

UI COUSEWORK 2 REPORT (GROUP 17)

{sc21anbm, sc21ewkc, sc20yx2, sc21ezqk, sc21wma}@leeds.ac.uk

UNIVERSITY OF LEEDS

In this report, we will discuss the iteration process and design decisions leading up to our final video player. This will involve three iterations, each having distinct goals and adding specific functionality to the product. We will explain the steps involved in developing and finalizing the prototype, starting with basic functions, improving the user experience, and ending with editing capabilities.

Our group decided to implement Agile methodology to facilitate each iteration because it encourages user feedback, and we were continuously improving our code and design. This was done by doing Scrum sprint cycles of one week's length each. We assigned one of our group members to be the Scrum master, who ensured we were on the right track and our targets were met. Kanban boards were used in conjunction with our sprints as can be seen below. This made it easier to organize the workload and for everyone to have some idea of the project's current progress.

Disclaimer: All surveys carried out during evaluation comply to university regulations by ensuring participants were informed of how their data would be used, stored, and processed, as well as anonymising the responses.

Iteration 1

Our first iteration consisted of analysing the initial Tomeo code provided and building upon it. The goal of this cycle was to add basic functionalities of a video player which were lacking in the initial code. This helped create a base layer for our video player for which could easily be built upon. Below is the Kanban board we used in this sprint.

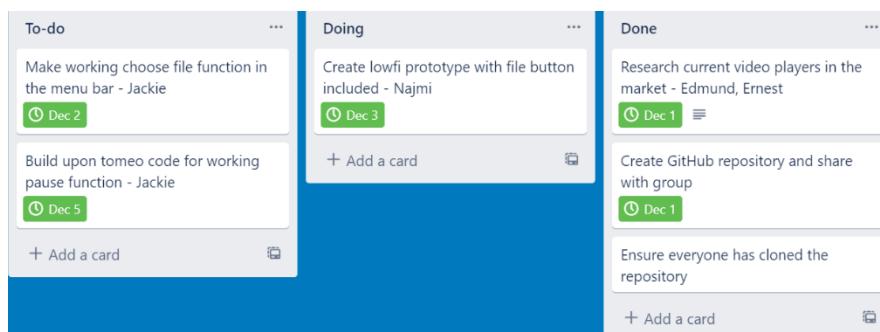


Figure 1 Kanban Board for Iteration 1

This stage involved us discussing use case scenarios for regular video players and performing heuristic evaluation on the Tomeo code. Usability problems with the UI included having to type the video file path in the command line, the resolution of the main video playing being too small, not being able to pause and play videos, not having a slider to manually start playing from a point in the video. We also performed market research on existing video players – like QuickTime Player – to gain UI and functionality inspiration as well as to identify other user needs and preferences.

Our evaluation showed that key requirements in a video player include: a button to choose the video file to be played and the ability to pause and play. We therefore added these features to a few sketching prototypes we developed. The use of sketching in the prototype phase meant we could quickly and easily create rough drafts of our designs and allowed us to make changes and improvements in a short amount of time. This helped us to move quickly through the sprint cycle and keep up with the fast-paced nature of scrum development. We produce two designs for our evaluation:



Figure 2.1 Design 1

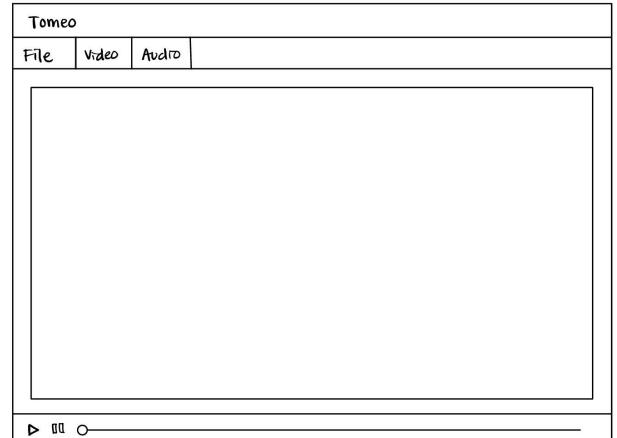


Figure 2.2 Design 2

We decided to use design 2 as the base layer for our app. We chose this design because it offers a more intuitive and user-friendly interface, with a clear and organised menu bar that includes a file tab for opening videos, a video tab for editing the video itself, and an audio tab for editing the audio track. By providing clear labels for each feature, this design helps ensure that our app's interface is easy to understand and navigate, which is crucial for providing a positive user experience. Additionally, the menu bar helps keep the interface organised and uncluttered, further enhancing its overall usability.

