

Virtual Room Decor



By

Kashan Sardar

Asad Ali Khan

Submitted to Faculty of Department of Computer Software Engineering National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering

June 2018

In the name of Allah, the Most Beneficent, the Most Merciful

ABSTRACT

Virtual Room Decor

Virtual Reality (VR) is the next big thing in technology industry and it will take over the gaming and multimedia industry. Virtual Room Décor application will be designed to run in Virtual Reality (VR) environment.

An android application, developed in Unity 3D, that runs in Virtual Reality (VR) environment. It shall provide the user a virtual room, where the user can decorate and furnish his room. Paint walls, paste wallpapers, add furniture to the room, add in doors and windows, change color and texture of all elements. Do all this virtually without moving a budge. Visualize their dream room. All of this functionality via simple drag-and-drop interface.

Virtual Reality (VR) headsets and motion controllers shall be used as designated hardware to bring forth a complete experience.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis – Virtual Room Decor carried out by **Kashan Sardar, Asad Ali Khan** under supervision of **Asst Prof Bilal Rauf** for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved By

Asst Prof Bilal Rauf

Department of CSE, MCS

Dated:

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation,

a work of this magnitude would not have been possible

To our supervisor, Asst Prof Bilal Rauf who has given us
great support and valuable suggestions throughout the
implementation process.

And finally, to our Friends and siblings for their encouragement.

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support and their critical reviews. We would like to thank our supervisor Asst Prof Bilal Rauf, for his continuous guidance and motivation throughout the course of our project. Without their help, we would have not been able to accomplish anything.

Contents

Chapter 1: Introduction.....	1
1.1 Overview	1
1.2 Problem Statement	1
1.3 Scope.....	1
1.4 Objective	2
1.5 Deliverables	2
Chapter 2: Literature Review	4
2.1 Technology	4
Chapter 3: Software Requirements Specification	5
3.1 Introduction.....	5
3.1.1 Purpose.....	5
3.1.2 Intended Audience and Reading Suggestions.....	5
3.1.3 Scope.....	5
3.1.4 References.....	6
3.2 Overall Description.....	6
3.2.1 Product Perspective.....	6
3.2.2 Product Functions	6
3.2.2.1 Select Rooms.....	6
3.2.2.2 Design Rooms	6
3.2.2.3 Interact with Objects	6
3.2.2.4 Take Screenshots.....	6
3.2.2.5 Save and Load Rooms.....	7
3.2.3 User Classes and Characteristics	7
3.2.3.1 Application User	7
3.2.4 Operating Environment.....	7
3.2.5 Design and Implementation Constraints.....	7

3.2.6	User Documentation	7
3.2.7	Assumptions and Dependencies	8
3.3	External Interface Requirements.....	8
3.3.1	User Interfaces	8
3.3.2	Hardware Interfaces	8
3.4	System Features	8
3.4.1	Accessing the Main Menu	8
3.4.1.1	Description and Priority	8
3.4.1.2	Stimulus/Response Sequences	8
3.4.1.3	Functional Requirements	9
3.4.2	Select Room	9
3.4.2.1	Description and Priority	9
3.4.2.2	Stimulus/Response Sequences	9
3.4.2.3	Functional Requirements.....	9
3.4.3	Room.....	10
3.4.3.1	Description and Priority	10
3.4.3.2	Stimulus/Response Sequences	10
3.4.3.3	Functional Requirements	10
3.4.4	Elements Menu	10
3.4.4.1	Description and Priority	10
3.4.4.2	Stimulus/Response Sequences	10
3.4.4.3	Functional Requirements.....	10
3.4.5	Interact With Objects	11
3.4.5.1	Description and Priority	11
3.4.5.2	Stimulus/Response Sequences	11
3.4.5.3	Functional Requirements.....	11
3.4.6	Save Room.....	12
3.4.6.1	Description and Priority	12
3.4.6.2	Stimulus/Response Sequences	12

3.4.6.3 Functional Requirements	12
3.4.7 Load Room.....	12
3.4.7.1 Description and Priority	12
3.4.7.2 Stimulus/Response Sequences	12
3.4.7.3 Functional Requirements.....	12
3.4.8 Take Screenshots	13
3.4.8.1 Description and Priority	13
3.4.8.2 Stimulus/Response Sequences	13
3.4.8.3 Functional Requirements	13
3.4.9 Help Screen	13
3.4.9.1 Description and Priority	13
3.4.9.2 Stimulus/Response Sequences	13
3.4.9.3 Functional Requirements	13
3.5 Other Nonfunctional Requirements	14
3.5.1 Performance Requirements	14
3.5.2 Safety Requirements	14
3.5.3 Security Requirements	14
3.5.4 Software Quality Attributes	14
3.5.4.1 Usability:.....	14
3.5.4.2 Accuracy:	14
3.5.4.3 Portability:.....	14
3.5.4.4 Availability	14
3.5.4.5 Flexibility	14
3.5.4.6 Data Integrity	15
3.5.4.7 Scalability	15
3.5.4.8 Confidentiality	15
3.5.4.9 Business Rules	15
Chapter 4: Design and Development	16
4.1 Introduction.....	16

4.1.1 Purpose.....	16
4.1.2 Scope.....	16
4.1.3 Definitions, acronyms, and abbreviations.....	16
4.2 System architecture description	17
4.2.1 Overview	17
4.2.1 GameObjects.....	17
4.2.2 UI	17
4.2.3 DataObjects.....	18
4.2.4 Enums	18
4.3 System Block Diagram.....	18
4.4 Component Diagram	19
4.5 User View (Use Case Diagram).....	20
4.6 Use Case Descriptions.....	20
4.6.1 Use Case 1.....	21
4.6.2 Use Case 2.....	22
4.6.3 Use Case 3.....	22
4.6.4 Use Case 4.....	24
4.6.5 Use Case 5.....	24
4.6.6 Use Case 6.....	24
4.6.7 Use Case 7.....	26
4.6.8 Use Case 8.....	26
4.6.9 Use Case 9.....	28
4.6.10 Use Case 10.....	28
4.6.11 Use Case 11.....	30
4.6.12 Use Case 12.....	30
4.6.13 Use Case 13.....	31
4.6.14 Use Case 14.....	32
4.7 Implementation View (Class Diagram)	32
4.7.1 Class Diagram.....	33

4.7.2	Class Diagram Description	34
4.8	Sequence Diagrams	35
4.8.1	Select Room	36
4.8.2	Spawn Room Object	37
4.8.3	Move/Delete/Rotate Room Object.....	37
4.8.4	Save Room	37
4.8.5	Load Room.....	38
4.8.6	Take and Save Screenshot	38
4.9.7	View Saved Screenshots	38
4.8.9	Access User Guide	39
4.9	User Interface	39
4.9.1	Main Menu.....	39
4.9.2	Elements Menu	40
4.9.3	Load Room.....	40
4.10	Detailed description of components	41
4.10.1	UI	41
4.10.2	Managers Classes.....	43
4.10.3	Unity Engine	47
4.11	Reuse and relationships to other products	49
4.12	Design decisions and tradeoffs	50
4.13	Pseudo code for components	50
Chapter 5: Project Test and Evaluation		52
5.1	Project Test and Evaluation	52
5.1.1	Introduction.....	52
5.1.2	Test Items	52
5.1.2.1	Features to be Tested	53
5.1.3	Approach.....	53
5.1.4	Pass/Fail Criteria for Test Items	53
5.1.5	Test Suspension Criteria	54

5.2 Test Deliverables	54
5.2.1 Testing tasks.....	54
5.2.2 Test cases	54
5.2.2.1 Displaying The Main menu	54
5.2.2.2 Connectivity Configuration Between Devices.....	55
5.2.2.3 Main Menu Navigation	56
5.2.2.4 Room Interactions	57
5.2.2.5 Room Elements Interactions	57
5.2.2.6 Saving the Room.....	58
5.2.2.7 Taking Screenshot and saving to the device	59
5.3 Environmental Needs.....	59
5.3.1 Hardware.....	59
5.3.2 Software	59
5.4 Responsibilities, Staffing and Training Needs.....	60
5.4.1 Responsibilities	60
5.4.2 Staffing and Training Needs	60
Chapter 6: Future Work	60
Bibliography	62
Appendix A: Project Proposal	63

