Coursework 3: The Process
Date set: 22.11.21
Date due: 19.12.21 - 12pm
Weighting: 50%

Goals for this CW:
- Apply the user interface theory that we have learnt
- Experience the interaction design lifecycle as a group
- Design, evaluate, and develop a complex GUI application in Qt

Getting started:
- Before the first meeting, everyone should get the *Tomeo* prototype running.
  - Watch the introductory [video](#).
    !! Beware: this program shows flashing videos !!
  - Download [the.zip](#) file and open the .pro project in QtCreator. You will probably have to configure Qt 5:
    - On Windows use Qt 5 to use the multimedia plugin ([video](#)).
  - Run the project. When a dialog pops up, click yes to open a link to OneDrive and download the video files. Extract the videos and set the absolute path of the *videos* directory as the first command line argument to the project.
  - Run the initial Tomeo prototype in Qt Creator, explore its (limited and buggy) functionality.
- Explore the code. In particular note the following classes, and read any Qt documentation you need to:
  - *tomeo.cpp*: contains the main method and creates a list of video files that have thumbnails.
  - *the_button.cpp*: a subclass of `QPushButton` which shows the icon, and has a signal (`jumpTo`) that is fired when someone clicks it.
  - *the_player.cpp*: a subclass `QMediaPlayer` which controls the playback of the video in the `QVideoWidget` class.
- Find a group of 5 people

Your task:
- Your group has been given the Tomeo prototype and asked to iterate it for the following opportunity:

  Our marketing department has identified an opportunity for a new product: Outdoor enthusiasts who wish to explore and organise a large personal video library. This group collects massive quantities of video from action cameras (e.g., GoPro), video drones (e.g., DJI Mavic), and mobile phones (e.g., selfie or slow-motion footage). The video is typically from a variety of locations, is of different lengths (ranging from a "moving photo" of a few seconds to 3 hours of raw skiing footage), and activities (e.g. cycling, parascending, football, or skiing).

- Arrange a first meeting time with your group members.
- In the first meeting:
  - Pick an empty group on Minerva; ensure each member has joined
  - Exchange contact details with your group members, set up a shared code repository (e.g., git, GitLab, Github….), and a shared process document (e.g. Office 365, Google docs….).
  - Arrange a weekly time to meet and update each other on your progress.
  - Explore the initial *Tomeo* prototype together.
  - Develop your requirements. For this coursework you should invent realistic requirements rather than gathering data yourself:
    - perform a *PACT analysis* to scope the possible requirements
    - write three use-case *scenarios* that you chose to develop the system for
  - Discuss alternative designs that satisfy your use-cases.

- You will then complete at least 4 development *cycles* of the Tomeo application before the due date above. For this coursework, one cycle consists of:

  i. prototyping an improvement
  ii. implementing the prototype
  iii. evaluating the prototype

- You will create a process document detailing the design process with the following numbered sections:
  1. The members of your group and their usernames.
  2. The PACT analysis and three use-case scenarios (one or two paragraphs per scenario) which refine your requirements.
  3. The platform you are targeting (desktop, mobile, web, etc…). All our *development* will take place on the Desktop with Qt and C++, but we can *design* for other hardware / software.
  4. A title and description of each cycle in the following format:
     i. prototype
        a. one paragraph describing the goal of this cycle and the reason this was selected as the highest priority.
        b. the name of the prototyping technique and any software used (technique: sketch, video, native…. software: Photoshop, Gimp, Qt Designer, etc…)
        c. one paragraph motivating the design shown in your prototype.
        d. one paragraph giving the reason for the chosen technique.
        e. evidence of the design (a photo, screenshot, a video of a paper prototype, etc...)
     ii. code
        a. a video illustrating the improvements
        b. a list describing any differences between the prototype and the implementation (because of time or technical difficulties).
     iii. evaluation

      a. the name of the evaluation technique used (heuristic evaluation, cognitive walkthrough, questionnaire, interview, etc...)
      b. one paragraph describing why this technique was chosen.
      c. one paragraph describing the outcomes of the evaluation and whether the changes this cycle were accepted or rejected moving forward.
      d. evidence of the evaluation (a table of the results, a video of the interactions, etc...)

