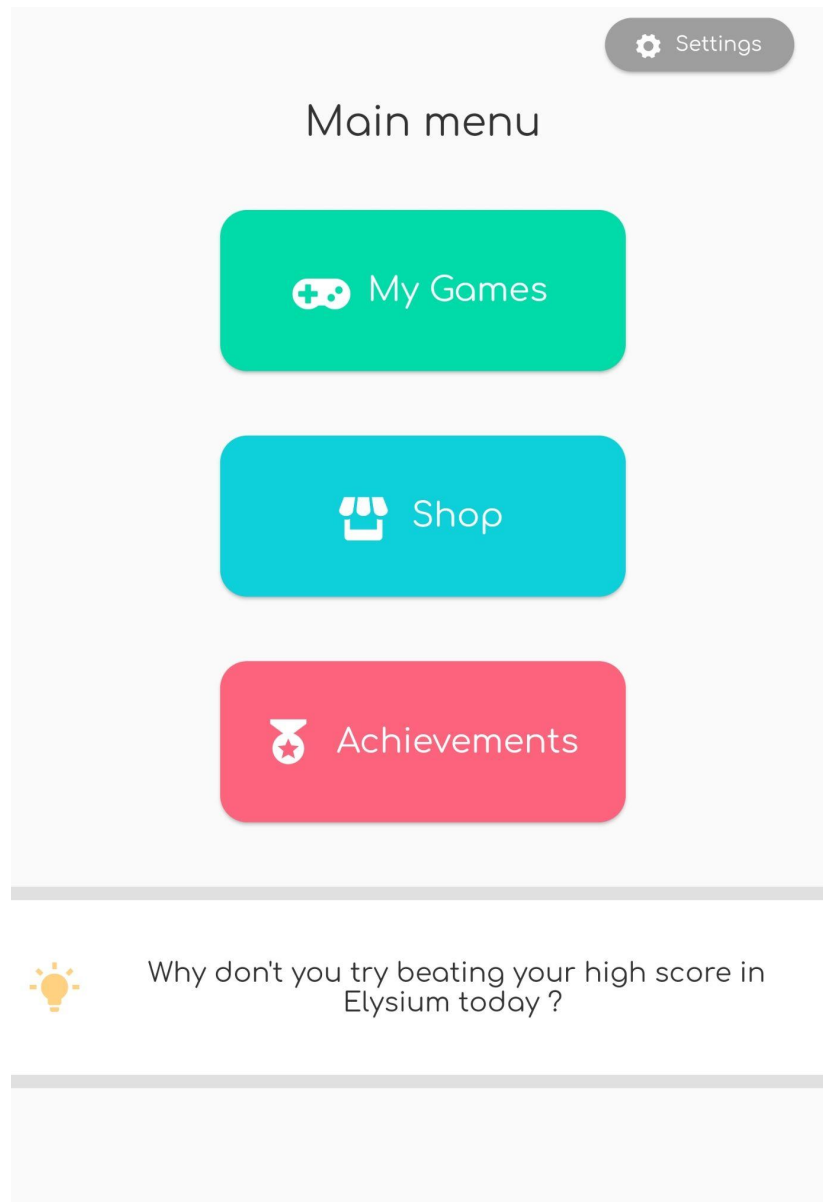

Cellulo Games Hub



Authors:

Baptiste Copros - 312639, Antoine Roger - 312065

Supervisor: Victor Borja

Professor: Pierre Dillenbourg

May 16, 2022

Contents

1. Introduction	2
2. Material and methods	4
2.1 Methodology	4
2.2 Coding choices	5
2.3 Firebase	6
2.3.1 Firebase Authentication	6
2.3.2 Firebase Firestore Database	6
2.3.3 Firebase Storage	7
3. Literature review	8
Visual acuity	8
Color perception	9
Visual field	10
Learning	11
Decision making	12
4. Results	13
4.1 Welcome screen	13
4.2 Authentication	14
4.2.1 Login	15
4.2.2 Sign up	15
4.2.3 Login with Facebook	16
4.3 Main menu	17
4.4 My Games	18
4.5 Shop	19
4.6 Achievements	20
4.7 Settings	21
4.8 Progress	22
4.9 Form	23
4.10 Unity plugin	24
4.11 End user interview	25
5. Future updates	27
5.1 Facebook login	27
5.2 More languages	27
5.3 Progress chart v2	27
5.4 Better suggestion system	27
5.5 Games update	27
6. Conclusion	28
7. References	29

1. Introduction

This report will cover how we created the Cellulo Hub application and how we made it as usable as possible for an elderly audience.

The initial goal of this project was the following: “Develop a multiplatform HUB for the games/activities of the robot to be showcased, and deployed in the device”, with Linux, Web and Android as the main platforms with a heavy focus on user experience and usability.

During the semester, we got new goals from weekly meetings and client meetings: having a menu where the user can track its number of hours played weekly, having achievements to encourage the user, having a friend list to play the multiplayer games easily and having a way for developers to add new games to the game pool. We also extended our work to make a working Windows version of the app. We had the opportunity to have actual users test the application and give us some feedback on some incoherent things that we thought would be understandable or confirm the intuitive parts of it.

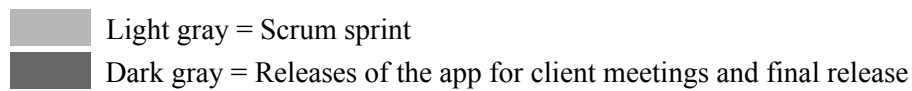
Cellulo Hub is a WIMP multi platform application for Cellulo users to manage their Cellulo games: they can discover new games, create a library of their favorite games, launch a game on any platform and track their overall progress as well as their progress in each game through an achievement system.

Cellulo Hub was designed for elderly people and their caretaker. This introduced one of the biggest challenges: the software should be easy to use and be usable by anyone with any limitations and any knowledge of software interaction. We will cover the ways we tackled this challenge in the first part of this report.

Cellulo Hub was a team project which means we had to use a multi-developer workflow: we will describe the Scrum methodology we used and the tools we used in a second part. We will also describe in detail how we equally split the work in this section with Baptiste mostly focusing on the Firebase backend and Android files management part of the app and Antoine mainly working on the user interface, the Unity plugin for developers and the desktop adaptation.

In the last part of this report, we will summarize what we were able to achieve in the amount of time we were given.

Here are the Gantt charts explaining what we did week after week during our 14 weeks of work:



Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Flutter/Dart discovery														
Welcome screen, main menu, games model														
My games & Shop menu														
Game platforms and shop enhancement														
Link games from Firebase & Linux shell scripts for installation														
Shop & Games cleanup for 1st meeting														
Shop enhancement and cleaning, form enhancement														
Project Cleanup & trending fixing														
Unity plugin start and visual fix														
Windows implementation of the app & form cleanup														
Unity plugin end, settings menu, form & Windows update														
Achievements menu														
CI and Testing														
Report														

Figure 1: Antoine's project timetable

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Flutter/Dart discovery														
Firebase/Flutterfire integration														
Authentication & game installation on Android														
Basic form to add new games														
Authentication menu implementation														
Form, authentication & game installation cleanup before 1st meeting														
Facebook Authentication & bug fix														
Game installation final version & authentication error displayed to the user														
Remove game from library & form fixing Android & number of downloads for trending														
Game suggestion implementation & bug fixing														
Android achievements & Progress menu														
CI and testing														
Report														

Figure 2: Baptiste's project timetable

2. Material and methods

2.1 Methodology

To take full advantage of this project being a team project, we decided to use the Scrum methodology to keep track of our progress and manage reasonable objectives we could reach before the end of the semester. We made a list of tasks we were going to need to achieve to reach our goal through a brainstorming session and a thorough comparison with other softwares of the same genre or meeting the same requirements. We based our tasks on our client's requirements and our user's feedback to reach the best final user experience.

Every week we both selected a task we were supposed to achieve either based either on the initial app requirements, on the new user stories from the clients or the user's feedback. The end of each sprint corresponded to our Monday meeting with Victor Borja, who was our first source of feedback and advice before we showed the new features to the clients. We were able to debate on our vision of the app with him and get his feeling on the new features we had just implemented.

For each of these meetings, we had to create a build of the application which we managed using Continuous Integration and Continuous Delivery.

We used GitHub to manage Continuous Delivery inside our project. GitHub is a Version Control System that helped us keep track of all modifications we did throughout the semester. GitHub also allows each developer to develop a new feature on his own branch while giving all developers the opportunity to look at everyone's code to help the others if they have any problem slowing down the development. Once a feature was finished, we merged our branch inside the Master branch which holds the latest releasable version of the app. For this merge to be possible, the other developer has to verify he does not find any flaws in the pull request's code and the new code needs to pass some automated tests.

These automated tests are part of the Continuous Integration tools we used in our project. We decided to go for a free test automation system integrated to GitHub called GitHub actions. Using GitHub actions, we are able to run a bash script we wrote on a Linux web server which simulates what we do to run tests locally and gives feedback on each test's success or failure. We added automated coverage computation to this bash script to help focus the next tests on the parts that need them the most.

The final methodology we had to choose was our approach towards User Experience since the main challenge of the app is to make it accessible to the biggest possible audience.

For this, we had 3 main sources of feedback to help us design the best possible interface. The first one as we said was Victor Borja, who gave us the first layer of feedback each week on the latest implemented features.

