Florida International University
School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

TAM - multi-Touch, mid-Air, and Motion for Virtual and Augmented Reality Version 6.0

Learning Data Structures with Augmented Reality for Computer Science Education (K-12)

Team Members: Tom Lam, Ramses Sanchez

Product Owner(s): Dr. Francisco Ortega

Mentor(s): Dr. Francisco Ortega

Instructor: Masoud Sadjadi

Abstract

This document is a compilation of work done for Dr. Francisco Ortega's VIP project TAM - multi-Touch, mid-Air, and Motion for Virtual and Augmented Reality Version 6.0: Learning Data Structures with Augmented Reality for Computer Science Education (K-12) using Agile-driven development. It provides a detailed plan of execution that includes hardware and software specifications, System design methodology and specifications, a detailed catalog of the Sprints undertaken to execute the features documented herein, a glossary of relevant terms, and finally a series of appendices that document UML diagrams, class interactions, screenshots and progress reports.

Table of Contents

Introduction

The use of augmented reality to assist in computer science education provides many benefits such as increased motivation and a better overall learning experience. This allows complicated topics to be understood in a much more efficient and simple manner. With the current hardware available, it is possible to use cheaper alternatives such as a tablet to express topics too complicated or too time consuming, rather than create a new specialized hardware platform specifically for augmented reality education. This will allow future developers and educators to freely use the platform and application to suit their needs.

Current System

There currently is no system prior to the creation of the version stated in this document; this will be the first iteration of the application, designated as “Learning Data Structures with Augmented Reality for Computer Science Education”.

Purpose of New System

Since there is no system prior to the creation of the version stated in this document, the purpose of the new system is to create a foundation for the application and to allow the application to be modular enough so that it can be modified and expanded upon in future updates. With this foundation level implementation, the user should be able to select a simple data structure and be able to view it in real-time through the device as a 3D object in the 3D virtual environment.

User Stories

The following section provides the detailed user stories that were implemented in this iteration of the "TAM - multi-Touch, mid-Air, and Motion: Learning Data Structures with Augmented Reality for Computer Science Education" project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

Sprint 1
(1/16/17 - 1/27/17)

User Story #1158 - Unreal Engine 4 with Augmented Reality

- As a Developer, I want to be able to learn about Unreal Engine 4 so that I can prototype an array data-structure model in Augmented Reality.

User Story #1159 - Learning Unity Game Engine with Augmented Reality

- As a Developer, I want to be able to learn about the Unity Game Engine so that I can prototype an array data-structure model in Augmented Reality.

User Story #1160 - Learning Unity Game Engine - ARToolKit

- As a Developer, I want to learn the ARToolKit plugin so that I am able to model an data structure in 3D-space.

User Story #1161 - Learning Unity Game Engine - ARToolKit

- As a Developer, I want to learn the ARToolKit plugin so that I am able to model an data structure in 3D-space.

Sprint 2
(1/30/17 - 2/10/17)

User Story #1162 - Create an AR visualization/demo

- As a User, I want to be able to point my camera at a AR-marker an see a visualization.

User Story #1163 - Create Fiducial Markers

- As a Developer, I want to be able to learn to design and implement fiducial markers for future projects.

User Story #1164 - Create a Multiple Marker Visualization

- As a Developer, I want to learn how to use and then implement multiple markers to visualize a complex AR environment.

Sprint 3
(2/13/17 - 2/24/17)

User Story #1182 - Research into Memory Retention

- As a Researcher, I want to find out if interactive systems lead to better memory retention, so that, I can determine whether our interactive application can aid in the learning of Computer Science data structure concepts as effectively as possible.

User Story #1183 - Design and Draft Research Study/Experiment

- As a Researcher, I want to design a research experiment that studies the memory retention of people before using our interactive software and after, so that we can gather statistics on the effectiveness of our interactive application.

User Story #1184 - Vuforia plugin Animation rendering

- As a Developer, I want to see the Vuforia plugin's Animations rendering capabilities so that I can determine its viability with respects to our AR project.

User Story #1185 - Research topics on Augmented Reality and Education

- As a researcher, I want to learn about topics such as augmented reality and education so I can better design an education product using augmented reality to learn.

User Story #1186 - Unity Game Engine with Vuforia

- As a developer, I want to learn to use Vuforia in Unity to create a simple demo for modeling a 3D object.

Sprint 4
(2/27/17 - 3/10/17)

User Story #1209 - Unity Game Engine Textures Learning

- As a Developer, I want to learn and create a prototype so that the 3D object has some kind of unique look and feel.

User Story #1213 - Prototype UI

- As a Developer, I want to create a prototype of our application's UI so that I can have an idea what the final UI could be.

User Story #1215 - Obtain Initial FIU IRB Approval

- As a Research, I want to obtain FIU IRB approval so as to be able to implement the proposed research proposal (see user story #1183).

User Story #1216 - Complete IRB Training Requirements

- As a Researcher, I want to complete all IRB training requirements to be able to execute my experiment.

User Story #1229 - Unity Game Engine Texture Creation

- As a Developer, I want to design and then create textures so that the 3D object has some kind of unique look and feel.

User Story #1240 - Unity/Vuforia Windows & Android Demo Deployment

- As a Developer, I want to deploy my demo application to a Windows Store and Android platform to ensure ease of deployment towards the end of the release cycle.

Sprint 5
(3/13/17 - 3/24/17)

User Story #1214 - Implement Splash Page GUI

- As a Developer, I want to be able to implement a Splash (or welcome page) for our application.

User Story #1248 - Compile research paper for First Draft

- As a Researcher, I want to take all the research papers I have previously found, including my teammates, and incorporate them into some kind of first draft.

User Story #1249 - Learning Animations in Unity Game Engine

- As a Developer, I want to learn to create some animations that allow some kind of transition between scenes/phases.

Sprint 6
(3/27/17 - 4/7/17)

User Story #1265 - Implement execution/loading page GUI

- As a Developer, I want to be able to implement an execution/loading for our application.

User Story #1266 - Implement Exit Page GUI

- As a Developer, I want to be able to implement an exit for our application.

User Story #1267 - Implement Options Page GUI

- As a Developer, I want to be able to implement an options page for our application.

User Story #1268 - Implement Front Page GUI

- As a Developer, I want to be able to implement a front page for our application.

User Story #1244 - Review Research and Create a summary

- As a Researcher, I want to come together with my partner, Tom, to discuss what he has researched, what I have research and catalog our research ideas to begin doing the Introduction section for our Senior Project research component.

User Story #1256 - Begin Introduction section (first draft) for Research Paper

- As a Researcher, I want to compile all the research papers proposed by Dr. Ortega, Tom, and I to incorporate them into the Introduction section of our research paper so as to begin Draft 1.

User Story #1290 - Unity Game Engine Animations

- As a Developer, I want to create animations for our design.

User Story #1295 - Compile a Second Draft for Research Paper

- As a Researcher, I want to compile a second draft for the research paper.

Pending User Stories

None.

Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

General Development Environment for Unity 5.6 and Vuforia 6.2:

- Windows 7 SP1 or newer editions / Mac OS X 10.8 or newer editions.
- Graphics Processing Unit (GPU) - DX9 capable with a shader model 3.0 or DX11 with feature level 9.3 capabilities.

All other recommendations below are dependent on the complexity of the project:

- iOS: OS X 10.9.4 or newer and Xcode 7.0 or newer.
- Android: Android SDK and Java Development Kit (JDK).
- Windows Store: Windows 8.1 (64-bit) and the appropriate Visual Studios version
    - Windows 8.1/ Windows Phone 8.1 - Visual Studios 2013 or newer and the Windows 8.1 SDK.
    - Universal Windows Platform (UWP) - Visual Studio 2015 or newer and Windows 10 SDK.
    - IL2CPP scripting backend also requires C++ compiler feature to be installed with Visual Studios.
- WebGL: Mac OS X 10.8 or newer / Windows 7 SP1 (64-bit editor only).

General Deployment Environment for Unity 5.6 and Vuforia 6.2:

- Desktop:
    - OS: Windows XP SP2 or newer, Mac OS X 10.8 or newer, Ubuntu 10.04 or newer, SteamOS.
    - Graphics Card: DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.
    - CPU: SSE2 instruction set support.
- iOS player requires iOS 7.0 or newer.
- Android: 4.1 or newer, ARMv7 (Cortex) CPU with NEON support or Atom CPU, OpenGL ES 2.0 or newer.
- WebGL: Any recent version of Firefox, Chrome, Edge, Safari.
- Windows Phone: 8.1 or newer.
- Windows Store Apps: 8.1 or newer.

We provide no hard drive space and memory recommendations because they are project dependent. Low-complexity projects will not require powerful hardware to operate while high-complexity projects will require a more powerful machine to run, consuming more resources.

This particular application was built and operated on a Microsoft Surface Pro 4:

- Intel Core i5-6300U.
- 8 GB of RAM.
- Samsung NVMe MZFLV256 SSD (256GB).
- Intel HD Graphics 520.
- 5.0 MP 1080p front-camera, 8.0 MP 1080p rear-camera.

All hardware and software specifications are gathered directly from the hardware/software websites respectively.

Sprint Plans

Sprint 1
(1/16/17 - 1/27/17)

User Story #1158 - Unreal Engine 4 with Augmented Reality

- As a Developer, I want to be able to learn about Unreal Engine 4 so that I can prototype an array data-structure model in Augmented Reality.

Acceptance Criteria:

1. Download Unreal Engine 4 https://www.unrealengine.com/.
2. Download AUGMENTED REALITY PLUGIN FOR UE4 http://www.unreal4ar.com/product/augmented-reality-plugin-for-ue4/.
3. Followed this Tutorial https://www.raywenderlich.com/151018/unreal-engine-4-tutorial-beginners.

Related Tasks/Stories:

None.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1159 - Learning Unity Game Engine with Augmented Reality

- As a Developer, I want to be able to learn about the Unity Game Engine so that I can prototype an array data-structure model in Augmented Reality.

Acceptance Criteria:

1. Installed from https://store.unity.com/products/unity-personal.

Related Tasks/Stories:

None.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1160 - Learning Unity Game Engine - ARToolKit

- As a Developer, I want to learn the ARToolKit plugin so that I am able to model an data structure in 3D-space.

Acceptance Criteria:

1. Import package into Unity.
2. Read the Documentation: http://artoolkit.org/documentation/.

Related Tasks/Stories:

None.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1161 - Learning Unity Game Engine - ARToolKit

- As a Developer, I want to learn the ARToolKit plugin so that I am able to model an data structure in 3D-space.

Acceptance Criteria:

1. Import package into Unity.
2. Read the Documentation: http://artoolkit.org/documentation/.

Related Tasks/Stories:

None.

Modeling:

Not applicable.

Testing:

Not applicable.

Sprint 2
(1/30/17 - 2/10/17)

User Story #1162 - Create an AR visualization/demo

- As a User, I want to be able to point my camera at a AR-marker an see a visualization.

Acceptance Criteria:

1. Place an AR Marker on a flat surface.
2. Point AR-capable device to (1).
3. Visualization manifests.

Related Tasks/Stories:

None.

Modeling:

See Figure 1, Appendix A.

Testing:

Not applicable.

User Story #1163 - Create Fiducial Markers

- As a Developer, I want to be able to learn to design and implement fiducial markers for future projects.

Acceptance Criteria:

1. Markers are square.
2. Recognized by Unity.

Related Tasks/Stories:

None.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1164 - Create a Multiple Marker Visualization

- As a Developer, I want to learn how to use and then implement multiple markers to visualize a complex AR environment.

Acceptance Criteria:

1. Use of multiple markers to visualize some kind of 3D object.
2. Create new unique markers only for multiple marker usage.

Related Tasks/Stories:

None.

Modeling:

See Figure 2, 5, and 6, Appendix A.

Testing:

Not applicable.

Sprint 3
(2/13/17 - 2/24/17)

User Story #1182 - Research into Memory Retention

- As a Researcher, I want to find out if interactive systems lead to better memory retention, so that, I can determine whether our interactive application can aid in the learning of Computer Science data structure concepts as effectively as possible.

Acceptance Criteria:

1. Must reference the Academic literature on (1) Interactive Software and (2) Memory Retention techniques (with respects to (1)).
2. Sprint review video https://www.youtube.com/watch?v=oSIPtFZI6G8&feature=youtu.be.
3. Product Owner's approval is a MUST.

Related Tasks/Stories:

1196 Incorporate found research into Sprint 3 Review Video.

1195 Document relevant research to reference later.

1194 Search for relevant research papers.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1183 - Design and Draft Research Study/Experiment

- As a Researcher, I want to design a research experiment that studies the memory retention of people before using our interactive software and after, so that we can gather statistics on the effectiveness of our interactive application.

Acceptance Criteria:

1. Must complete Story #1182.
2. Design a study to measure memory retention, after having conducted research on the matter.
3. Provide a draft of the experiment to our Product Owner.
4. If accepted, mark story as 'Done' if rejected, improve Study and mark as 'In Development'.
5. Sprint review video https://www.youtube.com/watch?v=oSIPtFZI6G8&feature=youtu.be

Related Tasks/Stories:

User Story #1182.

1206 Document proposal into Sprint 3 Review Video.

1202 Submit Proposal for Review.

1192 Create a Research Proposal.

Testing:

Not applicable.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1184 - Vuforia plugin Animation rendering

- As a Developer, I want to see the Vuforia plugin's Animations rendering capabilities so that I can determine its viability with respects to our AR project.

Acceptance Criteria:

1. Animation must add, and not detract, to our visualizations.
    1. Should be computationally cheap (hardware compliant).
    2. Must be accessible (easy to use).
2. Animation should not take a long time to render/execute.
3. Sprint review video https://www.youtube.com/watch?v=oSIPtFZI6G8&feature=youtu.be.

Related Tasks/Stories:

1204 Incorporate Animation Intro Sprint 3 Review Video.

1203 Follow Tutorial.

Testing:

Not applicable.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1185 - Research topics on Augmented Reality and Education

- As a researcher, I want to learn about topics such as augmented reality and education so I can better design an education product using augmented reality to learn.

Acceptance Criteria:

1. Are there certain methods of teaching/educating a student that shows higher results of retention of knowledge as well as understanding the topics better than others? If so, what are they?

2. Are there certain methods of teaching/educating a student that should be avoided due to little or no retention of knowledge as well as a lack of understanding topics? If so, what are they?