6. A demo video. A final 45 second demo video that can be shared with the class. It should illustrate the main ways it satisfies your requirements.
7. An ethics statement, explaining how you complied with the [university regulations](#) for ethical research on humans (Lecture 12). Include the [information sheets](#) and template [consent](#) form(s) any participants have completed. Do not include the completed forms themselves.
8. A description of how to run your software.

Notes:
- Ask questions on MS Teams "Questions" channels as usual.
- The *paragraph* limit on documentation is intended to make you choose your language and conclusions carefully. A paragraph is typically 1-2 sentences giving a statement, followed by 6-8 sentences *supporting* the statement.
- The paragraph should illustrate your understanding of UI theory and terminology learnt in the module.
- Note that our emphasis here is not on writing code or the final output. We care (and award marks for) the design process you go through. It is acceptable to reject the result of a cycle if there is a well argued reason.
- Examples of a cycle might be developing…
  - … a workflow (e.g., how to find video created at a certain time and place).
  - …the visual design of the system (e.g., colours, spacing, icons).
  - …a map showing video locations.
- We are developing a prototype, so it is acceptable for certain parts to be partially developed. You should develop the all necessary components of the system to evaluate the central concept of the cycle. However peripheral functionality may be partial. For example:
  - A "load" button may just show a dialog saying "you selected a video" before showing the loaded video, rather than creating a full file chooser.
  - A "slow-motion" button may just load a pre-computed slow-motion file rather than computing the slow-motion effect itself
  - Video thumbnail images may be loaded from disk rather than computed (the given prototype does this).
- You can run different cycles in parallel (at the same time) to spread the work between group-members, as long as the cycles can be merged into a single code base if successful. For example:
  - different people could perform parallel cycles to prototype, evaluate, and code the layout of the home screen - then one is picked to be used.

- ○ different components could be built by different pairs of people - a dialog box could be a parallel cycle to the main screen visual design.
- Video evidence should be less than 45 seconds long.
- Be sure to obtain permission to include people in videos. Find alternatives if they do not want to be recorded (e.g., a transcript, voice only, etc…)
- We do not care about the quality of the videos (noise, camerawork etc… are beyond the scope of this assignment). But they should be understandable on a first viewing when the accompanying paragraph is read.
- Check all the links in your document before submission - ensure they all link the correct video and are accessible (i.e., they are visible to people using an incognito browser tab which is not logged into YouTube).
- Use a sensible and consistent code style. Ensure comments are helpful.

Marks will be awarded for:
- The quality of your PACT scoping analysis and scenarios for the given task.
- The quality of your prototypes and evaluations, as well as how well they match your chosen scenarios.
- The application of a wide range of user experience theory (as taught in the module) to your motivations/decisions.
- The variety of prototyping and evaluation techniques applied.
- Completing at least four cycles. Extension marks are available for a fifth cycle.
- Developing significant and creative improvements to the Tomeo application which match your scenarios.
- The quality of your code.
- The quality of writing in your process document:
  - ○ structure
  - ○ arguments and supporting statements
  - ○ written English
- Your ethics compliance documents

Contributions:
- All members of the group will normally be assigned the same mark, unless there is disagreement…
- … in this case, complete [this form](#) with your submission, and the varying contributions will be taken into account when grading. It must be signed (images of signatures are acceptable) by all group members.

To submit:
- You submission should be a zip file containing:
  - ○ the PDF document in the format detailed above
  - ○ your final code
  - ○ your evidence and demo videos
  - ○ (if there is disagreement: a signed and scanned contribution form)
- Name your zip file for your usernames, e.g.:

```
scstke_scs16pop_sc12ghj_sc49ksh_sc88zxy.zip
```

- A single member of your group should submit the zip file by the deadline at the top of this page via Minerva