List of Tables

Table 1-1 : Deliverables.....	2
Table 4.6-1 Use Case 1	21
Table 4.6-2 Use Case 2	22
Table 4.6-3 Use Case 3	22
Table 4.6-4 Use Case 4	24
Table 4.6-5 Use Case 5	24
Table 4.6-6 Use Case 6	24
Table 4.6-7 Use Case 7	26
Table 4.6-8 Use Case 8	26
Table 4.6-9 Use Case 9	28
Table 4.6-10 Use Case 10.....	28
Table 4.6-11 Use Case 11	30
Table 4.6-12 Use Case 12.....	30
Table 4.6-13 Use Case 13	31
Table 4.6-14 Use Case 14.....	32
Table 4.7-1 Class Diagram Description.....	34
Table 5-1 Displaying The Main menu	54
Table 5-2 Connectivity Configuration Between Devices	55
Table 5-3 Main Menu Navigation.....	56
Table 5-4 Room Interactions	57
Table 5-5 Room Elements Interactions.....	57
Table 5-6 Saving the Room	58
Table 5-7 Taking Screenshot and saving to the device.....	59

Table of Figures

Figure 4-1-1 Mind Map	17
Figure 4.3-1 System Block Diagram	18
Figure 4.4-1 Component Diagram	19
Figure 4.5-1 User View Diagram.....	20
Figure 4.7-1 Class Diagram	33
Figure 4.8-1 Select Room	36
Figure 4.8-2 Spawn Room Object	37
Figure 4.8-3 Move/Delete/Rotate Room Object.....	37
Figure 4.8-4 Save Room	37
Figure 4.8-5 Load Room.....	38
Figure 4.8-6 Take and Save Screenshot.....	38
Figure 4.8-7 View Saved Screenshots	38
Figure 4.8-8 Access User Guide	39

Chapter 1: Introduction

1.1 Overview

An android application, developed in Unity 3D, that runs in Virtual Reality (VR) environment. It shall provide the user a virtual room, where the user can decorate and furnish his room. Paint walls, paste wallpapers, add furniture to the room, add in doors and windows, change color and texture of all elements. Do all this virtually without moving a budge. Visualize their dream room. All of this functionality via simple drag-and-drop interface.

1.2 Problem Statement

The purpose is to facilitate users in décor of rooms without the practical hassles of carrying heavy furniture and performing more than one shift to get the required décor. And wall painting without spending a penny in testing of suiting colors. All this can be perform virtually to satisfying needs.

1.3 Scope

The following are the basic features that will be provided to the users by the application.

- Design rooms from scratch.
- Select different room dimensions , grid based.
- Select and edit ready-made designed rooms.
- Place furniture objects, doors, windows and a lot of other fun stuff into your room.
- Apply textures for walls, floor and furniture objects.
- Interact with objects in your room: Throw a pillow onto the couch or lift your cupboard and place it somewhere else.
- Take screenshots and photos of your room from every angle.
- Save and load your rooms.

Extended Features

- Import Custom 3D Models and Textures

1.4 Objective

At the end of the project we'll be able to have command over

- Unity 3D Engine
- Advance C# Programming
- 3D Modeling
- 3D Image Rendering
- Connecting and Programming Hardware APIs
- Create Application for Virtual Reality (VR)

The following objectives are expected of the project in the end:

- Make it a marketable product.
- Product can be used at home.
- Product to be polished for entertainment purpose.
- Users can virtually design their dream rooms.
- Users can visualize their room settings and rearrange and refurnish them to see how they look after the changes.
- Interior designers in training can virtually design and furnish rooms for training purpose.

1.5 Deliverables

Table 1-1 : Deliverables

Deliverable Name	Deliverable Summary Description
Software Requirements Specification(SRS) Document	Complete Description of what the system will do, who will use it. Detailed description of functional and non-functional requirements and the system

	features.
Design Document	Complete description of how the system will be implemented i.e. the detailed design.
Code	Complete code with the API.
Testing Document	The whole system is tested according to the specification described in the SRS document. Black box, unit and System integration testing is done.

Chapter 2: Literature Review

Virtual Reality (VR) is the use of computer **technology** to create a simulated environment. Unlike traditional user interfaces, VR places the user inside an experience. Instead of viewing a screen in front of them, users are immersed and able to interact with 3D worlds.

2.1 Technology

Samsung Gear VR

The Samsung Gear VR is a mobile virtual reality headset developed by Samsung Electronics, in collaboration with Oculus, and manufactured by Samsung.

Bluetooth Controller

Use the Gear VR controller as a remote control to navigate your virtual reality with ease or use it as a gamepad complete with trigger to win battles. The controller is designed to be used naturally with one hand.

How to control

- Touchpad Press and swipe to select and navigate.
- Back Key + Home Key Press to go back or to go to Home
- Volume Key Press to raise or lower volume
- Trigger Press to initiate an action

UNITY 3D

Unity is a multipurpose game engine that supports 2D and **3D** graphics, drag-and-drop functionality and scripting using C#. Unity supports building to 27 different platforms. The platforms are listed in the following: iOS , Android , Tizen, Windows, Universal Windows Platform, Mac, Linux, WebGL, PlayStation 4, PlayStation Vita, Xbox One, Wii U, 3DS, Oculus Rift, Google Cardboard, Steam VR, PlayStation VR, Gear VR, Windows Mixed Reality, Daydream, Android TV, Samsung Smart TV, tvOS, Nintendo Switch, Fire OS, Facebook Game room, Apple ARKit, Google ARCore, and Vuforia.

Chapter 3: Software Requirements Specification

3.1 Introduction

3.1.1 Purpose

The purpose of the Software Requirements Specification (SRS) is to document precise description of the functionality of Virtual Room Decor, an android application that shall be developed and released on Google Play Store.

This document is aimed to eliminate ambiguities and misunderstandings that may exist. The document will be a reference point during software design, development, testing and maintenance.

3.1.2 Intended Audience and Reading Suggestions

The intended readers of the SRS are all the stakeholders.

- **Project Supervisor:** It will help to supervise the project and guide the team in a better way.
- **Developers:** Project developers have an advantage of quickly understanding the methodology adopted and personalizing the product.
- **Testers:** The testers of the system can check user requirements from this SRS and develop test scenarios accordingly.
- **Documentation writers:** The document can serve as a future reference for other versions of the SRS.
- **Project Testers:** Project testers can use this document as a base for making test cases.

3.1.3 Scope

Virtual Reality (VR) is the next big thing in technology industry and it will take over the gaming and multimedia industry. This specification establishes the functional,

performance, and development requirements for Release 1 of the Virtual Room Décor Application. The application will be designed to run in Virtual Reality (VR) environment.

3.1.4 References

- [Virtual Reality](#)
- [Google Daydream](#)
- [Gear VR](#)
- [Android](#)

3.2 Overall Description

3.2.1 Product Perspective

An android application, developed in Unity 3D, that runs in Virtual Reality (VR) environment. It shall provide the user a virtual room, where the user can decorate and furnish his room. Paint walls, paste wallpapers, add furniture to the room, add in doors and windows, change color and texture of all elements. Do all this virtually without moving a budge. Visualize their dream room. All of this functionality via simple drag-and-drop interface.

3.2.2 Product Functions

The following are the basic *features* that will be provided to the users by the application.

3.2.2.1 Select Rooms

Select from different sizes of the rooms in terms of dimensions. Select and edit ready-made designed rooms.

3.2.2.2 Design Rooms

Apply textures for walls, floor and furniture object. Place furniture objects, doors, windows and a lot of other fun stuff into your room.

3.2.2.3 Interact with Objects

Interact with objects in your room: Throw a pillow onto the couch or lift your cupboard and place it somewhere else.

3.2.2.4 Take Screenshots

Take screenshots and photos of your room from every angle.

3.2.2.5 Save and Load Rooms

Save and load your rooms.

3.2.3 User Classes and Characteristics

The application will be available on Google Play Store. Therefore anyone who downloads and uses the application will be its user.

3.2.3.1 Application User

The user will be able to use all the features of the application.

3.2.4 Operating Environment

OE-1: The application will on android smart phones.

OE-2: Virtual Reality headset is required for the VR capabilities of the application.

OE-3: Bluetooth Motion Controller required, that has support for the VR headset.

3.2.5 Design and Implementation Constraints

CO-1: Application will only run on android smart phones that have support for Gear VR or Google Daydream.

CO-2: Google Daydream or Gear VR are the only supported VR headsets.