Related Tasks/Stories:

1199 Incorporate the documented research in Sprint 3 Review Video.

1198 Document the research for future reference.

1197 Find relevant research papers.

Testing:

Not applicable.

Modeling:

See Figure 7, Appendix A.

Testing:

Not applicable.

User Story #1186 - Unity Game Engine with Vuforia

- As a developer, I want to learn to use Vuforia in Unity to create a simple demo for modeling a 3D object.

Acceptance Criteria:

1. Should be similar to #1164.
2. Install from https://www.vuforia.com/.
3. Must including training custom markers.

Related Tasks/Stories:

User Story #1164.

1245 Test to make sure there are no bugs or errors.

1201 Present the demo in the Sprint 3 Review Video.

1200 Document the features in the demo.

Modeling:

See Figure 8 and 9, Appendix A.

Testing:

Not applicable.

Sprint 4
(2/27/17 - 3/10/17)

User Story #1209 - Unity Game Engine Textures Learning

- As a Developer, I want to learn and create a prototype so that the 3D object has some kind of unique look and feel.

Acceptance Criteria:

1. Prototype should be simple and not too complicated.
2. Textures must be able to be separated from the prototype as a modular unit so that it can be imported over to other projects to be used.

Related Tasks/Stories:

1232 Present the textures in the Sprint 4 Review video.

1231 Document the features in the demo.

1230 Make sure that the textures work regardless of the size of the object.

Modeling:

See Figures 10, Appendix A.

Testing:

Not applicable.

User Story #1213 - Prototype UI

- As a Developer, I want to create a prototype of our application's UI so that I can have an idea what the final UI could be.

Acceptance Criteria:

1. Envision what the end-product UI and GUI looks like by providing:
    - Provide sketches of:
        1. splash page.
        2. front page.
        3. options page.
        4. execution page (or main page).
        5. exit page.
2. Provide a sketch of all possible page transitions from every page.
3. https://youtu.be/xsRUVAmVqjs.

Related Tasks/Stories:

1222 Incorporate work performed into Sprint 4 Review Video.

1221 Prototype exit page.

1220 Prototype execution/loading page.

1219 Prototype options page.

1218 Prototype front page.

1217 Prototype splash page.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1215 - Obtain Initial FIU IRB Approval

- As a Research, I want to obtain FIU IRB approval so as to be able to implement the proposed research proposal (see user story #1183).

Acceptance Criteria:

1. Complete all the steps at http://research.fiu.edu/irb/obtaining-approval/.
2. https://youtu.be/xsRUVAmVqjs.

Related Tasks/Stories:

None.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1216 - Complete IRB Training Requirements

- As a Researcher, I want to complete all IRB training requirements to be able to execute my experiment.

Acceptance Criteria:

1. Complete all the steps at http://research.fiu.edu/irb/obtaining-approval/.
2. https://youtu.be/xsRUVAmVqjs.

Related Tasks/Stories:

None.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1229 - Unity Game Engine Texture Creation

- As a Developer, I want to design and then create textures so that the 3D object has some kind of unique look and feel.

Acceptance Criteria:

1. Must not generic/default textures (grey).
2. Must be easily viewable in any environment and lighting regardless of background.
3. Must be simple and not difficult to look at or distracting.

Related Tasks/Stories:

1246 Test to make sure there are no bugs or errors.

1236 Present the textures in the Sprint 4 Review Video.

1235 Document the features in the demo.

1234 Make sure that the textures work regardless of the size of the object.

Modeling:

See Figure 11, Appendix A.

Testing:

Not applicable.

User Story #1240 - Unity/Vuforia Windows & Android Demo Deployment

- As a Developer, I want to deploy my demo application to a Windows Store and Android platform to ensure ease of deployment towards the end of the release cycle.

Acceptance Criteria:

1. Runs on Android and Windows platforms.
2. Software requirements: Windows SDK, Android SDK, Java SDK, Visual Studios, etc.

Related Tasks/Stories:

1247 Test to make sure there are no bugs or errors.

1243 Present in Sprint 4 Review video.

1242 Document the features in the demo.

1241 When launched, there are no errors preventing it from functioning.

Modeling:

Not applicable.

Testing:

Not applicable.

Sprint 5
(3/13/17 - 3/24/17)

User Story #1214 - Implement Splash Page GUI

- As a Developer, I want to be able to implement a Splash (or welcome page) for our application.

Acceptance Criteria:

1. Finished User Story #1213.
2. The GUI should be consistent in its theme and implementation.
3. The GUI must be thoroughly tested (see tasks below).
4. https://youtu.be/mB3Bn0HZgdw.

Related Tasks/Stories:

1264 Capture progress made in Sprint 5 Review Video.

1263 Document the feature.

1262 Test the feature for errors and bugs.

Modeling:

See Figure 3 and 4 Appendix A.

Testing:

Not applicable.

User Story #1248 - Compile research paper for First Draft

- As a Researcher, I want to take all the research papers I have previously found, including my teammates, and incorporate them into some kind of first draft.

Acceptance Criteria:

1. Must be approved by team mates.
2. Must be use relevant papers; discard the ones that are not.
3. Must be approved by Product Owner.

Related Tasks/Stories:

1255 Have paper approved by the Product Owner.

1251 Create video explaining what as included in the paper as well as the thought process to writing the paper.

1250 Document in demo.

Modeling:

See Figure 12, Appendix A.

Testing:

Not applicable.

User Story #1249 - Learning Animations in Unity Game Engine

- As a Developer, I want to learn to create some animations that allow some kind of transition between scenes/phases.

Acceptance Criteria:

1. Must not be too distracting.
2. Must be simple and pleasing to the viewer.
3. Must be modular to be imported or easily recreated.

Related Tasks/Stories:

1254 Create a preview of the feature in a video to showcase.

1253 Document the features in the demo.

1252 Test to make sure there are no bugs or errors.

Modeling:

See Figure 12 and 13, Appendix A.

Testing:

Not applicable.

Sprint 6
(3/27/17 - 4/7/17)

User Story #1265 - Implement execution/loading page GUI

- As a Developer, I want to be able to implement an execution/loading for our application.

Acceptance Criteria:

1. Finished User Story #1213.
2. The GUI should be consistent in its theme and implementation.
3. The GUI must be thoroughly tested (see tasks below).

Related Tasks/Stories:

None.

Modeling:

See Figure 15 and 16, Appendix A.

Testing:

See UT1265-1 and See UT1265-2.

User Story #1266 - Implement Exit Page GUI

- As a Developer, I want to be able to implement an exit for our application.

Acceptance Criteria:

1. Finished User Story #1213.
2. The GUI should be consistent in its theme and implementation.
3. The GUI must be thoroughly tested (see tasks below).

Related Tasks/Stories:

1274 Create a preview of the feature in a video to showcase.

1273 Document the features in the demo.

1272 Test to make sure there are no bugs or errors.

Modeling:

See Figure 17 and 18, Appendix A.

Testing:

See UT1266-1 and See UT1266-2.

User Story #1267 - Implement Options Page GUI

- As a Developer, I want to be able to implement an options page for our application.

Acceptance Criteria:

1. Finished User Story #1213.
2. The GUI should be consistent in its theme and implementation.
3. The GUI must be thoroughly tested (see tasks below).

Related Tasks/Stories:

1280 Create a preview of the feature in a video to showcase.

1279 Document the features in the demo.

1278 Test to make sure there are no bugs or errors.

Modeling:

See Figure 19, and 20, Appendix A.

Testing:

See UT1267-1.

User Story #1268 - Implement Front Page GUI

- As a Developer, I want to be able to implement a front page for our application.

Acceptance Criteria:

1. Finished User Story #1213.
2. The GUI should be consistent in its theme and implementation.
3. The GUI must be thoroughly tested (see tasks below).

Related Tasks/Stories:

1277 Create a preview of the feature in a video to showcase.

1276 Document the features in the demo.

1275 Test to make sure there are no bugs or errors.

Modeling:

See Figure 21, 22, and 23 Appendix A.

Testing:

UT1268-1, UT1268-2, UT1268-3, and UT1268-4

User Story #1244 - Review Research and Create a summary

- As a Researcher, I want to come together with my partner, Tom, to discuss what he has researched, what I have research and catalog our research ideas to begin doing the Introduction section for our Senior Project research component.

Acceptance Criteria:

1. Read and summarize partner's research.
2. Read and summarize own research.
3. Meet with partner to create a collaborative summary (of our research) and discuss the direction of our research paper.

Related Tasks/Stories:

1259 Capture progress made in Sprint 5 Review Video.

1258 Summarize relevant research.

1257 Go through and summarize the research papers proposed by Dr. Ortega and Tom.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1256 - Begin Introduction section (first draft) for Research Paper

- As a Researcher, I want to compile all the research papers proposed by Dr. Ortega, Tom, and I to incorporate them into the Introduction section of our research paper so as to begin Draft 1.

Acceptance Criteria:

1. Must finish User Story ##1244 first.
2. Must be approved by Product Owner and Tom.
3. Must use all (or almost all) of the papers provided; the introduction is where a survey of the literature is done.

Related Tasks/Stories:

1261 Capture progress made in Sprint 5 Review Video.

1260 Must be reviewed and approved by Dr. Ortega.

Modeling:

Not applicable.

Testing:

Not applicable.

User Story #1290 - Unity Game Engine Animations

- As a Developer, I want to create animations for our design.

Acceptance Criteria:

1. Must be simple and not distracting.
2. Portable; otherwise, easily recreated.

Related Tasks/Stories:

1293 Present in video demo.

1292 Document the features in the demo.

1291 Test to make sure there are no bugs or errors.

Modeling:

See Figure 14, Appendix A.

Testing:

Not applicable.

User Story #1295 - Compile a Second Draft for Research Paper

- As a Researcher, I want to compile a second draft for the research paper.

Acceptance Criteria:

1. Must be accepted by PO.
2. Must follow any rubric.

Related Tasks/Stories:

None.

Modeling:

Not applicable.

Testing:

Not applicable.

System Design

Because we are using Unity, our application essentially follows the compositional structure of a video game, since we are making an interactive application dependent on user input. Video game system design paradigm can thus be summarized into two camps (with respects to representing game entities which are essentially everything our application, and video games, are composed of): (i) a deep-class hierarchy or (ii) game entity objects composed of aggregate components.

Some thoughts on why we did not choose (i).

> It is necessarily the case that deep class hierarchies, when the height of a class (or how many classes depend on it) is longer than its width (how many classes it defines), take an extra layer of complexity and abstraction to an already complicated problem, lengthen compile-time, increases coupling, and hamper testing and bug-fixing. In the gaming environment, while most assets are shared, they do not necessarily have to be coupled since said assets are primarily textures, sounds, animations, and scripts that string scenes together. The coding elements behind the controllers that manage such tasks can be just as effective without being interdependent (note this does not necessarily mean that code cannot be reused across classes).

We used the component approach to model our Object-Oriented Programming, which consists of separating the functionality into individual components that are mostly independent of one another. The traditional object hierarchy is dispensed with, and an object is now created as an aggregation (a collection) of independent components (within the context of our application, assets, scripts, and rigs are said components). Each object now only has the functionality that it needs. Any distinct new functionality is implemented by adding a component, each game entity is comprised of a collection of components; the components themselves can be treated as being independent of the objects they make up.

Architectural Patterns

Our software uses the Model-View-Controller (MVC) architectural pattern for implementing interactive user interfaces. The Model is the various database markers linked with specific data structures (in our case Augmented Reality Visualizations). The view is the specific scenes modeled in Unity, which forms the User Interface experience. And the Controller are the scripts, Device Target Database, Cloud Target Database, and other Logic controllers handled natively by Unity and Vuforia. Our application is rather simplistic, in that it only has 5 scenes each with its own Camera and Light Manager, Canvas, Event System Handler, UI Manager, and Audio Manager scripts and other related entities.

System and Subsystem Decomposition

Our application is thus divided into the following systems and subsystems.

Main Camera – the main device that captures and displays the world to the player. By customizing and manipulating cameras, the game is presented exactly as we desire. Although we can have unlimited numbers of cameras in a scene and they can be set to render in any order, at any place on the screen, or only certain parts of the screen, for the purposes of our application we only had one.

Light Inspector – an essential part of every scene; defines the color and mood of the 3D environment; fundamental part of graphical rendering since they determine the shading of an object and the shadows it casts

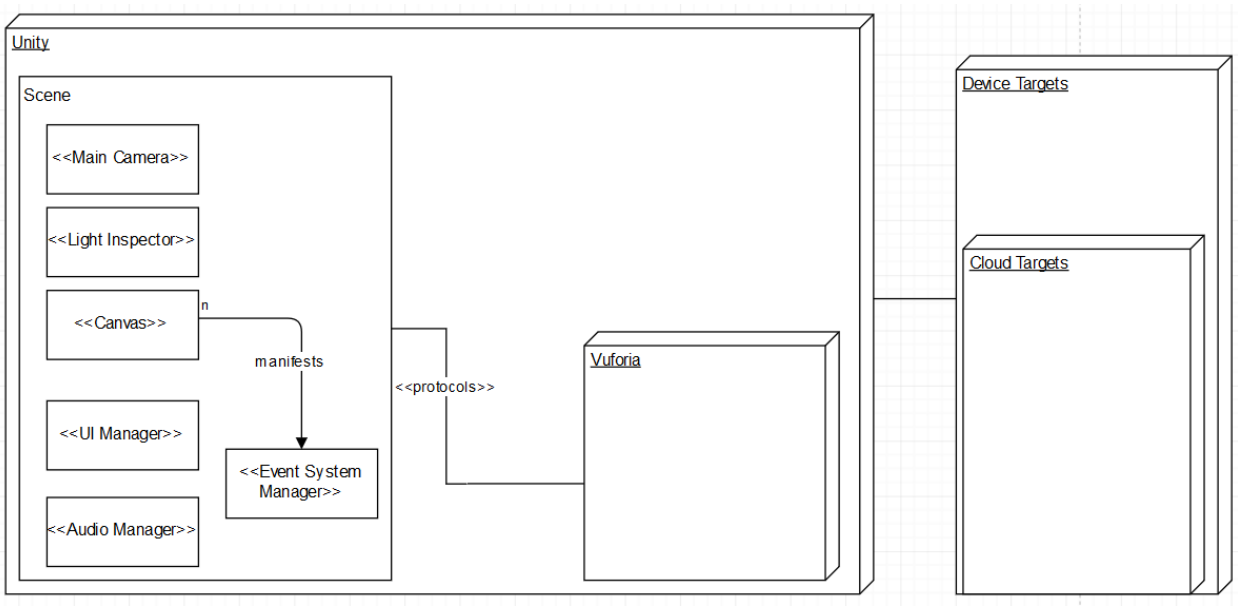
Canvas – is the area that all UI elements share. The Canvas is a Game Object with a Canvas component on it, and all UI elements are children of such a Canvas; canvas uses the EventSystem object to help the Messaging System.

