

Mobile Application Design 2022 project

Board Aid

Team: **Petr Šilling**, xsilli01
David Holas, xholas11
František Maštera, xmaste02
Vojtěch Čoupek, xcoupe01

Project Goal

- With board games, it is quite common to find extremely specific designs for timers, dice, etc. This could lead to gameplay issues when such items get lost or forgotten.
- It is also common for board games to require several such items (especially dice) in order to play, which may be cumbersome for travel and increases the chances of some getting lost.
- We believe these issues can be (at least partially) remediated by digital widgets providing the same functionality.
- We aimed at creating a minimalist application that can assist players of board games by providing customisable presets of digitalised timers, counters, number generators (dice), etc. These presets could then be used to play board games instead of their standard counterparts.

Used Technologies

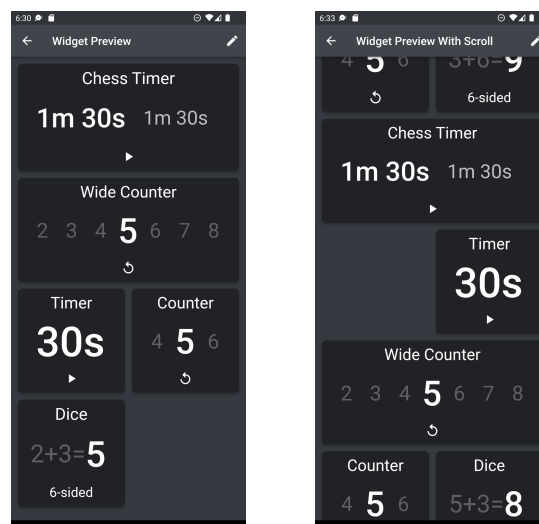
- We used Flutter to create our application (a functional Flutter environment should be the only project requirement). All source code files are written in Dart.
- For development, we used a mix of Android Studio, Visual Studio Code, and Xcode. All of those provided excellent support for Flutter development (including device emulators for testing and prototyping).
- Our mockups were created in Figma.
- To cooperate, we used Git, GitHub, Trello, and Discord.
 - GitHub repository: <https://github.com/PSilling/fit-tama-1>

Used Resources

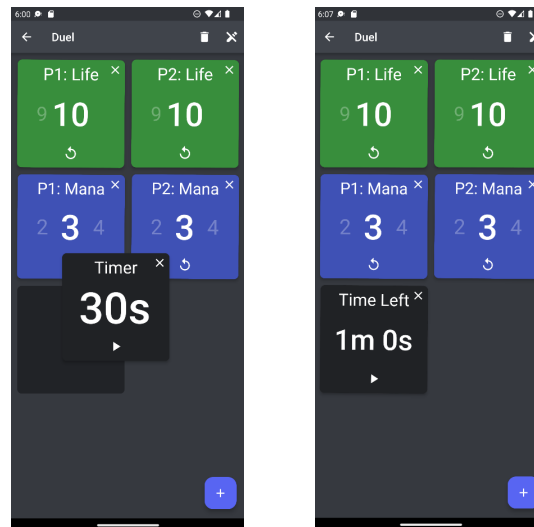
- Flutter documentation: <https://docs.flutter.dev/>
- Material Design documentation for Flutter: <https://m3.material.io/develop/flutter>
- Dashboard for preset editing: <https://pub.dev/packages/dashboard>
 - Our own fork with additional features: <https://github.com/xcoupe01/dashboard>
- Flutter Form Builder for easier form creation: https://pub.dev/packages/flutter_form_builder
- Flutter Speed Dial for dynamic wheel: https://pub.dev/packages/flutter_speed_dial

