Software Design Document

Version 2

Project Birdboard

Multidimensional Data Visualization in iOS Using Google Cardboard

Sponsor:
Dr. Razib Iqbal

Development Team:
Ryan Bagby
Chad Brewer
Leonard Museau
Darren Williams

Submitted to
Dr. Razib Iqbal
Assistant Professor of Computer Science
Missouri State University
in partial fulfillment of the
CSC450 Course Project
Summer 2017

Table of Contents

1.0 Introduction

This document will act as a reference for the software design of the Birdboard application and informs any interested persons about Birdboard architecture and Birdboard bird database. Additional details will be discussed to aid in this documents ability to inform and describe the overall design of the Birdboard application.

This document is designed for a technical audience with development experience in object oriented programming.

Further, the Birdboard application's data design, architectural and component-level design, user-interface design, restrictions, limitations, constraints, and testing issues will be included.

Goals and objectives

The major goal of Project Birdboard is to allow mobile users of both the Android operating system and iOS users to use and access all the features provided by the application. By providing access to both mobile operative systems, the application will be able to reach more users.

The second goal of the application is to add on to the 3-Dimensional or 3D aspects of the application. The application already provides a fairly basic 3D environment for the user to play in. By adding forestry to the Table View and GIS Point/Bird data View, this will provide a more immersive experience for the user and also provide a more realistic experience for the user.

1.2 Statement of scope

Birdboard is a virtual reality software paired with an informative wiki to provide useful information to the user. Birdboard uses a variety of interacting class objects to fulfill its intended purposes including classes for loading wiki information and updating Unity's Monobehavior for all assets containing scripts within a scene. Much of Birdboard focuses on taking user input in the form of directional cues and button input to allow users to navigate menus as well as location and wikis.

The major processing functionalities of the software can be classified into three categories as essential functions, desirable functions, and future requirements. Essential functions include: availability on both Android and iOS platforms, interactive diorama of the environment for overhead view and exploration. Desirable functions include: additional audio/music to further the immersive experience, additional visual assets to enhance the view and texture of the environment.

A description of the software is presented. Major inputs, processing functionality and outputs are described without regards to implementation detail. Rank the major processing functionality from the developer's point of view. Use a simple ranking system such as: essential, desirable and future requirements. This should represent what you think your team can accomplish in the time frame of a semester. The essential requirements, you are sure you can complete. The desirable requirements you hope to complete, but are not sure about. The future requirements, you have strong doubts about. Strive to balance the desires of your client with the reality of the time it takes to develop a SW product.

1.3 Software context

Birdboard acts as a product to aid institutions of higher learning. Birdboard strategically fills the niche of informative environmental information and an immersive virtual reality experience.

1.4 Major constraints

The only major constraint this project will have is that the product will only run on Android and IOS devices only. Windows smartphone users will not unfortunately have the capabilities to run the program on their devices.

2.0 Data Design

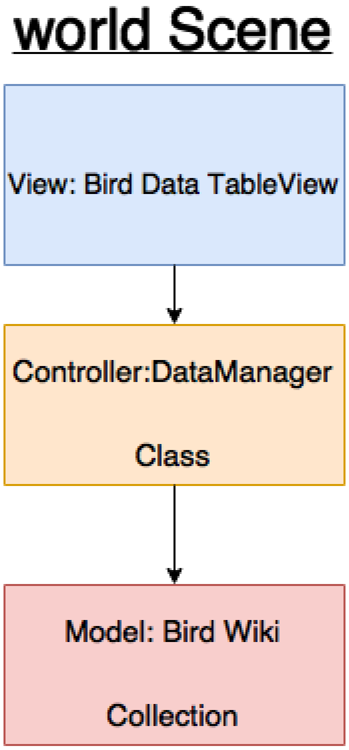
Unity structures the data within itself using Monobehaviour. The objects created within Unity have scripts and events attached to them, depending on their use.

2.4 Database description

The application uses CSV files to contain image files of the bird data. The applications pulls and displays the files when the user chooses a GIS point.

3.0 Architectural and component-level design

With the way that Unity's API works we find that by working with Unity's own preferred architecture that the model view controller (MVC) would work best for our project. Below is an example of this architecture in our application.


world Scene
View: Bird Data TableView
Controller:DataManager Class
Model: Bird Wiki Collection

4.0 User interface design

The user interface is designed to let the user move back and forth between each seen using Unity’s GazeClick asset or Google Cardboard’s button. The user can interact with menu options and GIS points but the surrounding forest environment will not be interactive.

