Veera Kansikas

TURN TRACKER
spring 2021

01/2021

TABLE OF CONTENTS

TERMINOLOGY

Action	An action that a player completes in the game, i.e. drawing cards or moving a character on a board.
Co-operative	A game where all players share a goal and they either win or lose together.
Phase	In co-operative games a single game round is often divided to a few distinct phases, which have different set of actions within them.
Ready	When a player is ‘ready’, they have completed all their actions in a turn.
Round	A board game often consists of many rounds of gameplay, during which each player has a turn. Unlike a phase, every round is identical to another.
Symmetric	A phase is symmetric, if players don’t have individual turns within it, but perform a set of actions together, i.e. enemy A.I. turn in co-operative games.
Turn	A player performs all their actions within their turn.
Versus	In a versus game, players play against each other. One player wins and others lose.

The number and functions of above vary from a game to game, but generally a game could consist of 5-20 game rounds, each of which consists of 1-5 distinct phases, during each of which all players have their turn, during which they complete 1-5 specific actions that are dictated by the current phase.

Many co-operative games have a structure described above, but most of versus games don’t have phases and each round of gameplay is identical in terms of game mechanics.

1. INTRODUCTION

The purpose of the app is to keep track of game phases and player turns during a board game. The app has two main views: main view and game view, which are shown in images 3 and 4. In main view the user selects the game they are playing, number of players and various other settings. The game view is then generated based on the settings and is used during the board game session. In short, the game view informs the players what game phase is currenlty in progress, whose turn it is and who players are ready.

The app runs on Android and iOS (at least in theory). By default, the app is running in one phone that is located somewhere where all players can reach it but running the app with multiple phones is an optional feature to be developed.

1.1 Background and Reasoning

One of my hobbies is playing board games, and my favorite ones are co-operative tactics games such as Spirit Island and Direwild. These games often have complex turns that require many actions from each player and one round of the game consists of many phases that all have different actions. Keeping track of current phase and completed actions can often be difficult, while the primary focus is on figuring out what to do in the game itself.

The normal way of keeping track of turns is that players announce “ready”, “done” or something similar to let the next person in order know that it’s their turn. In practice though, often players forget to announce that they are ready, or pay attention to who has a turn and if theirs is coming up soon. Because of my personality, I often end up being the “turn tracker” myself, announcing aloud what phase we are currently in, whose turn it is, and keeping track of all players if they seem ready with their turns. It adds a lot of mental load to the game for me and can also be annoying both to me and other players.

This creates a need for an external tool, which would do most of the work for me. I don’t have to be as on top of all the players actions, if there is a tool that tracks that and reminds me and everyone else what phase and whose turn it is.

1.2 Users and Use Context

Primary user group for the app are people who play board games regurlarly. The idea for the project began from my personal need, but I believe there are also other people dealing with the same problem to some extent.

The type and complexity of played board games isn’t restricted to just some category. For me the use case is often with just two players and a more complex game, but I could see the benefits when playing a simpler game with more players, for example Carcassonne or Ticket to Ride.

However, the need probably isn’t there for a simple 2-player game, so either more players or a more complex game is needed for the app to be useful.

Use context

Use context for the app is while playing a board game, which means that all players are around the same table, which is often a large dinner table. Depending on the size of the table and how much space the game itself takes, all players may or may not be able to reach a single phone on the table. This creates a need to the possibility of running the app on multiple phones that the players can position near themselves.

Kuva 1. *Sometimes available space is limited.*

The tempo of the game also varies with different games. In some games a single turn can be done in less than a minute, while in others it can take 15 minutes or more. This creates some flexibility needs for the app. Changing turns should be fast and noticeable enough for fast paced games, but also the app should also stay on when left alone for long periods of time, preferrably without draining the phones battery.

Kuva 2. *The app prototype used in context, a single player game of Spirit Island.*

Board games can also be played in different environments and lightning can vary. The app should be pleasant to use both in broad daylight where light colors both better blend in with the surroundings and are more visible (image 1), but also with dim evening lights when light dark colors are easier on the eyes (image 2).

2. REQUIREMENTS

This chapter lists all requirements for the project. Used screenshots are used to better illustrate the requirements of each view and are not final.

The requirements are divided to functional and non-functional requirements. Functional requirements define all the thing that the app should do, while non-functional requirements define how it should do them.

2.1 Functional Requirements

These requirements are divided into mandatory and optional features. Mandatory features are required for the app to fulfill tis purpose and to be a minimal viable product, whereas optional features are features that would improve the app in one way or another, but are not necessarily required and some of them can be dropped from the final product.