Event System Manager – is responsible for controlling all the other elements that make up eventing. It coordinates which Input Module is currently active, which GameObject is currently considered 'selected', and a host of other high level Event System concepts.Each 'Update' the Event System receives the call, looks through its Input Modules and figures out which is the Input Module that should be used for this tick. It then delegates the processing to the modules.

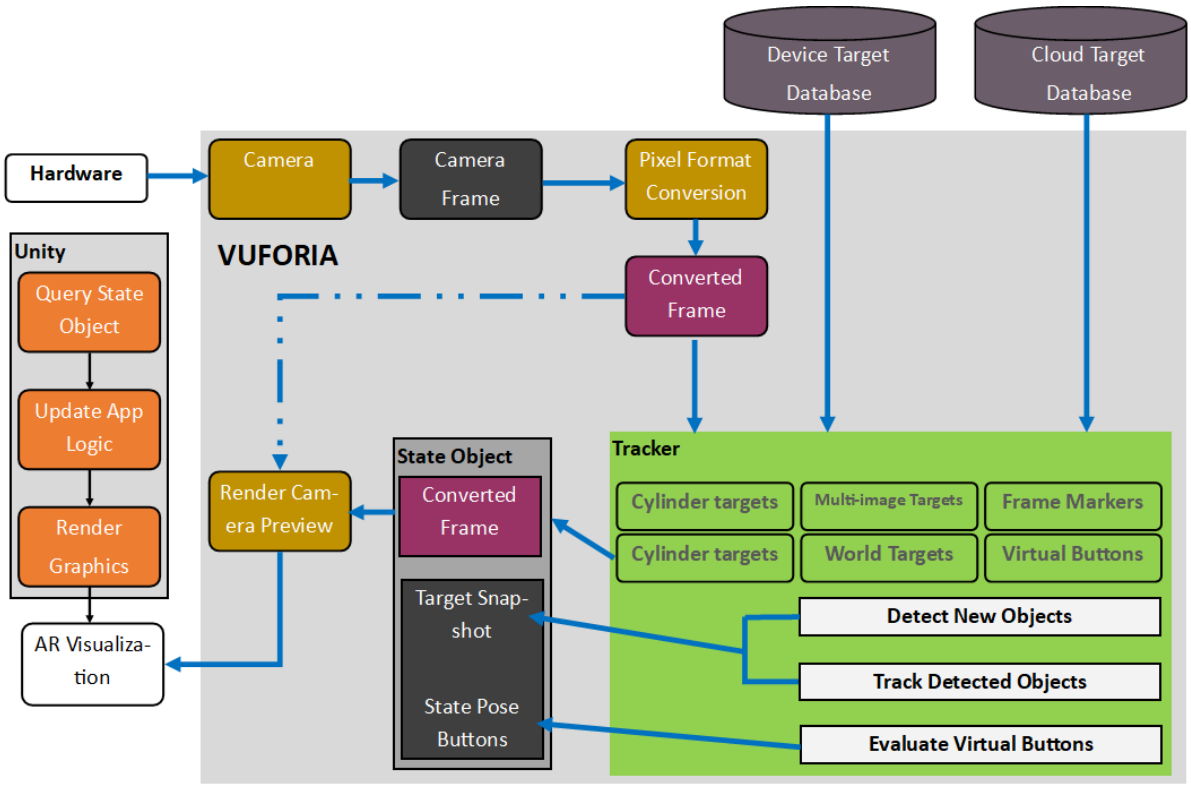
UIManager – A programmer defined scrip that handles Scenes and Canvas transitions in between launching and loading; sets up and controls every Canvas element.

Audio Manager – A programmer defined script responsible for controlling sound elements paired with GameObjects and Unity Scenes and keeping a Master-level Audio Mix throughout.

Deployment Diagram



Deployment Diagram



System Architecture Deployment Diagram

Design Patterns

The following design patterns were used:

- Adapter: Incorporated Vuforia library and wrapped it in a layer to cover existing Scenes and Scripts.
- Composite: Game assets (scripts, textures, game objects, audio, canvas UI elements) were aggregated together to make Scenes.
- State: Each Scene was linked together allowing for transitions between them.
- Builder: Modules were built using component-based entities (one component at a time) based on data.

System Validation

Unit testing is essential to test all the layers within the domain encompassing our application. All test cases were used to help validate features resulting from user stories.

ID	**UT1265-1**
Purpose	Sunny-day test to check whether there is a loading scene every time a scene changes to confirming that a scene-change took place.
Precondition	User must be at a static scene where no loading is taking place.
Input	User clicks on 'Start', 'Exit', 'Options', or 'Back' button.
Expected Output	Whenever there is a scene transition, there must be some feedback to the user that the scene is changing and that loading is taking place.
Status	PASS

ID	**UT1265-2**
Purpose	Rainy-day test to check that there is no unintended loading or scene change unless initiated by the user.
Precondition	User must be at a static scene where no loading is taking place.
Input	User clicks anywhere but the 'Start', 'Exit', 'Options', or 'Back' button.
Expected Output	No scene transition takes place since no loading has taken place.
Status	PASS

ID	**UT1266-1**
Purpose	Sunny-day test to check whether program exits when the 'Exit' button is clicked by the user.
Precondition	User must be at a page with an 'Exit' button.
Input	User clicks on the 'Exit' button.
Expected Output	Program closes down by exiting (screen changes to black and window minimizes and disappears – no memory traces are left of the program).
Status	PASS

ID	**UT1266-2**
Purpose	Rainy-day test to check that the program does not exit when buttons other than the 'Exit' button are clicked by the user.
Precondition	User must be at a page with an 'Exit' button.
Input	User clicks anywhere but on the 'Exit' button.
Expected Output	Program does not close down by exiting (screen changes to black and window minimizes and disappears – no memory traces are left of the program).
Status	PASS

ID	**UT1267-1**
Purpose	Sunny-day test to check that the program transitions to the Options page when the ‘Options’ button is clicked by the User.
Precondition	User must bet at a page with an ‘Options’ button.
Input	User clicks on the ‘Options’ button.
Expected Output	Program transitions from current scene to the Options page scene.
Status	PASS

ID	**UT1214-1**
Purpose	Sunny-day test to check that the program launches and that the Splash page manifests when the program initializes
Precondition	Program must be closed.
Input	User starts the program.
Expected Output	The program loads and the splash page manifests.
Status	PASS

ID	**UT1214-2**
Purpose	Rainy-day test to check that the program does not launch and that the Splash page does not manifests due to the program not initializing
Precondition	Program must be closed.
Input	User does not start the program.
Expected Output	The program does not load and the splash page does not manifest.
Status	PASS

ID	**UT1268-1**
Purpose	Sunny-day test to check that the program transitions from the Splash page to the Front/Menu page.
Precondition	User must be at the Splash page.
Input	User clicks on the 'Start' button.
Expected Output	The program transitions from the Splash page to the Front Page as denoted by the Front page having three buttons: Select Module, Options, Exit.
Status	PASS

ID	**UT1268-2**
Purpose	Rainy-day test to check that the program does not transition from the Splash page to the Front/Menu page.
Precondition	User must be at the Splash page.
Input	User does not click on the 'Start' button.
Expected Output	The program does not transition from the Splash page to the Front Page as denoted by the Front page having three buttons: Select Module, Options, Exit.
Status	PASS

ID	**UT1268-3**
Purpose	Sunny-day test to check that the program initializes a module.
Precondition	User must be at the Front page.
Input	User clicks on the 'Select Module' button. User clicks on the desired module.
Expected Output	The program loads desired module.
Status	PASS

ID	**UT1268-4**
Purpose	Rainy-day test to check that the program does not initialize a module.
Precondition	User must be at the Front page.
Input	User does not click on the 'Select Module' button.
Expected Output	The program does not load desired module.
Status	PASS

Glossary

AR: Augmented reality is a live direct or indirect view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even diminished rather than augmented) by a computer.

Fiducial marker: is an object placed in the field of view of an imaging system which appears in the image produced, for use as a point of reference or a measure

SDK: is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform

Unity: is a cross-platform game engine developed by Unity Technologies used to develop video games for PC, consoles, mobile devices and websites.

Visualization: An AR image displaying from a fiducial marker.

Vuforia: is an Augmented Reality Software Development Kit (SDK) for mobile devices that enables the creation of Augmented Reality applications.

Appendix

Appendix A – UML Diagrams

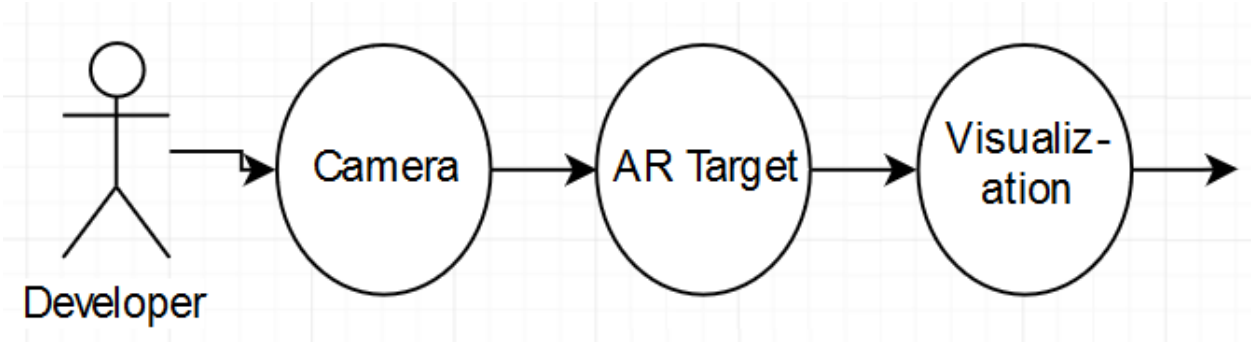


Figure 1

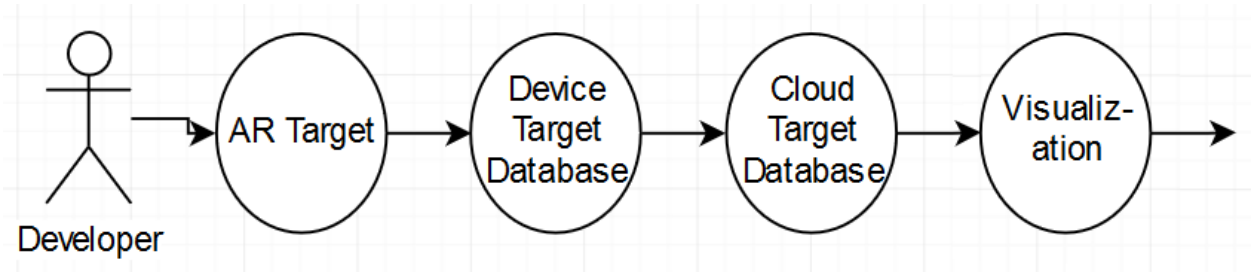


Figure 2

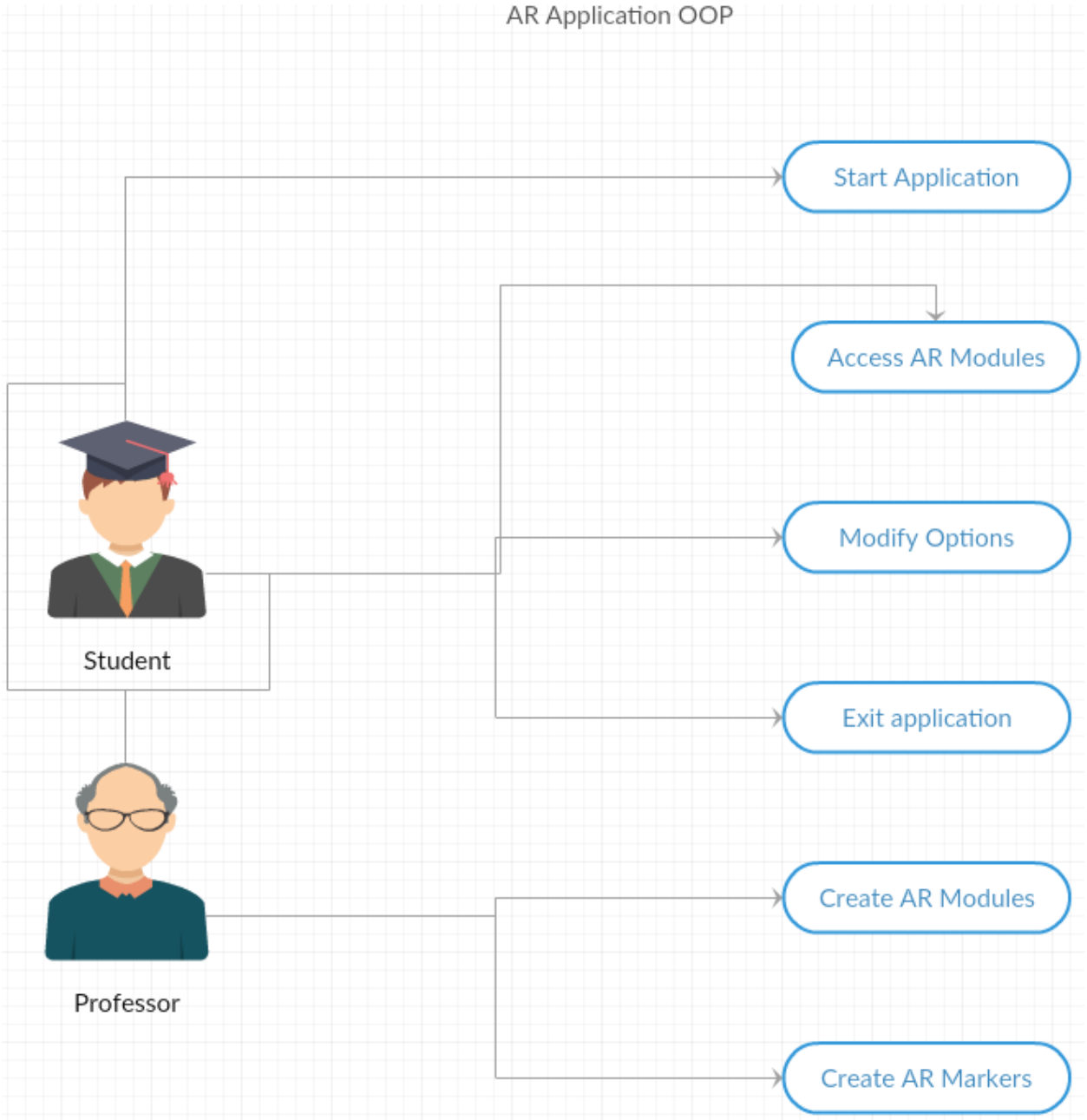

AR Application OOP
Start Application
Access AR Modules
Modify Options
Exit application
Student
Professor
Create AR Modules
Create AR Markers