4.1 Description of the user interface

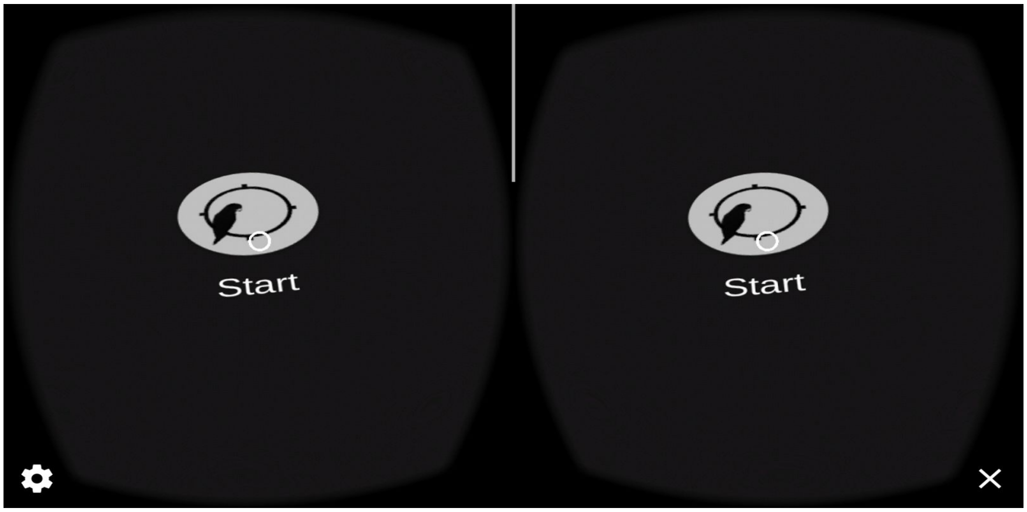
Once the user starts the application and inserts their mobile device into a Google Cardboard device, they are presented with the option to start along with the logo of the application. Once the user selects start, they are brought to the main menu view.
At the main menu view, the user is allowed to choose from four different options. The first option is to select the Table View, the second option is Instructions, the third option is Credits, and the fourth option is to exit the application.

If the user selects Instructions, they are brought to a view that informs them on how to operate the application. If the user selects Credits, they are brought to a view that gives them information on the design teams of the application.

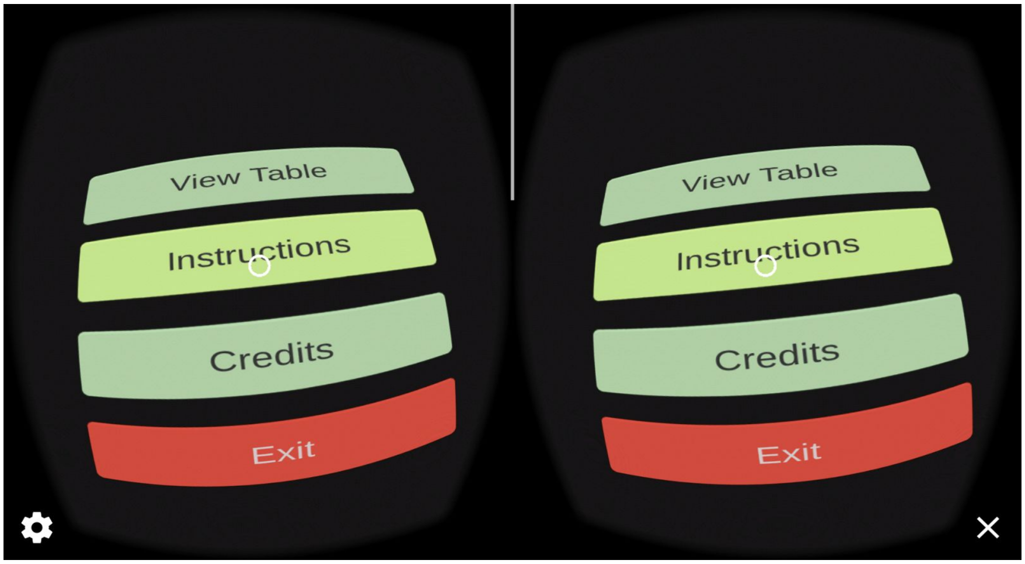
If the user selects the Table View option, they are brought to an forest environment. In this view a table is presented to the user where they can choose a point to be teleported to, they can choose to view the instructions of the application, or they can choose to move back to the main menu.

