Assignment 3
(due 3/29 at 11:59 pm)

CMSC 428 – Mobile Programming (iOS)
Spring 2022

In this assignment you will be building a kids's app called Kids Joy Center. It will be designed only for iPad Pro 9.7 inch screen (**no autolayout**) in landscape mode with width=1024 and height=768.

Initial Screen:

It will show icons for three games, level of games (easy, medium, hard), play button and high scores button. Once the player selects one of the games and level of difficulty and clicks the play button, it will open the related view controller and the game. **The user can only select one of the games and one of the difficulty levels. So, make sure this relation is maintained.** There should be an **indication** that the specific game or level of difficulty is selected (similar to segmented control object working principle in UIKit). Navigation controller should be used to have back navigation capability from each new screen opened. Set the navigation controller title view to app logo.

3 Games: a) memory game, b) sorting game, c) popping the balloons game

View:
- All items should have images.
- Only landscape mode should work.
- Use background image with some **high transparency.**
- This is the **only screen that is allowed to be designed using Xcode interface builder** (IB). Others should be designed programmatically. For others you are allowed to use IB only to create the view controllers and arrange necessary segues from initial screen.

Functions:
- Clicking play button will start the selected game with the selected game difficulty. User should **not** be able to start the game if either the game type or difficulty level **is not selected.**
- High scores button will **open a pop up page** showing the best five scores of the player in all games.

The content of view controllers for each game **should be designed programmatically** rather than using storyboard. You can use storyboard only for linking segues from home screen to corresponding empty view controllers.

Memory (or Pairing) Game Page (View controller):

The player is expected to find the pairs of images. When the first image is clicked, the underlying image is shown. With the click to another image, its underlying image is also shown and if it matches with the previous one, a sound (e.g., cheer) is played and **images stay opened** and their user interaction is disabled. **If they do not match, after 1 second,** the images are **closed**.

View:
- Images should be added to the view programmatically. There will be 4x3 (easy), 4x4 (medium) and 4x5 (hard) images, depending on the difficulty level selected. **Do not use Collection View for this.** Images should be **placed on screen evenly** in any level of difficulty (distance to left and right sides, and distance to up and down sides should be equal).
- The name of the game will be seen at the top (as title).
- Labels for time and score will be shown at corners. Time will be set to 2:00 minutes for easy, 1:45 for medium and 1:30 for hard.

Functions:
- You can use **touches or gesture recognizers** to interact with user.
- There will be a timer counting to zero (updated every second with Timer class) and score part. Every time a new pair is found in 3 sec (after previous found), 5 points added to score. If it is found in more than 3 seconds but less than or equal to 7 seconds, 4 points is added. If it is found more than 7 seconds, 3 points added.
- If user goes back to initial screen, the game will be lost.
- If user achieves finding all pairs before time expires, score will be recorded in high scores if it is within top 5 (in terms of score only). Also, **it will ask user if he/she wants to play again.** Answering "No" will take the user to initial screen. If the game ends before user finishes opening all, it will say, you lost the game and ask if he/she wants to play again (as alert).
- Every time the game starts it will **randomly** decide the locations of image pairs. 10 animal images are given in image resources. They can cover up to 20 images. However, if fewer number of images is needed (as in the case of easy and medium difficulty levels), you will need to **randomly select** which of the 10 images will be shown. **You are free to choose your own images (but use png files with no backgrounds)**.

Sorting (the Vehicles) Game Page (View controller):

The player is expected to sort the vehicles based on the part of the earth they are used, namely, air, water, land. The images will be **randomly** selected from a pool of 15 images and located at the top part. Player will be able to drag them with one finger to the right areas he/she thinks they are used. An appropriate sound should be played if the vehicle is dragged to correct area and finger is lifted there. If it is not correct area, the image **will slowly go back to initial location** with an animation (say in 2-3 sec) **from the location the finger is left.**

View:
- Images should be added to the view programmatically. There will be 8 (easy), 10 (medium) and 12 (hard) images, depending on the difficulty level selected. The distance between the images (at their initial location) should be arranged equally, so appropriate frame locations for images should be calculated while adding the images to the view.
- The name of the game will be seen at the top (as title).
- Labels for time and score will be shown at corners. Time will be set to 1:00 minutes for easy, 45 sec for medium and 30 sec for hard.

- You can approximately define the correct areas for each of air, water, and land in the given background image using 1-2 rectangular views.

Functions:
- You can use touches or gesture recognizers to interact with user. Custom cocoa touch classes can be written.
- There will be a timer counting to zero (updated every second) and score part. Every time a vehicle is located to the correct area in 2 sec (after previous correct placement) 5 points is added to score. If more than 2 seconds but less than or equal to 4 seconds has passed, 4 points is added. If more than 4 seconds has passed, 3 points is added.
- If user goes back to initial screen, the game will be lost.
- If user achieves locating all the vehicles to correct areas before time expires, score will be recorded in high scores if it is within top 5 (in terms of score). Also, it will ask user if he/she wants to play again. Answering "No" will take the user to initial screen. If the game ends before user drags all items to correct areas, it will say, you lost the game and ask if he/she wants to play again (as alert).
- Every time the game starts, the images will be **randomly determined** from the pool of 15 images. You are free to choose your own images (but use png files with no backgrounds).

