ASSIGNMENT 2

GUESS GAME

# OBJECTIVE

In this assignment, you will create the UI for the Guess Game, or the whole game, it’s up to you.

The goal of this assignment is to get familiar to auto layout concepts using Xcode IDE.

Additionally, for extra credit, you will let your users to actually play the Guess Game.

# MATERIALS

For this assignment, you’ll need the presentation slides form session 6 (you can find it on Slack **endavamobile** (#**isobootcamp**) channel).

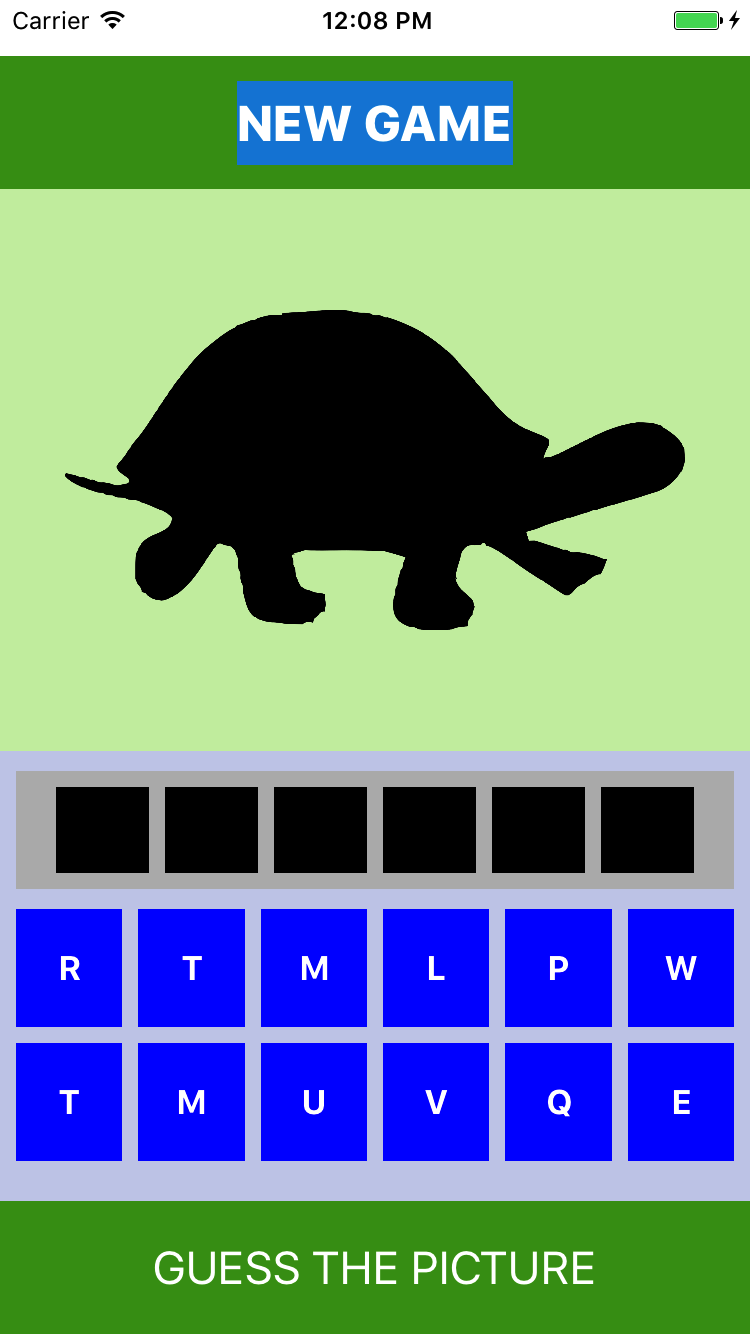
Also, the start-up project with all necessary assets is provided. Short demo video is also provided as reference. You can found both on the GitHub.

Game rules in short:

* 1. Image presenting the animal is showed
  2. User must guess the animal picking the right offered letters.
  3. Only 3 wrong entries are allowed per animal
  4. If user has guessed the animal name, a green view is presented saying “Correct!”. If user failed (has more than 3 wrong attempts) the red view saying “Game over” is presented
  5. Tapping on “New Game” -> game is restarted

# Required tasks

1. Create a game scene in **storyboard**.
2. Game scene should be properly presented in **portrait** mode for all device screen sizes (iphone4, SE, 7, 7+, iPad). For now, don’t be bothered with landscape orientation mods, that will be part of your next assignment
3. Use of **stack view** is **forbidden** for this assignment. The goal is to master the basic auto layout skills.
4. The views presenting the “Correct”/”Game Over” texts should be created and presented **in code**, these views should **not** be created in **storyboard**.. Don’t forget to add all necessary constraints in code!
5. Game should support animals with up to **6 letters** long names.
6. Game should provide **exactly** **10** proposal **letters**.
7. Some basic scene guidelines:



C

D = 0.8 \* C

A

B = 0.1 \* A

B

E

E

E

1. Solution letters (6 black ones) are equally spaced with same width and height.
2. Proposal letters (10 blue ones) are split in two rows, equally spaced with same width and height.
3. The rest of the positions and sizes are free to choose, it’s up to you to improve this design basic.
4. The view presenting the end result (Correct/Game Over) should be presented right over the centre of the animal image (look at the demo video).
5. If you get stuck with proposal/solution letters layout, I recommend you to watch iOS8 Stanford course, 2. session, starts about 44:00 (https://www.youtube.com/watch?v=QLJtT7eSykg&index=2&list=PLy7oRd3ashWodnpf8rjfYEkTgwbOEsKfU)

# ADDITIONAL tasks

Make this game actually to work (replicate the demo video)

1. Recall MVC pattern. Don’t do everything in View Controller. Design a nice game model with exposed API.
2. Think twice how you will model the game level – an animal. Hint: It can be the class, or the struct, or the enum …

Extra additional task:

1. User has total 3 attempts to pick the wrong letter, not per level.
2. Rename the “New Game” to “Restart level”. Tapping on that button, an alert dialog is show asking the user if she want to restart the level (Yes/No).
3. Remove presenting the end result views (Correct/Game Over). Instead of that, implement next:

* When the user successfully pass the level, present the alert dialog with title “Level Clear” asking the user to replay the level or to try next one (another animal).
* When the user failed the game, present the alert dialog with title “Game Over” asking the user to start from the begging (first level, attempt number is restarted to 3).

For the extra additional task, you’ll need to get familiar with **UIAlertController** (presentation of alert view)