CO-3: Daydream Controller or Gear VR Controller for respective VR headset is required.

The supported Android Smart phones list is included in the *bibliography*.

3.2.6 User Documentation

UD-1 User guide will be integrated in the application.

3.2.7 Assumptions and Dependencies

AS-1 The user shall have the specified android smart phone.

AS-2 The user shall have the specified VR headset with controller.

3.3 External Interface Requirements

3.3.1 User Interfaces

- Main Menu
- Screens for each item on the Main Menu.
- Controller Interface
- 3D Room Space.
- Elements Selection Panel

3.3.2 Hardware Interfaces

- The specified android smart phone shall be connected properly with the specified VR headset.
- Specified controller shall be connected the android smart phone via Bluetooth.

3.4 System Features

3.4.1 Accessing the Main Menu

3.4.1.1 Description and Priority

After the start-up of application, the main menu shall be displayed to the user.

User shall be able to access the rest of the system through this main screen. It's priority will be medium.

3.4.1.2 Stimulus/Response Sequences

1. Start the System
2. Display main menu

3.4.1.3 Functional Requirements

REQ-1: The application must be properly installed on the android Smartphone.

REQ-2: The different options available shall be:

- Select Room
- Load Room
- Screenshots
- Help
- Exit

3.4.2 Select Room

3.4.2.1 Description and Priority

A screen shall be displayed to the user. The screen displays different room dimensions for the user to select from.

3.4.2.2 Stimulus/Response Sequences

1. Select Room is selected in the Main Menu.
2. Select Room screen display where user shall choose a room dimension from the given options.
3. The selected room is then displayed.

3.4.2.3 Functional Requirements

REQ-3.1: The Select Room option shall be selected in the main menu.

REQ-3.2: A screen for Select Room option shall be displayed.

REQ-4: Select Room screen shall provide the user option to select room dimensions.

REQ-5: The user shall only be able to choose from given room dimension options.

3.4.3 Room

3.4.3.1 Description and Priority

It is the virtual room where the user shall perform different actions.

3.4.3.2 Stimulus/Response Sequences

Select Room

1. A room dimension is selected in the Select Room screen.
2. Room is displayed having selected dimensions.

Load Room

1. A saved room is selected in the Load Room screen.
2. Selected room is displayed.

3.4.3.3 Functional Requirements

REQ-7: A room shall be selected from Select Room screen.

REQ-8: A room shall be selected from Load Room screen.

REQ-9: User shall be provided 360 degree view of the room.

3.4.4 Elements Menu

3.4.4.1 Description and Priority

The user can pop the elements menu by triggering the controller. The menu allows the user to select from different room elements, such as textures for the walls or furniture etc. The selected element can then be applied to the room or be placed in the room. The save room option can be accessed from the elements menu.

3.4.4.2 Stimulus/Response Sequences

1. The user is in a room.
2. The user presses the key on the controller that triggers element menu to pop up.
3. The elements menu then appears.

3.4.4.3 Functional Requirements

REQ-10: The user shall be in room.

REQ-11.1: The element menu trigger key on the controller shall be pressed to make the element menu appear.

REQ-11.2: The element menu trigger key on the controller shall be pressed to make the element menu disappear.

REQ-12.1: The user shall select elements from the element menu.

REQ-12.2: The selected element shall be placed to the room, if place able element.

REQ-12.3: The selected element shall be applied to the room, if applicable element.

REQ-12.4: The elements shall be categorized as follows:

- Colors
- Textures
- Furniture
- Wall Items
- Miscellaneous

REQ-13: The elements menu shall have the save room option

3.4.5 Interact With Objects

3.4.5.1 Description and Priority

The user can interact with objects in the room. Move the objects around, change their position or rotate objects.

3.4.5.2 Stimulus/Response Sequences

1. User is in a room.
2. User selects an object.

3.4.5.3 Functional Requirements

REQ-14: The user shall be in a room.

REQ-15: The user shall be able to select objects and move them around the room.

REQ-16: The user shall be able to rotate the objects.

REQ-17: The user shall be able to remove objects from the room.

3.4.6 Save Room

3.4.6.1 Description and Priority

The user can save the current state of the room.

3.4.6.2 Stimulus/Response Sequences

1. User is in a room.
2. User pops the elements menu.
3. User selects the save room option.

3.4.6.3 Functional Requirements

REQ-18: The user shall be in a room.

REQ-19: The user shall trigger the elements menu to appear.

REQ-20.1: The user shall select the save room option in the elements menu.

REQ-20.2: The room shall be saved and shall be available in the load room screen.

3.4.7 Load Room

3.4.7.1 Description and Priority

The user can load saved rooms.

3.4.7.2 Stimulus/Response Sequences

1. User selects load room option in the main menu.
2. Load Room screen appears.
3. User selects the room that is to be loaded.

3.4.7.3 Functional Requirements

REQ-21: The user shall select a room from the load room screen.

REQ-22: The selected room shall be loaded and displayed

3.4.8 Take Screenshots

3.4.8.1 Description and Priority

The user can take screenshots inside any room from any view. The saved screenshots can be accessed from the screenshots screen, which can be accessed from the main menu.

3.4.8.2 Stimulus/Response Sequences

1. User is in a room.
2. User triggers the key on the controller to save screenshot.
3. User access the screenshots screen from the main menu.
4. User can view all saved screenshots on this screen.

3.4.8.3 Functional Requirements

REQ-23: User shall be in a room.

REQ-24.1: User shall trigger the assigned key on the controller to take screenshot.

REQ-24.2: The screenshot will be saved at the current view of the camera in the room.

REQ-25: User shall access all saved screenshots from screenshot screen, accessed from main menu.

3.4.9 Help Screen

3.4.9.1 Description and Priority

The user can access the help screen from the main menu, the help screen shall contain the user guide.

3.4.9.2 Stimulus/Response Sequences

1. User is in the main menu.
2. User selects the help option.
3. Help screen is displayed.

3.4.9.3 Functional Requirements

REQ-26.1: User shall select Help option from the main menu.

REQ-26.2: The help screen shall contain the user guide.

3.5 Other Nonfunctional Requirements

3.5.1 Performance Requirements

The performance of the application shall be top-notch, since it runs on android smart phones with high-end hardware specifications.

3.5.2 Safety Requirements

The use of the application shall have no harms whatsoever; nor shall it have any possibility of loss or damage that might be inflicted during the use of the application. If the application crashes during any phase, there will be no change in the database.

3.5.3 Security Requirements

There isn't any security required; the application shall be standalone on the android smart phone. Product is not connected to internet to protect from malicious attacks.

3.5.4 Software Quality Attributes

3.5.4.1 Usability:

The application will be easy to operate for any user with minimum technical knowledge.

3.5.4.2 Accuracy:

To ensure reliability and correctness, there will be zero tolerance for errors in the algorithm that computes results.

3.5.4.3 Portability:

The application will run on limited number android smart phones.

3.5.4.4 Availability

The application will be available from start-up to exit, provided android smart phone is in working state and the product is installed and configured properly.

3.5.4.5 Flexibility

The design and architecture of the product will be flexible enough for catering any new requirements, if any at some later stage or for the product enhancement.

3.5.4.6 Data Integrity

If the product crashes during any phase, there will be no change in the database.

3.5.4.7 Scalability

Only one instance of the application can run at a time.

3.5.4.8 Confidentiality

The application doesn't require an confidentiality requirements.

3.5.4.9 Business Rules

BR-1: This product shall be developed to target the android users.

Chapter 4: Design and Development

4.1 Introduction

4.1.1 Purpose

The purpose of the Software Design Specification (SDS) is to document precise description of the design of Virtual Room Decor, an android application that shall be developed and released on Google Play Store.

This document is aimed to eliminate ambiguities and misunderstandings that may exist. The document will be a reference point during software design, development, testing and maintenance.

4.1.2 Scope

Virtual Reality (VR) is the next big thing in technology industry and it will take over the gaming and multimedia industry. This specification establishes the functional, performance, and development requirements for Release 1 of the Virtual Room Décor Application. The application will be designed to run in Virtual Reality (VR) environment.