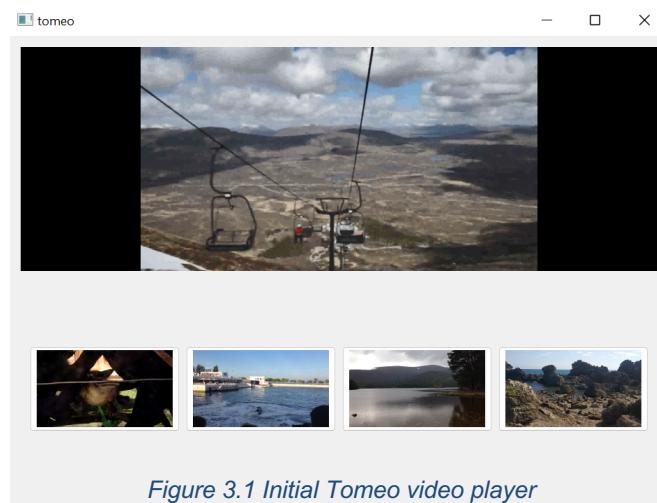


Figure 3.1 Initial Tomeo video player

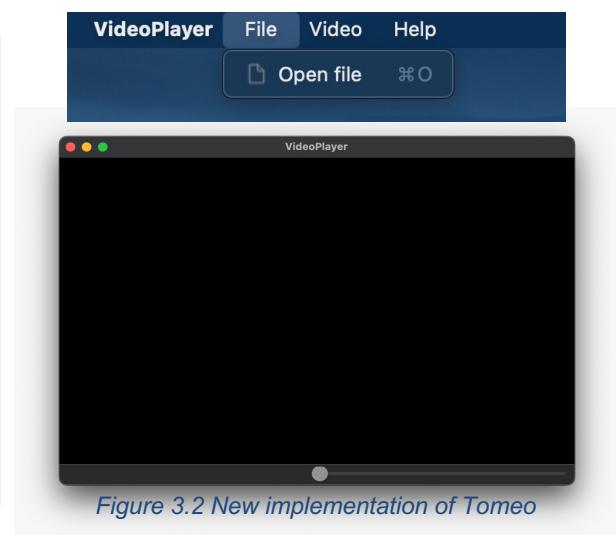


Figure 3.2 New implementation of Tomeo

As can be seen from the diagrams above, where the initial Tomeo interface had the top half of the screen displaying the video, and the bottom half displaying other videos, our improved UI allows the video to be played full screen with a slider at the bottom. We also added the file button in the menu bar making it more convenient for the user to choose the file to play, as the initial Tomeo prototype required the user to type in the file path in the command line. By using slots in Qt, we also added the ability to pause and play the video when the spacebar is toggled.

Iteration 2

In our second iteration, our main goal was to improve the user experience by providing a simple and intuitive design for our video player. To achieve this, we added several new features and functionalities to the player. These new features not only enhanced the overall design of the player, but also made it more user-friendly and enjoyable to use. This was our

main priority because we wanted to create a well-designed video player for our users. We also wanted to make sure our app is responsive.

