Project 14

“ARCADE GAMES FOR TACTILE TABLE”

BRACQUIER Benjamin
DUCROS Vincent
BOMPARD Noah
KÜRKLÜ Fikret

Tutors:
Didier Donsez, Nicolas Palix

Department: INFO4 - Polytech Grenoble - UGA

Table of content

Introduction

The objective of this project is to put on the touch table a certain number of games (arcade games, pinball machines, board games, card games, MCQ like Polystar) and simple applications (photo sorter, slideshow, information site Polytech, ADE of the day for the Polytech Grenoble sectors, CROUS menu of the day, etc. on the 3 touch-sensitive coffee tables (present in room 106). The team on this project had to set up the tables that haven't been used for a long time, make an inventory of the existing applications and study the possibilities of using these tables in public places.

Our first demonstration was during Polytech Grenoble's 2023 Open Day.

Every documentation made to explain the processes will be featured within this report.

Analysis of the existing

First, before establishing the specifications, we must understand in which state we recovered the 3 touch tables, so the team made an inventory of the existing capabilities of the project:

On the 3 tables :

- One missing wifi module
- One internal clock battery down (cmos)
- Old ubuntu
- Unknown password
- Some physical damage (including a touch dead zone at the top of one of the tables)
- Some wires damaged

Internal Inventory :

- No files useful (only pre installed games from the app store)
- we made a save of an old project made by the previous team that worked on these tables, but the project's not usable since it needs another machine as a server (which has been down ever since)

Characteristic of the 3 touch tables (with the ubuntu version updated):

Table 1 (Polytech) i5 4250u / 4GB ram.

Table 2 (Polytech): i5 4250u / 4GB RAM

Table 3 (Fablab): i5 4250u / 6GB of ram

note : there was a 3rd table at polytech, but it is damaged and thus not used throughout our project.

Specifications

After doing an analysis of the touch table,we started defining the goals and specifications of our project taking into account the original requests of the tutors and ordering them by priority :

Team	Specifications (global)	LVL
H/S	Operate the table	1
H	Install a button to flip the screen	1
S	Porting a number of simple games and applications on the tables	1
S	Create 3d apps via three.js: polytech 3d modeling	3
H	Documentation of the installations for setting up the tablet	1
H	Update touch tables to Ubuntu 22.04	1
H	Any application can be imported and used on the tablet	2
S	Website with the CROUS references	1
S	Polytech integration (ade)	1
S	Evaluate the portability of the native (app store) ubuntu applications.	2
S	Having a kiosk mode	1
H	Manage the starting applications and setup	2
S	Create an interface to manage games, apps.	1
H/S	Evaluate and improve the security of the table’s usage	3