The second one was the meetings with our clients: we had 2 meetings with Barbara Bruno and Hala Khodr, one towards the early middle of the semester to check the project was heading in the right direction and another one towards the end to state what the focus should be on for the last 2 weeks of development. These meetings were especially useful to give us general feedback on the app: what important features are missing, why the flow may or may not be coherent, ...

Finally, the last important source of feedback was feedback from elderly people and middle-aged people that we interpreted as caretaker's feedback. This feedback allowed us to verify our interface was easy enough to use and fix some issues we thought would be clear enough but revealed not to be. It was very useful for small details that could have easily gone unnoticed and to give us feedback on the time it took a user to navigate the app.

2.2 Coding choices

First of all, we had to choose which language/platform we will be using to create the multiplatform hub. Multiple choices were possible, such as Xamarin, React Native, Flutter and more. We decided to stick with our supervisor's initial idea, which was Flutter, mainly because it was using Dart language, which is very similar to Java and Scala, which are two languages we learned recently, and because Firebase is, as Flutter, developed by Google which facilitates the link with the application. Dart is especially performant when working with data collections because it provides

some functional programming tools to iterate, filter or operate over entire arrays in a very elegant and efficient way.

During the project, we also needed to use other languages to accomplish our goals. Indeed, we had to use shell scripts and its Windows equivalent, in order to install/launch/uninstall games on Windows and Linux. In addition to script shells, we had to create a plugin for Unity in C#, which enables a game developer to add achievements to its game which can be read by the Cellulo hub app to show the user his progression on each achievement.

2.3 Firebase

We decided to use Firebase for a lot of purposes for our app. Its use was facilitated thanks to the Flutter packages Flutterfire and Firedart. We created a whole class which contains functions to handle every Firebase functionalities call. Here are the Firebase functionalities we used in our app:

2.3.1 Firebase Authentication

We used Firebase Authentication to handle the authentication in our app. It enables us to choose the Sign-in methods we want to implement in our app. We then chose to use Email/password authentication, where all the emails are saved and handled by the Firebase project, and Facebook authentication.

Firebase Authentication enables a Cellulo hub user who forgot its password, to reset it thanks to an email generated by Firebase.

2.3.2 Firebase Firestore Database

Cellulo hub needs a database in order to store all the games, which user has which game in its library, and how much time has each user played each day. We then decided to use Firebase Firestore Database in order to store all this data. In addition to being an intuitive and easy database to manage, using it enables us to create a direct link with the users in the Firebase Authentication part, thanks to an auto-generated id created during the creation of a new account.

Firestore is designed with collections, made of documents, made of fields. Our collections are :

- Games, made of all the games
- Owns, the relations from user to game
- TimePlayed, which stores the time played per user

However, Firestore is not made to store large files (MBs), so we needed a Storage for these files and download links to them inside Firestore fields.

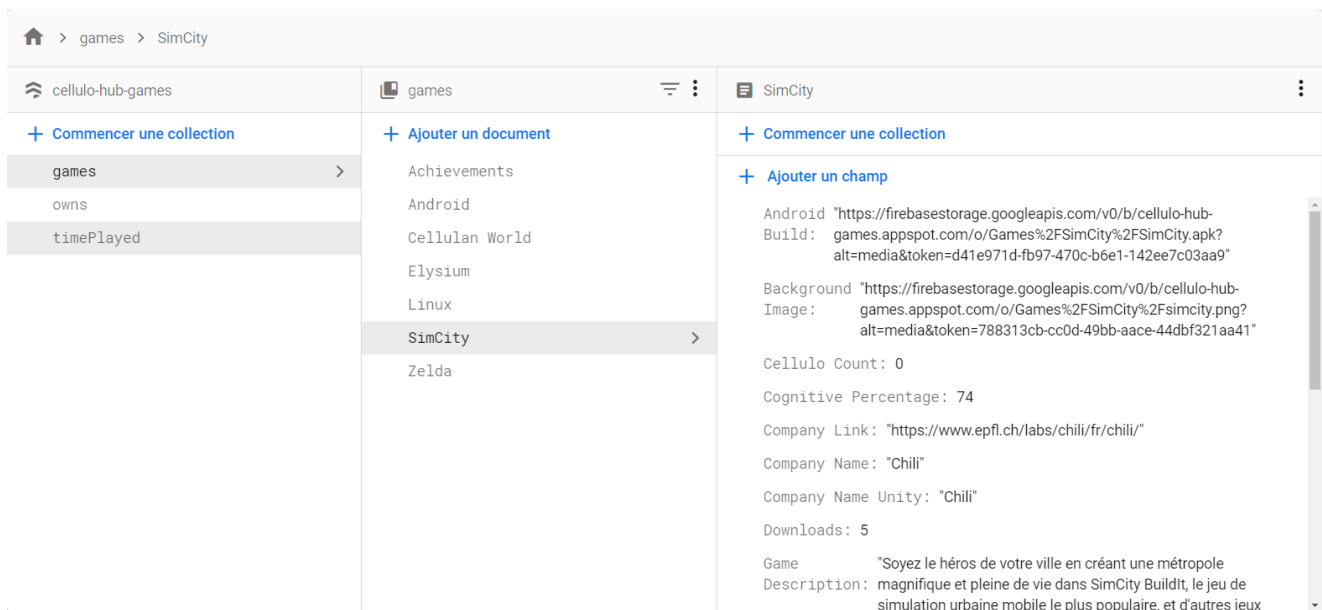


Figure 3: Cellulo hub Firestore project

2.3.3 Firebase Storage

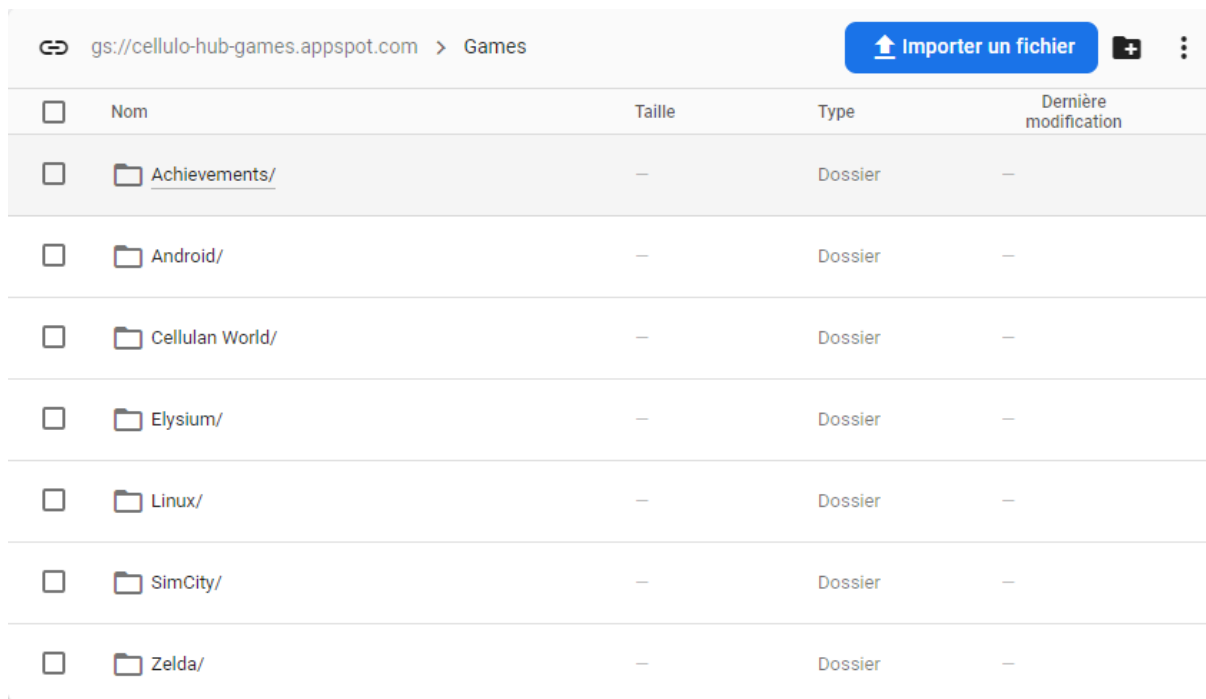
To store large files, we decided to use Firebase Storage. This enables us to store all the games' builds, background image, the Achievements plugin and the tutorial to help developers using the form.

In order to get/download a file from this storage, we need the download URL of an item. When a developer adds a new game to the Firestore Database, in addition to storing the builds and background image, we create the download URL of all of them, and store them in the corresponding fields in Firestore Database. Then when one needs a build or image from the storage in the app, it uses the download URL to get the item, and then it is retrieved.

We decided to arrange the Storage in directories, a main one called Games, within which resides a directory per game.

gs://cellulo-hub-games.appspot.com				
Importer un fichier				
<input type="checkbox"/>	Nom	Taille	Type	Dernière modification
<input type="checkbox"/>	Games/	—	Dossier	—
<input type="checkbox"/>	Achievements.zip	3.9 KB	application/x-zip-compressed	19 mai 2022
<input type="checkbox"/>	Tutorial for developers.pdf	22.7 KB	application/pdf	20 mai 2022

Figure 4: Cellulo hub Firebase Storage bucket










<input type="checkbox"/>	Nom	Taille	Type	Dernière modification
<input type="checkbox"/>	 <u>Achievements/</u>	—	Dossier	—
<input type="checkbox"/>	 <u>Android/</u>	—	Dossier	—
<input type="checkbox"/>	 <u>Cellulan World/</u>	—	Dossier	—
<input type="checkbox"/>	 <u>Elysium/</u>	—	Dossier	—
<input type="checkbox"/>	 <u>Linux/</u>	—	Dossier	—
<input type="checkbox"/>	 <u>SimCity/</u>	—	Dossier	—
<input type="checkbox"/>	 <u>Zelda/</u>	—	Dossier	—

Figure 5: Cellulo hub Firebase Storage Games directory

3. Literature review

Our app is mainly focused towards elderly people and this makes focusing on user experience an especially important part of our project. Designing for older people brings a big accessibility challenge to the development: as people get older, their senses tend to diminish, making it more difficult to use software with complex interaction mechanisms or intricate visuals. We thus needed to adapt our design to help reduce the impact of these physical limitations.

