

Assignment 3

Elm project

Deadline: Wednesday, November 11, 23:55

3.1 Submission instructions

1. Unzip the `Project.zip` folder. You should find:
 - 2 files in the `src` folder:
 - `Main.elm` - for the application itself
 - `Card.elm` - for the functions and types you defined in the previous assignments
 - 1 file in the `tests` folder:
 - `Tests.elm` - for testing your implementation
2. Edit the first line of each of the source files as described in the comments.
3. Edit the source files in the `src` folder with your solutions.
4. When done, zip this `Project` folder and name the zip archive with the following format:

Project_⟨FirstName⟩_⟨LastName⟩_⟨Group⟩

Examples of valid names:

- `Project_John_Doe_30432.zip`
- `Project_Ion_Popescu_30434.zip`
- `Project_Gigel-Dorel_Petrescu_30431.zip`

Examples of invalid names:

- `Solutions.zip`
- `Project.zip`
- `Solutii_Project_Ion_Popescu.zip`

3.2 Project resources

Table 3.1: Lab Resources

Resource	Link
Blackjack rules	https://en.wikipedia.org/wiki/Blackjack
Elm guide	https://guide.elm-lang.org/
Elm core library	https://package.elm-lang.org/packages/elm/core/1.0.5/
Elm Random package	https://package.elm-lang.org/packages/elm/random/latest/
Elm Html package	https://package.elm-lang.org/packages/elm/html/latest/Html

3.3 Project description

In this project you will develop a basic blackjack web app. You will be given some starting (boilerplate) code, which you will have to complete in order to obtain a functional application.

The user of the app should be able to click on a button to draw a card and the drawn cards (the user's hand) should be shown as huge unicode characters. The user should also be able to toggle with a second button whether the cards remaining in the deck should also be shown or not. The user should see their current score and a message when they win (obtain a score of 21) or lose (obtain a score over 21) the game.

3.4 Project tasks

Exercise 3.4.1

10p

Implement the `calculateScore` function, as described in the comments.

Grading:

- 2 points for get the value of each card
- 4 points for generating a list of all possible scores
- 4 points for returning the score closest to 21

Exercise 3.4.2

5p

Complete the `update` function.

Grading:

- 1 point for handling the `Toggle` message
- 3 point for handling the `NewCard` message
- 1 point for using the record update syntax

Implement the view function.

Grading:

- 2 points for adding a button that is used to draw a card
- 2 points for adding a button that can be used to toggle the deck visibility (hidden or shown)
- 2 points for showing the player hand
- 2 points for showing the score
- 2 points for displaying an appropriate message when the player wins or loses