Below is the Kanban board we used for this iteration:

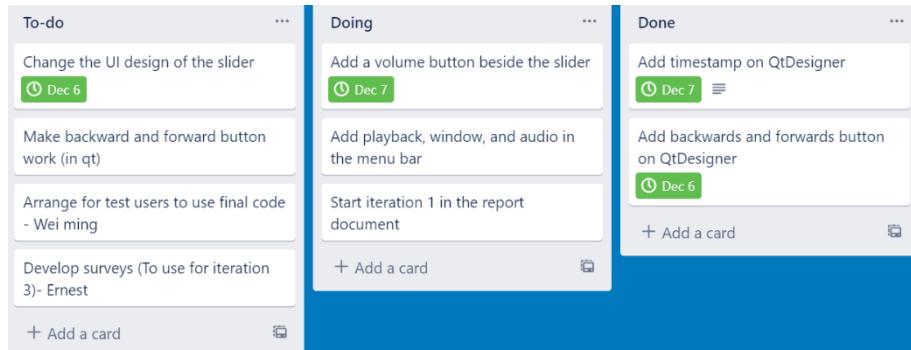


Figure 4 Kanban Board for Iteration 2

We evaluated QuickTime Player to find ways to make our product different. We also asked users how complex they wanted the video player to be. We used a questionnaire because it allowed us to ask specific questions. From the results, we decided to make two prototype designs that met our users' expectations. We decided to incorporate sketching into our prototype as we can quickly and easily explore different design ideas. In this iteration, we produced two designs using this method. This allowed us to quickly compare and evaluate different approaches and decide which direction to pursue for our design much quicker.

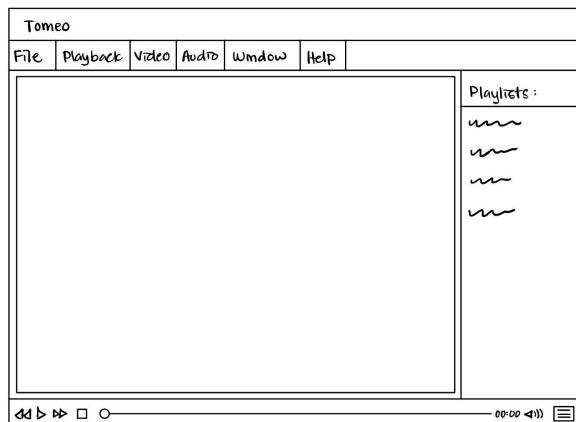


Figure 5.1 Design 1

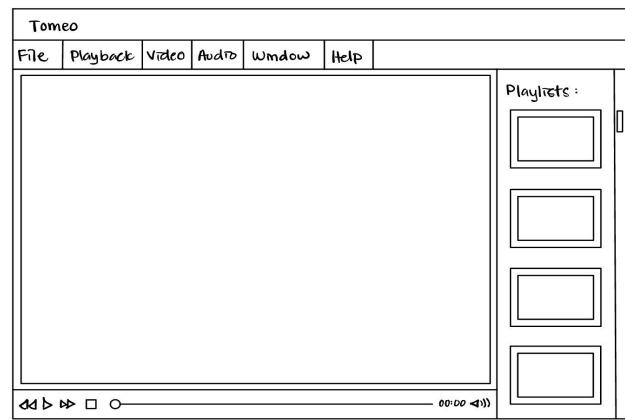


Figure 5.2 Design 2

We decided to proceed with the first design. This is because one key aspect of usability of our video player is that the user interface is simple and minimal. The playlist tab of the chosen design shows the title of each video alphabetically. On the other hand, the alternative design previews all the media on the tab. This will make the appearance of our app cluttered and unorganised. This will likely be overwhelming and difficult for users to navigate and will likely result in a poor user experience. A simple design improves usability as it is easier to understand, more user-friendly and accessible. Other than that, a simple design provides better aesthetics. Our focus is to provide a clean and modern look to our app. This would make it more visually appealing and create a positive first impression of our users.