Figure 3

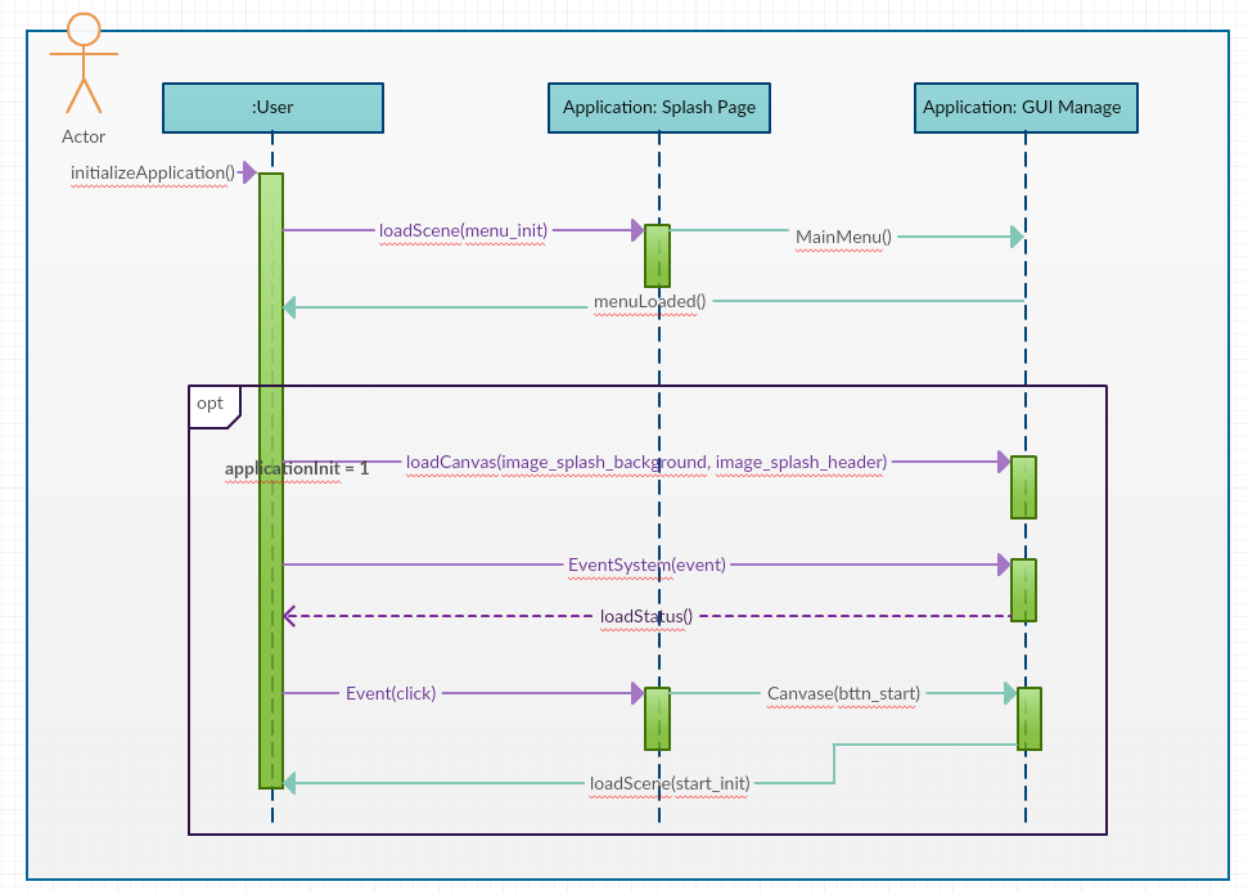


Figure 4

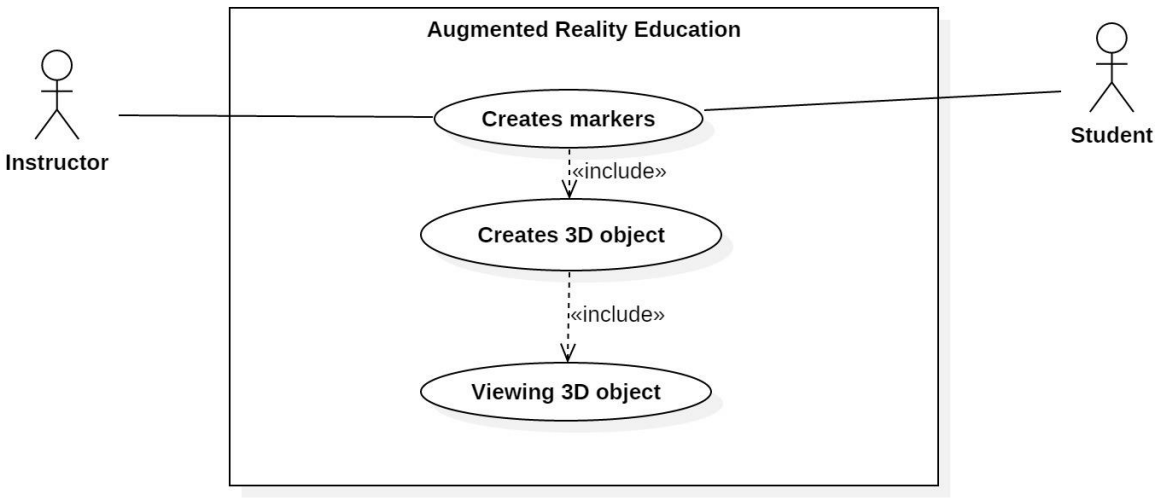


Figure 5

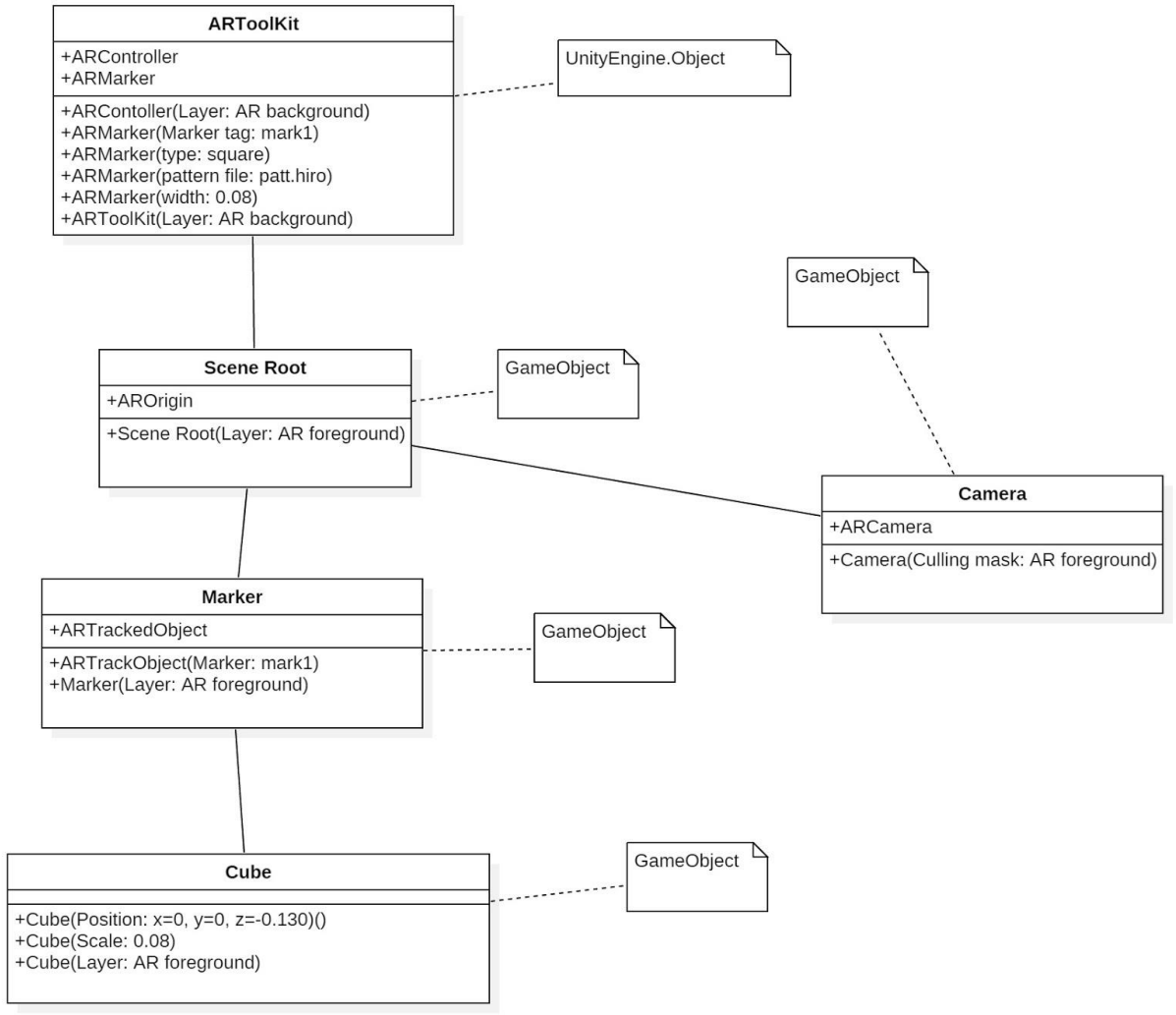

ARToolKit
+ARController
+ARMarker
+ARContoller(Layer: AR background)
+ARMarker(Marker tag: mark1)
+ARMarker(type: square)
+ARMarker(pattern file: patt.hiro)
+ARMarker(width: 0.08)
+ARToolKit(Layer: AR background)
UnityEngine.Object
GameObject
Scene Root
+AROrigin
+Scene Root(Layer: AR foreground)
GameObject
Camera
+ARCamera
+Camera(Culling mask: AR foreground)
Marker
+ARTrackedObject
+ARTrackObject(Marker: mark1)
+Marker(Layer: AR foreground)
GameObject
Cube
+Cube(Position: x=0, y=0, z=-0.130)()
+Cube(Scale: 0.08)
+Cube(Layer: AR foreground)
GameObject