When the user selects a GIS point, they are teleported to another forest scene where the GIS point is presented in front of them. They are also given the geographical location of where the point is located. In this view the user is able to either choose the point and view the data of the birds that can be found in the region or they can choose to go back to the table view.

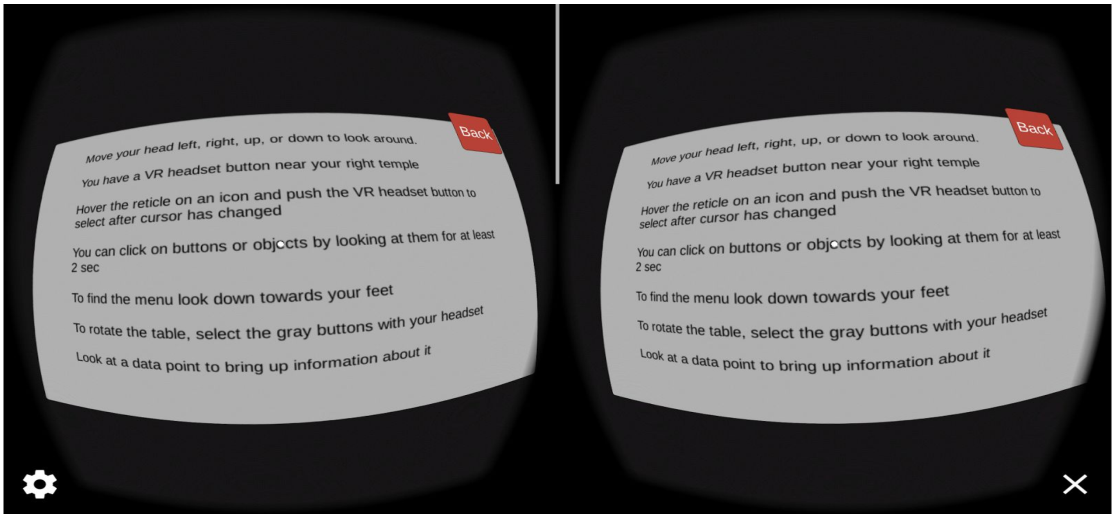
4.1.1 Screen images



Start view. The first image is what the user sees when they first open the application.



Menu View. When the user selects start, they are brought to the menu view where they can select from a list of options.



Move your head left, right, up, or down to look around.

You have a VR headset button near your right temple

Hover the reticle on an icon and push the VR headset button to select after cursor has changed

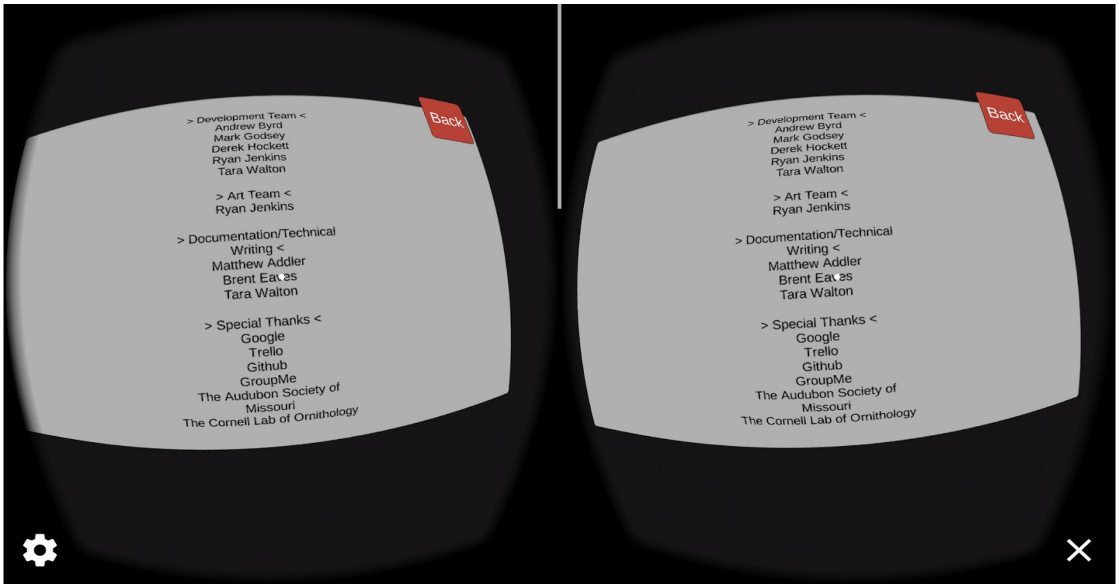
You can click on buttons or objects by looking at them for at least 2 sec

To find the menu look down towards your feet

To rotate the table, select the gray buttons with your headset

Look at a data point to bring up information about it

Instructions view. When the user chooses instructions from either the main menu or the table view, they are presented with a view that gives them instructions on how to operate the application.