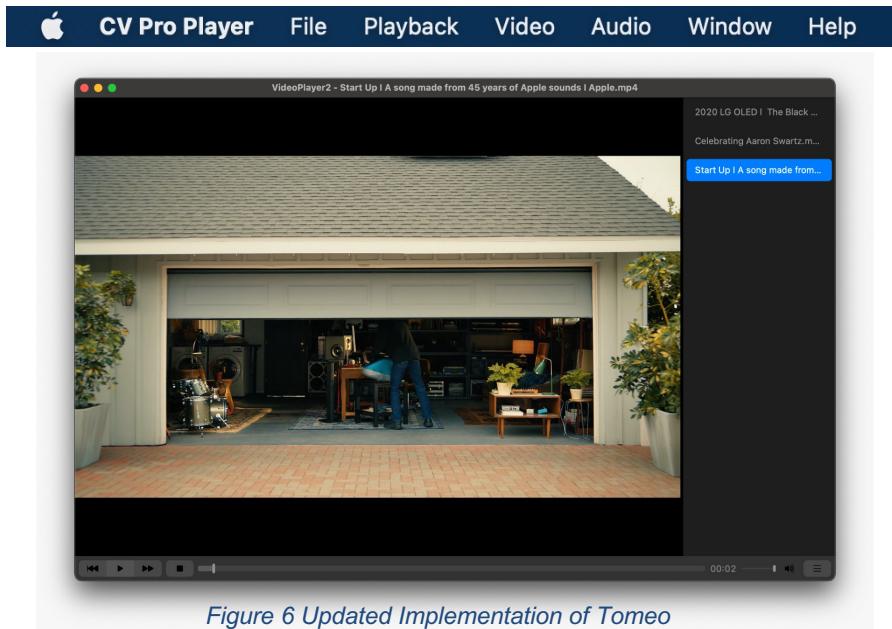


Figure 6 Updated Implementation of Tomeo

The diagrams above show the updated design of our video player, which includes several new features and improvements compared to the initial design. One key change was the inclusion of a playlist tab, which allows users to easily access and manage their videos. This simple and clean design was a significant improvement over the initial Tomeo code provided, which was more cluttered and difficult to use. In addition to the playlist tab, we have also added several new buttons to the player, including forward and backward buttons for easy navigation, a stop button to stop the video, and a playlist button that allows users to access their videos. These new features enhance the functionality and usability of the player and make it more user-friendly. Overall, our updated design provides a simple and intuitive interface that improves the user experience and makes it easier for users to access and manage their videos.

Iteration 3

In our third iteration, our main goal was to implement editing capabilities to the app. This followed from a survey we sent out asking how likely the user would edit their videos if they were to record outdoors, as well as what features they were most likely to use. We decided a questionnaire would be fitting because we were going to ask multiple choice close ended questions. The results were as follows:



Figure 7 Results of Survey

From the results, we established that our users were more likely to edit their videos and the editing requirements in our video player would include a rotating feature, trimming feature, remove video, remove audio as well as colour grading.

As we focused on usability and simplicity, we want to create a user-friendly interface for our editor. Our aim is to make it as easy as possible for users to access the tools and features

they need to edit their videos. We designed the interface with clear, recognizable icons and intuitive menu structures, so that users can quickly find what they need without getting lost or confused. We also designed our app to be simple and intuitive, with a focus on the core functions that users need to perform. This will reduce the learning curve for new users and make it more accessible for people with different levels of technical expertise. By keeping the design streamlined and focused, we can help users focus on the creative aspects of video editing, rather than getting bogged down by complicated or unnecessary features. By focusing on both functionality and simplicity, we want to provide a high-quality user experience that is enjoyable and visually appealing for all users.

Below are two sketches of the video player with the editing requirements mentioned:

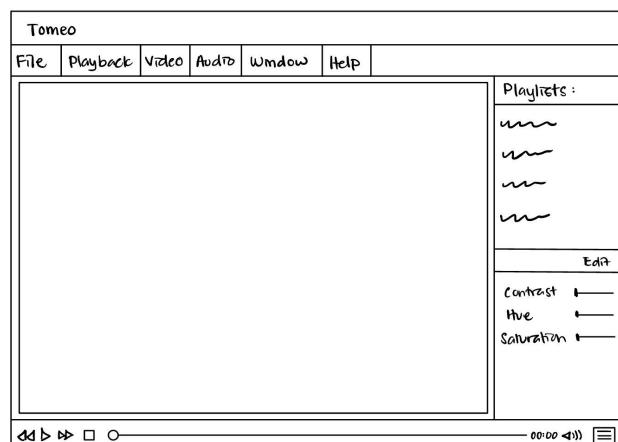


Figure 8.1 Design 1

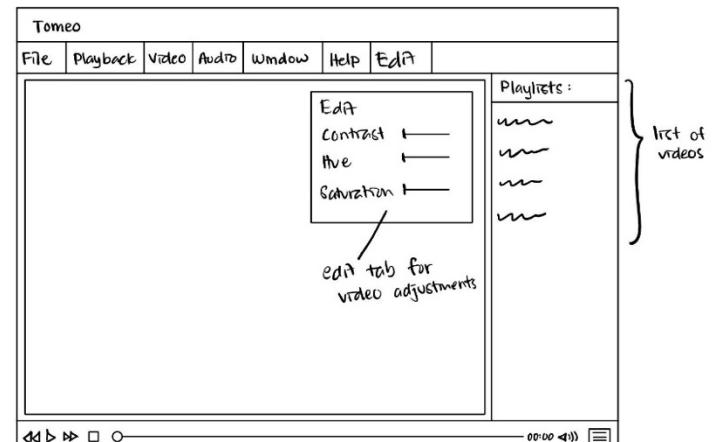


Figure 8.2 Design 2

From the prototypes, we sent out a questionnaire asking which editor design was preferred, with the results showing 62% being in favour with the first design. We therefore decided to follow through with the first design.

Coding our final design, our group decided to split the tasks by each of us doing an editing feature. Utilising GitHub's branch and merge features, we were able to run iterations in parallel increasing our efficiency, as well as keep backups and track the changes made in the code. Below is the kanban board we used for this iteration.

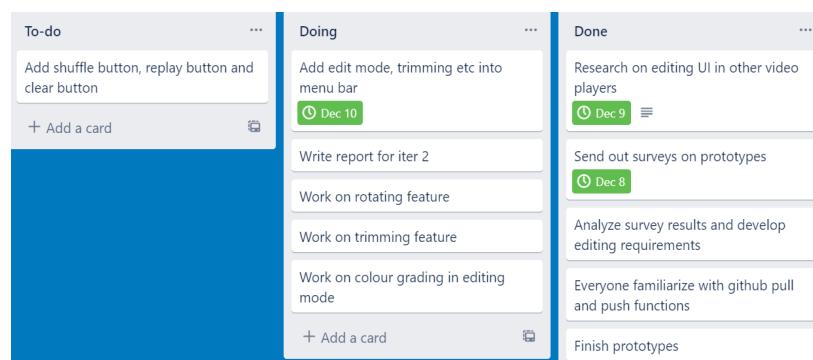


Figure 9 Kanban Board for Iteration 3

Upon further research, we found out that trimming and rotating the video file could not be done without external libraries like OpenCV or FFmpeg, which would require downloading the sources. As a team, we decided it would be too much of a risk, given we could not guarantee these libraries would work smoothly with our code on the test machines. As such, we focused on implementing just the UI of the trim and the colour grading features. We also

made shuffling, clearing, and replaying possible by toggling buttons at the top of the playlist section. This gives the user more control over their watching experience and adds a hint of professionalism to our video player's UI.