Based on the user's feedback we were able to get throughout development, here are the solutions we used to counter the effect of each decreasing sense:

Visual acuity

Visual acuity is the ability to recognize small details of an object or interface. Starting around an age of about 40, a phenomenon known as "Presbyopia" starts to appear in the human eye: the lens of the eye begins to harden. The *eye's near point*, the closest point at which an object can be brought into focus is pushed back and leads to a loss of visual acuity.

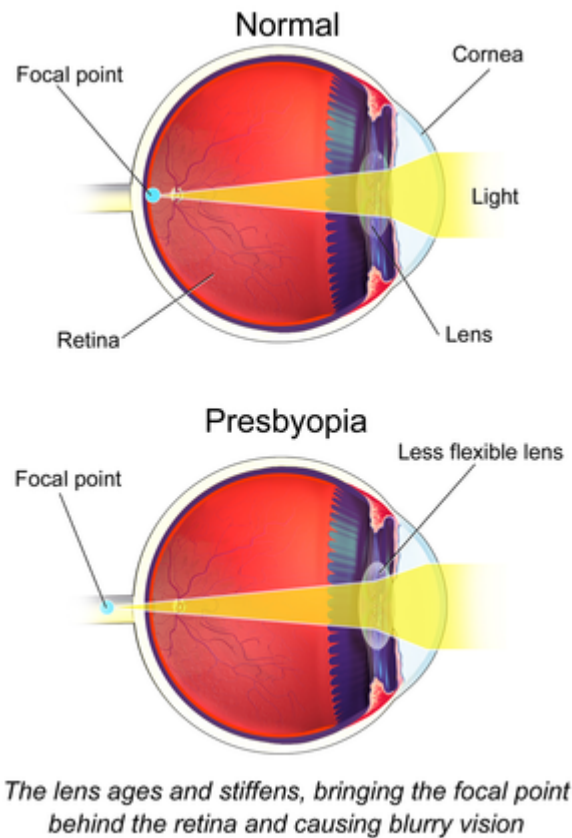


Figure 6: Schematic representation of Presbyopia [1]

This loss can be helped by designing interfaces with bigger elements: bigger texts, buttons and icons. Bigger elements simulate the distance to the eye's *near point* decreasing and compensate for this loss. This solution limits the amount of elements which can be displayed on screen at a given time but we will see later this is not really a limitation since it makes interfaces simpler and thus easier to use. We also decided to go for a font with no serif and big round characters to make it as readable as possible with a low *visual acuity*. [2]

Color perception

In addition to our lens hardening, our eyes' pupils shrink throughout our life. Elder people receive as low as 15% of the light a teenager receives. Receiving less light makes it harder for the eye to detect low light level variations and low contrast variations. *Contrast* is the difference in color or brightness that makes an object distinguishable from its surroundings.

Contrast is more complex to compensate because we need to fit as many contrast levels as possible: if we were to simply increase the contrast of our design's colors using automatic contrast increasing

tools, we would make it more usable to people with very low contrast perception but make it really visually unpleasant for people that still have a high contrast perception and thus less usable.

To fit both of these needs, we opted to pick colors as far as possible to one another on the *color wheel* while trying to make them fit to a coherent *color palette*. This limits visual variations with various shades of the same color which would look good considering every user has a high contrast perception but would look either too flat or too contrasted to people with varying contrast perception. In case the color choice is not enough for people with very low contrast perception, we also added a high contrast mode with colors further away from each other without worrying about the *palette* aesthetic. [3]

Example: here are the normal and high contrast palettes used in the app

Normal contrast					
High contrast					

Visual field

Finally, according to the Royal National Institute for the Blind in the UK, approximately 16% of people between 65 and 74 years suffer from some visual field restriction.

To name a few: “cataracts”, “glaucoma” and “optic neuritis” all affect the *field of vision* in some negative way.

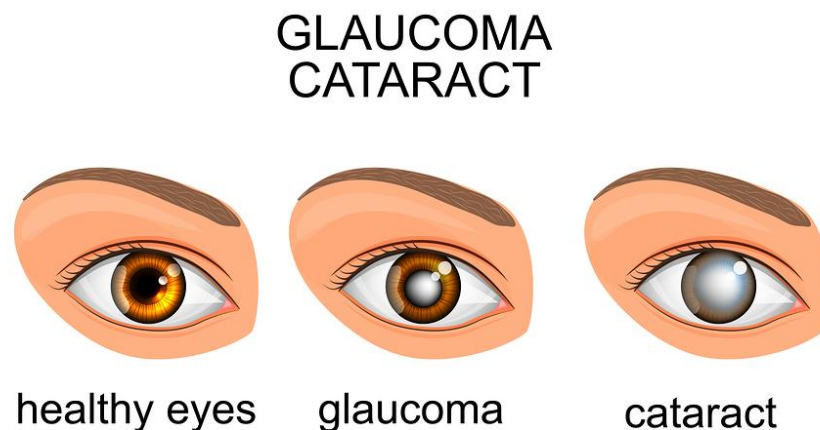


Figure 7: Schematic representation of visual fields impairments [4]

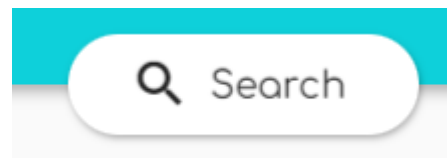
These limitations affect the user's ability to use an interface by limiting the amount of elements that can be seen at once: to accommodate for this disability, we need to minimize the amount of actions required to quickly look at a particular place of the screen or make their interaction optional. As this is also required due to the limited speed at which elderly people navigate an application, this problem was solved by the fact that our users never need to execute an action quickly which gives them time to process where the required action is supposed to take place. [5]

Learning

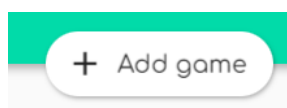
According to a study of information and communication technologies used by the elderly from the “Current Gerontology and Geriatrics Research” in 2015, elder people are as capable of learning as younger people given more learning time. Elderly people find learning technology more rewarding than younger people. Learning takes time and a good memory to be able to use an unknown software fluently: 66.2% of the people interrogated in the study declared barely or never using the Internet.

To make learning motivating and as enjoyable as possible, we used multiple different techniques:

- Elder people are most comfortable with what they already know: the fastest way to make them understand a new technology concept is thus to relate it to something they already know. To do this, we visually mapped important parts of the app to their real life equivalent: the My Games library uses a gamepad icon, the Shop uses a Shop front icon, the search bar uses a magnifying glass icon and so on.



- Older people tend to look at everything on the screen to minimize their chance of pressing the wrong button. Thus, we limited the screen complexity to a minimum: every button is highlighted and has free space around it to make sure the user has the highest chance of seeing it. This also helps with memory complexity: each path you can take in the app is never longer than 4 consecutive clicks and each of the screens the user went through is simple enough to get the right input consistently. This reduces the chances of getting lost to a minimum.
- To help the users learn faster, we tried adding informative feedback wherever possible in the Log In part for example when we differentiate between wrong password and wrong username and explain to the user which input they should re enter and why.
- Finally, to reward people who made the effort of learning the app, some more interesting features can be used in the top right button of some screens: we have implemented shortcuts between screens, a search bar in the shop to avoid scrolling all existing games and a settings menu to track your progress and change the color theme. [6]



Example: “Add game” goes directly from My Games to the Shop but is only a faster way to do what users can already do through the main menu.

Decision making

With age comes a decrease in decision making ability accompanied by a loss in *visual thinking*. Visual thinking is the ability to correlate visual elements in an interface. This decrease is due to a decreasing pattern recognition ability. Older people will not look for new kinds of inputs or naturally click on a text whose positioning only indicates it is clickable.

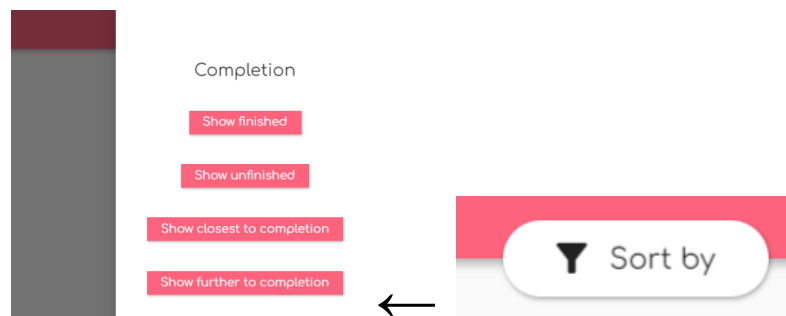
To help respect these constraints and reduce the cognitive load to a minimum, we opted for an interface that would be composed only of the very simplest elements almost everyone has seen somewhere.

We tried to reduce any screen to a basic list of buttons and when really needed an input text field for the connection process.

Flutter already provides micro-informative feedback such as color changing on hovering a button to indicate a possible interaction and we tried to elevate every button to compensate for this hovering property on touch screens.