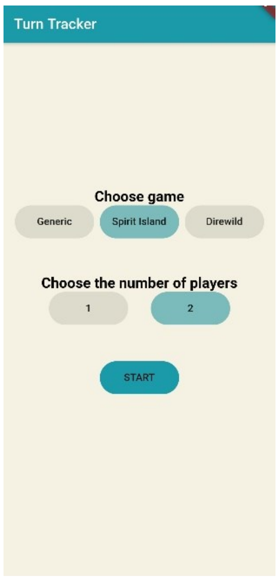
2.1.1 Mandatory features

All mandatory features of the app are:

- selecting options for the game,
- starting a new turn tracker,
- completing an action,
- being ready,
- going to next phase or turn,
- going to previous phase or turn and
- exiting the game view.

Selecting options

Main view of the app displays different options that the user can change. When the user clicks the ‘start’ button, the selected options are used to generate a game view. Prototype version of the main view is shown below in image 3.



Kuva 3. *Main view options.*

List of options in main view is:

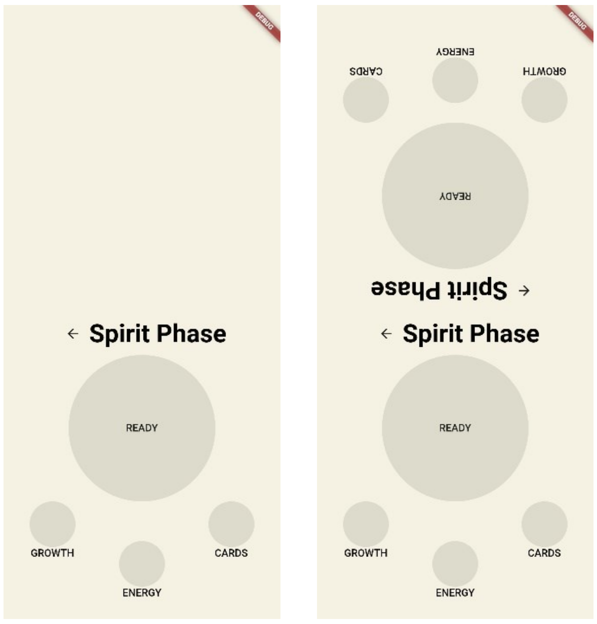
- game name / generic co-op / generic versus,
- number of players,
- sound on / off and
- dark / light theme.

Starting a new turn tracker

When the user has clicked 'start' on the main view, a game view is then generated and displayed.

The view is divided to sections according to player countm one section consists of name of the current phase (if the game has phases), one big ready button and 1-5 small action buttons. The sections are laid out as follows:

- 1 player: one section, right way up
- 2 players: two sections, one right way up and one upside down
- 3 and 4 players: three to four sections, two at 90 angle and one/two at 270 angle orientation



Kuva 4. *Game view layout with one and two players.*

Other variables are then set based on the chosen game. These define how many buttons are displayed, how they behave and what text is shown at any time. The variables are:

- Number and names of phases
- If a phase is symmetric, so that the players complete it together. i.e. enemy A.I. phase
- Number and names of actions in each phase

Completing an action

When a player has completed a single action that has a toggle button in the game view, they can click the button to mark the action as completed. The button then changes color. Clicking a button that is marked as completed, resets its state to uncompleted.

Marking all actions as completed before clicking the ready button is not required, and it’s purpose is purely to remind players of necessary actions during a phase, to remind what actions the player has completed and whether they have completed all actions in the phase and could be ready.

Being ready

When a player has completed all their actions in a turn (whether or not they have marked them as completed with the app), they can click the big ‘ready’ button. The button then makes a short clicking sound, changes color and triggers some additional visual changes within the player section to indicate to others that the player is ready. Clicking the ready button again resets its state to not ready.

Going to next phase or turn

In a co-operative game, when all players are ready, the last player to click the ready button triggers the game view to change to next phase. This resets all ready and action buttons to unselected states and goes to the next phase, which means changing what action buttons are available, their titles and the title of the phase.

In a versus game, only the player whose turn it is has their buttons available. When they then click the ready button, all their buttons reset to unselected and the turn goes to the next player in order and their buttons become available.

Going back to previous phase or turn

The game view has a back button, that the players can press to go back to the previous turn or phase in case they triggered a ready button by accident.