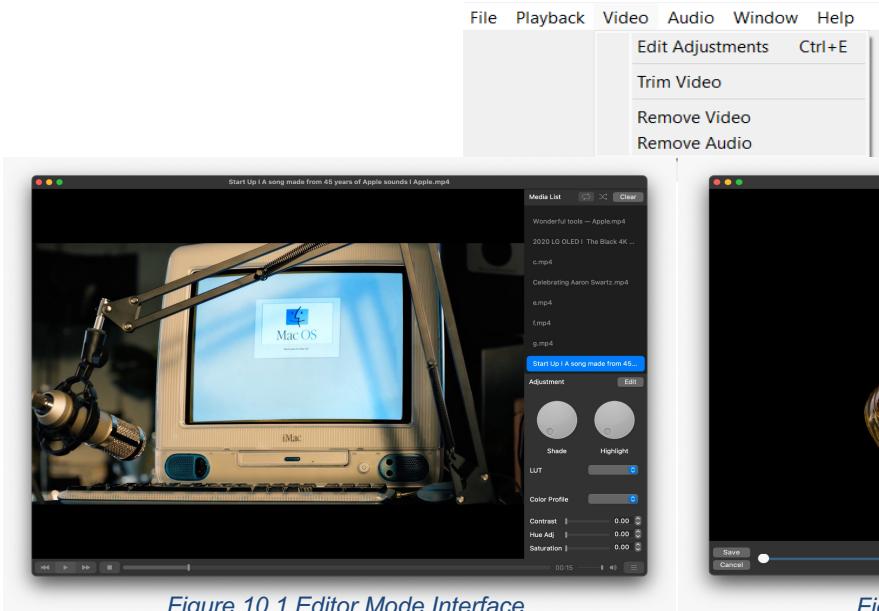


Figure 10.1 Editor Mode Interface

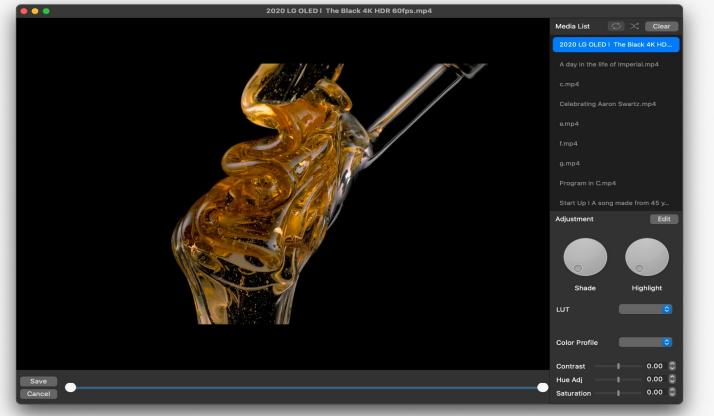


Figure 10.2 Trim Function Interface

The diagram above shows our final implementation of our video editor. It shows the menu bar with the edit buttons, the UI of our edit mode and our trim feature in edit mode. The interface has clear and intuitive "Adjustment" tabs that allow users to edit their videos using features like contrast, hue, saturation, and LUT adjustments. The interface also has a trimming function and save and cancel buttons. The "Video" tab has features for editing videos. The design provides users with a clear and intuitive interface that makes it easy to edit videos.

Midway through the iteration, a re-evaluation of the scenario gave us the idea to add a "Language Settings" button, allowing the user to choose from a list of languages if English is not their preferred language, as well as a subtitles button to promote Internationalisation. This once again showcased an advantage of using an agile framework like Scrum, because it allowed us to make changes with our evaluations late in the iteration.

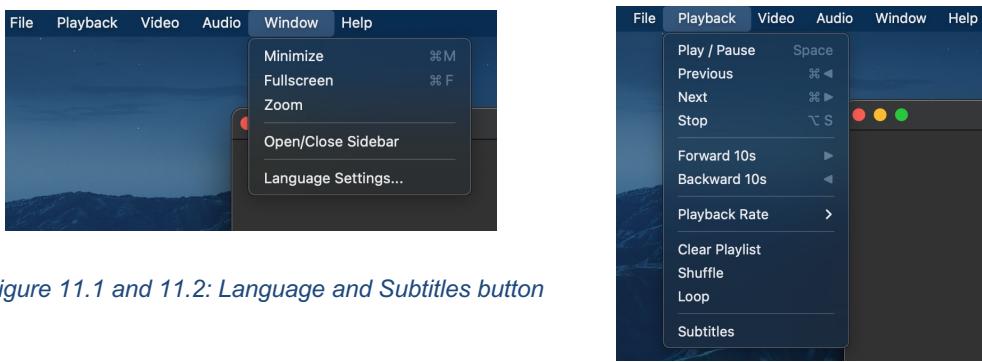


Figure 11.1 and 11.2: Language and Subtitles button

In this report we have looked at how our design follows some of Shneiderman's eight golden rules of interface design, for example ensuring that the aesthetic of our interfaces was consistent and that all widgets had uniform lengths and gaps. Our decision to implement a minimalistic interface also ensures the user will have a heuristic experience navigating and exploring the app. This was all done throughout 3 iterations, each undergoing evaluations to ensure key requirements were met before finalizing our design.