The main exception to this is the scroll input to browse the games list which we reused whenever possible to avoid confusing the users with different types of scrolling.

We tried to keep the animations to a minimum to have the interface seem fluid and more elegant while not disturbing the user. [7]



Example: the drawer to sort achievements types is usually opened through sliding a finger from the left of the screen towards the right but we added a clear button to open it. The swipe method is still there for more experienced users.

4. Results

In 14 weeks, we managed to achieve the main goal, which was having a multiplatform hub, containing all the cellulo games, which can be installed and launched on multiple platforms. Here is the app flow diagram:

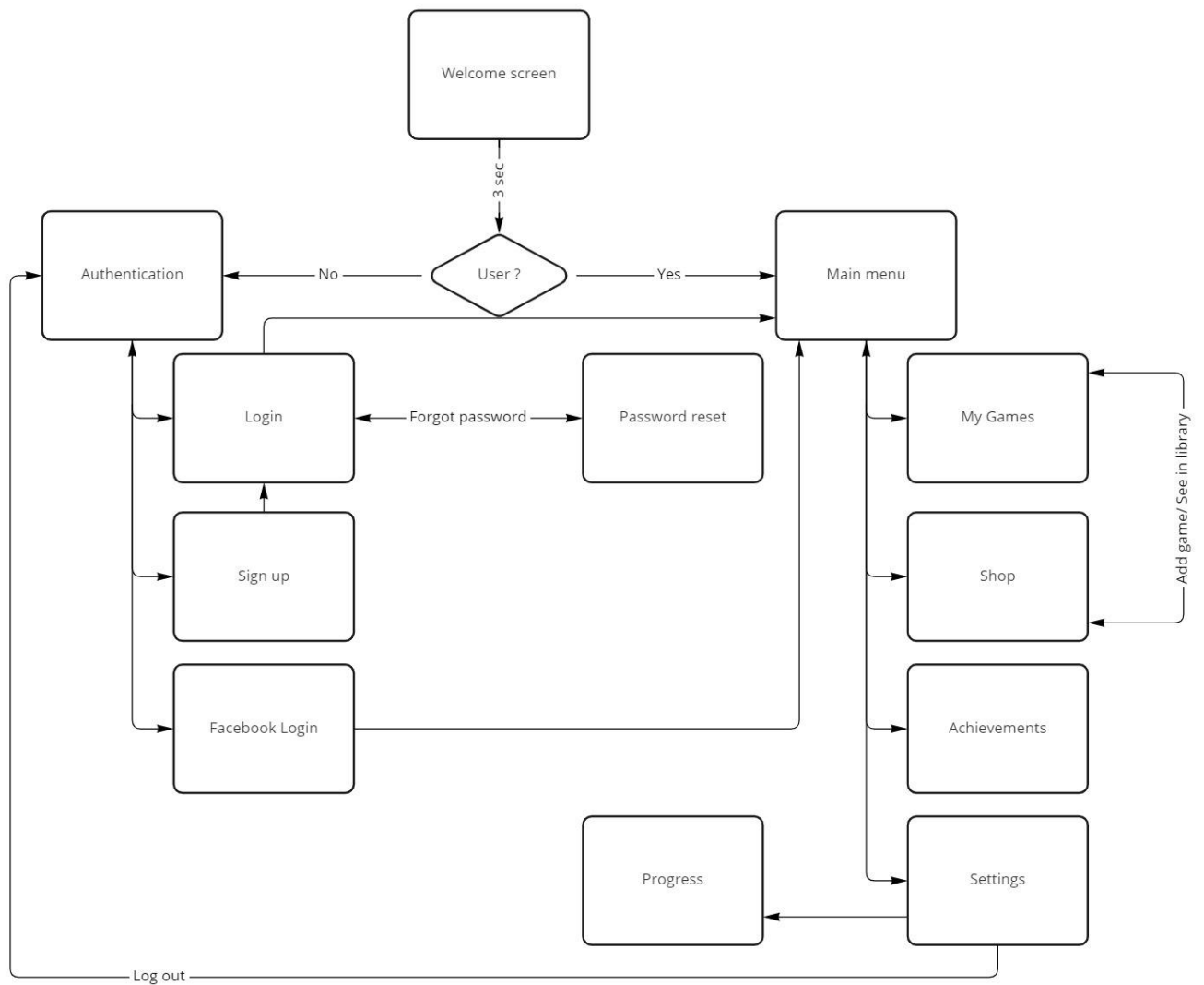


Figure 8: App flow diagram

4.1 Welcome screen

The welcome screen is the first screen the user will be facing when launching the app. It is displayed for 3 seconds, during which the program checks if the user is already logged in, and determines which screen to show afterward: If a user is logged in, the main menu is displayed and the list of all games and the user game list are loaded, otherwise the authentication menu is displayed.



Figure 9: Welcome screen of Cellulo hub application

4.2 Authentication

The authentication menu of the app has been made as simple as possible. Even though we don't expect elderly people to do the authentication setup, the caretaker or the person who will do it shouldn't have any problem.

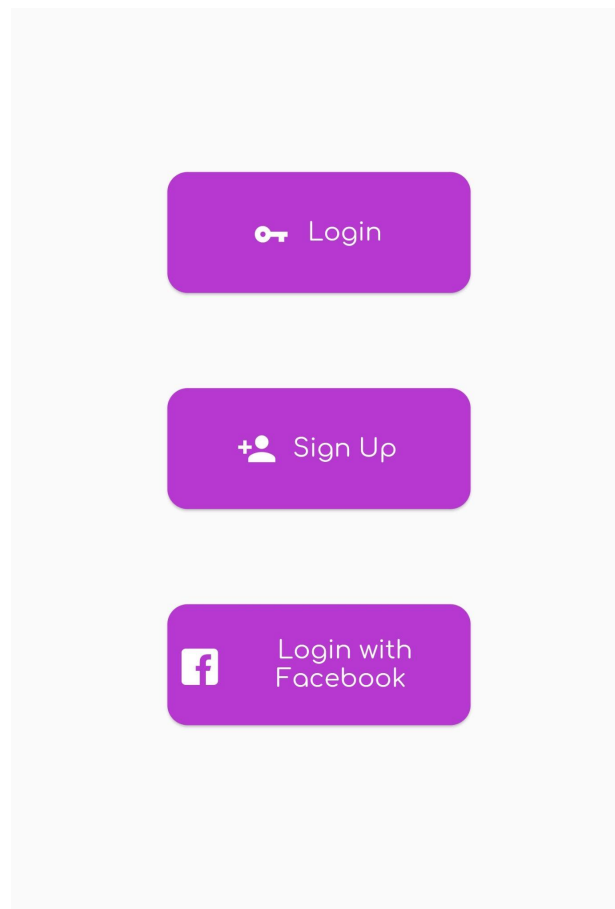
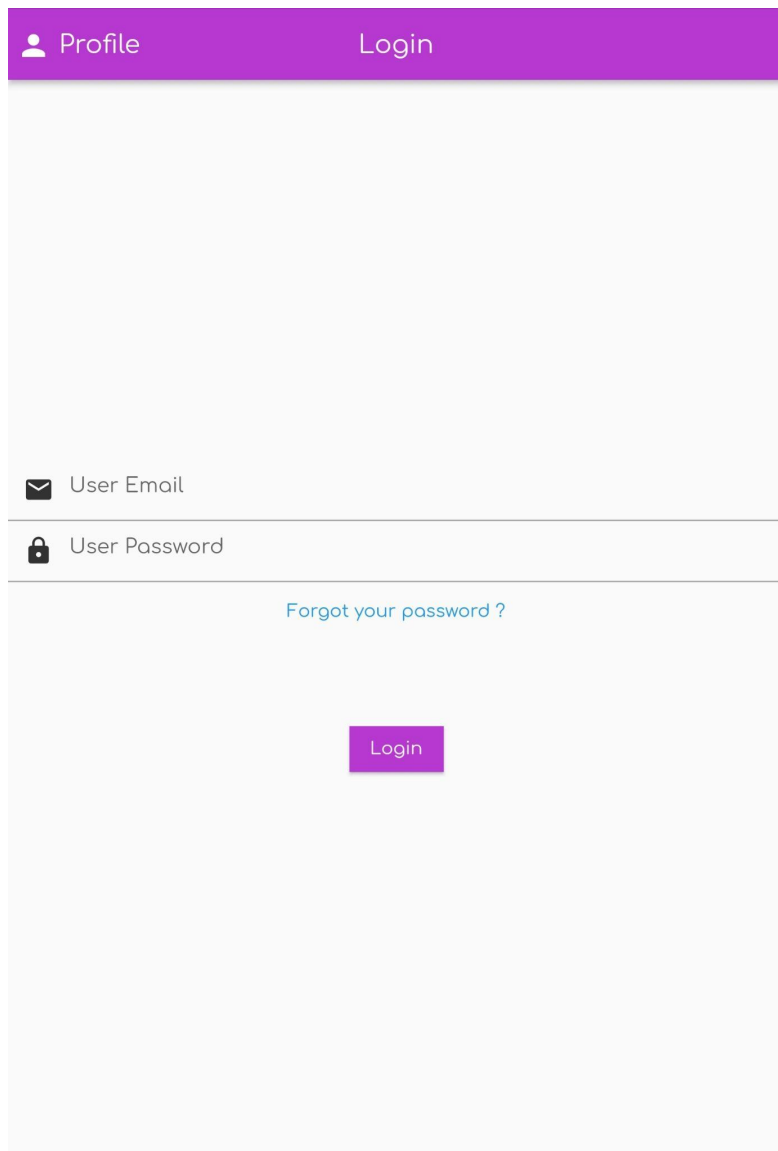