Figure 6

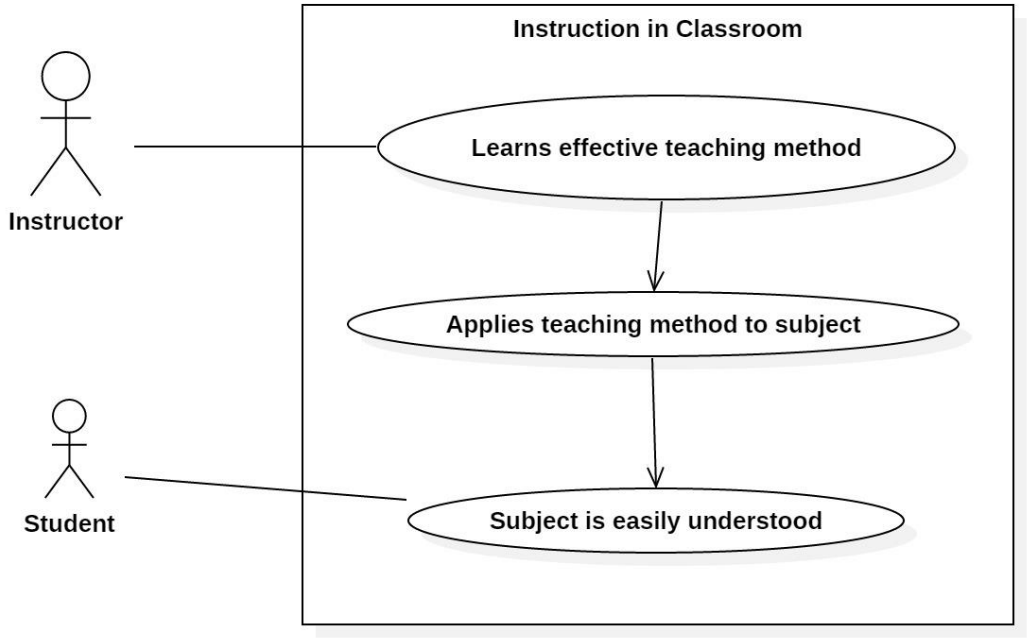


Figure 7

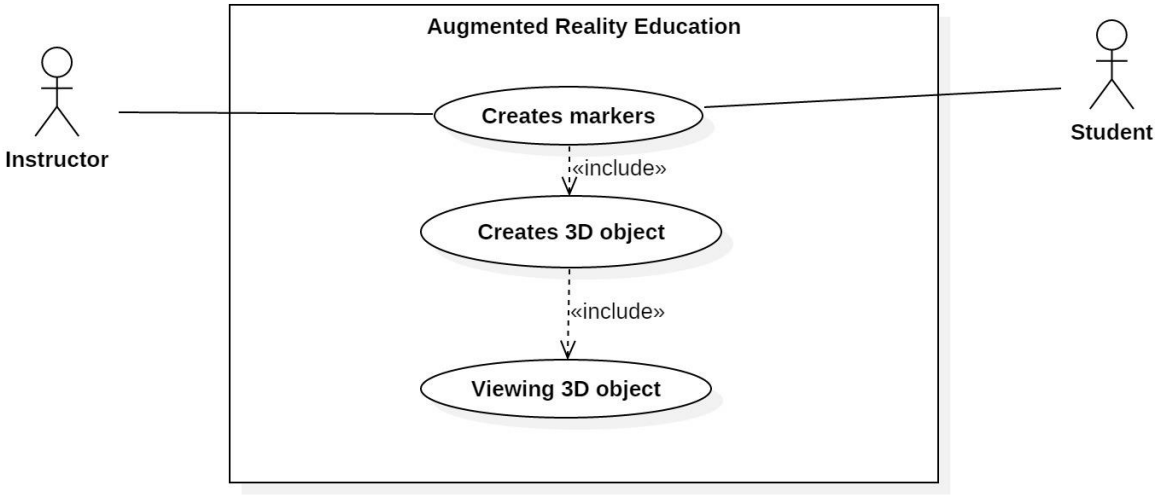


Figure 8

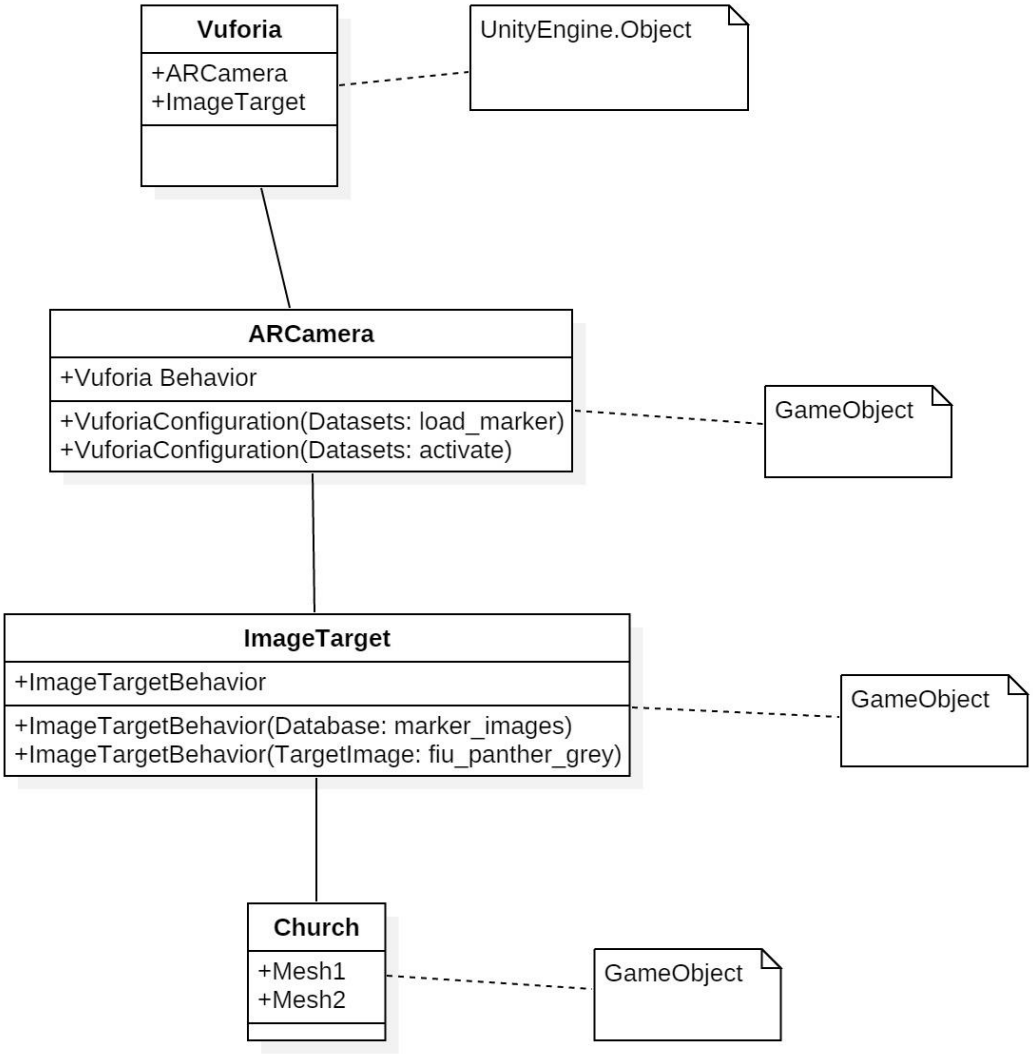

Vuforia
+ARCamera
+ImageTarget
UnityEngine.Object
ARCamera
+Vuforia Behavior
+VuforiaConfiguration(Datasets: load_marker)
+VuforiaConfiguration(Datasets: activate)
GameObject
ImageTarget
+ImageTargetBehavior
+ImageTargetBehavior(Database: marker_images)
+ImageTargetBehavior(TargetImage: fiu_panther_grey)
GameObject
Church
+Mesh1
+Mesh2
GameObject

Figure 9

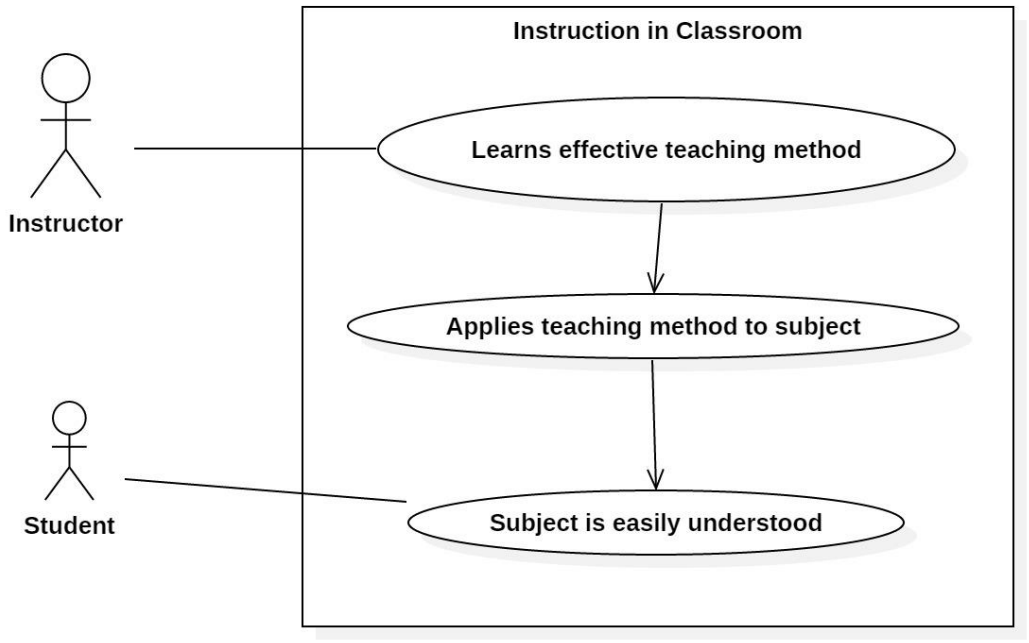

Instruction in Classroom
Learns effective teaching method
Applies teaching method to subject
Subject is easily understood
Instructor
Student

Figure 10

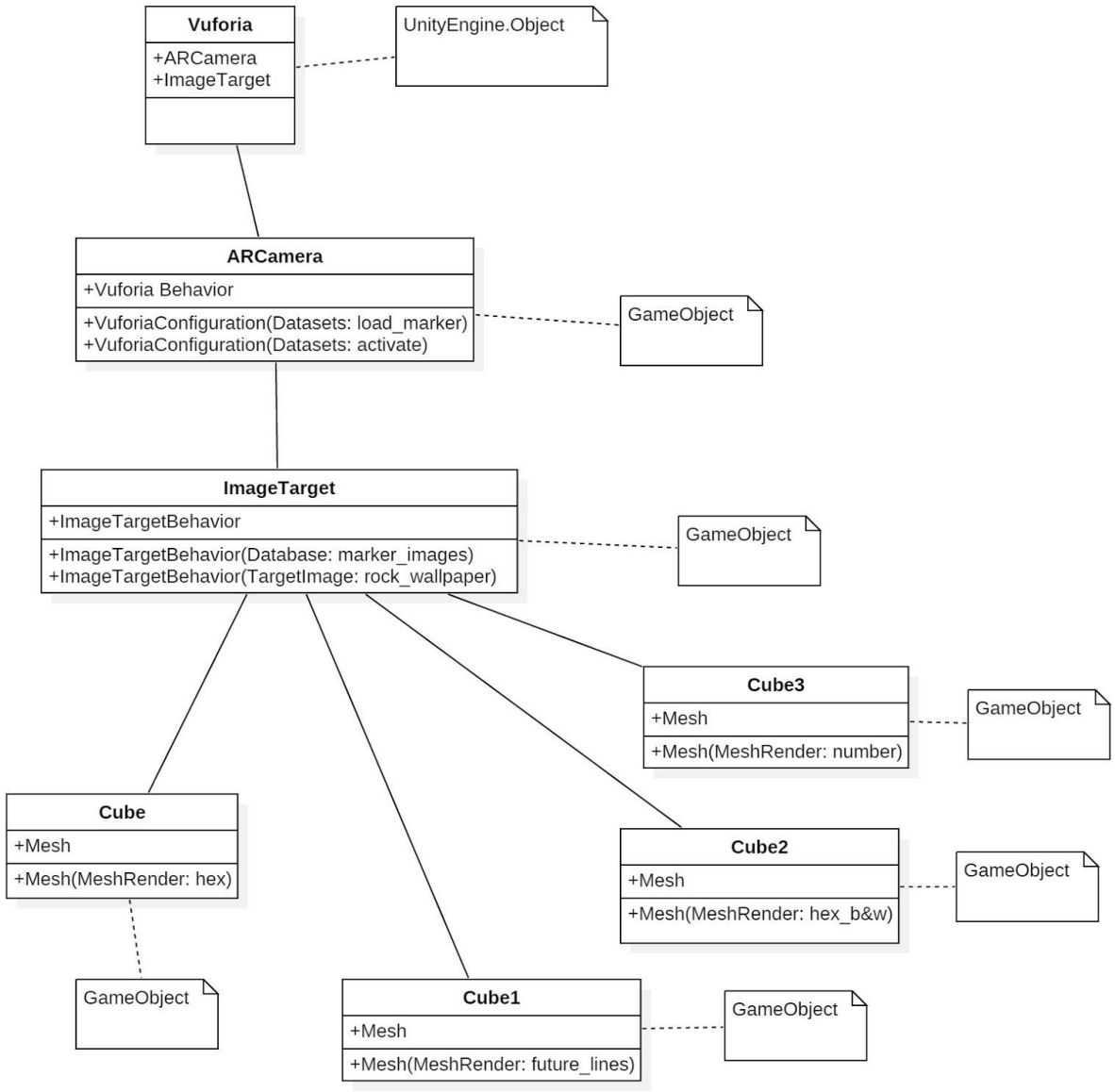

Vuforia
+ARCamera
+ImageTarget
UnityEngine.Object
ARCamera
+Vuforia Behavior
+VuforiaConfiguration(Datasets: load_marker)
+VuforiaConfiguration(Datasets: activate)
GameObject
ImageTarget
+ImageTargetBehavior
+ImageTargetBehavior(Database: marker_images)
+ImageTargetBehavior(TargetImage: rock_wallpaper)
GameObject
Cube3
+Mesh
+Mesh(MeshRender: number)
GameObject
Cube
+Mesh
+Mesh(MeshRender: hex)
Cube2
+Mesh
+Mesh(MeshRender: hex_b&w)
GameObject
GameObject
Cube1
+Mesh
+Mesh(MeshRender: future_lines)
GameObject

Figure 11

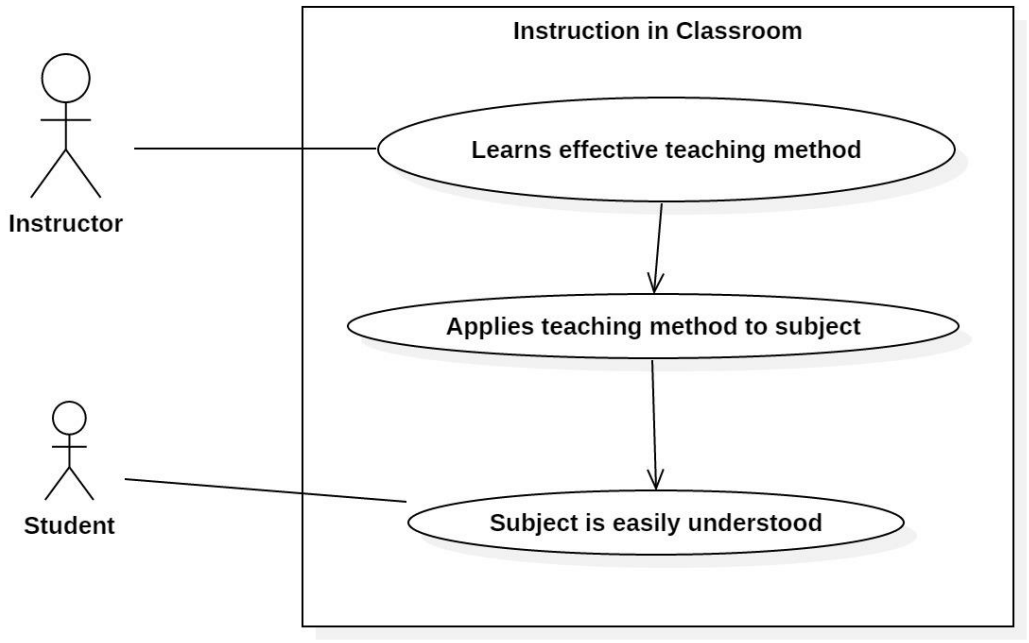

Instruction in Classroom
Learns effective teaching method
Applies teaching method to subject
Subject is easily understood
Instructor
Student