Organisation

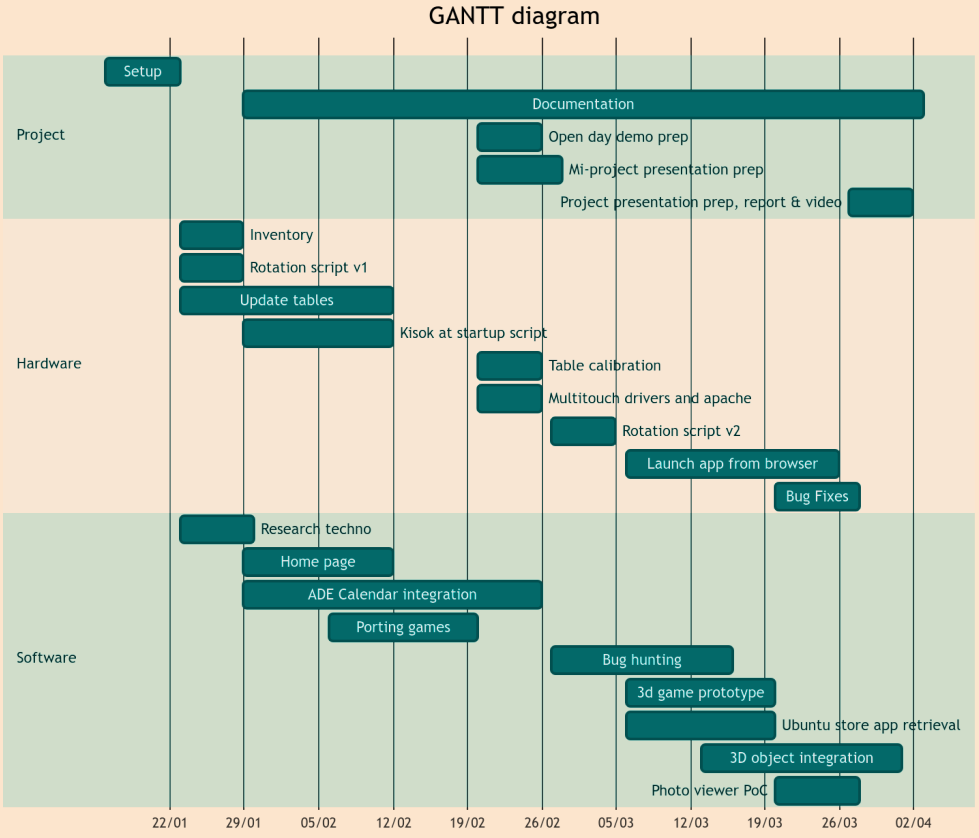
In order to manage the organization of our group and not to deviate from our specifications, we organized our group in two teams :

- One dedicated to the hardware problems, located on the tactile table, which was in charge first to make an inventory, then to evaluate and implement the available solutions to answer Mr.Donsez's expectations (for example, a method to orientate the screen to manage the multitouch, choose an OS, etc.)

- The other one, dedicated to the software of the table, had to implement the software solutions, to make the choices which best answers the specifications, by putting forward the points defined in the specifications by level of importance.

Splitting into two teams to work efficiently on two targeted problems was not a reason to avoid knowledge sharing between the two teams. Indeed, with permanent contact on remote working software such as Discord, each team had the duty to report on the progress made during the week (which was also reported in the progress sheet) and to indicate to the other members of the group when a feature had been fully implemented, while explaining its overall functioning.
To avoid getting lost in the progress of the project, a weekly meeting was held with our project manager in charge, Mr. Donsez, to report on our progress, the problems encountered and especially the next objectives to be achieved, it was inspired by the scrum method studied.

A change of organization was made throughout our project, in order to adapt to the needs, it was to switch Noah Bompard on the software team to make a prototype of the photo viewer, the 3d integration and the kiosk mode at start , the workload of the hardware team having been reduced to one person.



Our work

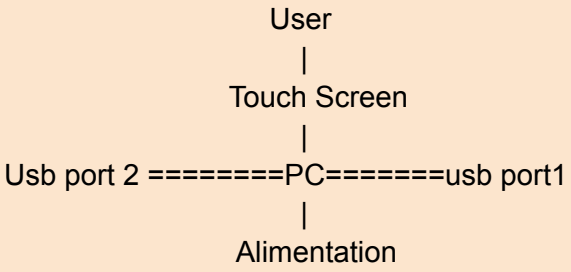
We will explain the work that has been done by our team in two parts, the first will explain the choices made by our team for each specification, then the other one will explain the problems encountered and the way to fix them (if there is one).

1. Choices

TEAM HARDWARE :

The most important choices for the hardware team were to choose the OS, and study the feasibility of each solution for the ”tactile” side of the table.

After dismantling the table, the team figured out the internal functioning of the table, and decided on a method to use multitouch afterward.
Architecture :



For security reasons, we set a password that is only typeable by a physical keyboard (it activates the tactile keyboard only after the session has started (1234 for project purposes..)