Most Important Achieved Results

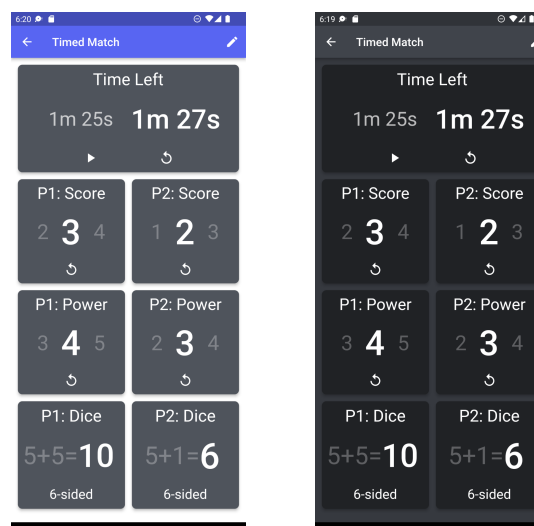
- We wanted to create a set of non-distracting, minimalist widgets that would cover as many board games as possible. In the end, we created five widgets: chess timer, classic timer, dice, small counter, and wide counter. We believe these widgets realise our goal quite well.



- While we were designing our application, we found out that creating a customisable grid of draggable widgets that supports different widget sizes is difficult. This is true even when third party packages get involved. Therefore, we extended a public Flutter package with more animations and scrolling bug fixes.



- We did not want our application to distract players from their game. Therefore, we created two unified themes—light and dark—so that every player can use the colours he or she prefers.



Controlling of Created Program

Preset List View

- The first view users see upon opening up the application.
- A list of created game preset cards (empty at the beginning, can be sorted, etc.).
- New presets are created using the floating add button.
- Using Customise in the dots icon menu on these cards opens Customise Preset View.
- When a card gets tapped, the corresponding Preset View is opened and a game can be played.

Customise Preset View

- Provides customisation options (changing the name, game title, colour, etc.) for preset cards.
- Changes are previewed and saved automatically.

Preset View

- Contains a customisable grid of widgets (timers, counters, number generators, etc.) for playing board games.
- Controls for each widget are based on its type and configuration.

- Can enter editing mode using the pencil icon. In editing mode, the grid can be modified, new widgets can be added using the floating add button, and Widget Edit View can be opened by tapping existing widgets.
- Preset title can be edited when tapped.

Widget Edit View

- Provides configuration options (changing the title, control type, number ranges, etc.) for game widgets (depending on their type).
- Changes are previewed and saved automatically.

Experience with the Selected Platform

We were pleasantly surprised by the level IDE support Flutter provides, considering it is a relatively new framework. After the initial setup (which was well documented), everything worked out of the box, including mobile device emulation. Additionally, since Flutter is quite popular, community support was also outstanding: most issues we faced were promptly resolved after a quick web search. The official documentation for Flutter and its integration with Material Design was also incredibly helpful and we did not need to use any other tutorials to familiarise ourselves with the platform.

That said, we did encounter some issues as well:

- Our IDEs often failed to detect changes in the list of required Flutter packages, preventing the application from compiling (without providing any meaningful error messages).
- Mobile device emulators (especially in Android Studio) were crashing quite frequently, often freezing the whole IDE.
- We found out that properly theming an application is a rather difficult task, especially since there were many attributes responsible for theming (and not all of them were named identically).
- Not everyone liked Flutter's code style and automatic formatting since it forces very short lines, which may feel unnecessary on today's large screens and can lead to some strange formatting.
- Finally, we had some issues with dynamic types. Our IDEs often failed to provide meaningful hints or error messages when dynamic types were involved.

Distribution of the Work in the Team

- Petr: designed initial application mocks, created the Preset List View and editing previews, handled documentation, led the team.
- David: designed and implemented the dice widget and the small counter widget, handled usability testing and iOS testing.
- František: designed and implemented the timer widget, created presentation materials, added theming support.
- Vojtěch: created the Preset View, designed and implemented the chess timer widget and the wide counter widget, and was our primary Android tester.

What Was the Biggest Challenge

- Polishing the controls to be user friendly. It is not only necessary to come up with a design and implement it, but also to show it to users, realise that it is far from ideal, redesign it, reimplement it, and show it to users again.
- Theming and design as a whole, as it required modifications across the majority of the codebase. Widgets provided by Flutter also often suffered from inconsistencies—some of them used colours from fields that were meant to be replaced with different ones, etc.
- Modification of an existing package because of a bug it contained that we did not want to have in the final application. Finding the bug in third-party code was also rather difficult.