Figure 12

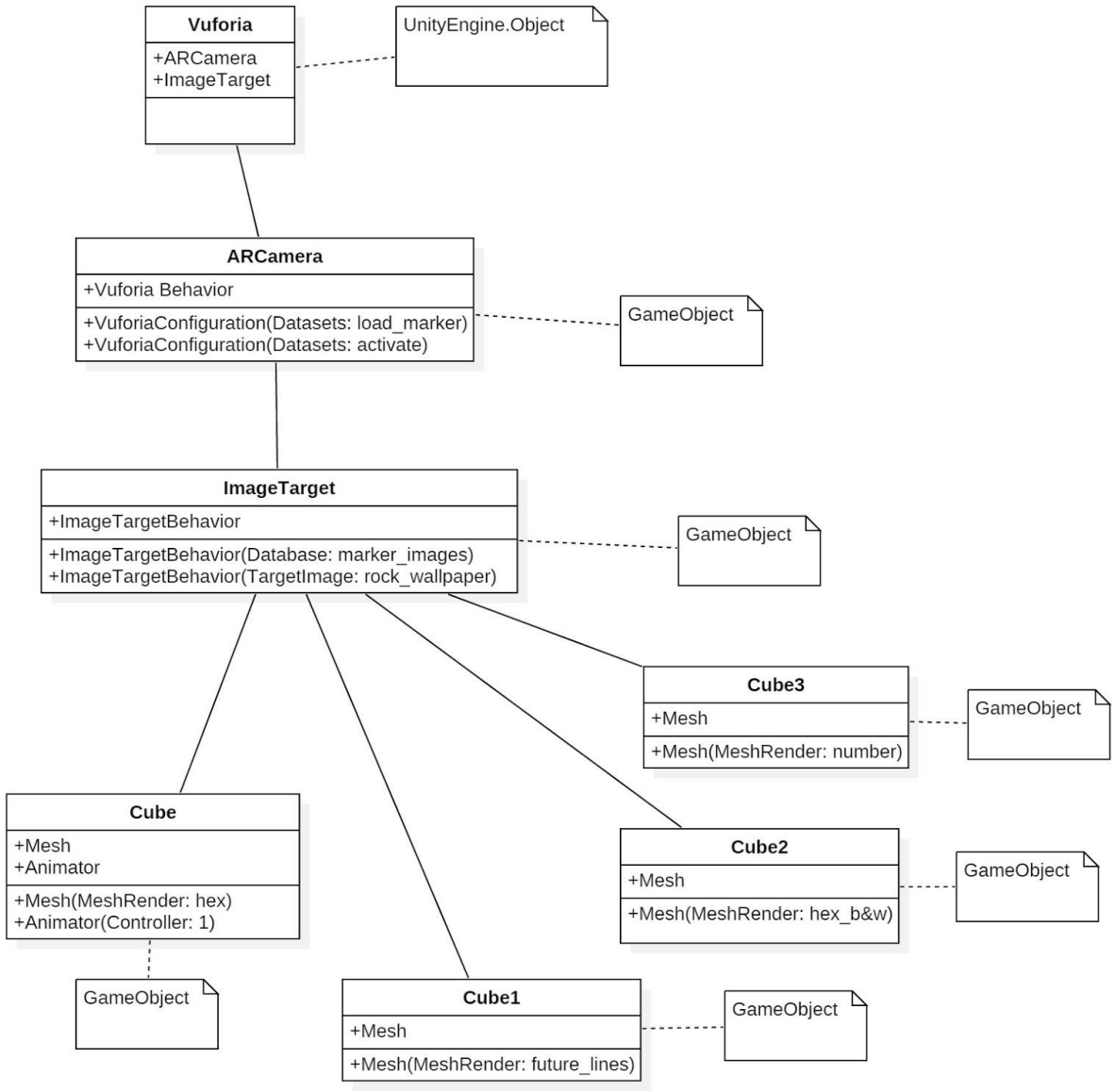

Vuforia
+ARCamera
+ImageTarget
UnityEngine.Object
ARCamera
+Vuforia Behavior
+VuforiaConfiguration(Datasets: load_marker)
+VuforiaConfiguration(Datasets: activate)
GameObject
ImageTarget
+ImageTargetBehavior
+ImageTargetBehavior(Database: marker_images)
+ImageTargetBehavior(TargetImage: rock_wallpaper)
GameObject
Cube3
+Mesh
+Mesh(MeshRender: number)
GameObject
Cube
+Mesh
+Animator
+Mesh(MeshRender: hex)
+Animator(Controller: 1)
Cube2
+Mesh
+Mesh(MeshRender: hex_b&w)
GameObject
GameObject
Cube1
+Mesh
+Mesh(MeshRender: future_lines)
GameObject

