# IN721 2019 5.1 Language Trainer

## Introduction

Research in human learning has demonstrated that memorisation is best achieved through “distributed practice” – that is short, frequent periods of study, rather than occasional, long periods of study (see e.g. Cain and Wiley, 1939; Mishra, 2002; Benjamin and Tullis, 2010). Mobile devices are uniquely suited to the support of distributed practice, because people generally have their mobiles with them all the time, including during frequent, short periods of leisure. In this practical, you are going to build a mobile app to support the memorisation of a simple language feature – noun gender in German. The app is intended to be used frequently for short periods of time, to promote effective learning.

Each noun in German has an associated ***gender*** – it is either masculine, feminine or neuter. The different noun genders have different associated articles (i.e. words for “the” and “a”). For example, the word for “the” is ***der*** for masculine nouns, ***die*** for feminine nouns, and ***das*** for neuter nouns. To speak correctly then, one must know the gender of each noun. Unfortunately, there is no predictable pattern to noun gender. For example, cake is masculine and houses are neuter (for no obvious reason). So noun genders must be memorised. Your job is to build a mobile app that allows people to memorise noun gender through distributed practice.

## Vocabulary

A complete mobile language learning trainer (for example, DuoLingo), would support a very large vocabulary, which would probably merit a backend database. In our simplified version, we are going to concentrate on learning the gender of only 11 different nouns. You may hard code these nouns into your application. ***It is not necessary to use any form of external data storage.*** The required nouns and their genders are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Gender** | **Article** | **Noun** | **English Translation** |
| Masculine | Der | Apfel | Apple |
| Neuter | Das | Auto | Car |
| Masculine | Der | Baum | Tree |
| Feminine | Die | Ente | Duck |
| Neuter | Das | Haus | House |
| Feminine | Die | Hexe | Witch |
| Feminine | Die | Kuh | Cow |
| Feminine | Die | Milch | Milk |
| Neuter | Das | Schaf | Sheep |
| Neuter | Das | Schloss | Castle |
| Masculine | Der | Stuhl | Chair |

You will need an image for each noun (see Functional Requirements, below). Images can be found on the I drive, or you may source your own as long as they are tasteful.

## Functional Requirements

For each noun in the learning set, your app must present to the user a picture of the item. During one iteration of the program, all 11 nouns will be presented in random order. For each presented image, the user will be able to select between ***der, die*** and ***das,*** attempting to choose the article of the correct gender for the visualised noun. The app must give feedback for each individual item, as well as a final score (i.e. total or percent correct) after all 11 nouns have been presented. It is not necessary to save any record of performance between uses of the app.

## In-class preparation (to be done on paper):

### The Interface

Sketch a ***complete*** set of wireframes showing the screen appearance over time in response to user interaction.

1. What ***controls*** will you provide to the user, both to submit their input (i.e. their choice of article for each noun) and for navigation (i.e. starting the quiz, moving from one item to the next, etc.) and how will they be placed on the screen?
2. What ***feedback*** will you present to the user, both to display their performance and to display the state of the system (e.g. when the quiz is over).
3. What ***constraints*** will your app provide to prevent errors. For example, can you reduce the chance that a user will inadvertently submit an answer other than the one s/he intends?
4. What is the ***timecourse*** of your application? That is, the timing sequence of events – show the new picture, accept the answer, present the feedback. Have you made it entirely ***visible*** to the user what is happening at all times?
5. What ***non-functional*** ***graphical elements*** do you need (colours, borders, logos, etc.)? Remember that screen space on mobile devices is extremely limited, so graphical elements must serve a clear purpose, either functional or aesthetic.

### The Implementation

You should pseudo-code all methods to transcription.

1. How many Activities do you need?
2. What data structure(s) do you need to manage the application data? This includes the nouns, their images, their genders, the user’s responses, etc.
3. What other non-Activity classes do you need?
4. What algorithm will you use for randomisation of the items? Remember that you must present each noun exactly once during each quiz, and the set of nouns must be presented in random order.
5. What algorithm will you use to mark each response as correct or incorrect?
6. How will you store the accumulated results to allow you to present the summary score at the end?