4.1.3 Definitions, acronyms, and abbreviations

3D view: 3-Dimensional view

APP: Application

GUI: Graphical User Interface

UML: The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system

VR: Virtual Reality

WBS: The project management Work Breakdown Structure

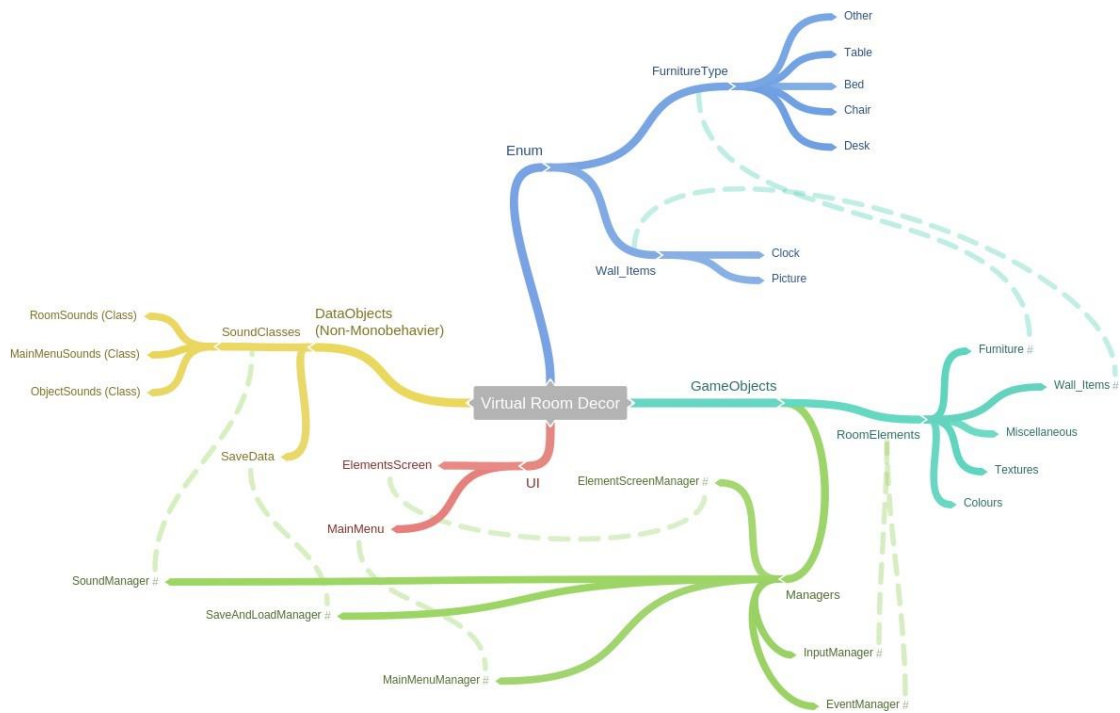


Figure 4-1-1 Mind Map

4.2 System architecture description

4.2.1 Overview

4.2.1 Game Objects

Game Objects are the fundamental objects in Unity that represent characters, props and scenery. They do not accomplish much in themselves but they act as containers for Components, which implement the real functionality.

4.2.2 UI

The main user interfaces include, the main menu and the elements menu.

4.2.3 Data Objects

These classes contain, all the reference data required for the manager classes.

4.2.4 Enums

Enumerations shall be designed for room objects.

4.3 System Block Diagram

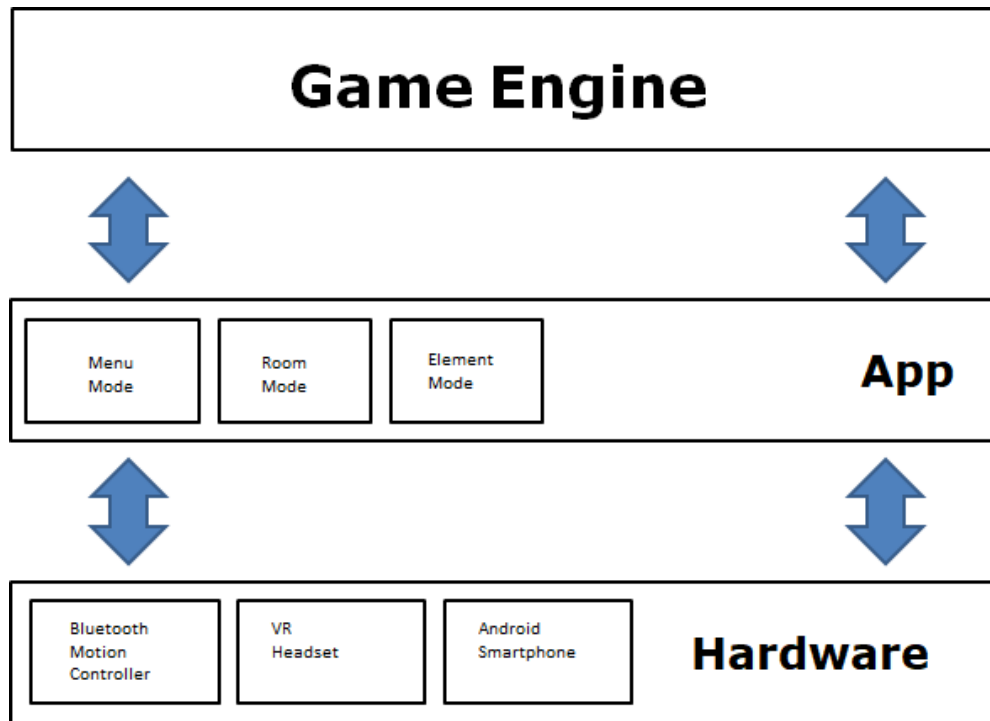


Figure 4.3-1 System Block Diagram

4.4 Component Diagram

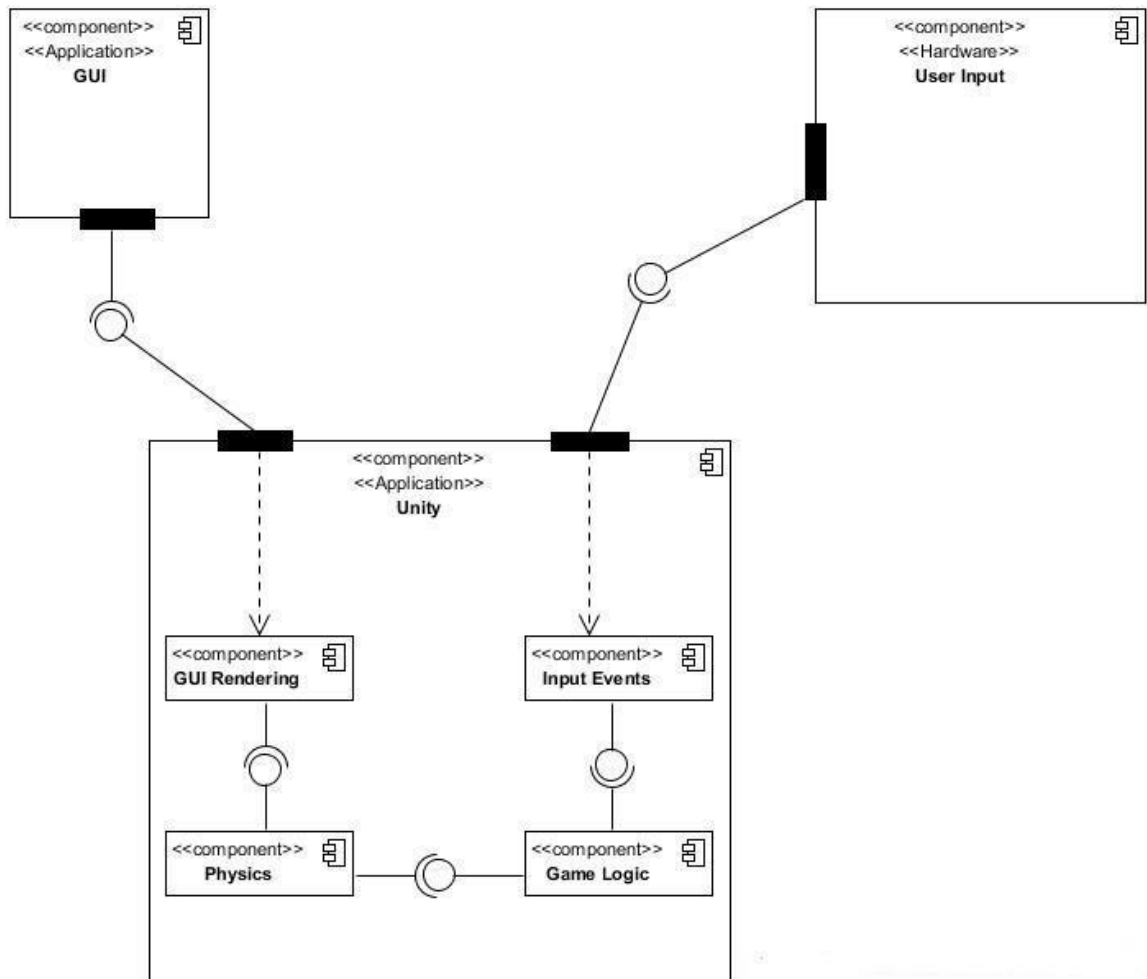


Figure 4.4-1 Component Diagram

The main components are

- User Input
- GUI
- Unity Engine

4.5 User View (Use Case Diagram)

Use cases tell how the user shall be able to interact with the system and how the system shall respond in return. They provide a means to show the functionalities of the system in a user- centric manner, without paying attention to the dynamic behavior of the system or its underlying working.

