Group Report

Introduction

In 2016, Durham University's Institute of Computational Cosmology successfully created the Galaxy Makers exhibit to communicate our computational cosmology and astronomy research. This project is prepared for the Royal Society Summer Science Exhibition 2020 and furtherly developed in an AR form.

Aim Outline

This project aims to develop an AR cosmology app, which will be widely used in the event of cosmology exhibition to educate young generation to increase their interest in cosmic knowledge. The app enables students to view these AR galaxy model via the AR technology.

Among the development process, these galaxy models are pre-simulated and will be imported into the app functions, which could be easily called when they are need. Specifically, the app will recognize a picture hanging on a wall with an anchor point. The procedure will firstly recognize the anchor point as an axis starting point. The initial interface of the app will show the real-time picture that the device camera is pointing to. By changing the different direction values of device. When obtained value meet the pre-set galaxy, more detailed information will be popped out in 3-d model and show the formation information in 3D and multiple media based on AGN feedback, mass stars, which define the structure of the universe model. All these values will be calculated by the computer as the settled algorithm, and they will be used to make a realistic model.Objectives

Objectives

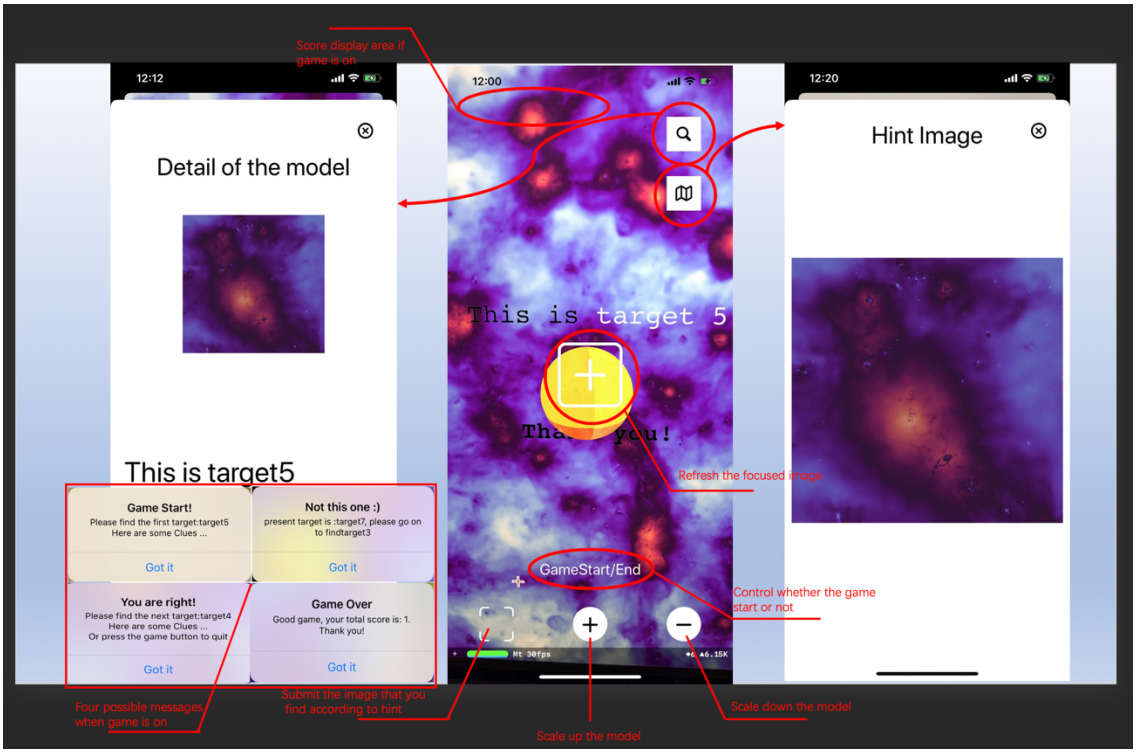
1. By using the Xcode as the development tool, which includes all the required packages and interfaces for the AR project development.
2. Swift is selected as the main development language. Owning to the advantages of simple grammar, easy to use and learn. It dramatically reduces the cost of learning developers. Suitable for this short-term group project.
3. GitHub was chosen as a version management tool to facilitate simultaneous code writing among team members to speed up the completion of this project.
4. A galaxy map is pre-given by the group supervisor, all subsequent programming, testing, and performance demonstrations will be based on this

Download

The app is only available for ios version now and only works on device with ios 11 or higher system version. Available device models include iPhone 6s/iPhone 6S Plus/iPhone 7/iPhone 7 Plus/iPhone 8/iPhone 8 Plus/iPhone X/iPhone XR/iPhone XS/iPhone XS Plus/iPhone 11/iPhone 11 Pro/iPhone 11 Pro Plus

1. Pull/download the AR-cosmology project from the GitHub or simply navigate to your directory of interest, then clone. Use 'bash$ git clone https://github.com/huangjjv1/AR-cosmology.git.'
2. Make sure your computer has Xcode pre-downloaded
3. Connect any one of the above device with your computer
4. Finally, open the `*.xcodeproj` file and build to your [supported device](#requirements)