Even if there were other solutions like Ubuntu touch, especially designed for tactile only devices, the fact that the internal core was a pc linked to a tactile screen made Ubuntu the best choice.
For reliability purposes, we installed the latest long-term support version of Ubuntu. We also chose it since we were most familiar with that linux distribution and that the project scope didn’t require us to look elsewhere.

To start in kiosk mode, we’ve set up a gnome extension that allows the admin to write scripts or instructions to execute when either loading the machine, or opening the user session.

As for the reorientation of the tactile table, we built a small shell script. This script uses functionalities from RandR, the official configuration tool for X11’s windowing system that allows users to rotate and change the size of their display. At first it was used in addition to a gnome shell extension javascript powered button that sat on the right side of the Ubuntu top bar. However, since we needed to launch the tablet in kiosk mode, we didn’t have access to it and needed to find another way to launch the script.

For that, we used TouchéGG, an application that we installed in the hope of fixing our multitouch troubles. But it also provides other functionalities such as linking a touch gesture with launching a command. With that, we were able to launch the rotation script using a

three-finger tap on the tactile table. The good thing is that it works anywhere you are on the OS.

Finally, to prove that we could launch and port any application to the tables, we needed to find a way to link the apps installed locally on the touch table with the homepage that we built. In order to do that, we relied on custom URI schemes, which is the same method Zoom uses to provide a link with which you can launch their app. To build your own custom scheme, you need to provide a web handler as a MIME type in the .desktop file of the application. Once you've done that, update your desktop environment to make your application the default one for the custom URI and do the same for the cached MIME types database of your computer.

TEAM SOFTWARE :

For the software team,the choices were how to regroup all the features in order to make everything easily accessible .We choose to have as an interface ,a website that acts as an home page so it could redirect toward others pages:
We have decided to create it in php, html and css and we used Apache to provide us with a local server. Remembering the inventory where we couldn't access the project of the previous group because the server had been shut down, we felt that having everything being accessible locally was the right move. In addition, this home page will display the time and the weather to quickly have the date or the time.
To make interacting with the touch table easier, we have decided to implement a slider on the home page where when a user presses a finger on the screen and moves it to one side, the bar that displays the different games, applications and sites scroll to the side that the finger moved and you can access all applications.

From the homepage, an app that we developed is accessible. It's a Next js web application that was designed for students and teachers to access the school calendar, and menus from CROUS. The app is responsive so it works well on pc and on the tablet. Thanks to Next.js we developed a single page app, it is way easier to communicate with a server and display the data properly this way, and the app doesn't need to reload while navigating through its pages. For the backend we developed a little api with python flask, so we can access from the front-end to the menu data easily. To get the menu from CROUS we used beautifulsoup, so we can do scraping from the CROUS website to list the menu. The user interface for the app is clean and simple, making it easy for students and teachers to navigate and find the information they need. The home page provides quick links to the calendar and the menu.

About the 3d integration, we thought that a simple 3d texture could be designed and then loaded in a scene using tree.js to demonstrate that it's working as intended.

For the proof of concept,we choose to put a lot of different games so that the user has a bigger range to choose from and all the moves available with multi-touch are presented and also that any application can be imported and used on the table like some will be recovered from the internet (flash game), others will be recovered from github ,others will be from local applications and we will also code an application.

As for the applications we ported, we tried to include a varied selection of technologies : some are flash/javascript games we gathered, others are locally installed applications launched using a custom URL built by the hardware team. Finally, we also programmed our own application to add functionality to the tactile table such as a photo viewer.

For the security of the table and the files,we choose to activate the kiosk mode at the start of the table and it will be activated in the home page website so that it's impossible to go anywhere else in the table.

2. Issues

Material problems

The hardware problems team started by making an inventory of the equipment and found that part of a table had been damaged. They then dismantled and repaired the connectors and the tables in general. The group also worked on setting up a method to make the table screen rotatable and handle multitouch.

They tested with different Linux distributions and eventually upgraded the machines to Ubuntu 22.04, ensuring compatibility with the touch and multitouch features. Team members also worked on touchscreen calibration, installation of specific drivers and multitouch repair, which was the most difficult problem to solve during the whole project.

