

# Implementing a Cosmological Simulation into Augmented Reality Using ARKit

Zibo Wang, Jinjie Huang, Wei Zeng, Siyi Han, Project Supervisor: Prof. Carlton Baugh

## Introduction

In 2016, Durham University's Institute of Computational Cosmology successfully created the Galaxy Makers exhibit to communicate our computational cosmology and astronomy research. We developed an app to enable students to view these AR galaxy model via the AR technology.

This project is prepared for the Royal Society Summer Science Exhibition 2020 and furtherly developed in an AR form.

## Aims

This project aims to develop an AR cosmology app, which will be widely used in the event of a cosmology exhibition, motivate the young generation interest in cosmic knowledge.

The app enables students to view these AR galaxy model via the AR technology. They can play the game and unconsciously learn the knowledge about the galaxy model. Concretely speaking, the initial aim for our project is to help people know the models in the galaxy map below.

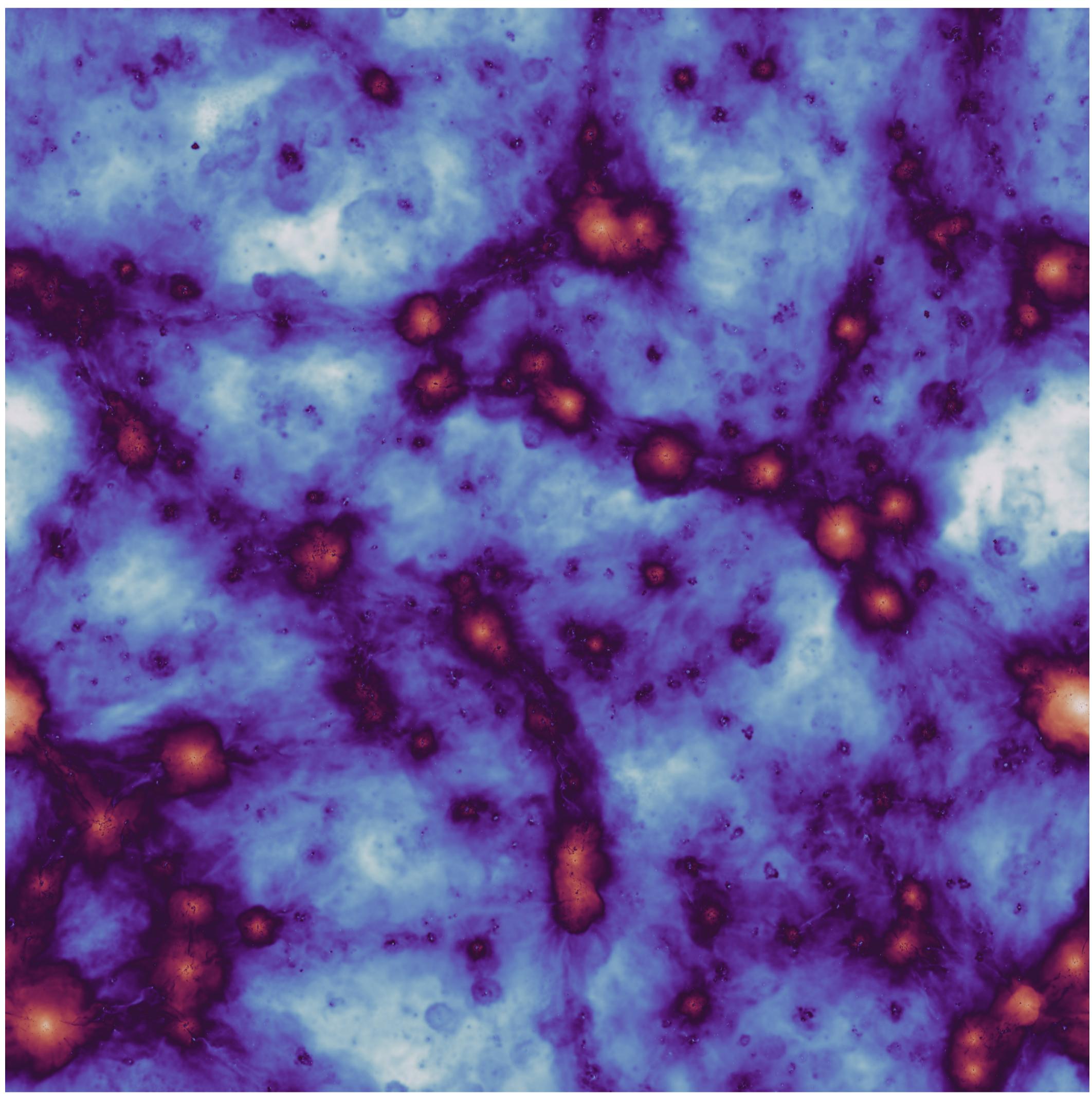


Figure 1: the specified galaxy simulation map

\* The app will be pre-installed on the provided devices

## Instruction for use

### Main interface illustration

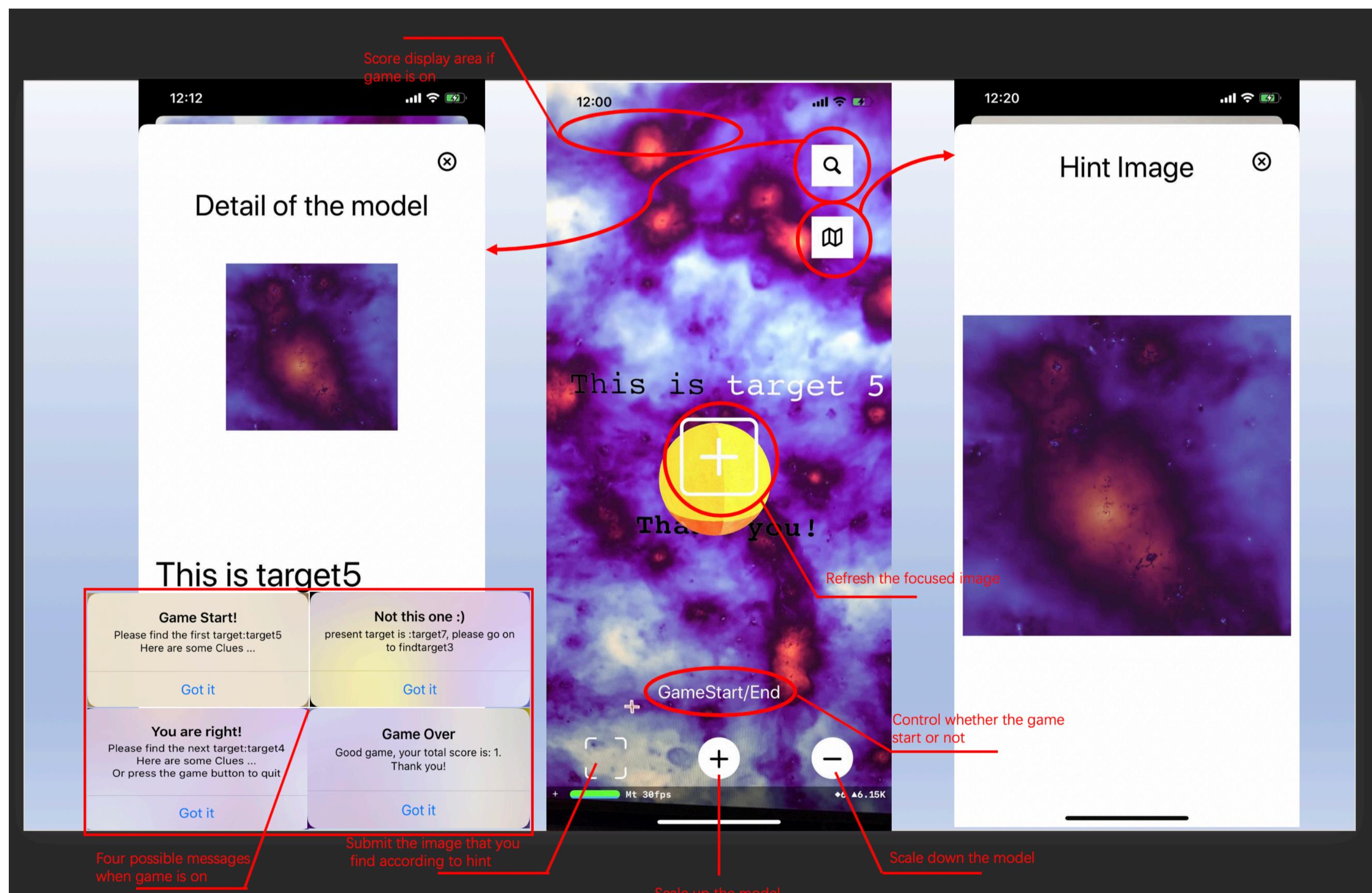


Figure 2: Main theme & detail window

1. There is a camera view page directly shown on the main page.
2. 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.
3. 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.
4. In the case of the AR model, by clicking the search button on the upper right, an up slide window will show the model details.

## Development Materials

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

## Functions Work Flow

### Targets displaying:

- Targets recognition. The ARScneneView is running in real-time form and recognizing the image in the viewing frame. Whenever it successfully get the target and its corresponding anchor name, renderer() function is called to render and display the model in augmented reality. When the user presses the detail button at the top right of the interface
- Action function. Whenever a target is recognized, it will be added a series of activities to the displayed model (We set it up as a spinning sphere by default). After pressing the scale-up/scale-down button (at the bottom of the interface). Function spawningmode() is called to add the function to the rendered model.
- Display function. After the target is recognized, there is a button on the upright, and when the button is pressed, the information of the target will be displayed on the screen.

