BLG439E – Computer Project 1

Mobile Application for Teaching Programming to Children

Group Members

ID	Name	Surname
150140718	Göksel	Çoban
150120138	Emre	Özdil
150140115	Arzugül Merve	Ecevit

Design Criteria

At the beginning of this project, we aimed to create an application for teaching programming to children. As it is explained in lesson, we should have taught some programming approaches like variables, data types loop structures.

We made our research in application stores as it is requested. We examined three applications which are:

- Box Island
- Coding for Kids
- Lightbot Hour

All of them was in a labyrinth about going from somewhere to somewhere but it was not the application that we wanted. In class, it is said that children should not write code from the keyboard since it is a little boring for a game. Therefore, we did this application which consists of descriptions and questions. By this way, we taught that we will keep the attention of children to episodes. He or she will try to finish the episodes easily and the aim will be accomplished.

Used Technologies

JSON

To get the questions from application, first we wrote an JSON code which consists of 1300 line. There are 3 steps, in each step there are a number of episodes. Episode consists of different type of cards.

Swift

Swift is a programming language to write software to phones which has iOS as operating system. It’s a fast and user interactive language type.

Xcode

Xcode is a software package which is used by programmers to write applications for iOS. It is a IDE which has editors, compilers, and other software tools that work together to help writing software, compiling, loading it onto a device, debugging, and submitting it to the App Store.

Cocoapods

CocoaPods is the dependency manager for Swift projects. It has so many libraries that can help scaling projects elegantly. The goal of cocoapod is to improve discoverability of, and engagement in, third party open-source libraries, by creating a more centralized ecosystem.

Implementation Steps

There are 3 steps in our game. The steps can be seen from the below:

- Training
- First Step
- Control Structures

Training consists of 9 episodes which teaches different programming commands like printing, variable, numbers, operations with numbers, arrays and if structure.

First step consists of 11 episodes which teaches different programming approaches like basic operations, float and string data types, input and output concepts, casting between data types and operation shortcuts.

Control structures consists of 5 episodes which teaches different programming approaches like comparing with bool data type, if-else code block, priority of operations and while loop.

In the opening of the application, steps will be show to children. If the step is selected, episodes collection view will be opened. Children have to go in episodes in order. For example, to enter episode 2, first the episode 1 should be finished.

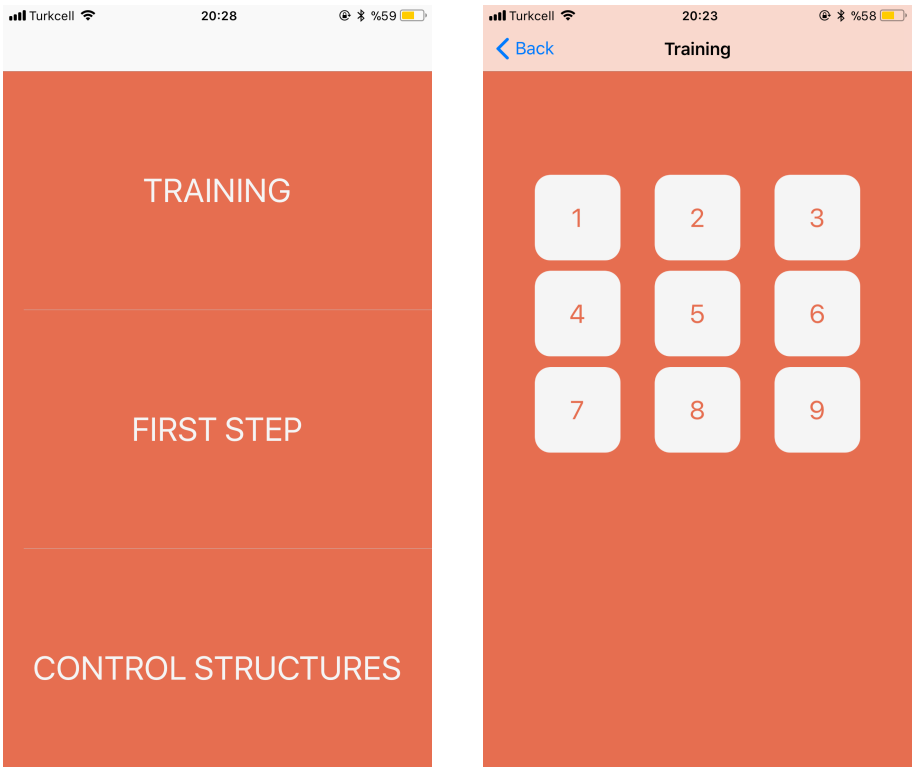

Figure 1 – MenuTableViewController – EpisodeViewController

In each episode, instructions will be shown to user which explains what will be learned in this episode. There are 6 type of cards which are:

- Description
- Description with Code
- Question
- Question with Code
- Executor
- Finish