Figure 13

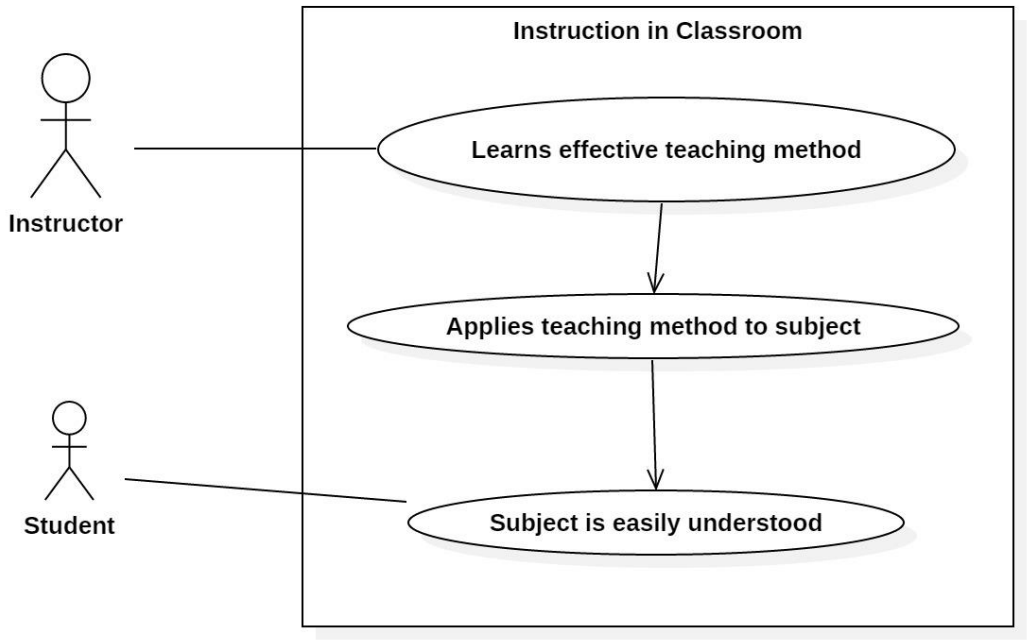


Figure 14

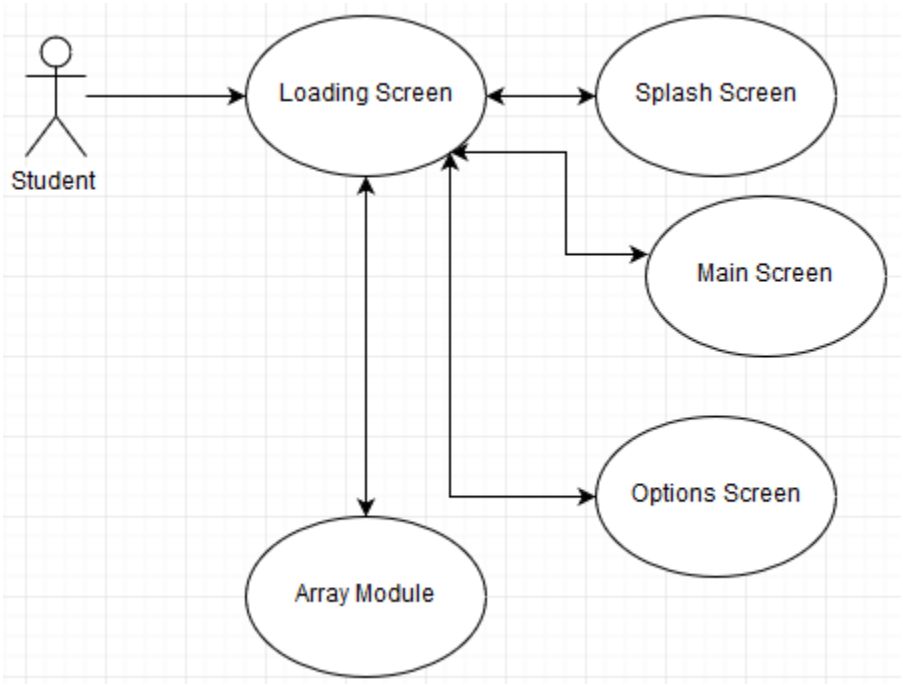


Figure 15

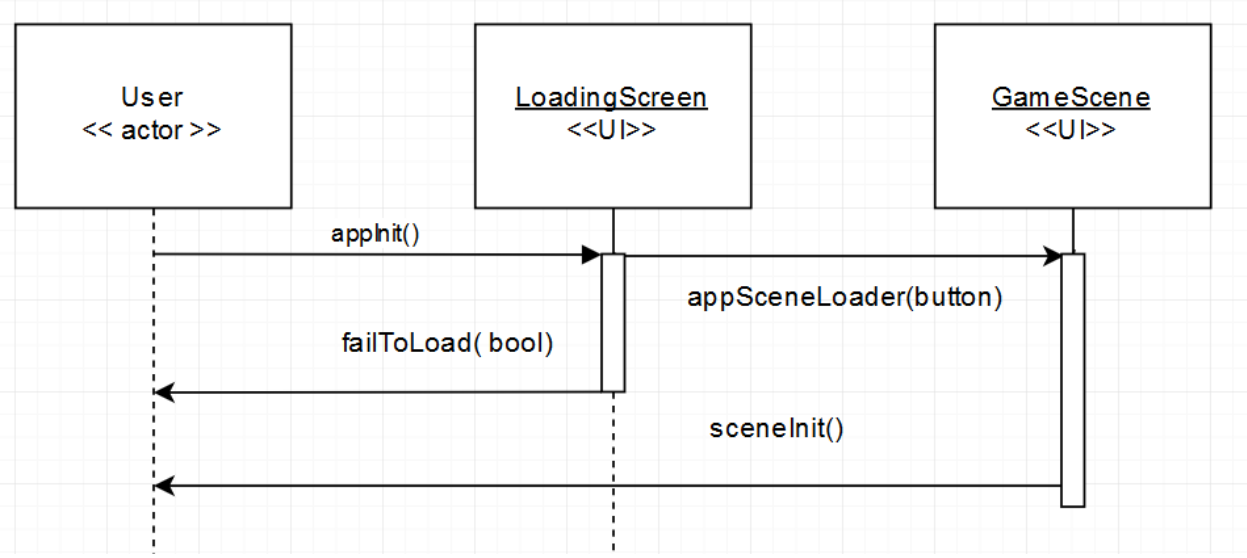


Figure 16

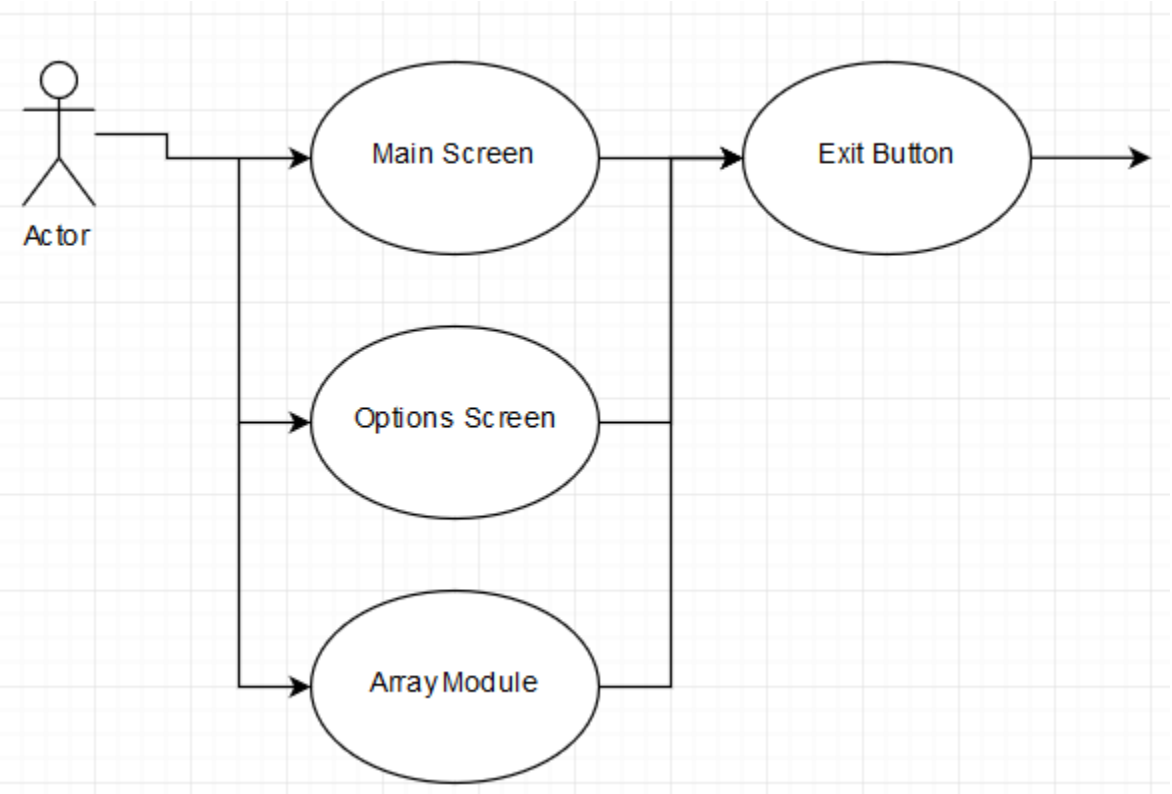


Figure 17

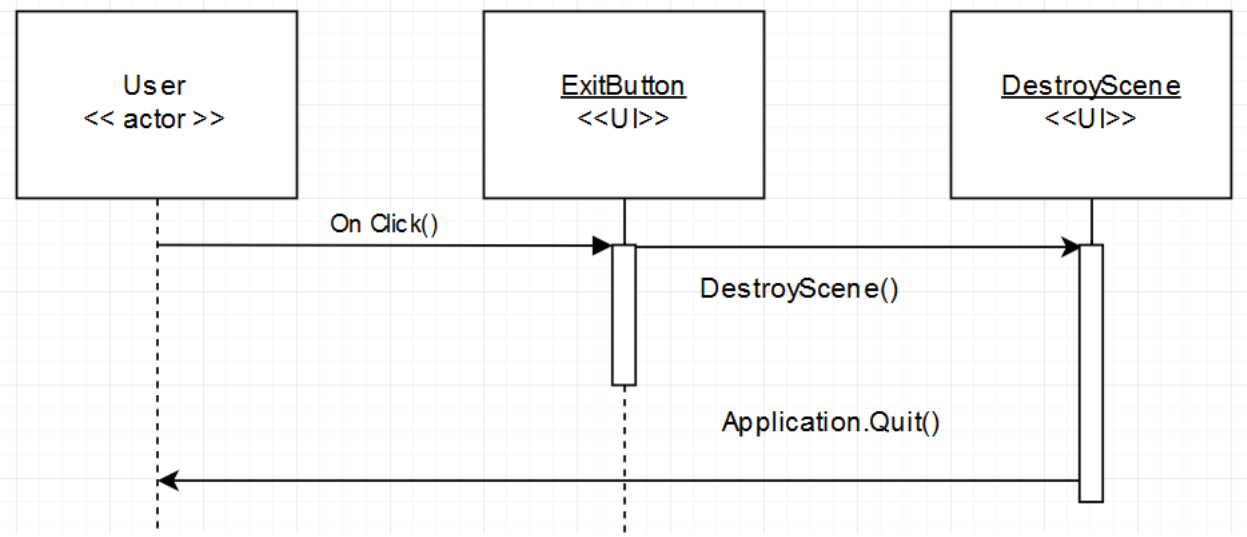


Figure 18

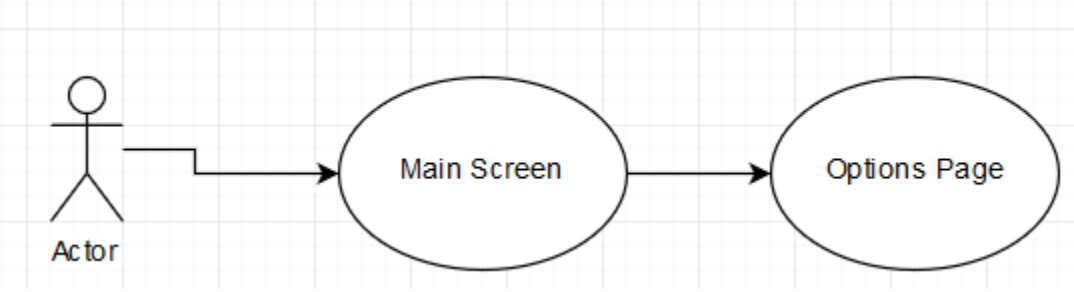


Figure 19

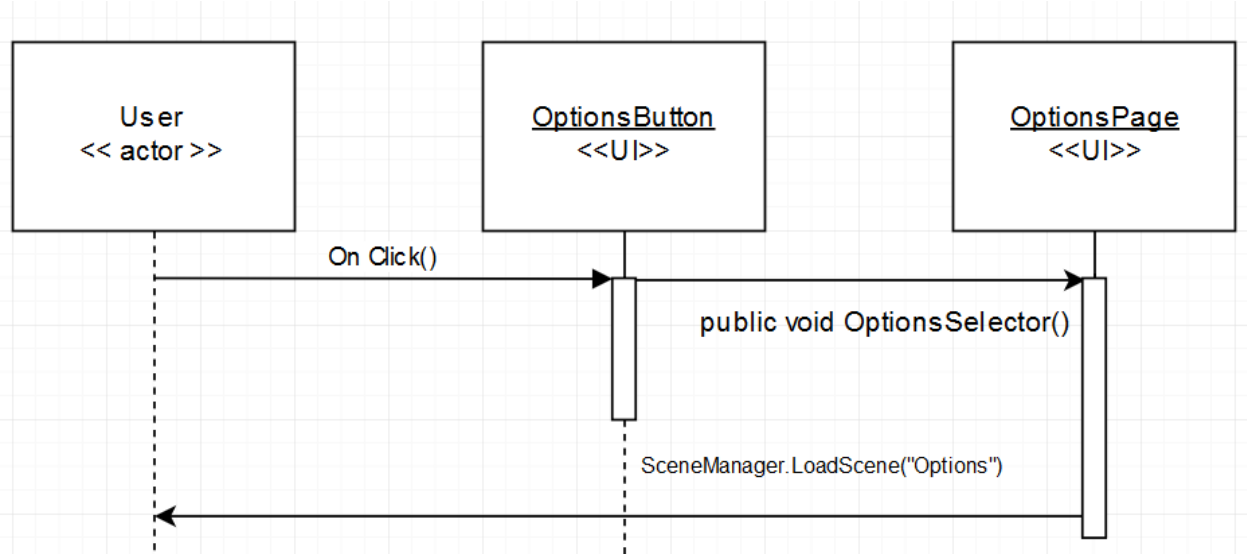


Figure 20

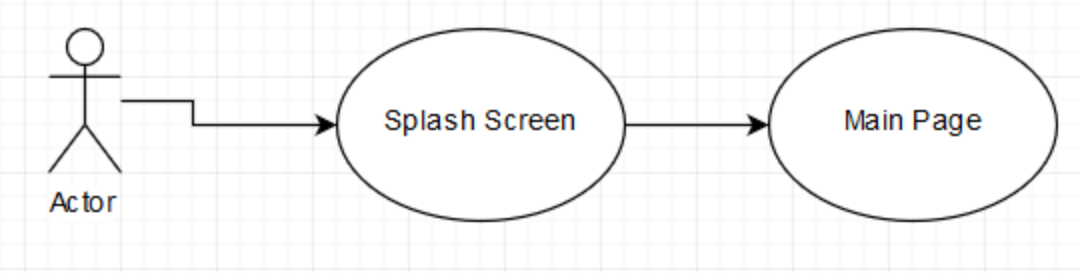


Figure 21

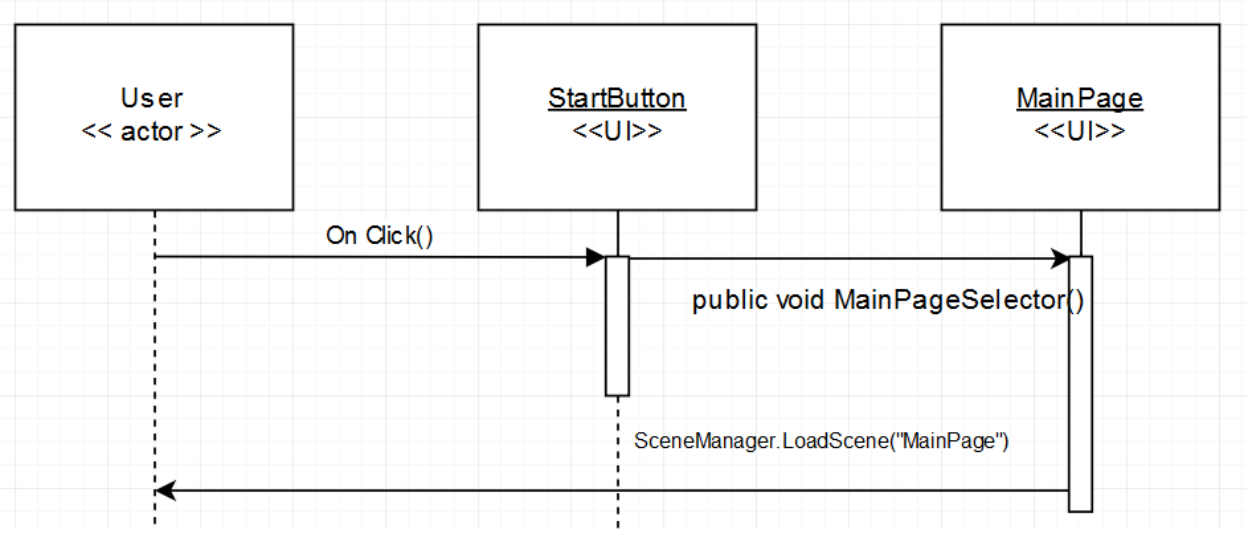


Figure 22

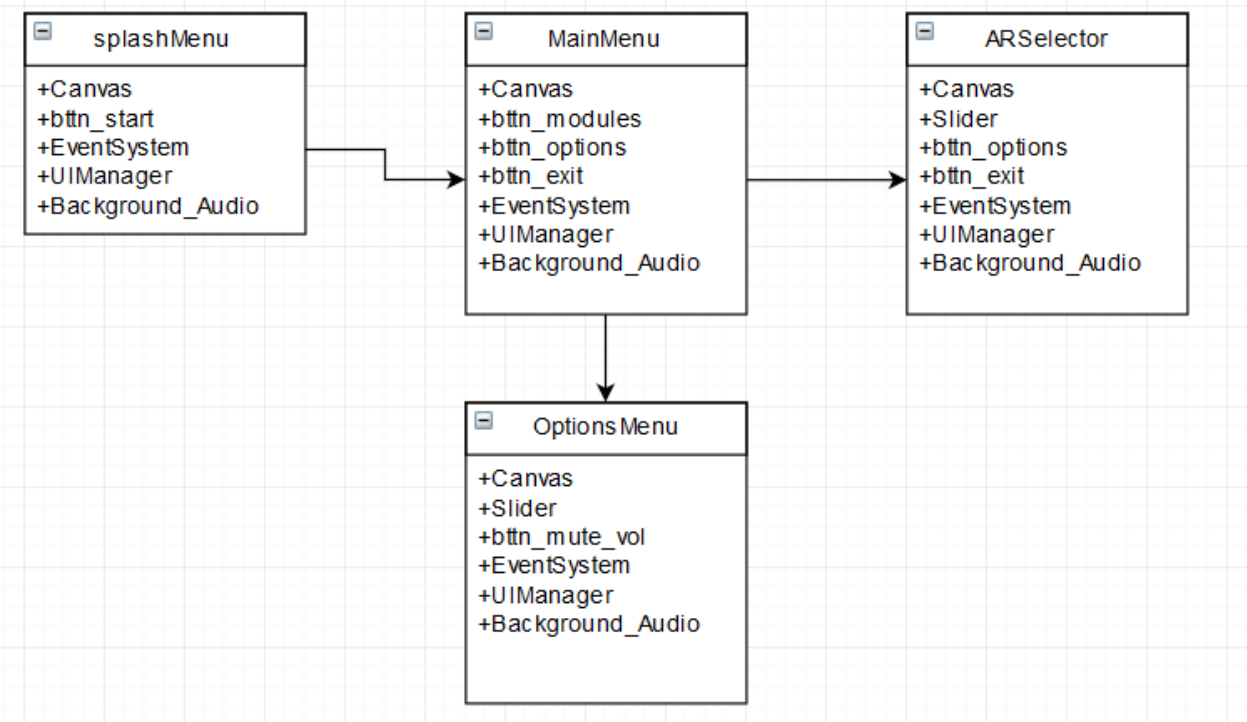

splashMenu
+Canvas
+bttn_start
+EventSystem
+UIManager
+Background_Audio
MainMenu
+Canvas
+bttn_modules
+bttn_options
+bttn_exit
+EventSystem
+UIManager
+Background_Audio
ARSelector
+Canvas
+Slider
+bttn_options
+bttn_exit
+EventSystem
+UIManager
+Background_Audio
OptionsMenu
+Canvas
+Slider
+bttn_mute_vol
+EventSystem
+UIManager
+Background_Audio

Figure 23

Appendix B

Figure 1: Splash Page.