They also encountered an issue with the custom URI scheme in that snap packages are locked and don't provide any way to modify their .desktop files. Since we wanted to prove that we could have access to any application from our homepage, we needed to find a way to be able to launch snap packages. Fortunately, the solution was quite easy : we just needed to override the original .desktop file of the snap by copying it and its content to another folder and adding the web handler there. Once that was done, we could follow the other steps as normal. Also, at the current time, we haven't found any way to automate this process, meaning that you need to do the same setup for every application that you wish to add, which is quite cumbersome.

Software issues

The software issues team worked on the implementation of software solutions to best meet the specifications. In particular, they implemented an ADE calendar with FullCalendar to display the timetables of the different courses. ADE isn't accessible from outside, and we didn't have any api key to communicate with it. So we just downloaded all the data, it was ics files, and with a node package, we converted them to json data. After that it was easy to load them inside next js and display them.
They also developed a homepage with different functionalities, such as the weather, the clock, an RSS feed and links to the Polytech website.

The group also researched games and applications to integrate into the homepage and worked on opening local applications from a browser in kiosk mode. The team members also studied the integration of 3D objects on the web and conducted tests on different 3D integration solutions available. they concluded that the hardware of the tactile table did not

meet the requirements to manage smoothly a 3d environment (also tried with a unity 3d app, that made a vulkan error not resolved)

An issue that we met with the CROUS menu, is that there isn't any server or data files which contains the menu. So we have to find a solution to access the data, that is the reason why we used scraping. But now we have some problems communicating with the api and the front-end. We have to find a solution in the future.

Progress and challenges

Throughout the project, the team faced several challenges, such as fixing the multitouch, which initially only worked via the driver test application and not in any other test, and debugging various issues with the tablet.

The team managed to fix the sliding behavior on Firefox, implement a prototype image viewer and test various 3D integration solutions. They also documented all the installation, update and repair processes to facilitate the monitoring and maintenance of the project.

Potential Improvements

Though we ticked off most if not all of the requirements laid out at the beginning with the time we had, the project still has room for improvement, for example even if the table's mainly secured for common use, someone with a keyboard can easily get out of the kiosk mode due to firefox's kiosk mode design. That issue could be removed with chromium or with other kiosk compatible internet browsers.

Here is a resume of all the improvements the team found, or any idea the team proposed to Mr.Donsez without being able to finish the implementations :

- We can try to switch to another OS that handles multitouch better. (but we didn't have the time to redo the whole installation)
- The team also thought of designing a multiplayer game, with communication between the different tables, but decided against including it as it was not as important
- We wanted to do a Plato integration (Plato is a mobile application that features a list of playable multiplayer games using only touchscreen)
- Moving the rotation to a finger gesture was a good idea, but for someone who discovers the table, it isn't explained that you can rotate it by tapping the screen with three fingers. We didn't have the time to integrate a small javascript button based on the gnome shell extension code to all the project pages which would have allowed the user to have another way to rotate the screen.
- We can try to move the json files from the front-end, to the api, and access them by doing api calls
- Correct the communication problems between the front-end and the back-end

Conclusion

At the end of our project, our tactile table is in a fairly usable shape with access to a selection of applications. We mainly focused on getting up and running one table, but we think that with the documentation we provided, someone with no prior knowledge could be able to set up a tactile table as we have done. As for the applications themselves, we at least provided proof of concepts for different implementations, so we think it is fairly easy to expand the scope of the project with the basis that we laid out.

In conclusion, this project allowed us to expand our skill set and apply the knowledge we learned during courses. On the hardware side, we got a better idea of the different ways to troubleshoot and set up an OS for a specific objective. On the software side, we have learned how to have an interface on a table that is practicable to the users and has all the specifications that we choose. Finally, on the team side, we also built up valuable experiences in communicating efficiently, managing the time that was given to us and helping each other out when in need.