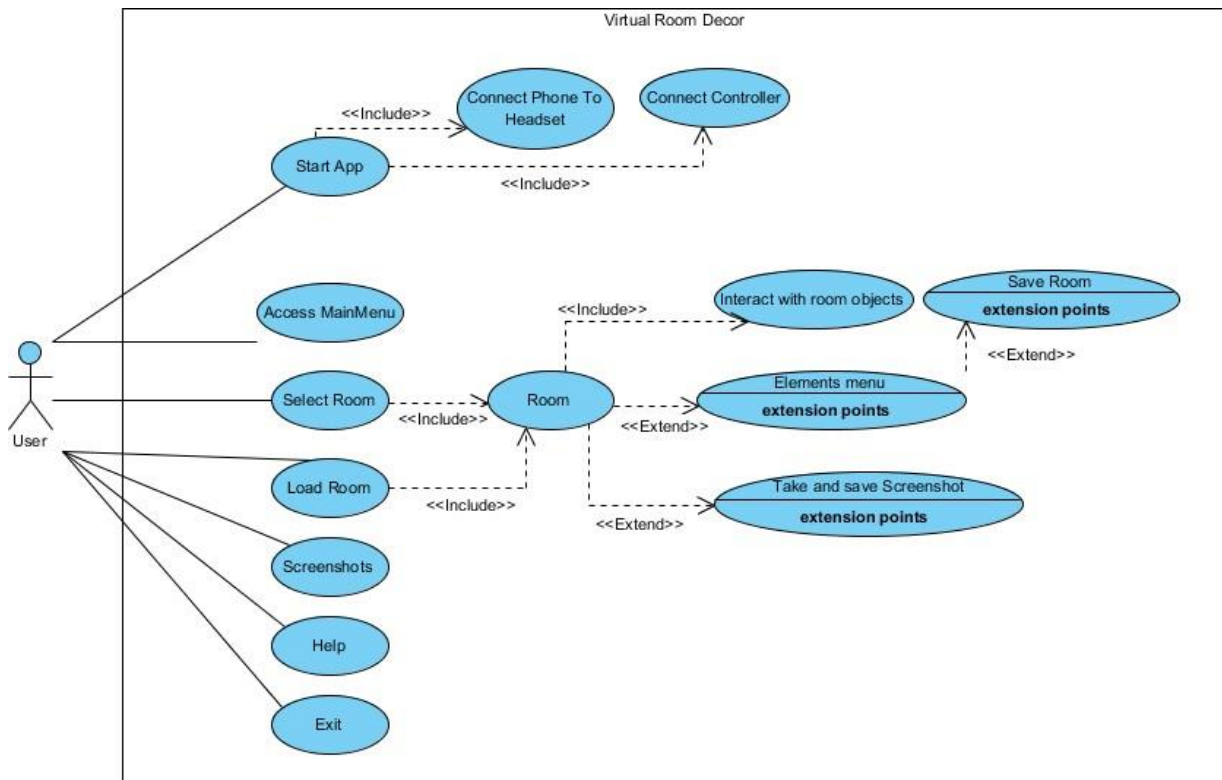


Figure 4.5-1 User View Diagram

4.6 Use Case Descriptions

The section will include the detailed description of all the individual use cases.

4.6.1 Use Case 1

Table 4.6-1 Use Case 1

USE CASE NAME	Start App
ACTOR	User
NORMAL CASE	1. Phone has application installed. 2. Bluetooth Motion Controller is connected.
ALTERNATE	1. The user will not be moved to next screen until devices are

CASE	successfully connected. 2. A prompt will show up to try again if the devices are not connected.
PRE-CONDITION	The app must be installed on the mobile device.
POST CONDITION	The phone is connected to the VR Headset.
EXTENDS	N/A
INCLUDE	Connect Controller, Connect Phone to VR Headset.
ASSUMPTIONS	Required hardware devices are connected and configured.

4.6.2 Use Case 2

Table 4.6-2 Use Case 2

USE CASE NAME	Connect Phone to Headset
ACTOR	User
NORMAL CASE	1. The phone is connected to the VR headset.
ALTERNATE CASE	N/A
PRE-CONDITION	User must start the app first.
POST CONDITION	User shall access the main menu.
EXTENDS	N/A
INCLUDE	N/A
ASSUMPTIONS	VR headset is configured properly.

4.6.3 Use Case 3

Table 4.6-3 Use Case 3

USE CASE NAME	Connect Controller
ACTOR	User
NORMAL CASE	1. Bluetooth motion controlled is connected with the phone.
ALTERNATE CASE	N/A
PRE-CONDITION	App must be installed in the phone.
POST CONDITION	Controller is connected with the phone.
EXTENDS	N/A

INCLUDE	N/A
ASSUMPTIONS	Bluetooth motion controller is configured and working properly.

4.6.4 Use Case 4

Table 4.6-4 Use Case 4

USE CASE NAME	Access Main Menu
ACTOR	User
NORMAL CASE	1. User access the main menu.
ALTERNATE CASE	N/A
PRE-CONDITION	1. App is running. 2. All hardware devices are connection.
POST CONDITION	User can now access all the main menu options.
EXTENDS	N/A
INCLUDE	N/A
ASSUMPTIONS	Phone is configured correctly for connection and application is installed on it. Required hardware devices are functioning properly.

4.6.5 Use Case 5

Table 4.6-5 Use Case 5

USE CASE NAME	Select Room
ACTOR	User
NORMAL CASE	User is prompted to select the “Select Room”, option in the main menu.
ALTERNATE CASE	N/A
PRE-CONDITION	User has access to the main menu.
POST CONDITION	User shall be directed to the room.
EXTENDS	N/A
INCLUDE	Room
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.6.6 Use Case 6

Table 4.6-6 Use Case 6

USE CASE NAME	Room
----------------------	------

ACTOR	User
NORMAL CASE	User has selected “Selected Room”, option in the main menu.
ALTERNATE	N/A

CASE	
PRE-CONDITION	User has selected “Selected Room”, option in the main menu.
POST CONDITION	User is now in the room.
EXTENDS	1. Elements Menu 2. Take and Save Screenshots
INCLUDE	N/A
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.6.7 Use Case 7

Table 4.6-7 Use Case 7

USE CASE NAME	Interact with room objects
ACTOR	User
NORMAL CASE	User selects an object in the room.
ALTERNATE CASE	User selects an object from the elements menu.
PRE-CONDITION	User must be in the room.
POST CONDITION	User can spawn, move, rotate or delete the object.
EXTENDS	N/A
INCLUDE	Room
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.6.8 Use Case 8

Table 4.6-8 Use Case 8

USE CASE NAME	Elements menu
ACTOR	User
NORMAL CASE	1. User is in room. 2. User access the elements menu by key trigger on the controller.
ALTERNATE CASE	N/A
PRE-CONDITION	User press “elements menu”, trigger key on the controller.

POST CONDITION	Elements menu pops up.
EXTENDS	Save room
INCLUDE	N/A

ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.
--------------------	--

4.6.9 Use Case 9

Table 4.6-9 Use Case 9

USE CASE NAME	Save room
ACTOR	User
NORMAL CASE	1. User has access to Elements menu. 2. User selects the save room option.
ALTERNATE CASE	N/A
PRE-CONDITION	User has accessed the Elements menu.
POST CONDITION	The room state is saved and can be access from “Load Room” option, from the main menu.
EXTENDS	N/A
INCLUDE	N/A
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.6.10 Use Case 10

Table 4.6-10 Use Case 10

USE CASE NAME	Take and save Screenshots
ACTOR	User
NORMAL CASE	1. User is in the room. 2. User trigger the key on the controller, programmed to take screenshot.
ALTERNATE CASE	N/A
PRE-CONDITION	User is in room.
POST CONDITION	Screenshot is saved, which can be accessed from the “Screenshots” option, from the main menu.