Figure 10: Authentication menu

The menu is composed by 3 buttons, each one of them redirecting to its own page:

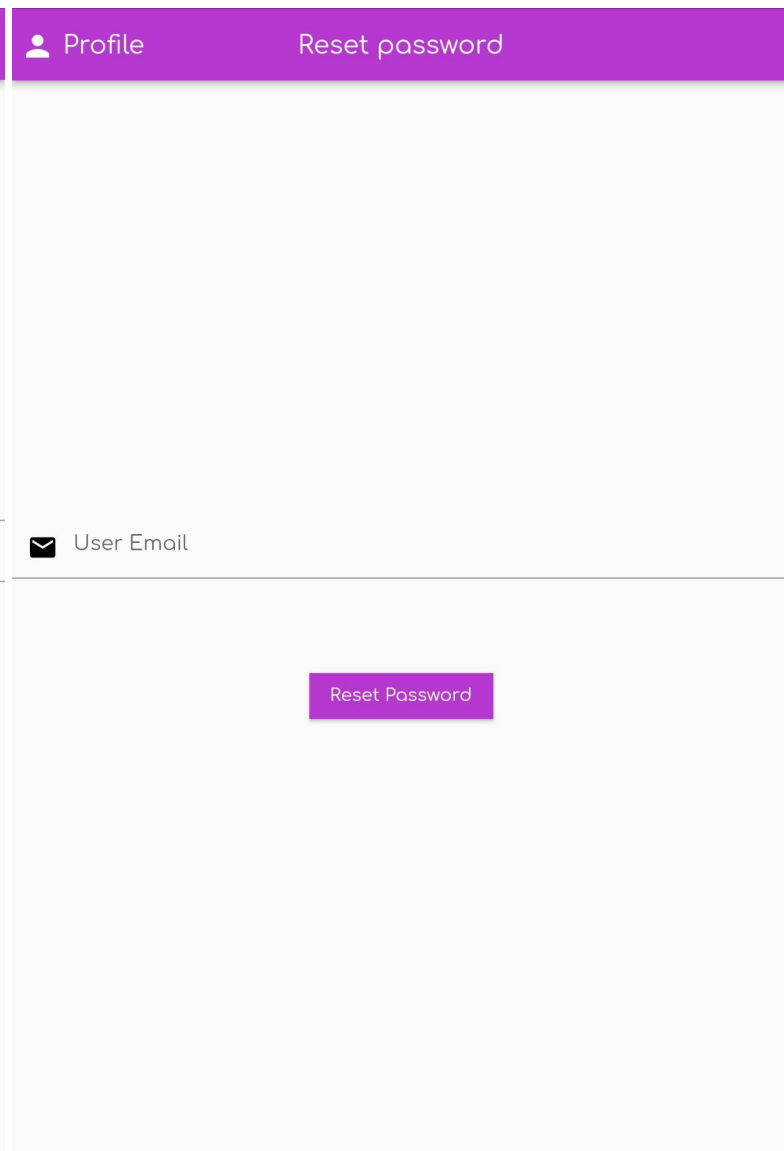
4.2.1 Login

The login screen is composed of 2 text fields for the email and the password of the user, a text button “Forgot your password ?” which redirects to the Reset Password page, and a Login button.



The login screen features a purple header bar with a profile icon and the text 'Profile' on the left, and the text 'Login' on the right. Below the header, there are two text input fields: 'User Email' with an envelope icon and 'User Password' with a lock icon. A link 'Forgot your password ?' is positioned below the password field. At the bottom center, there is a purple 'Login' button.

Figure 11: Login screen



The reset password screen features a purple header bar with a profile icon and the text 'Profile' on the left, and the text 'Reset password' on the right. Below the header, there is a single text input field 'User Email' with an envelope icon. At the bottom center, there is a purple 'Reset Password' button.

Figure 12: Reset password screen

4.2.2 Sign up

The Sign up screen is composed of 3 text fields, for the email and password of the user, as well as a confirmation for the password, and a button to create the account.

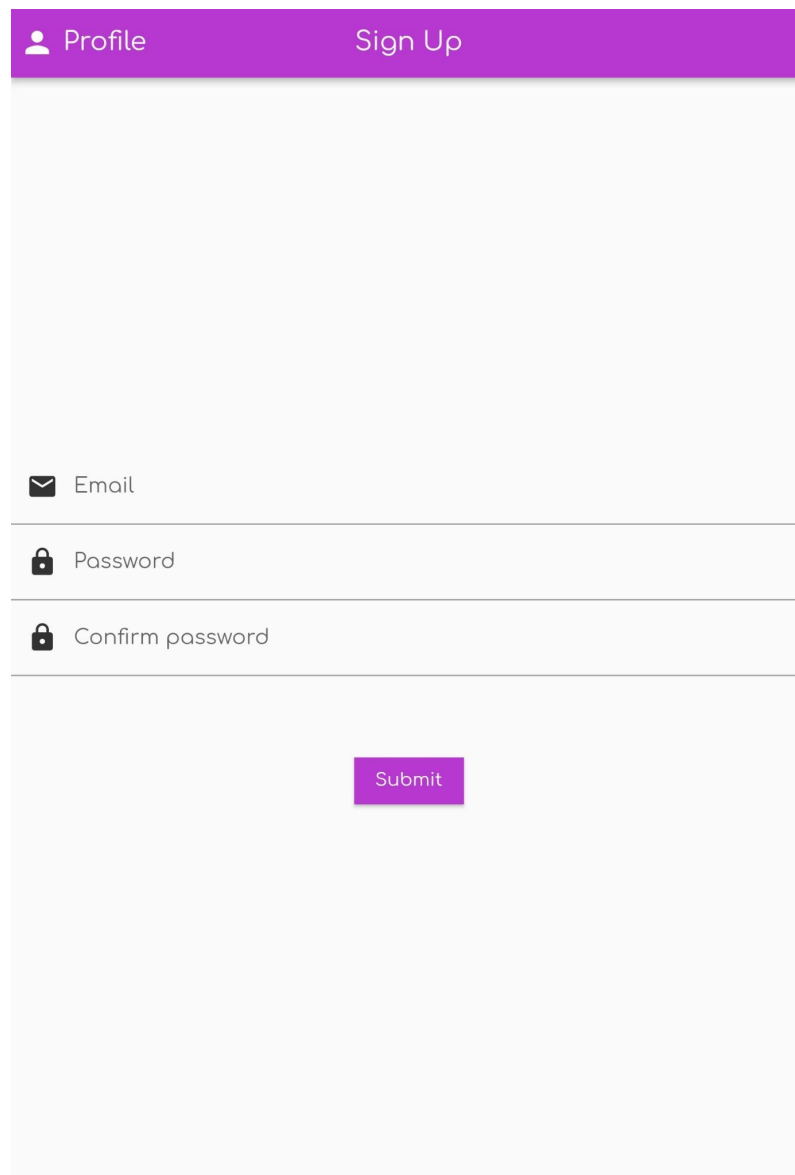
The image shows a mobile application sign-up screen. At the top, there is a purple header bar with two options: 'Profile' (with a person icon) and 'Sign Up'. Below the header is a large, empty light gray rectangular area. Further down, there are three input fields, each with a small icon on the left and a label on the right: an envelope icon for 'Email', a lock icon for 'Password', and another lock icon for 'Confirm password'. At the bottom center of the screen is a purple rectangular button with the word 'Submit' in white text.

Figure 13: Sign up screen

4.2.3 Login with Facebook

The “Login with Facebook” button will redirect you to an official Facebook page, where you would be asked to login, and then accept that the Cellulo hub application collects some of your basic data (Profile, friend list...). Once done, you will be redirected to the Cellulo hub application, logged in with your Facebook account.

This part used to be working, until the Meta for developer website blocked the application, because we needed to add a “Confidentiality politic URL”, which could be done in the future. We then stopped implementing the friend list since we couldn’t get Facebook friends.

This button is also not available for Linux and Windows, since the Api doesn’t support them. However the web version has its implementation done.

4.3 Main menu

Once the user is logged in, the main menu is shown to the user. It is composed of 3 main buttons, which are :

- “My Games”, which redirects to the user game library
- “Shop”, which redirects to the shop containing all the game available on the app
- “Achievements”, which redirects to the list of achievements for all the games the user played.

Moreover, at the bottom of the screen, we added a suggestion that encourages the user to play one of the games he has in his library. Clicking on the suggestion will then redirect the user to its library, with the suggested game panel opened. This suggestion system enables a user to launch a game in 2 clicks once in the main menu.

Finally, on the top right corner of the screen is located the “Settings” button, which redirects the user to the settings screen.