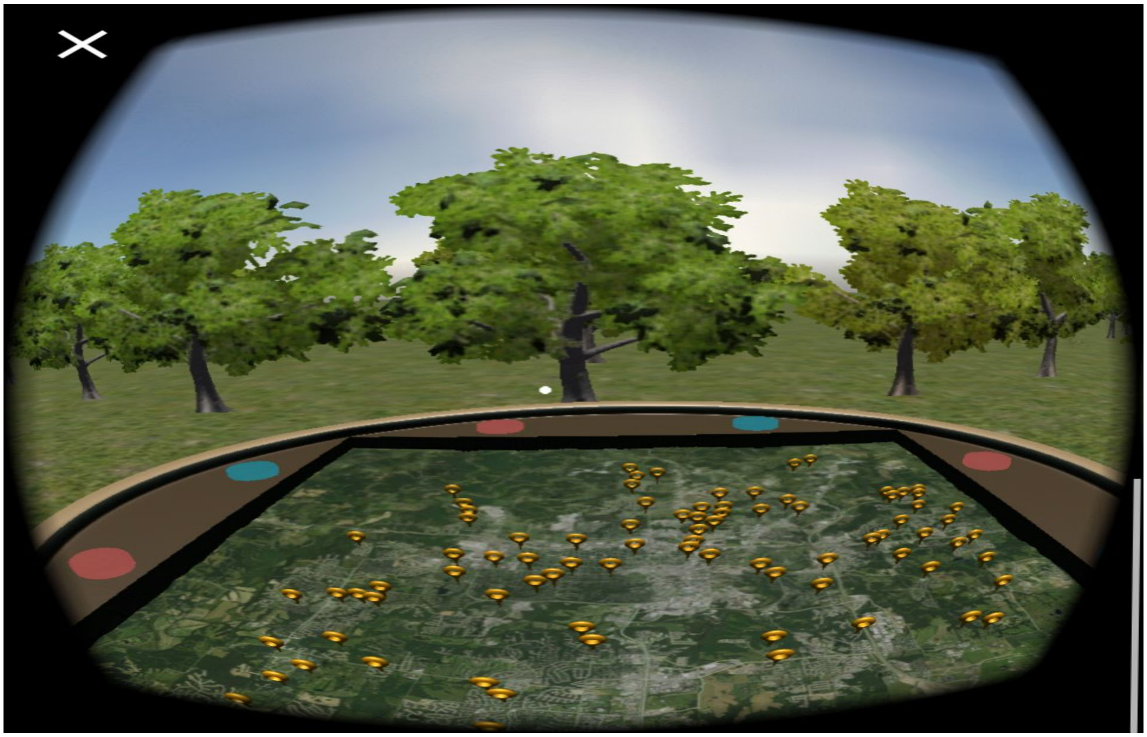
> Development Team <
Andrew Byrd
Mark Godsey
Derek Hockett
Ryan Jenkins
Tara Walton