Clicking the button resets all ready and action buttons and in a co-operative game the view goes back to the previous phase, whereas in a versus game, the turn goes to the previous player in order.

Exiting the game view

Exiting the game view is done with the back button of the phone. This can be in Android 10 be done by a gesture, or in older phones it can be a physical button. The app then navigates back to the main view.

2.1.2 Additional features

Additional features for the app are:

- user preferences,
- dark and light themes,
- storing game information in a database,
- creating a local custom game,
- connection to nearby phones, and
- timed turns

These features are not strictly necessary for the app to be functional and some of them may be dropped during development. They however enhance the usefulness and flexibility of the app.

User preferences

The user can access their user preference setting through the main view. The preferences are stored locally permanently and include:

- favorite games, which are then displayed more prominently in main view,
- most used player count, which is then selected by default in main view,
- sound on/off, whether the sound on button in main view is toggled on or off by default and
- dark/light theme (if implemented), which changes the theme of the app

Dark and light themes

The user can change the app's theme between a light and a dark variant. The change is stored permanently in user preferences.

Storing game information in a database

Information about different games is stored in a database, from which it is fetched based on the selected games id when starting a new turn tracker. The information in the database can't be altered by users.

Alternatively, the information could be stored locally in a json file that is read when starting a turn tracker.

Creating a local custom game

The user can create a local custom game with their own values for all variables used in game view generation. The settings can be stored locally.

Connection to nearby phones

The user can from the main view navigate to a connection view, from which they can create a connection to 1-3 phones nearby i.e. via bluetooth. The app then uses all connected phones to display the game view.

This increases the number of possible players upwards from 4 and all players can have their turn tracker near them, instead of having to reach to the center of the table.

Timed turns

Some board games use a sand timer or electrical one to give player turns a time limit. This could also be included as an option to the turn tracker.

2.2 Non-Functional Requirements

The purpose of non-functional requirements is to define how various aspects of the app should be implemented and they serve as principles that quide the design and implementation process.

The non-functional requirements are:

- starting a new turn tracker should be fast,
- turn tracking should adapt to the game and players,
- usage should be simple during a game,
- turn tracking should always be reliable,
- display shouldn’t turn off during a game and
- battery usage should be low.

Starting a new turn tracker should be fast

Setting up some complex board games can in itself be time consuming, some heavy games taking up almost an hour to setup and the turn tracker should not add to this time considerably. The same is true for very easy to setup games, if starting the game without the turn tracker takes a couple of minutes and a single game takes only 15 minutes, getting the turn tracker running should be equally fast.

Therefore, the basic setup of the app should only take a few seconds, and even a more complex use case shouldn’t take more than a few minutes.

Turn tracking should adapt to the game and players

Different games can vary a lot in how they use the consepts of game rounds, phases, player turns and actions. For example, some games have a limited turn number of game rounds and some have simultaneous player turns. This means that the logic behind turn tracking should be extremely flexible to accommodate all different game systems.

Different players also have very different habits. For example, some players always complete their actions in the intended order, while others vary the order constantly. Some players also naturally pay much more attention to tracking phases and turns, while others are more passive and only “wake up” when someone informs them that it’s their turn. The functionality and design of turn tracker should try to accommodate different players as well as possible.

Usage should be simple during a game

Some board games can be very complicated and require a lot of thinking and/or physical actions from the player. The usage of the turn tracker should not complicate playing that much that keeping track of the game would be easier without it. Adding more complexity would also defeat the purpose of lessening the mental load of players.

Turn tracking should be reliable

The players should be able to rely on the turn tracker to keep track of turns and phases. This means that any errors that mess up the turn order or skip a phase could result to players skipping player turns or game phases, which could have large impacts on the game.

If the turn tracker can't be trusted to reliably keep track of turns and players still have to double check that everything is going as it should, the app doesn't fulfill its purpose.

Display shouldn't turn off during a game

In some games, there can be long pauses between any user input to the turn tracker as players are thinking their turns and planning tactics. The app should remain open during all this time, so that the player in turn doesn't have to wake the phone, which could also require the owner of the phone to unlock the screen lock.

Having to wake and unlock the phone multiple times during a game would distract the players and easily become annoying. This also ties to the requirement that the app should be as unintrusive and simple to use during a game as possible.

Battery usage should be low

A single session of playing board games can often be 4-6 hours with more complex games, and the app should be on the whole time without turning off the display. Most of the battery usage during this time comes from simply keeping the display on, but the app should do its best to minimize its impact.

Possible solutions include dimming the display and giving the user an option to not keep the display on after a period of inactivity.

