

Practical Course: Logic App Development

Exercise sheet week 6

To be completed before 24th November 2023, 11am.

1. The solution to this week's task should be in a folder **week6** in the main branch of your repository; copy all necessary parts from last week to a new project. If your app does not yet satisfy all the mandatory properties of week 5, you also need to include the missing functionality. This week's task is to refactor your app and to add state management.
 - Your app should have (at least) one widget per mandatory item of last week's exercise sheet (i.e., dice, number of throws, switch, reset button, button for 1000 throws, and the two visualizations of the distributions). Every class which you create (including this week's classes) should have their own file.
 - Your app should run using both Chrome and at least one emulator (Android or iOS).
 - Your app should not fail (trigger exceptions) at any screen sizes.
 - Your app should have an undo functionality for the last action (i.e., it should either undo one throw of the dice, undo 1000 throws, undo switching the mode between normal distribution and equal distribution, or undo a reset). This should be possible until there are no further actions to undo in which case the functionality should be de-activated (and re-activated as soon as there are again actions to undo). There are various ways for storing the current state; the most obvious choices might be a list or a stream of **Dice** objects. The actual approach is up to you.
 - You should manage the state of your app using Riverpod:

`https://pub.dev/packages/riverpod`

Have a look at the documentation on how to use this package. The state should comprise of at least the **Dice** objects from the previous item, but it can also have other properties. Your app should not use any stateful widgets (except for the built-in stateful widgets like **MaterialApp**).
2. The following requirement is optional. You do not have to implement this feature but it certainly would add even more value to your learning experience.
 - You can add a “re-do” functionality to your app which undoes the last undo action if available.