> Art Team <
Ryan Jenkins

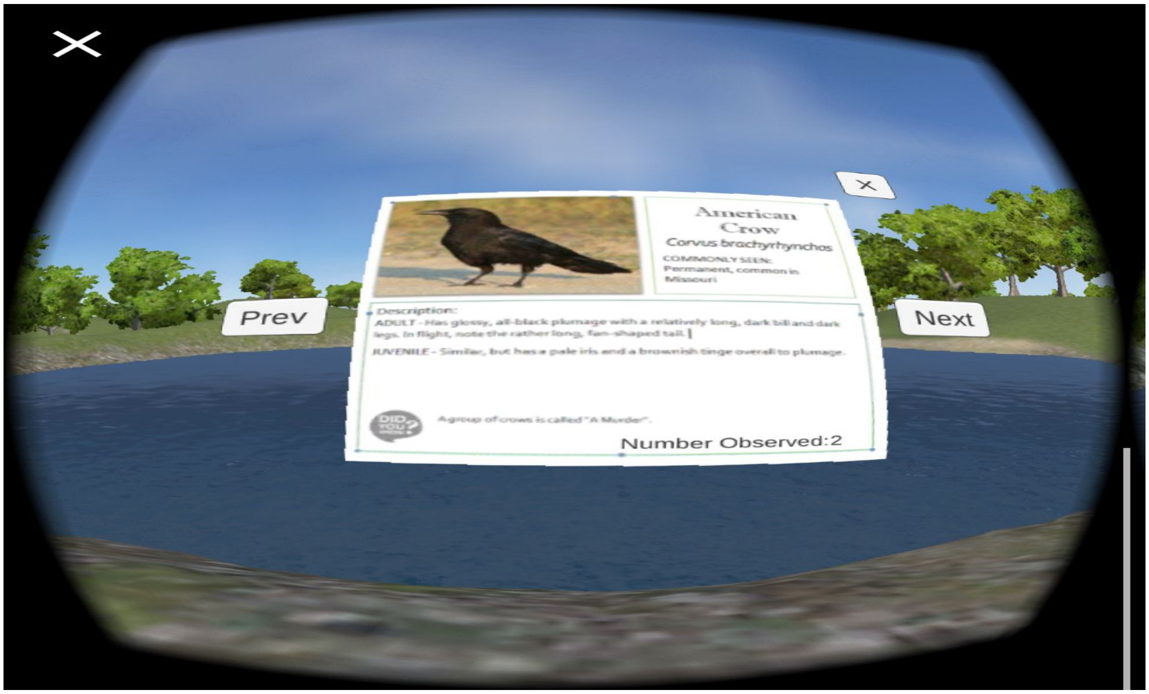
> Documentation/Technical Writing <
Matthew Addler
Brent Eaves
Tara Walton

> Special Thanks <
Google
Trello
Github
GroupMe
The Audubon Society of Missouri
The Cornell Lab of Ornithology

Credits view. When the user selects Credits from the main menu, they are presented with a view of the application’s credits, who helped with the creation of the application.



The Table view. The user can select the table view from the main menu. Once the Table View is selected, the user is transported in front of an interactive table that allows them to rotate the table, select instructions, go back to the previous view (Menu View), or to select a GIS point that will teleport them to the point and allow them to view information about the birds that can be found in that area. This view has a forest environment added to it to provide the user with a more immersive experience.



American Crow

Corvus brachyrhynchos

COMMONLY SEEN:
Permanent, common in Missouri

Description:
ADULT - Has glossy, all-black plumage with a relatively long, dark bill and dark legs. In flight, note the rather long, fan-shaped tail.

JUVENILE - Similar, but has a pale iris and a brownish tinge overall to plumage.

A group of crows is called "A Murder".

Number Observed:2

Bird data view. Once the user chooses the GIS point from the Table View, information about the birds found in that area will be presented to them. This view had a forest scene added to it to provide a more immersive experience.

4.1.2 Objects and actions

Unity's internal Monobehavior acts as a foundational object so that assets containing scripts are updated via an update function. In practice, this results in the user viewing his environment which is updated upon an action/interaction (i.e. movement).

5.0 Restrictions, limitations, and constraints

Constraints are minimal for the Birdboard application: the external application Gaze Click it utilized in the core functions necessary in Birdboard performance.

6.0 Testing Issues

To test the application, Unity Game Engine allows for the application to be ran within Unity itself. The developer is able to move throughout the different views and test whether the added features are working properly. The only testing that needs to be done is whether the application works on iOS and if the added features are displaying properly on iOS as well as Android.

6.1 Classes of tests

Since the added features require little to no extra code, black-box testing can be implemented. Since Unity also allows for building and running the application on a target device such as an Iphone or an Android phone, we can connect a device and use the device to navigate through the application to make sure the added features are working properly.

6.2 Expected software response

The application will work on iOS mobile platforms and will display the added forestry features that will be implemented in the Table View and GIS point/Bird data views.

6.3 Performance bounds

There should be no changes in performance from the Android build to the iOS build.
There will be added load times from when loading the Table View scene and the GIS point/Bird data scene due to the added environment features.

6.4 Identification of critical components

Critical components that should be closely watched during testing are any differences in how the newly added features may change or differ when the application is ported to the iOS operating system from the Android operating system.