Figure 2: Loading Screen

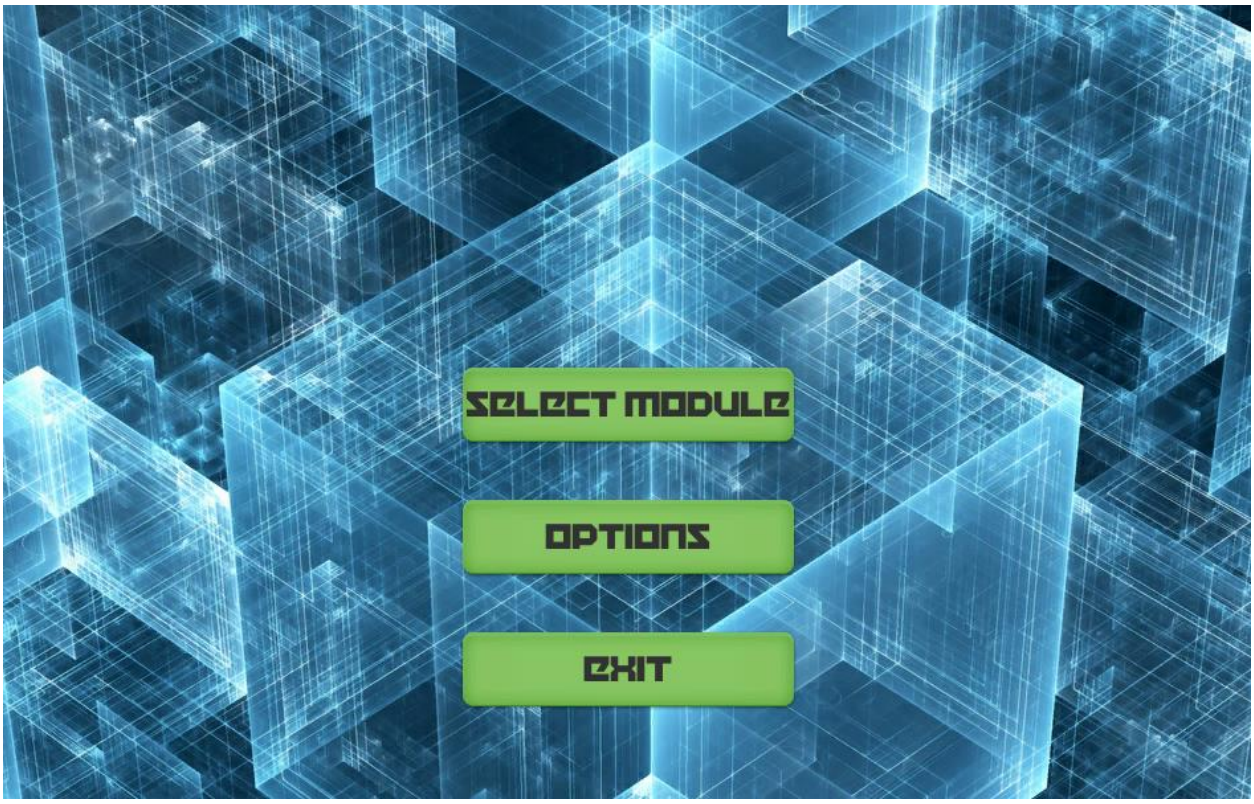


Figure 3: Front Page

Figure 4: Options Page

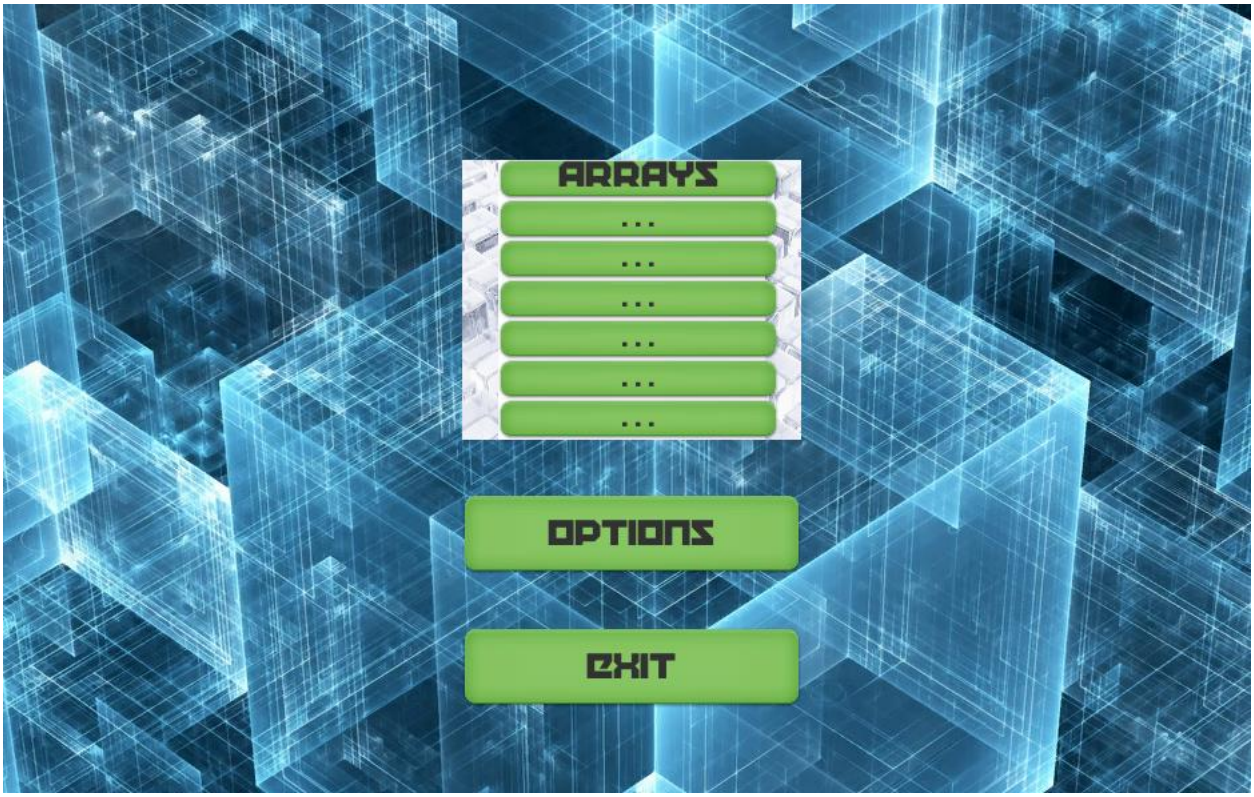


Figure 5: Select Module Page

Figure 6: Array Visualization

Appendix C

Sprint Review

Attendees: Tom Lam, Ramses Sanchez
Start time: 2/10/17 - 9:00pm
End time: 2/10/17 - 9:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

Sprint 1:

- #1158 - Learning Unreal Game Engine 4 with AR (Ramses)
- #1159 - Learning Unity Game Engine with AR (Tom)
- #1160 - Learning Unity Game Engine with ARToolKit (Ramses)
- #1161 - Learning Unity Game Engine with ARToolKit (Tom)

Sprint 2:

- #1162 - Create an AR visualization demo (Ramses)
- #1163 - Create fiducial markers (Tom)
- #1164 - Create a multiple marker visualization (Tom)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Attendees: Tom Lam, Ramses Sanchez
Start time: 2/26/17 - 9:00pm
End time: 2/26/17 - 9:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

Sprint 3:

- #1182 -Research into Memory Retention (Ramses)
- #1183 - Design and Draft Research Study/Experiment (Ramses)
- #1184 - Vuforia plugin Animations rendering (Ramses)
- #1185 - Research topics on Augmented Reality and Education (Tom)
- #1186 - Unity Game Engine with Vuforia (Tom)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Attendees: Tom Lam, Ramses Sanchez
Start time: 3/10/17 - 9:00pm
End time: 3/10/17 - 9:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

Sprint 4:

- #1209 - (AR_subgroup) Unity Game Engine Textures Learning (Tom)
- #1213 - (AR_subgroup) Prototype UI (Ramses)
- #1214 - (AR_subgroup) Implement GUI (Ramses)
- #1215 - (AR_subgroup) Obtain Initial FIU IRB Approval (Ramses)
- #1216 - (AR_subgroup) Complete IRB Training Requirements (Ramses)
- #1229 - (AR_subgroup) Unity Game Engine Textures Creation (Tom)
- #1240 - (AR_subgroup) Unity/Vuforia Windows & Android Demo Deployment (Tom)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Attendees: Tom Lam, Ramses Sanchez
Start time: 3/24/17 - 9:00pm
End time: 3/24/17 - 9:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

Sprint 5:

- (AR_subgroup) Unity Game Engine (User Story #1248)
- (AR_subgroup) Unity Game Engine (User Story #1249)
- (AR_subgroup) Implement Splash Page GUI (User Story #1214)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Attendees: Tom Lam, Ramses Sanchez
Start time: 4/10/17 - 9:00pm
End time: 4/10/17 - 9:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

Sprint 6:

- (AR_subgroup) Implement execution/loading page GUI (User Story #1265)
- (AR_subgroup) Implement Exit Page GUI (User Story #1266)
- (AR_subgroup) Implement Options Page GUI (User Story #1267)
- (AR_subgroup) Implement Front Page GUI (User Story #1268)
- (AR_subgroup) Review Research and Create a summary (User Story #1244)
- (AR_subgroup) Begin Introduction section (first draft) for Research Paper (User Story #1256)
- (AR_subgroup) Unity Game Engine Animations (User Story #1290)
- (AR_subgroup) Compile a Second Draft for Research Paper (User Story #1295)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Sprint Retrospective

Sprint 1 & 2
Attendees: Tom Lam, Ramses Sanchez
Start time: 2/10/2017 @ 10:00 PM
End time: 2/10/2017 @ 11:00 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - We did a good job estimating our team's velocity: our user stories were adequate and covered the length (2 weeks) of the sprint.
- Did we do a good job estimating the points (time required) for each user story?
    - Our stories did not have the points required to account for testing and documentation: just implementation which led to accruing technical debt.
- Did each team member work as scheduled?
    - Yes; everyone performed their work diligently.

What went right?

- Tom and I are committed to meeting daily - which limits the amount of surprises as the Sprint is winding down; thus we work efficiently together since we are honest with our progress.

How to address the issues in the next sprint?

- How to improve the process?
    - It is better to overestimate the points for each user story than to underestimate them - that way we won't accrue technical debt.
- How to improve the product?
    - We need to meet more with our Product Owner to understand how to fully realize his vision.

Sprint 3
Attendees: Tom Lam, Ramses Sanchez
Start time: 2/24/2017 @ 4:00 PM
End time: 2/24/2017 @ 5:00 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - We did a good job estimating our team's velocity: our user stories were adequate and covered the length (2 weeks) of the sprint.
- Did we do a good job estimating the points (time required) for each user story?
    - Yes, not all of the stories required most of the time allotted to them but it was helpful in the sense that there was no pressure due to time constraints and that the user stories could be adequately fulfilled and done properly.
- Did each team member work as scheduled?
    - Yes, everyone performed their work diligently.

What went right?

- Our commitment to meeting daily has allowed us to remain focused on our goals as the end of the sprint draws closer. We communicated in areas where we were not doing very well in and as a result we were able to mitigate repercussions.

How to address the issues in the next sprint?

- How to improve the process?
    - More communication about details would be a good place to start as it is often that unless specific details are reviewed, they are often forgotten or misunderstood.
- How to improve the product?
    - More meetings with the Product Owner to help provide if not inspire the development process would be extremely helpful.

Sprint 4
Attendees: Tom Lam, Ramses Sanchez
Start time: 3/10/2017 @ 4:00 PM
End time: 3/10/2017 @ 5:00 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - We did a good job estimating our team's velocity: our user stories were adequate and covered the length (2 weeks) of the sprint.
- Did we do a good job estimating the points (time required) for each user story?
    - Yes, not all of the stories required most of the time allotted to them but it was helpful in the sense that there was no pressure due to time constraints and that the user stories could be adequately fulfilled and done properly.
- Did each team member work as scheduled?
    - Yes, everyone performed their work diligently.

What went right?

- Our commitment to meeting daily has allowed us to remain focused on our goals as the end of the sprint draws closer. We communicated in areas where we were not doing very well in and as a result we were able to mitigate repercussions.

How to address the issues in the next sprint?

- How to improve the process?
    - More communication about details would be a good place to start as it is often that unless specific details are reviewed, they are often forgotten or misunderstood.
- How to improve the product?
    - More meetings with the Product Owner to help provide if not inspire the development process would be extremely helpful.

Sprint 5
Attendees: Tom Lam, Ramses Sanchez
Start time: 3/27/2017 @ 4:00 PM
End time: 3/27/2017 @ 5:00 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - We did not do a good job estimating our team's velocity: while our user stories were adequate and descriptive and covered the length (1 week) of the sprint, some unforeseen hurdles did come up; on the GUI design stories, we did not account for the creative aspects of GUI design: the design of buttons (32-bit), images, and other buttons.
- Did we do a good job estimating the points (time required) for each user story?
    - No. We did not properly estimate the amount or complexity required in making GUI buttons and related-images which were easy but time consuming.
- Did each team member work as scheduled?
    - Yes, everyone performed their work diligently.

What went right?

- Our commitment to meeting daily has allowed us to remain focused on our goals as the end of the sprint draws closer. We communicated in areas where we were not doing very well in and as a result we were able to mitigate repercussions.

How to address the issues in the next sprint?

- How to improve the process?
    - More communication about details would be a good place to start as it is often that unless specific details are reviewed, they are often forgotten or misunderstood.
- How to improve the product?
    - More meetings with the Product Owner to help provide if not inspire the development process would be extremely helpful.

Sprint 6
Attendees: Tom Lam, Ramses Sanchez
Start time: 4/10/2017 @ 4:00 PM
End time: 4/10/2017 @ 5:00 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - In this sprint the team was left with a lot of work to do due to complications in design and implementation.
- Did we do a good job estimating the points (time required) for each user story?
    - Due to unexpected difficulty in some of the user stories as well as the need for final integration, our timing was off which left much of the work to be done in overtime rather than on schedule.
- Did each team member work as scheduled?
    - Yes, everyone performed their work diligently.

What went right?

- Development was straightforward, if not, time-consuming, Our designs were fully realized in during development and functioned as they should. Communication was once again very important at this point in time because of the time constraint remaining the the project. It kept everyone on task and allowed us to complete the sprint with minimum concerns.

How to address the issues in the next sprint?

- How to improve the process?
    - Maintain communication and workload.
- How to improve the product?
    - Focus on bug fixes if there are any, attempt to introduce quality-of-life improvements to the application.

Appendix D

User Manual

Launching the "Array" Data Structure:

1. Launch application.
2. Click "Select Module" button.
3. Select "Arrays".
4. Point camera towards AR Marker.
5. View 3D Array Object.

Viewing "Options":

1. Launch application.
2. Click "Options" button.

Exiting the application from main menu:

1. Launch application.
2. Select "Exit" button.

Installation/Maintenance

You must first install ALL required software before the application will be able to compile and become operable.

Minimum software required:

- Unity Game Engine Version 5.6 or newer.
- Vuforia for Unity Plugin Version 6.2 or newer.
- Visual Studio Community 2017 or better.

To run the application in the Unity Game Engine Preview Mode:

1. Download the project code from the GitHub repository.
2. Save project to any location.
3. Open the project in Unity.
4. Click the "Play" button at the top center of the screen.

To run the application as a Windows Store App:

1. Be sure to have all the required software, including the Windows 10 Universal SDK.
2. Download the project code from the GitHub repository.
3. Save project to any location.
4. Open the project in Unity.
5. Go to "File" > "Build Settings" > Select "Windows Store" .
6. Select SDK "Universal 10".
7. Select Build and Run on "Local Machine".
8. Select the Checkbox for "Development Build" under the Debugging section.
9. Select "Build & Run".
10. Find a save location; the location itself does not matter, only that the process is performed.
11. The application should run and be operable.

To run the application through Visual Studios:

1. Follow the exact steps for running the application as a Windows Store app.
2. Open the folder where the Windows Store app was save.
3. Launch the ".sln" file using Visual Studio. This should be the first file in the save's root folder.
4. At the top of Visual Studio, select "Release" from the dropdown menu, as well as "x64" if you are running on a 64-bit computer.
5. Launch by clicking on the green play button titled "Local Machine" to the right of the "x64" dropdown menu
6. The application should compile and be operable.

To run the application on Android:

1. Be sure to have all the required software, include the Android SDK and the Java Development Kit.
2. Be sure that the Android device is in USB debugging mode. In Android 5.0 and newer, you can enable Developer Mode by going into Settings > About Phone > Build number > Tap it 7 times to become developer.
3. Scroll down the list in Developer Mode and enable “USB Debugging”.
4. Download the project code from the GitHub repository.
5. Open the project in Unity.
6. Go to “File” > “Build Settings” > Select “Android”.
7. Select “Development Build”.
8. Select “Player Settings” > Give the name to the Company Name and Product Name fields.
9. Under the “Other Settings” tab, under the “Identification” section, fill in the “Bundle Identifier” field.
10. Click “Build & Run”.
11. Find a save location; the location itself does not matter, only that the process is performed.
12. The application should auto-run and is operable.

NOTE: It is currently not possible to deploy any Vuforia applications as a Windows Standalone Executable (.exe). The only supported platforms are: Windows Store App (tested), Android (tested), and iOS (not tested).

Shortcomings/Wishlist

Since this is the first iteration of the application, there are many items on the wishlist we would like to see in the future:

- Ability to select more data structures.
- Ability to provide field-of-view option to user allowing greater view angles.
- Ability to have the data structure sort through itself depending what inputs is given.
- Ability to complex animations selectable by the user.
- Ability to input data into the data structure by the user.

References

Unity Documentation:

https://docs.unity3d.com/Manual/index.html

Vuforia Documentation:

https://library.vuforia.com/getting-started

Deployment to Android Tutorial:

https://www.sitepoint.com/how-to-build-an-ar-android-app-with-vuforia-and-unity/
https://www.sitepoint.com/building-a-google-cardboard-vr-app-in-unity/