EXTENDS	N/A
INCLUDE	N/A
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.6.11 Use Case 11

Table 4.6-11 Use Case 11

USE CASE NAME	Load Room
ACTOR	User
NORMAL CASE	1. User has access to main menu. 2. User selects the “Load Room” option.
ALTERNATE CASE	N/A
PRE-CONDITION	User has access to the main menu.
POST CONDITION	User can now load a previously saved room.
EXTENDS	N/A
INCLUDE	Room
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.6.12 Use Case 12

Table 4.6-12 Use Case 12

USE CASE NAME	Screenshots
ACTOR	User
NORMAL CASE	1. User has access to main menu. 2. User selects the “Screenshots” option.
ALTERNATE CASE	N/A
PRE-CONDITION	User has access to the main menu.
POST CONDITION	User shall view all the saved screenshots.
EXTENDS	N/A
INCLUDE	N/A
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.6.13 Use Case 13

Table 4.6-13 Use Case 13

USE CASE NAME	Help
ACTOR	User
NORMAL CASE	1. User has access to main menu.

	2. User selects the “Help” option.
ALTERNATE CASE	N/A
PRE-CONDITION	User has access to the main menu.
POST CONDITION	User guide shall be displayed to the user.
EXTENDS	N/A
INCLUDE	N/A
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.6.14 Use Case 14

Table 4.6-14 Use Case 14

USE CASE NAME	Exit.
ACTOR	User
NORMAL CASE	1. User has access to main menu. 2. User selects the “Exit” option.
ALTERNATE CASE	N/A
PRE-CONDITION	User has access to the main menu.
POST CONDITION	The application is terminated.
EXTENDS	N/A
INCLUDE	N/A
ASSUMPTIONS	App is functioning properly. Required hardware devices are functioning properly.

4.7 Implementation View (Class Diagram)

These diagrams will show the main classes of our systems. It will also show the attributes and functions associated with each class. Class diagrams are integral for the description of low level components of the software that include data storage and state details. Classes provide the means for the system to perform its functions.

4.7.1 Class Diagram

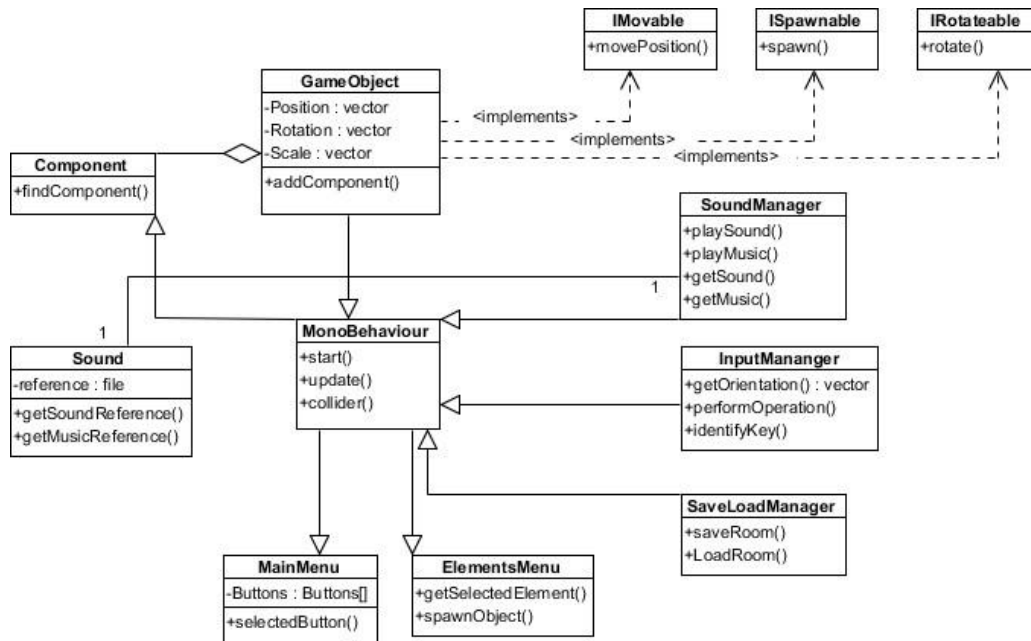


Figure 4.7-1 Class Diagram

4.7.2 Class Diagram Description

Table 4.7-1 Class Diagram Description

Name	Description
GameObject	This class is responsible for defining a unity game object. This class contains a series of properties which can be used to customize a unity game object. These properties are known as components. An example of a game object in this game is a racket. A simple example of a component is a „transform“ which defines a game object’s position, rotation and scale.
Component	This class is a base class for everything that can be attached to a game object. This class provides the facility to store the entire component connected to a game object in as a single structure.
MonoBehaviour	This class inherits form a component. This means that it can be attached to a game object. This class defines the behavior of the game object it is attached to. Multiple MonoBehaviors can be attached to a game object.
ISpawnable	This interface is implemented by GameObjects. To ensure those objects to be spawn able.
IMoveable	This interface is implemented by GameObjects. To ensure those objects to be moveable.
IRotateable	This interface is implemented by GameObjects. To ensure those objects to be rotate able.
SoundManager	This class is responsible for managing all in-app sounds. It is associated with sound

	class for reference data.
InputManager	This class responsible for exposing all kinds of user input for example accelerometer data, gyroscope data, touch input.
SaveLoadManager	This class Extends from MonoBehavior. This class saving and loading the state of a room.
MainMenu	This class is responsible for providing the user with a menu to navigate through different options in the game. It displays different buttons depending on the situation will trigger events accordingly.
ElementsMenu	This class is responsible for providing user an interface, to navigate thorough object that can be spawned into the room.
Sound	The class has the all the reference data of the sounds of the games. It is associated with SoundManager , to provide the required data.

4.8 Sequence Diagrams

Sequence diagrams show how different objects are involved in the completion of a functionality of the system. They have a unique format that allows the reader to see how many objects are used and for how long for the completion of a system requirement.