(Popping the) Balloons Game Page (View controller):

The player is expected to pop the balloons as they come from the bottom of the screen. The balloons will come with numbers on them determined randomly. There are 10 balloon images with different colors, so each balloon color will also be determined randomly. When a specific balloon is popped with user touch, the number on the balloon will be added to the score (also **add a balloon pop sound** at that moment). There will also be bonus balloons (with star image on it) or killer balloons (with skull image). Bonus ones should have **higher speed than** regular balloons and if popped, they will **slow down the speed of the new balloons in the next 5 seconds.** Killer balloons will have **slower** speed than regular balloons and if popped, they will end the game.

View:
- Balloons should be added to the view programmatically **at every sec.**
- The numbers on the balloons will be selected from range 1-9 in easy, 1-7 in medium and 1-5 in hard level.
- The speed of the balloons should increase as the difficulty level increases (you can decide reasonable speeds considering that this app is for kids).
- Moreover, there will be one balloon added (animated from bottom to top) at every sec in easy level. In medium mode, there can be one or two balloons added (**with 50% chance**) at every sec. In hard mode, there can be one, two or three balloons added (w**ith 33.3% chance**) at every sec.
- You can define 10 different spots (that do not overlap) of entrance for the balloons from the bottom of the screen. The entrance location should be **decided randomly** out of these 10 possible spots. Also, if multiple balloons will be added at current sec, then their selected **locations should be different.**
- The name of the game will be seen at the top (as title).
- Labels for time and score will be shown at corners. Time will be set to 1:00 minutes for easy, 45 sec for medium and 30 sec for hard.

Functions:
- You can use touches or gesture recognizers to interact with user.
- There will be a timer counting to zero (updated every second with Timer class) and score part. Every time a balloon is popped, the number on the balloon will be added to the score.
- If user goes back to initial screen, the game will be lost.
- The game will end when the time expires or the user **cannot pop a balloon in the last 10 sec.** When if it ends because the time expires, score will be recorded in high scores **if it is within top 5**. Also, it will ask user if he/she wants to play again. Answering "No" will take the user to initial screen. If the game ends because player did not pop any balloon in last 10 sec, it will say, you lost the game and ask if he/she wants to play again (as alert).
- Every time the game starts it will **randomly decide the colors of balloons**, their **entrance spot from bottom and numbers on them.** 10 balloon images and number images are given in image resources. You can use your own.
- **Bonus balloons will appear randomly in every 20-25 sec**. They will have higher speed than regular balloons and if popped, it will slow down the speed of the new balloons in the next 5 seconds.
- **Killer balloons will appear randomly in every 20-25 sec** and will have slower speed than regular balloons and if popped, they will end the game (do not save the score even if it is in top 5).

High scores page:

View: This will be shown **as a pop up view** with an appropriate size. It will show the top 5 scores so far. **Each row will show four labels**: "order", "game type played", "level of difficulty", and "score". Ex: "1", "Sorting", "Hard", "40"

Functions:
- User will close the high scores pop up window by clicking cancel.
- When the app is closed and reopened, the high scores will not be deleted (Apply **data persistence** with UserDefaults to save the high scores and read when the app is re-opened).

General Guidelines:

- Define your custom classes as needed.
- Auto layout and constraints are **NOT needed** as you need to design programmatically on a single iPad screen only (iPad Pro 9.7 inch).
- Do not make your images stretched. Use aspect fit or fill for the content mode of the image views.
- Most image resources are provided but you are free to create/use your own.

Submission Guidelines:

- Make sure your app is running before submitting.
- Submissions should be made in Canvas before midnight on 3/29.
- Up to 2 days late submissions only (1 day 20% penalty, 2 days 40% penalty).

- Make sure all your images and other resources used are inside the project folder. When copying an image inside the Assets folder, always make sure "copy" is checked and select the target as the project.
- Don't forget to set the app icon properly.

General Grading Criteria:
1- Appearance: (50%)
    a. Initial Screen
    b. Memory Game Page
    c. Sorting Game Page
    d. Balloons Game Page
    e. High scores as pop up
2- Functionality: (50%)
    a. Correct playing of games
    b. Timer's counting to zero
    c. Randomization
    d. Updates for level of difficulty
    e. Adding to high scores
    f. Data persistence application for high scores
    g. Alerts

BONUS:
Implement scoreboard updates with some animations. For ex: when a point is earned, show this point on the score and move it up/down while fading and with a money sound etc. Also, specific to balloon game, when a balloon is popped, make the number from that balloon move to score **(with same speed for all popped balloons).**

Some screenshots.

High Scores

Easy Medium Hard

PLAY

Back

TIME 1:27   SCORE 0