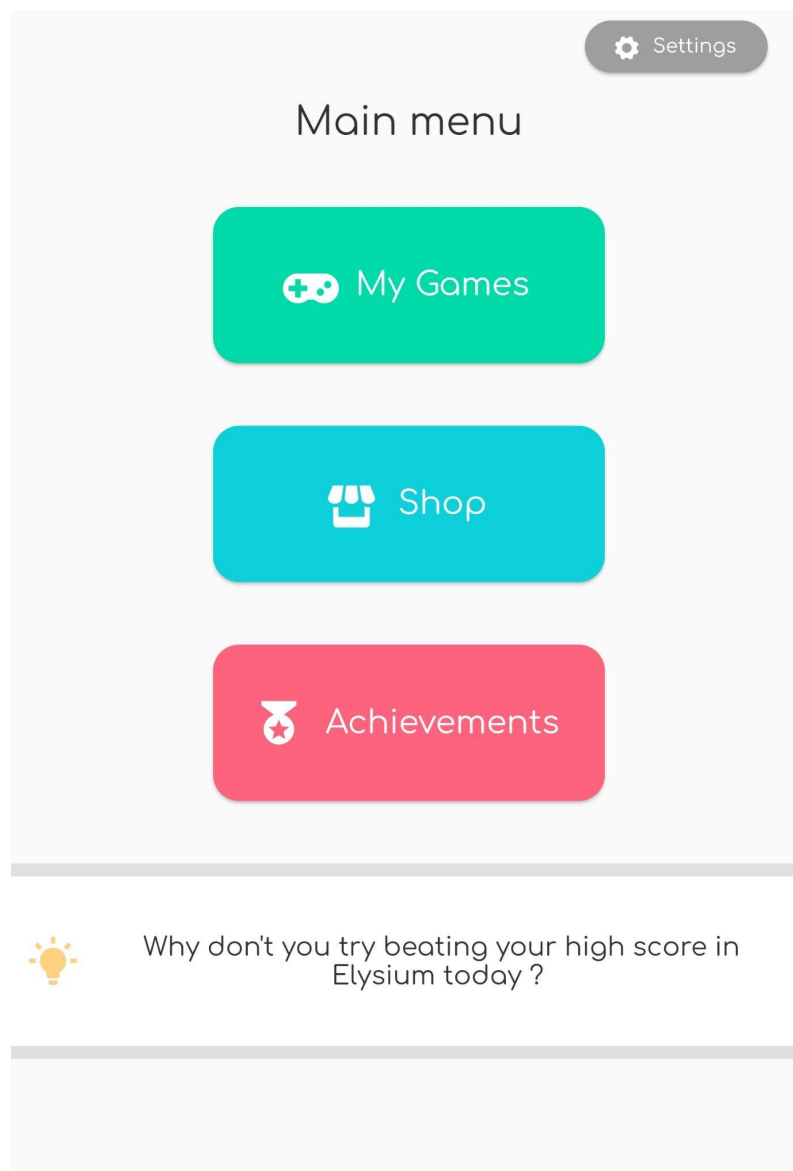


Figure 14: Main menu

4.4 My Games

The “My Games” menu is the library containing all the games the user added from the shop. It contains a list of all the games, and a button “Add game”, which redirects the user to the Shop.

The games shown are lightened if installed, and darkened if not. We also have a little warning for each game that is unavailable for the platform the user is currently using. However, games that have a web version, can still be launched from any platform, but the web site of a game is not high priority compared to the current platform build if any, which means that the Android app will be launch if any, or the web site if not, if the user is on an Android device.

Each game can be expanded with a single click, which will show the user:

- All the platforms the game is available on,
- Some information about the game such as: the creator, with a link to its website if any, the game description, and the physical/cognitive/social specifications,
- The “Install/Uninstall” and “Launch” buttons, the “Remove” one which remove the game from the user library, and the “See more” button which brings the user to the Cellulo requirements and Achievements of a game. The Install and Uninstall buttons are mapped to an indicator explaining to the user which task is being executed to avoid confusion.

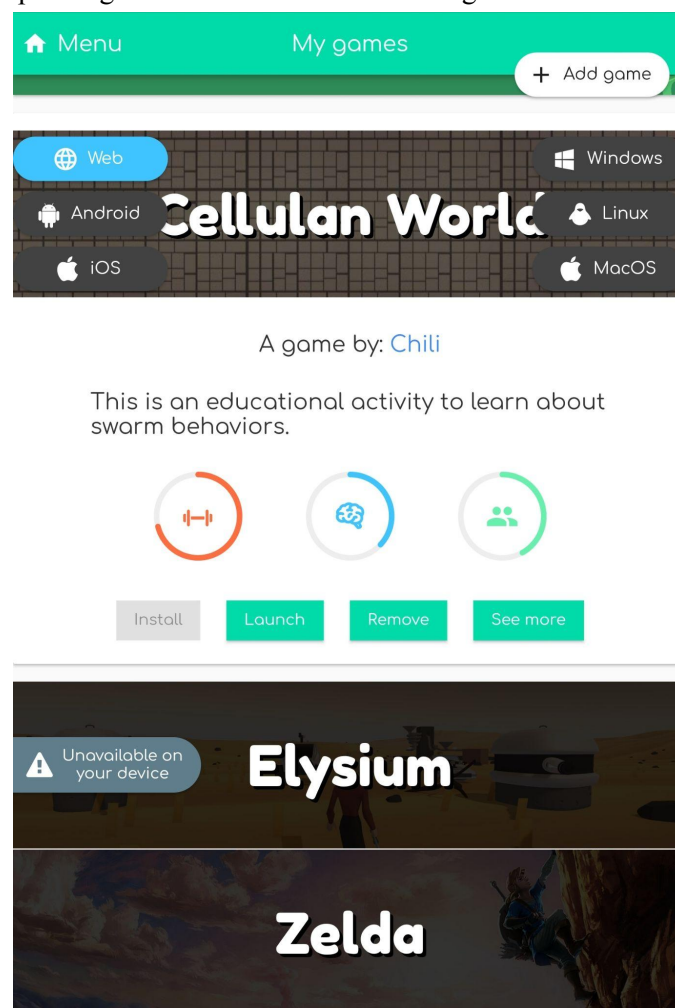


Figure 15: My Games screen

4.5 Shop

The “Shop” menu contains all the verified games available on the app. It first has a Trending scrollable bar, which displays the most downloaded games of the app. Then, it contains 4 tabs, containing : All games and the most physical/cognitive/social games. Finally, it contains a “Search” option, which enables the user to filter the games per name. One can click on its desired game, which displays the availables platforms, information about the game, and the button to add the game to the user game library.

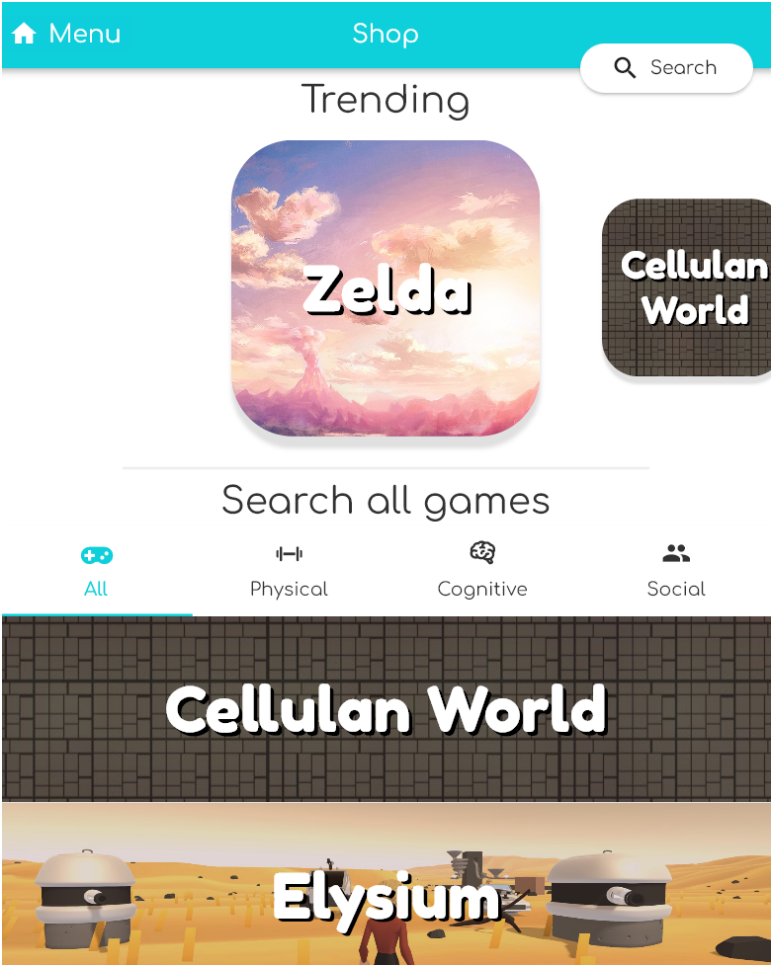


Figure 16: Shop screen



Figure 17: Shop game expanded

4.6 Achievements

The achievements menu contains all the achievements of a user. It was made to encourage users playing more, by displaying the milestones they achieved. Achievements are displayed game per game, showing if they are completed, in progress or is a high score.