Main interface illustration



Main theme & detail window

AR-cosmology App Utilisation Guidance

1. Opening the AR-cosmology app, the one with galaxy map icon
2. The main page will be a camera page, and it only works with the iPhone back camera. A focus square is located in the middle, which is helping to increase the accuracy of target capture.
3. While the specified galaxy simulation map is given, a user could use the camera to capture any galaxy dot on the map. Thus the corresponding AR model will be shown on the screen.
4. In the case of the AR model, by clicking the “plus” and “minus” top to zoom in or zoom out the model, which enable the user to observe the model details.
5. Under the game mode, the target galaxy model will be pre-given by the system. User only need to use the device to find the proper model on the map. When the correct model is found, the number of scoreboards in the upper left will increase.
6. In the case of AR model, by clicking the search button on the upper right, an up slide window will be presented to show the model details, like the black matter quality, number of planets, etc.
7. The 6 and 7 options will be widely used in the event of an exhibition to educate the young generation to increase their interest in cosmic knowledge

* If running error happens, you could click the highest root directory option in the feature bar on the right-hand side. Thus, under the “Signing & Capabilities”, just change the team name and bundle identifier by your own individual information.

Functions Work Flow

Setup

Application starts: It automatically calls the function **viewWillAppear()**, in which configuration will be initialized to **ARImageTrackingConfiguration**, reference images will be referred. After that, the function **viewDidLoad()** is automatically called next. In which, some global variable such as score, transparency.

Targets displaying

1. **Targets recognition**. The **ARScneneView** is running in real-time form and recognizing the image in the viewing frame. Whenever it successfully has the target and its corresponding anchor name, **renderer()** function is called to render and display the model in augmented reality. When the user press the detail button at the top right of the interface

2. **Action function**. Whenever a target is recognized, it will be added a series of action to the displayed model (We set it up as a spinning sphere by default). After pressing the scale up and down button (at the bottom of the interface). Function spawningmode() is called to add the function to the rendered model.

Game system

1. **Game initialization**. Game start/end button (in main interface) and the corresponding button function will give the first target to find and set score to zero.

2. **Game UI**. Submit button submitting the current scanned target to the system and the corresponding function; The pop-up window calls the function giveClue() to show the next target that user should look for and whether the user has chosen the right one or not.

3. **Recapture button**: The focus frame is deigned to be a button as well, which has both the functions of alignment and recapturing the target. When it is pressed, the corresponding reRecognize() function will be called to run viewWillDisappear() to make a pause then run the viewWillAppear() to go on detecting.

Functional Requirements

Image Recognition

- The front camera should have the function to scan the map and do the image recognition.
- The back camera should identify the target quickly and correctly.
- If the wrong target has been captured, a warning box should pop up.
- There must have a focus window at the camera page, which help to increase the recognition accuracy.
- The front cannot be used among the processing of the app.

Game mode

- While the right target has been scanned, the score on the score board will be increased by 1.
- While the wrong target has been scanned, the score on the scoreboard does not change.

Interface adapter

- When the user rotates the device, the interface rotates accordingly to fit the device and the app still work properly.

Model information page

- While the right target has been scanned, the target information page will pop up to show the target information.
- While the wrong target has been scanned, the target information box will remain a constant.

Button functions
- While the “Plus” button is pressed, the presented model will be zoomed in.
- While the “Minus” button is pressed, the presented model will be zoomed out.
- While the “Refresh” button is pressed, which mean the user is ready to scan the next target. The page will be refreshed, and the presented model will be removed.

Non-functional Requirements

Usability
- The language of the app must be English.
- The app must have a refreshing interface that is as quick as possible.
- The app must be able to give a high-level utilisation experience for users, which include ease of use and maintenance.

Efficiency – Performance
- The camera must respond to the users’ feedback less than 1 second while they open the app.
- The game mode must strictly follow the rules as the developers requires.
- After the user clicks on the “Plus” button, the presented model must be zoomed in less than 2 seconds.
- After the user clicks on the “Minus” button, the presented model must be zoomed out less than 2 seconds.
- While the target in map is captured into the focus window, the corresponding model must be popped up less than 2 seconds.
- The recognition accuracy rate should be less than two errors in the process of 20 times recognition.
- While the user clicks on the “Search” option, the model information popup should appear in two seconds.
- While the user clicks on the “Refresh” option, the page should be refreshed less than 1 second.

Efficiency – Space
- Adding extra functions into the game must require minimum changes to the whole function logical code.

Dependability – Reliability
- The app must not crash more than one time after doing 20 game rounds.

Scientific Method

1. Question generation
    a. What is AR technology?
    b. How to implement AR in this project?
    c. How to do the galaxy model simulation?
    d. What matter does a galaxy will contain?
2. Background research
    a. We do the fundamental background research on the cosmological simulation and related technology.
3. Construct a hypothesis
    a. At the beginning, we built a simple version app with Swift, which only brought the function on image detection and recognition.
4. Test hypothesis by doing experiment (Testing)
    a. In accordance of the given galaxy map, the recognition function worked but the accuracy was quite low.

    b. Testing data gathered.
5. Refine, alter and expand software functions
    a. Increase the customer requirements.
    b. Add corresponding functions to improve the utilisation experience of the software.
    c. Function and performance testing
6. Data analyse and conclusion drawn
7. Report results