Razzmatazz: Smart Frisbee

Mobile App Development and User Interfaces

By Cole Taylor, ECE '21

Introduction

The Rediscoulous smart disc, seeks to provide disc golf players with quantitative feedback on the performance of their throws. A suite of sensors on the disc will gather spatial data about the throw in three dimensions. The information will then be transmitted off the disc via Bluetooth to an iPhone. Once the data is on the iPhone, a mobile application will process the data and display the results to the user.

This article will focus on the iPhone stage of the project. It will first go over a few common architectural design patterns for developing mobile application to demonstrate how user interfaces are decoupled from the data processing. From there, the report will delve deep into the concepts and process of developing a successful user interface.

Design Process

To Understand iOS development, it is important to look at the established architectural patterns used by people in industry. There are three main patterns that are used to break mobile development into more accessible chunks. These three patterns are Model View Controller (MVC), Model View Presenter (MVP), and Model ViewModel (MVVM). The latter two architectural patterns are newer variants on the original MVC pattern.

The goal of all of these architectural patterns is to separate the computational logic from the presentation layer. In other words, it creates a barrier between the user and the data the drives the application. Since MVP and MVVM are derivatives of MVC, all three architectures share many similarities. The following sections will examine the general structure of each pattern and the differences between them. Figure 1 provides a visualization of the three structures.

Model View Controller

MVC is made up of three layers. As the name suggests, the three layers are the Model, the View, and the Controller. In order to decouple these three components as much as possible, the layers are not aware of the state of the other layers. The layers can only communicate through notifications. The Model holds the application's data, logic, and rules. The View displays the information to the user in an accessible format. It is updated when the Model or the Controller sends it a notification. The Controller takes input from the user and converts it to notifications. It sends these notifications to the Model and the View so that they can update the state of the application. Figure 1 shows the direction that notifications are sent between the three layers.

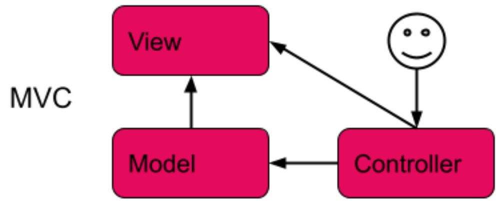

Figure 1. Diagram of MVC layer communication

Model View Presenter

MVP is also made up of three layers. The layers are slightly different than the MVC pattern. The three layers are the Model, the View, and the Presenter. This variant provides more decoupling between the View and the Model. The Model has essentiallt the same function as in the MVC pattern. However, the View layer takes on more responsibility. The View layer takes on the role of the Controller from MVC. In MVP the View is not connected to the Model layer. Thus, it has to delegate any model changes to the Presenter layer. The Presenter is in charge of communicating with the Model and the View. Figure 2 shows how notifications are sent in MVP.

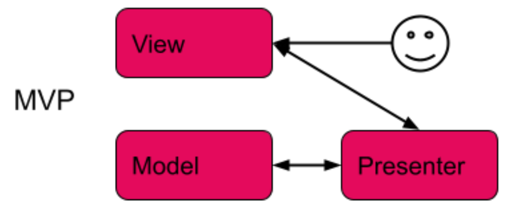

Figure 2. Diagram of MVP layer communication

Model View ViewController

MVVC is made of three layer much like the other two variants. It is made up of the Model, the View, and the ViewModel. The Model is again unchanged. It performs the same tasks in all three variants. The View is as passive as possible in this variant. Any logic that needs to be performed is delegated to the ViewModel. The ViewModel is a model of the View. It is unaware of the state of the View. The ViewModel defines what happens when there are user inputs and translates the model so that is can be displayed in the View. Figure 3, shows how notifications are sent in MVVC.

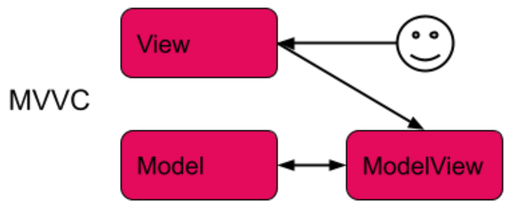

Figure 3. Diagram of MVVC layer communication

User Interfaces

A user interface is how humans interact with technology including mobile applications Successful user interfaces make it easy for humans to navigate through the application and understand the information that the application is trying to convey. The two main components of user interfaces are data visualization and usability.

Data Visualization

Data visualization has been around for thousands of years. Ancient humans used images carved into cave walls to keep track of hunting statistics. Today, data visualization is even more important as we attempt to understand large data sets in applications like machine learning. Additionally, visual representations of data can communicate information faster and more effectively to a broad range of knowledge backgrounds and cultures. This makes successful data visualization important in any product that wishers to appeal to a wide variety of people.

One of the most important choices in data visualization is the choice of color. Color is important when mapping data elements to a more approachable visual form. It allows humans to quickly distinguish between data elements in order to find the desired information. Color separation plays the largest role in speed and accuracy when identifying separate data elements.

In order to understand color separation, it is important to first understand the CIED LUV color model. Colors are defined with three dimensions. The first dimension is the luminance, L. L gives a value to the brightness of a color. The seconds and third dimensions define the chromaticity, u and v. Chromaticity gives a way of defining colors independent of the brightness. This method for defining colors is advantageous for understanding the perceived color serperation. The main advantage is that it is easy to calculate the separation. The following equation can be used to find a value that represents the perceived separation of any two colors, ΔE.

$$\Delta E = \sqrt{(\Delta L)^2 + (\Delta u)^2 + (\Delta v)^2}$$

Using color separation when assigning colors to data elements can improve the speed and accuracy that a user can identify the information relevant to them. Once main case that receives the aforementioned speed improvements is color legends for graphs or maps. Making sure that colors on the legend maximize the ΔE between them, can make the data visualization much more effective.