The user has the option to sort the achievements by:

- Finished achievements
- Unfinished achievements
- Closest to completion
- Furthest to completion
- Highest high score
- Lowest high score

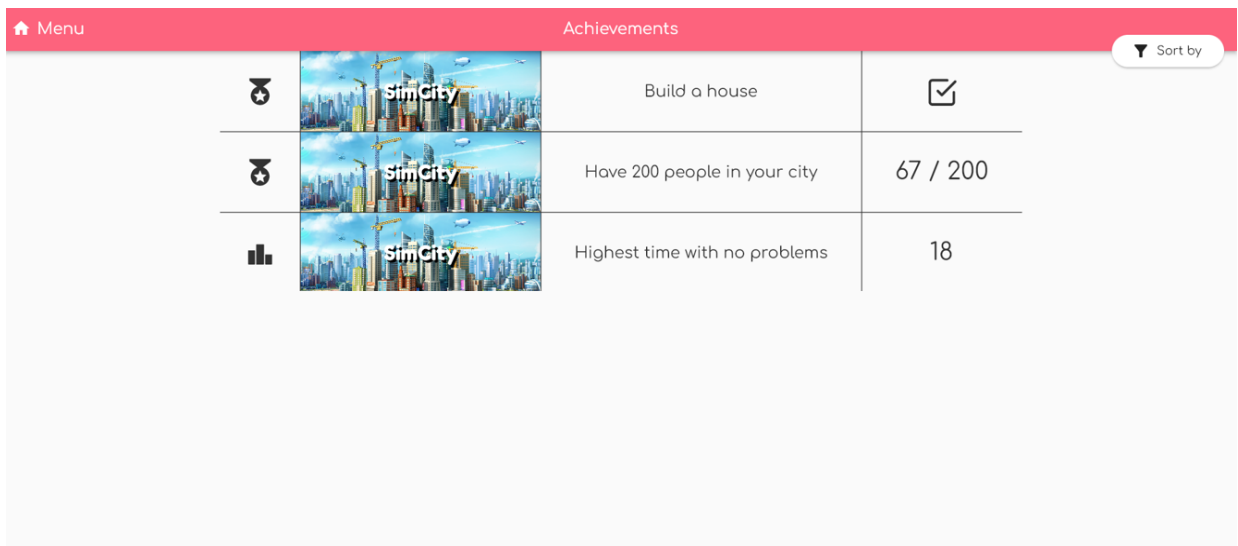


Figure 18: Achievements screen

4.7 Settings

The settings menu is splitted in 2 parts: the Visual settings and the Profile settings. The visual ones allow the user to switch between the Day/Night theme and activate/deactivate the High contrast theme. The profile ones allow the user to go to their progress screen, and to Log out from the current account.

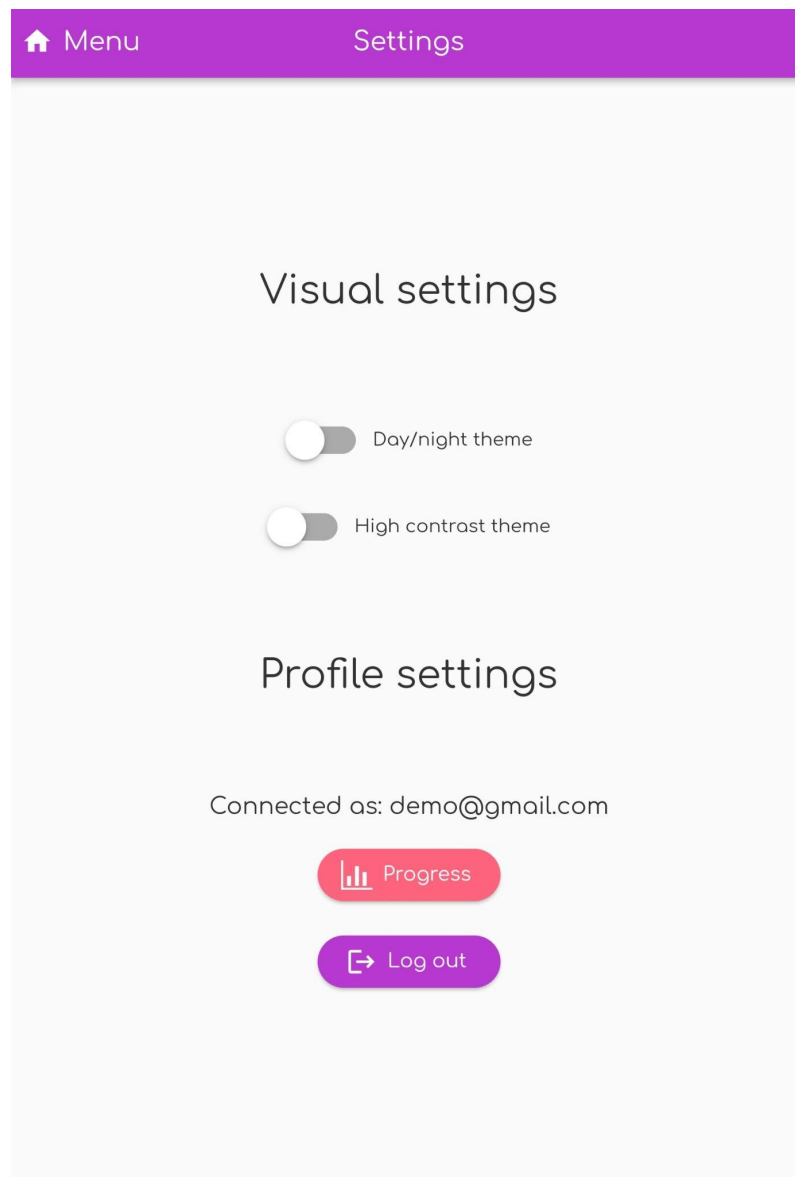


Figure 19: Settings screen

4.8 Progress

The progress menu, which is more dedicated to caretakers, is composed of the statistics about the game time of the user. We can find a chart, which displays the number of hours played this week, and the total hours played overall and this week.

This menu is only working on Android because we had to use a different package than Flutterfire (Firedart) for the other platforms (except Web), which makes it difficult to use the same logic as for

Android, and we chose not to spend our remaining time fixing these platforms since Android is the most important one.

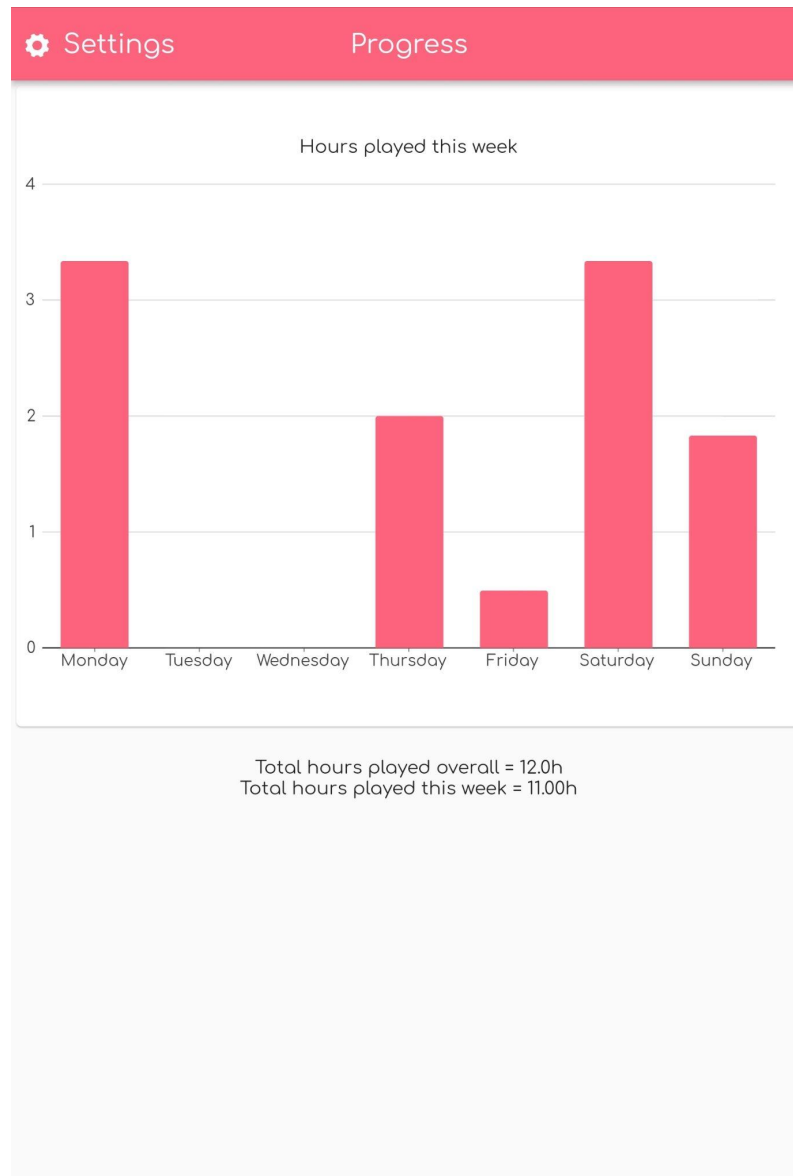


Figure 20: Progress screen

4.9 Form

In addition to the Cellulo hub app, we created a form for the game developers, so that they can add new games to the Cellulo hub game pool.

The form contains the information needed for the creation of the game, with both required and optional fields, the Unity plugin for the achievements and the tutorial to upload a game.

At the end of the form, the developer can visualize a preview of the game in the app, which is composed of the background image with the game name on top.

To guide the user as best as possible we highlighted required fields and rejected every unsupported file type or any incomplete submission.

When a developer fills the form correctly and presses the “Submit” button, the game is created on the database, but it is marked as non verified, and is not yet displayed in the app. Then its content can be verified before being part of the official Cellulo hub game pool.