Experience Gained From the Project

First of all, we learned more about the whole process of developing a mobile application (especially since, for most of us, this was the first mobile application created)—from coming up with an idea to achieving the final product. We believe we learned how to work with Flutter well enough to design even more complex user interfaces (possibly in other declarative UI frameworks as well). Second, we tried proper user testing and are quite satisfied with the results. For example, it showed us that it really is true that if you need to explain your UI, it is probably not that good. Finally, we learned how to extend third-party packages to make them fit our needs.

Autoevaluation

Technical Design: 90%

We believe the problem was properly analysed from the very beginning. Therefore, we were able to simply apply our initial mockup designs and application structure ideas. The selected development, design, and planning tools were generally quite reliable.

Programming: 80%

Our application feels quite stable with no known major bugs. Implementation of each widget is also completely separated from other elements of the application, so additions should also not be an issue. However, we do believe it is often hard to read the code.

Usability of the Created Solution: 85%

Usability tests showed that our application is generally simple to use and has a pleasing design. We also managed to remediate all issues our testers brought to our attention. However, there might still be board games that require more widgets than our application currently provides.

Use of Resources: 90%

We believe we handled Flutter libraries quite well: we used reputable ones for often fairly common features, and extended one less robust one to suit our needs. Most of the time, we were able to get all necessary information directly from the official Flutter and Material Design documentations.

Time Management: 75%

Since we were diligently tracking our progress via Trello and cooperated well, we were able to make steady progress. However, we did end up pushing several features back by a few weeks (some even multiple times).

Team Cooperation: 95%

Communication via Discord was seamless. Everyone was very productive, kept their word, and came up with their own ideas to improve the application. Everyone also worked for about the same amount of time.

Chances of Publishing the App: 80%

The application should be ready for publishing as we feel it is quite polished and bug-free. However, to do so we first need to decide on a suitable approach for registration expense handling, etc.

Overall Impression: 85%

Overall, we believe the project took a considerable amount of time. However, we think the resulting application fulfills its intended role, and we gained valuable experience with Flutter and mobile application development. We can use this experience later to develop even more robust applications.

Five Main Questions of TAM

What drives this sector of IT?

Accessibility – most people have a phone but not everyone has a laptop or PC. Most people also carry phones with them pretty much everywhere (compared to, e.g., desktops, which are stationary). This means that the mobile user base is the largest, which in turn directs more money and more talented people to the mobile sector.

What will it be like in five years?

We will most likely see more matured foldable devices and wearables (e.g., smart watches) and more streamlined applications for them. Gesture-based UI controls might also become even more dominant. We do not think that virtual or augmented reality features will become mainstream in this time, however.

What slows it down, speeds it up?

Growing competition and computing power definitely speed it up. On the other hand, technological limitations (e.g., front camera holes/notches are still present because hiding a camera behind a screen is hard.) continue to slow it down.

What ideas are dead (though they appeared great once)?

Windows Phone is most certainly dead. Same goes for multiplayer games on a single device. VR using your phone as the display (with lenses inside cardboard goggles) also does not seem to be picking up any real use cases.

Where do new ideas come from?

From analysis of user behavior and of their usage of existing solutions (with modern technology, analysis tools have become extremely powerful). Inspiration from day-to-day life frustrations, business competition, and open source innovation may help as well. Money also plays a big factor in what ideas can actually be realised.

Recommendation for Assigning Future Projects

It would be nice to have more detailed information about the project schedule and requirements for each workshop (more ahead of time). Concrete examples of project topics which are not ideal would also help tremendously. Since topic selection is a big challenge, it might help to require teams to send their application ideas before the first workshop. Relaxing the idea uniqueness condition might help quite a bit as well since it is getting harder and harder to come up with new ideas each year. We also liked the feedback given at project workshops (generally really good points). The delivery could sometimes feel quite harsh, however.

Recommendation for Future Students

Start working on your project as soon as possible. Ask the teacher to give opinions about your ideas before presenting them (it is too late to find out that your idea is not ideal there). Keep in mind that no idea is too simple, even a basic counter application will turn into a pretty challenging project if done with fluid animations, intuitive UI, proper theming, etc.