Usability

Usability is important to think about on any platform. Mobile applications stand out among all platforms, because they require even more attention. The small screen size and limited user input methods make usability a critical issue. Usability can be broken down into three metrics according to the International Organization of Standards. It can be broken down into effectiveness, efficiency, and satisfaction. Effectiveness looks at whether or not users can accurately achieve their goals. Efficiency examines the resources that users require to achieve their goals. Finally, satisfaction assesses the attitude that the user has towards the product after use. In most cases satisfaction follows from the first two metrics. Thus, the main focus for usability is effectiveness and efficiency.

Two main design trends are used in mobile applications to better usability: skeuomorphism and flat design. The goal of skeuomorphism is to attempt to replicate the physical world. One the other hand, the goal of flat design is to abandon the physical world and rely on metaphors and symbols. Figure 4 shows the two design trends. The calculator on the left implements skeuomorphic design by using drop shadows and gradients to make it look like there are physical buttons. The calculator on the right uses flat design.

Skeuomorphic design and flat design both have advantages and disadvantages. Skeuomorphic design enables mobile phone novices to more effectively navigate applications. The realism is more approachable for people who have little experience with the mobile platform. The downside is that the realism comes at a space and complexity cost. Figure 4 does a good job of demonstrating the space cost. In order for the buttons to have effective gradients, the text must be smaller. Flat design has the advantage of simplicity. The buttons on the calculator don't have any gradients, so the text can be larger. The main challenge with flat design is that it is not as intuitive to new users.

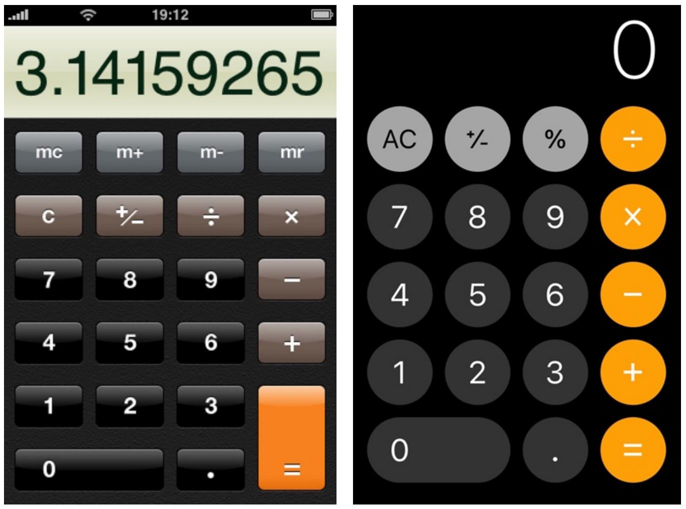

Figure 4. Skeuomorphism vs flat design

Conclusion

In conclusion, the smart disk application will aim to incorporate everything discussed in this report in order to move towards an effective and easy to use application. The smart disc application will use a variant of MVC to make sure that the user interface is decoupled from the data logic. This will allow us to focus on the different elements in parallel. The user interface will develop design elements in a way that integrates the data visualization and usability in the aforementioned guidelines. Color separation will be used so that disc golf players can quickly identify key information. A mixture of flat design and skeuomorphic design will be used when implementing user interactive features. The two methods will be used in tandem to attempt to combine the strengths and mitigate the weaknesses. Creating a mostly flat designs will simplify each element and make the app more streamline. However, including some elements of skeuomorphism will add a touch of reality so that new users can easily understand how to interact with the application. With all of these design criteria, the smart disc application will be an effective and efficient way for users to improve their disc golf performance.

References

1. Manuela Aparicio and Carlos J. Costa. Data visualization. 3(1):7–11.

2. Autumn. Flat vs. skeuomorphism.

3. Leonard A. Breslow, Raj M. Ratwani, and J. Gregory Trafton. Cognitive models of the influence of color scale on data visualization tasks. 51(3):321–338.

4. Aymen Daoudi, Ghizlane El Boussaidi, Naouel Moha, and S`egla Kpodjedo. An exploratory study of MVC-based architectural patterns in android apps. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC ’19, pages 1711–1720. Association for Computing Machinery.

5. Rachel Harrison, Derek Flood, and David Duce. Usability of mobile applications: literature review and rationale for a new usability model. 1(1):1

6. C.G. Healey. Choosing effective colours for data visualization. In Proceedings of Seventh Annual IEEE Visualization ’96, pages 263–270,. ACM.

7. Ankit Sinhal. MVC, MVP and MVVM design pattern.

8. Konstantinos Spiliotopoulos, Maria Rigou, and Spiros Sirmakessis. A comparative studyof skeuomorphic and flat design from a UX perspective. 2(2):31.

9. Alfredo Vellido Alcacena, Jos ́e David Mart ́ın, Fabrice Rossi, and Paulo J. G. Lisboa.Seeing is believing: the importance of visualization in real-world machine learning ap-plications. pages 219–226.

10. Dominic Alves. iPhone Calculator Screen Print, November 2009.