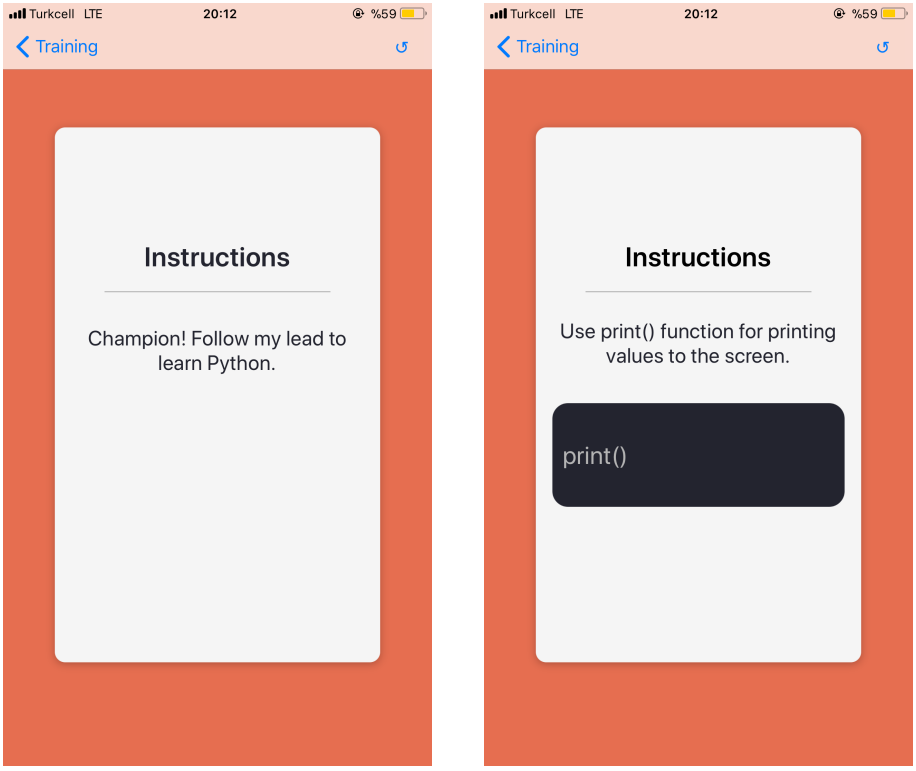

Figure 2 – DescriptionCardView – DescriptionWithCodeCardView

- In description card, there is a header and text explaining what will be learned which are taken from following JSON code.

```
{
        "type" : "description",
         "text" : "Champion! Follow my lead to learn Python.",
         "header" : "Instructions"
},
```

- In description with code card, there is a header, code piece and text explaining code which are taken from following JSON code.

```
{
         "type" : "descriptionWithCode",
         "text" : "Use print() function for printing values to the screen.",
         "header" : "Instructions",
         "code" : "print()"
},
```