### Game system:

- Game initialization. Game start/end button (in main interface) and the corresponding button function will give the first target to find and set the score to zero.
- 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. On the up right, there is a button connecting to hint image and the next image we need to find will show on the screen.
- Recapture button. The focus frame is designed 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.

## Results

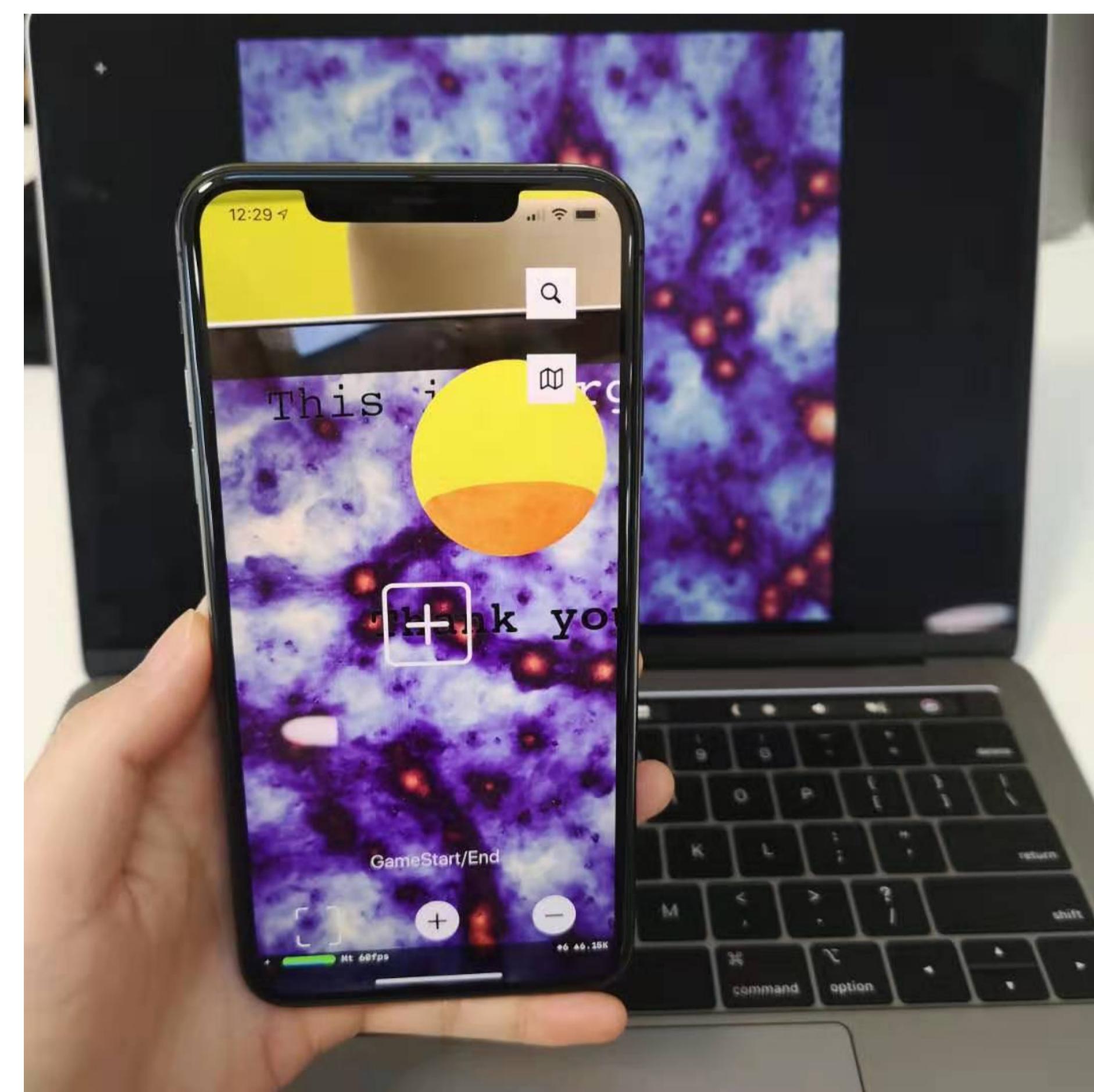


Figure 3: Result map

### Testing Results

Number	Test steps	Test results
1.	Running multiple apps at the same time, such as playing music.	Multiple apps running at the same time will not affect normal functions.
2.	Switch between front and background when the app is running.	Does not affect normal function.
3.	Call and send a message while the app is running.	The app can work correctly.
4.	Plug and unplug the charger while the app is running.	It will not fall back.
5.	Switch from landscape to portrait	The app can work correctly.