3. DESIGN PROCESS

This chapter describes the design process of the application and reasonings behind the design choices made.

3.1 Design principles

To guide the design process, here are the most important principles:

- the app should be simple to use and not add complexity to a board game session,
- the visual design should be minimalistic, but blend with the table or the game, and
- the design should have flexibility with different games and player counts.

These principles have mostly risen from my own experiences with board gaming and how I feel an external tool should fit the context. As many board games are very thematic and role playing can also be a big part of the experience, I would want the app to be as unnoticeable as possible. Many players can also feel that mobile phones don’t belong in the table, so my aim would be to design the app in a way that makes it blend in and feel less intrusive.

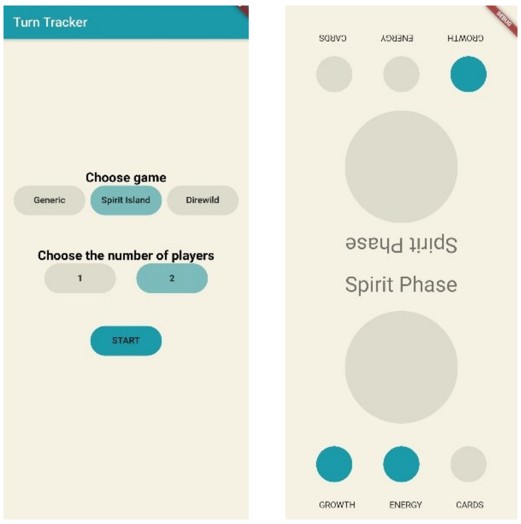
3.2 User tests

First tests were just of me and various friends of mine playing the board games we usually do, but this time with the app. The tests are described in more detail in chapter 4, but in short, they concluded that the basic concept of the app was working as intended and development could be continued.

In the next steps more user testers should be gathered.

3.3 Implementation of the Design

The first look of the design was made just by coding the app in Flutter and defining some simple rules for page layouts to get a very simple, but functional look for the app prototype. Pictured below in image 5.



Kuva 5. *First look of the app.*

4. IMPLEMENTATION

This chapter describes how the implementation of the app was done.

4.1 Tools

During the development process the following tools were used:

- UI design: Adobe XD,
- illustrations: Adobe Illustrator,
- IDE: Android Studio,
- framework: Flutter,
- coding language: Dart,
- testing device: Android 10 phone.

I have used most of the listed tools in some previous projects, so no setup was needed, and I knew that the tools would fit this project.

4.2 Implementation Phases

The implementation has been divided to a few different phases, which are documented here, along with all possible problems that arose.

Phase 1: Prototype

In the first phase of the development, my first goal was to figure out the basic consept of the app and code a prototype of it. With the prototype fuctional, the basic consept of the app could then be tested in the correct use context and problems could be identified in extremely early phase. Below is a picture taken of the first testing session.

Kuva 6. *One of the first tests of the app prototype.*

The first tests verified that the basic consept of the app was functional and extremely helpful while playing. Before we have been constantly asking each other “Are you ready?” “Are we done?”, “What were we doing again?” and “What phase were we in?”, but during these tests those were almost completely eliminated, 1-3 times a game one of us had to remind the other to click the ready button. The initial impression was then positive, we felt that the app relieved a lot of brain power used to track what game phase it is and what actions have been done, which made the otherwise quite difficult game feel less tiring.

After a couple of test games, the prototype already identified some problems:

- We sometimes triggered a ready button by accident, so I added a “back” button to each phase, which goes back to previous phase.
- Almost all of the phases included 2-3 action buttons, with which a player could track what actions they have completed. In many phases completing all listed actions ment that the player would be ready, but in one phase only some of the actions had a button. This often resulted in marking all shown actions as done and then immetiatedly clicking the ready button, allthough not all actual actions were completed. This was fixed by adding all actions as buttons.

All in all, the prototype tests showed that the application would be useful and that its core functionality is sound.

5. FINAL PRODUCT

- Description of the final product.
- Is the product complete and in use.

5.1 Documentation

- What documentation exists and where can it be found.
    - This document and a Dartdoc html documentation

5.2 User Feedback

- Hopefully from users in BGG forum
- Reviews from app stores?

5.3 Release

- Available as an open source project on GitHub
- Firebase?
- Google Play Store
- No Apple Store, as I don’t own a MacBook or an iPhone.

6. FUTURE WORK

- How will the app be managed after this project.
- What improvements could be made.

REFERENCES

[1]