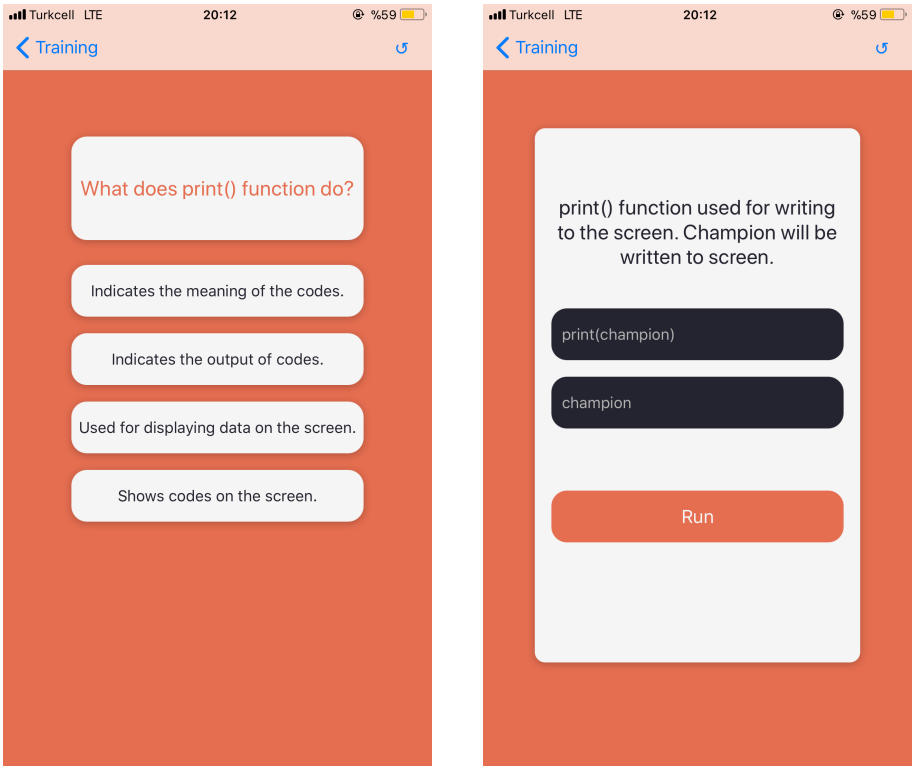

Figure 3 – QuestionCardView – ExecutorCardView

- In question card, there is a text consists of question, answers array and index of correct answer which are taken from following JSON code. Child will select the answer, if the answer is incorrect an alert will pop up will say that the answer is not correct. After child touches the correct answer, view will go to next card.

```
{
        "type" : "question",
        "text" : "What is print()?",
        "answers": ["Function", "Variable", "Value", "Data"],
        "correctAnswerIndex" : 0
},
```

- In executor card, there is a code piece, result of the code and text explaining code which are taken from following JSON code. In the opening of the screen result will not be seen. After children clicks the run button, result will be show to kid. By this way, child can see the result of the code instantly and will learn easily.

```
{
        "type" : "executor",
        "text" : "print() function used for writing to the screen. Champion
will be written to screen.",
        "code" : "print(champion)",
        "result" : "champion"
},
```

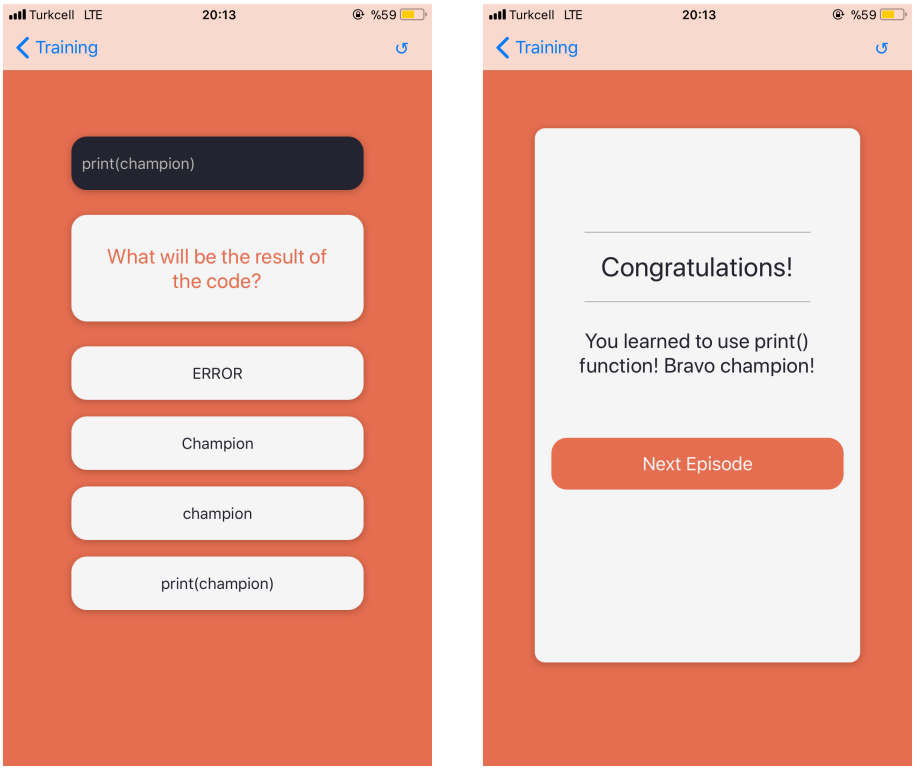

Figure 4 – QuestionWithCodeCardView – FinishCardView

- In question with code card, there is a code piece, text consists of question, answers array and index of correct answer which are taken from following JSON code. Child will select the answer, if the answer is incorrect an alert will pop up will say that the answer is not correct. After child touches the correct answer, view will go to next card.

```
{
 "type" : "questionWithCode",
 "code" : "print(champion)",
 "text" : "What will be the result of the code?",
 "answers": ["ERROR", "Champion", "champion", "print(champion)"],
 "correctAnswerIndex" : 2
},
```

- In finish card, there is a header and text explaining is learned which are taken from following JSON code. There is also a button in order to go next episode.

```
{
 "type" : "finished",
 "text" : "You learned to use print() function! Bravo champion!",
 "header": "Congratulations!"
}
```

Parts Implemented by Each Member of The Group

Merve Ecevit

- JSON code for “training” part questions
- QuestionWithCodeCardView and ExecutorCardView design
- EpisodeViewController.swift file writing

Emre Özdil

- JSON code for “first step” part questions
- QuestionCardView and FinishCardView design
- MenuTableViewController.swift file writing

Göksel Çoban

- JSON code for “control structures” part questions
- DescriptionCardView and DescriptionWithCodeCardView design
- Extension.swift file writing