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

4.8.1 Select Room

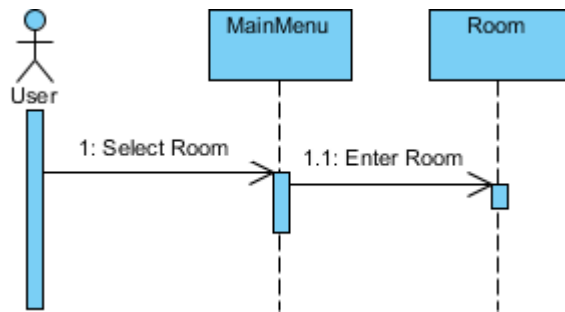


Figure 4.8-1 Select Room

4.8.2 Spawn Room Object

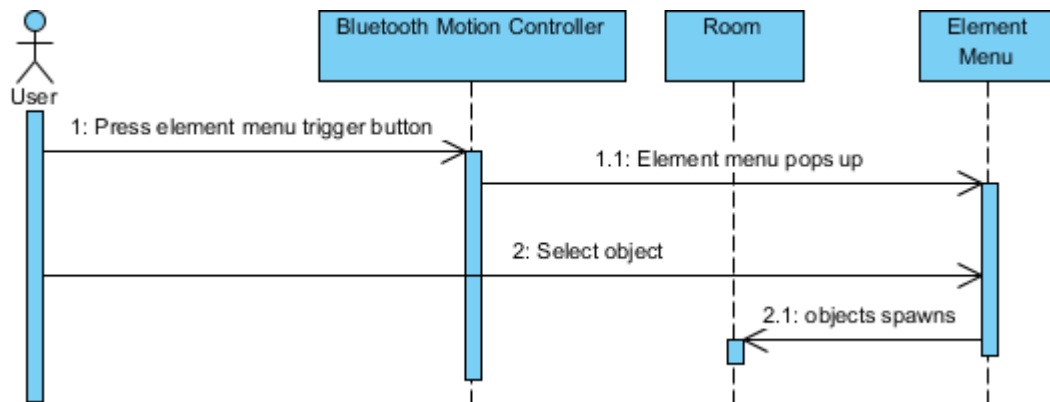


Figure 4.8-2 Spawn Room Object

4.8.3 Move/Delete/Rotate Room Object

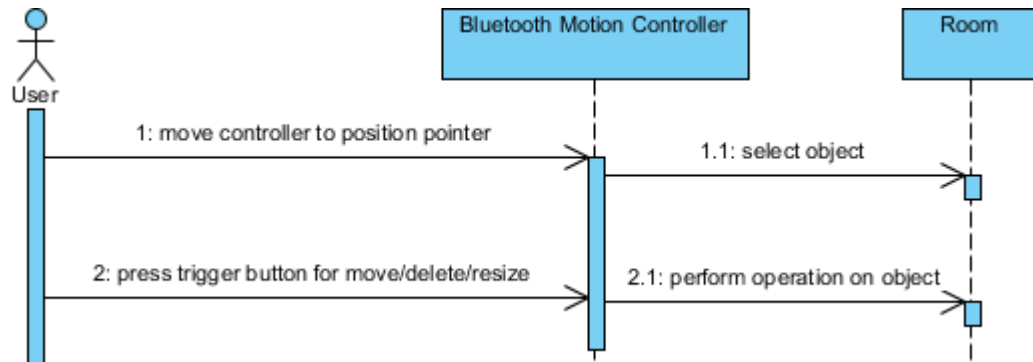


Figure 4.8-3 Move/Delete/Rotate Room Object

4.8.4 Save Room

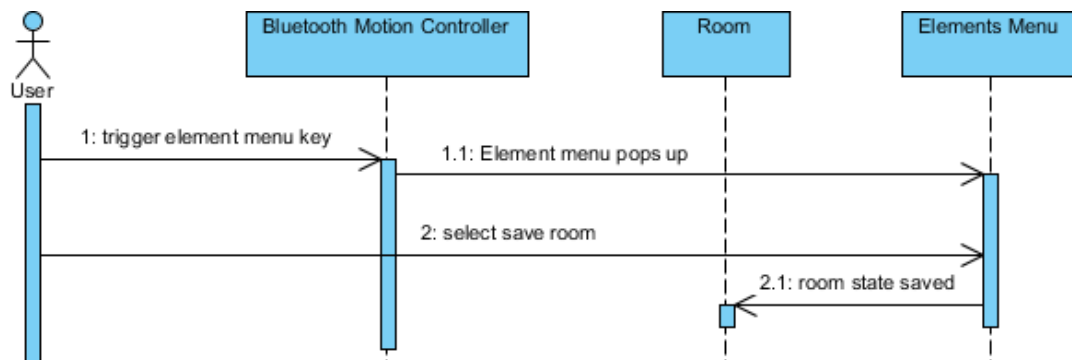


Figure 4.8-4 Save Room

4.8.5 Load Room

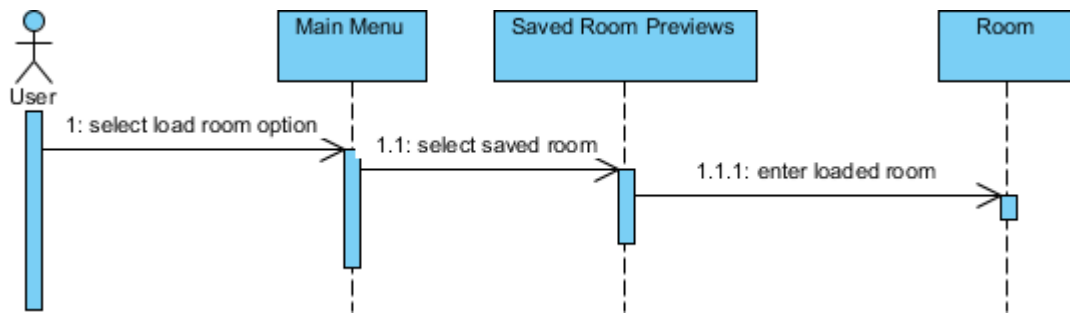


Figure 4.8-5 Load Room

4.8.6 Take and Save Screenshot

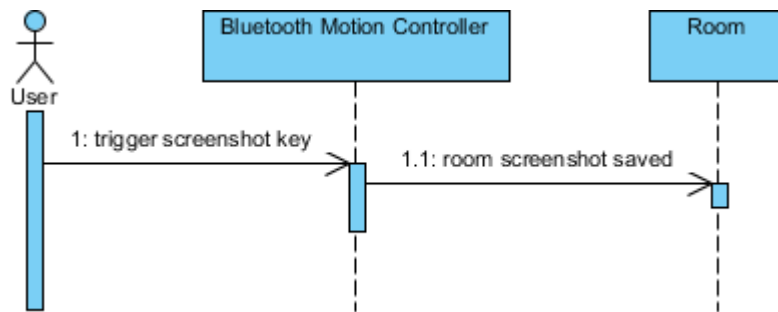


Figure 4.8-6 Take and Save Screenshot

4.9.7 View Saved Screenshots

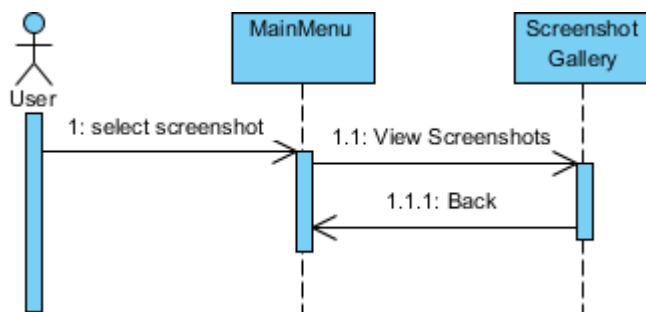


Figure 4.8-7 View Saved Screenshots

4.8.9 Access User Guide

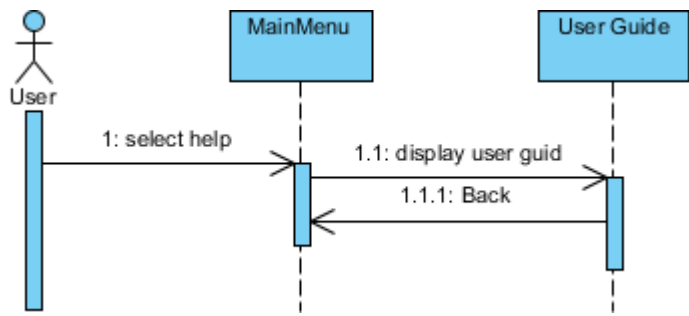
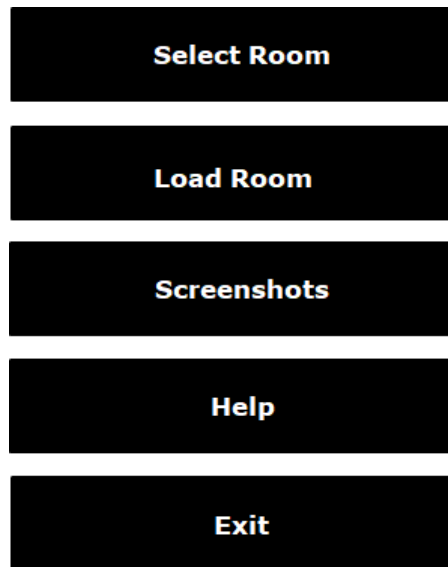