Upload a game

Download: [Unity plugin](#)
Download: [Upload tutorial](#)

Game name [REQUIRED]
Sonic Cellulo

13/50

Game name in Unity [REQUIRED]

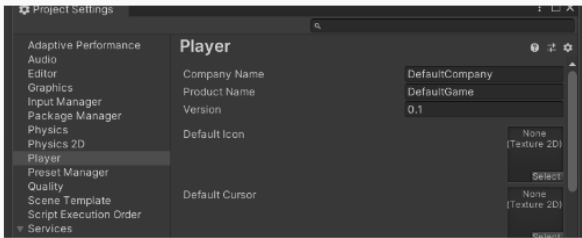
0/50

Company name [REQUIRED]


0/50

Company name in Unity [REQUIRED]

0/50



0/50


 Personal website link

Game description [REQUIRED]

0/300

Game instructions (how to play) [REQUIRED]

0/300

 Web build


 Web link

Figure 21: Form part 1

Upload a game

Game instructions (how to play) [REQUIRED]

0/300

Web build


Web link

Linux build

Windows build

Android build

Background image [REQUIRED]
sonic.png



How physical is your game ?

0

100

How cognitive is your game ?

0

56

100

How social is your game ?

0

100

How many Cellulos does your game use ?

0

8

Submit

Figure 22: Form part 2

4.10 Unity plugin

In order to help developers add achievements to their Cellulo games, we created a Unity plugin providing them with a visual interface to create a list of achievements of 3 different types: a “boolean” achievement that corresponds to something you have either done or not done yet in the game (for example: Scoring a goal while in the air), a “multiple steps” achievement which is a goal you reach in a given amount of steps (for example: Scoring a 100 goals) and a “high-score” achievement which is the highest score you got on some part of the game (for example: Highest number of goals scored in 5 minutes).

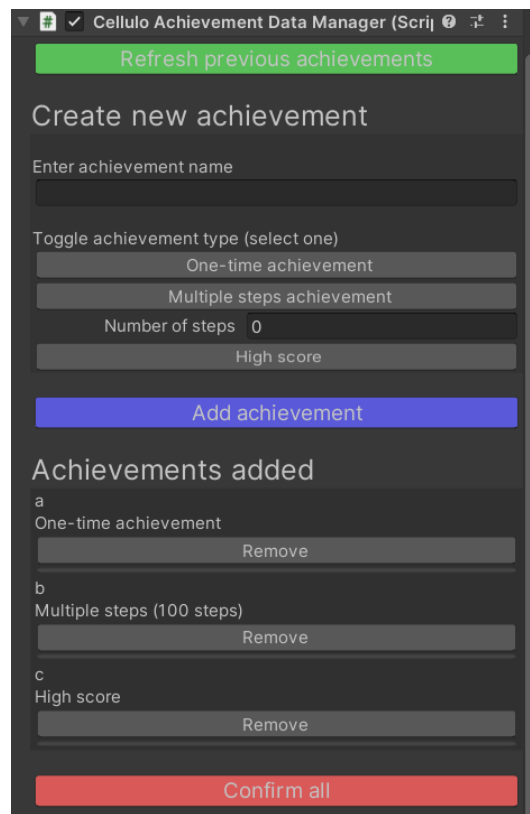


Figure 23: Plugin visual inspector

This file generation system is accompanied by a set of 3 methods allowing the developer to update each of the previously defined achievements inside his code using only their name.

```
// Changes value of achievement
public static void UpdateAchievementValue(String label, int value)
{ ...
}

// Increase value of achievement by 1
public static void IncreaseAchievementValue(String label)
{ ...
}

// Set achievement to completed
public static void SetAchievementTrue(String label)
{ ...
}
```

Figure 24: Plugin available methods

4.11 End user interview

In addition to the standup and clients meeting, we needed to test UX by getting feedback from end users, which are elderly people and caretakers. Their feedback was extremely useful, and really meant we were going in the right direction with our app. It allowed us to confirm the user experience was good and the design choices we made were coherent. We interviewed 2 elderly people aged 79

and 85, at 2 distant developing times of the project, and one 57 years old woman at the late time of development.

We asked these 3 people the same task to execute, which where :

- Go to all the different menus and come back to the main menu, with as an instruction “Where would you go if you wanted to access your games ?” or “Where would you go if you wanted to see all the games ?”,
- Inside My Games menu, Install/Launch/Uninstall a specific game,
- Inside Shop menu, look for a specific game with text search, look for the game with the highest Physical/Cognitive/Social percentage, add a specific game to the library and go to the library after adding it,
- Go to the settings and log out from the current account

The first elderly person we interviewed was a 79 woman during the Easter break, who uses a computer and a smartphone for basic purposes. Overall, she executed most of the instructions well, except those which needed to press on an icon to go to the textual search in the shop, and going back to the main menu from another menu.

Her feedback was the following: Icon should always be accompanied by a descriptive text such as “Home” for the home button or “Search” for the textual search and she was not convinced by the utility of showing the available platforms for each game.

The first issue has been fixed and icons are accompanied by text in our final version. The second one has been discussed with the clients and our supervisor, and we decided to keep the available platforms shown.

The second elderly person we interviewed was a 81 years old man during the 11th week, with presbyopia and reduced vision to one of his eyes due to an accident, who is using a smartphone daily for a lot of purposes. He overall executed most of the instructions correctly, except that he was having trouble with English and that he was clicking on the text next to an Icon instead of the Icon itself.

Finally, we interviewed a 57 years old woman during the 11th week, with presbyopia, who is using her smartphone everyday for mainly Facebook. She executed all the instructions almost instantaneously, but had the same problems as the 81 years old man, which were English and pressing text next to icons.

From these last two interviews, we decided to keep the core of the app the same, since it was pretty simple to navigate for them, but we did change the icon and text to make them into a single button instead of an icon button only.

However, the language was not an issue we decided to solve, since English was the language desired in the initial goal.

5. Future updates

Here are some ideas for future updates the app could need:

5.1 Facebook login

The Facebook login part is almost implemented, but it has to be finished if the app wants a friend list with the Facebook Friends. Right now, the feature is missing: access to the API thanks to a confidentiality URL on the Meta for developers website, a link to Firebase Auth to create Firebase user id for facebook accounts, and finally the implementation of the friend list system with an invitation system.

5.2 More languages

We saw during the interviews that English can be hard for French speaking people. Depending on the choices made for a future release of the app, more languages could be helpful.

5.3 Progress chart v2

Right now, the progress chart only works on Android because Google didn't finish implementing the Flutterfire package for Linux and Windows. In addition to this, the method we use to keep track of the time played is based on a difference between the local time played and time played saved online. Each time a user opens the app, the program checks if the time played is the same locally and on Firebase Firestore, and if not, a new field is created with the time and the difference between the two times played. This way we can get all the times that are in the current week to get the week time played. But this system could probably be simplified and less space consuming in the Database.

5.4 Better suggestion system

Our suggestion system randomly takes a game in the library and suggests it. In the future, one could create an algorithm that chooses the suggested game on more precise conditions, for example it could choose a game in the shop that is not in the library yet, but which fits more to the user's needs. Also, user preferences could be added, such that a user who needs to train physically will be suggested physical games more.

5.5 Games update

In addition to the form, something should be added to enable developers to send updates of the game into the Database. Right now, it is technically doable if the developer fills the form again with the same name, but everyone could override any of the games in the game pool, but the update should be validated first. This means that the form should be accompanied by an update form that has a user system to keep track of who created which game, so that no one can override someone else's game.

6. Conclusion

During this project, we managed to reach the initial goals, which was to create a multiplatform hub for Cellulo's games, which enables the user to Install and launch a game on Android, Linux and Web.

Additional goals have been achieved, except one. We managed to:

- Create a Progress menu with the time played per week per user,
- Create an achievements system with a Unity plugin for developers and a menu which displays all the achievements users have,
- Create a form for developers to add new games to the Cellulo hub game pool,
- Create a working Windows build for the app.

However, we didn't finish implementing a friend list system.

The final version of the application contains 3 main screens created for elderly people, which are the My Games library, the Shop and Achievements. It has been designed in a way that a user can launch an already installed game in 2 clicks (suggestion bar + Launch) if the game is the one suggested in the main menu or 3 clicks : My games + desired game panel + Launch, if the game is not suggested. If the app is not installed, a user can launch a game in 5 clicks : Shop + desired game panel + Add to My Games + (install if on Android) + See in library + Launch. The install click on Android is compensated by the "Open" option in the install popup while the Android app has been installed.

As we expect the most frequent user being an elderly person opening the app on an Android tablet already connected to an account, this user will mostly click 2 or 3 times, which can be considered as simple.

All things considered, we would like to conclude by saying that this project brought us a lot of experience, not only on the development part where we had to learn everything from scratch, but also in the communication within the project and with the clients.

Thanks to Barbara Bruno and Hala Khodr for their enthusiasm about our project.

Special thanks to Victor Borja who supervised us and was super encouraging during the whole project !

7. References

[1]

<https://upload.wikimedia.org/wikipedia/commons/thumb/8/8b/Presbyopia.png/290px-Presbyopia.png>

[2]

<https://medlineplus.gov/ency/article/001026.htm>

[3]

<https://www.cns.nyu.edu/~david/courses/perception/lecturenotes/brightness-contrast/brightness-contrast.html>

[5]

<https://www.lindavision.com/wp-content/uploads/2021/08/healthy-eye-glaucoma-cataract-min.jpg>

[4]

<https://www.britannica.com/science/visual-field-defect>

[6]

https://www.researchgate.net/journal/Current-Gerontology-and-Geriatrics-Research-1687-7071/publication/281644299_ICT_Learning_by_Older_Adults_and_Their_Attitudes_toward_Computer_Use/links/618bb8bd3068c54fa5c8ca51/ICT-Learning-by-Older-Adults-and-Their-Attitudes-toward-Computer-Use.pdf

[7]

<https://www.sciencedaily.com/releases/2009/01/090128092341.htm>