Figure 4.8-8 Access User Guide

4.9 User Interface

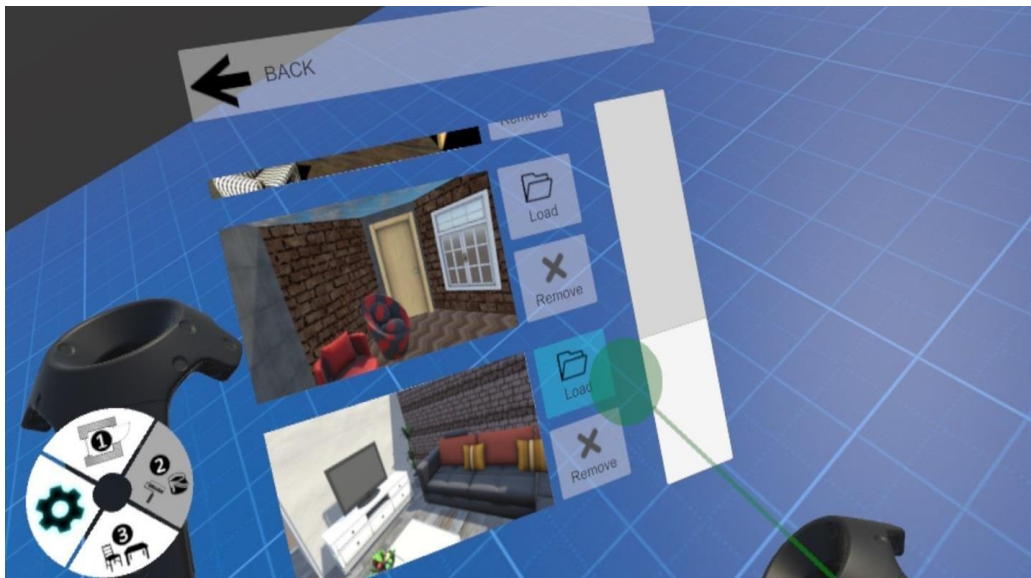
4.9.1 Main Menu



4.9.2 Elements Menu



4.9.3 Load Room



4.10 Detailed description of components

4.10.1 UI

Identification	<p>Name: UI (User Interface) Location: Application layer</p>
Type	Component
Purpose	To display the app.
Function	<p>What the component does: The complete game environment is displayed by this component.</p>
Subordinates	<p>Constituents of the component: UI basically contains all the game objects which includes:</p> <ol style="list-style-type: none"> 1. Furniture 2. Wall Items 3. Textures 4. Colors 5. Miscellaneous Items <p>Functional Requirements: REQ-1: The application must be properly installed on the android smartphone. REQ-2: The different options available shall be:</p> <ul style="list-style-type: none"> • Select Room • Load Room • Screenshots • Help • Exit <p>REQ-26.1: User shall select Help option from the main menu. REQ-26.2: The help screen shall contain the user guide. REQ-10: The user shall be in room. REQ-11.1: The element menu trigger key on the controller shall be pressed to make the element menu appear. REQ-11.2: The element menu trigger key on the controller shall be pressed to make the element menu disappear. REQ-12.1: The user shall select elements from the element menu. REQ-12.2: The selected element shall be placed to the room, if place able element.</p>

	<p>REQ-12.3: The selected element shall be applied to the room, if applicable element.</p> <p>REQ-12.4: The elements shall be categorized as follows:</p> <ul style="list-style-type: none"> • Colors • Textures • Furniture • Wall Items • Miscellaneous <p>REQ-13: The elements menu shall have the save room option</p>
Dependencies	UI is dependent on unity engine. The rendering component of the unity will be providing input to the UI.
Interfaces	<p>External interface requirement for UI</p> <p>An output display screen which it will be using to display all the objects created by the unity engine and the entire menu and app mode screen.</p>
Resources	<ol style="list-style-type: none"> 1. It will be using graphics engine of the system as per unity requirements. 2. Android smartphone screen
Processing	The processing required of this component is to respond and display results of player input during the game and also during navigation.
Data	User input (touch, hand gestures, head gestures) , and motion controller

4.10.2 Managers Classes

Identification	Name: Manager Classes Location: Application layer
Type	Class
Purpose	The manager classes are responsible for coordination of different events. The following are manager classes: 1.InputManager Responsible for detection of input in phone orientation, head movement, hand gesture and controller inputs.

	<p>2.SoundManager Responsible for managing in app sounds.</p> <p>3.MainMenuManager Detects selection on main menu screen and coordinate responses.</p> <p>4.SaveLoadManager Responsible for saving and loading the state of a room, in app.</p> <p>5.EventManager Responsible for managing the event triggered on objects.</p>
Function	<p>What the component does: Each manager class is related to handling specific types of inputs.</p>
Subordinates	<p>Constituents of the component: The component has no sub-components.</p> <p>Functional Requirements:</p> <p>REQ-1: The application must be properly installed on the android smartphone.</p> <p>REQ-2: The different options available shall be:</p> <ul style="list-style-type: none"> • Select Room • Load Room • Screenshots • Help • Exit <p>REQ-3.1: The Select Room option shall be selected in the main menu.</p> <p>REQ-3.2: A screen for Select Room option shall be displayed.</p> <p>REQ-4: Select Room screen shall provide the user option to select room dimensions.</p> <p>REQ-5: The user shall only be able to choose from given room dimension options.</p> <p>REQ-7: A room shall be selected from Select Room screen.</p> <p>REQ-8: A room shall be selected from Load Room screen.</p> <p>REQ-14: The user shall be in a room.</p> <p>REQ-15: The user shall be able to select objects and move them around the room.</p>

	<p>REQ-16: The user shall be able to rotate the objects.</p> <p>REQ-17: The user shall be able to remove objects from the room.</p> <p>REQ-18: The user shall be in a room.</p> <p>REQ-19: The user shall trigger the elements menu to appear.</p> <p>REQ-20.1: The user shall select the save room option in the elements menu.</p> <p>REQ-20.2: The room shall be saved and shall be available in the load room screen.</p> <p>REQ-21: The user shall select a room from the load room screen.</p> <p>REQ-22: The selected room shall be loaded and displayed</p> <p>REQ-23: User shall be in a room.</p> <p>REQ-24.1: User shall trigger the assigned key on the controller to take screenshot.</p> <p>REQ-24.2: The screenshot will be saved at the current view of the camera in the room.</p> <p>REQ-25: User shall access all saved screenshots from screenshot screen, accessed from main menu.</p>
Dependencies	<p>Components using this component: The sub component input events of the unity component will get its input from this module. This input will be further processed by input events component.</p>

Interfaces	<p>The external hardware interfaces interacting with this component will be:</p> <ul style="list-style-type: none">• Android smartphone <p>Error messages: Connection not established.</p>
------------	---

Resources	The resources used by this component are touch screen (for touch inputs) and sensors (for racket movement), and motion controller.
Processing	The processing required for this component is receiving the user input and giving this input to the input event module of the unity game engine
Data	User inputs via controller and headset.

4.10.3 Unity Engine

Identification	Name: Unity Engine Location: Processing layer
Type	Module
Purpose	Unity is a game development ecosystem. It is a powerful rendering engine fully integrated with complete set of intuitive tools and rapid workflows to create interactive 3D and 2D content. Unity built in physics engine provides component that handles physics simulation with just a few parameters setting. Physics can be controlled from scripts
Function	What the component does: The main functions of this module are as follows: <ol style="list-style-type: none"> 1. Handle game physics 2. Game play rendering 3. Manage game resources on system 4. Manage all other major components of game

Subordinates	<p>Constituents of the component: The other components using this component are:</p> <ol style="list-style-type: none"> 1. UI 2. User Input 3. Network Manager <p>Functional requirements: REQ-10: The user shall be in room.</p> <p>REQ-11.1: The element menu trigger key on the controller shall be pressed to make the element menu appear.</p> <p>REQ-11.2: The element menu trigger key on the controller shall be pressed to make the element menu disappear.</p> <p>REQ-12.1: The user shall select elements from the element menu.</p> <p>REQ-12.2: The selected element shall be placed to the room, if place able element.</p> <p>REQ-12.3: The selected element shall be applied to the room, if</p>
--------------	---

	applicable element.
Dependencies	Components using this component: Player input controls GUI rendering
Interfaces	The external interfaces interacting with unity game engine are <ol style="list-style-type: none"> 1. touch screen and sensors to get input data 2. Speakers 3. Motion Controller
Resources	The resources used by this module are RAM Graphic Memory, CPU Usage, Sensors
Processing	The processing done by this module is that it: <ul style="list-style-type: none"> • Initiates game, game objects and all required global variables • Allocates required system resources for game • Manages memory requirements
Data	Float values, integer values, strings

4.11 Reuse and relationships to other products

Virtual Room Decor is completely a new project. It is not an extension of already existing projects. However, materials already available on internet are being used to aid the modeling and enhance the graphics of the game. Special tools including „Unity 3D“ are used in our project.

Existing and available models are not used in our project as our project is according to our own requirements and choices.

4.12 Design decisions and tradeoffs

The design was kept as simple as possible, facilitating the UI. Graphics and hardware selection was top-notch in order to provide high quality VR experience.

4.13 Pseudo code for components

```
using UnityEngine;

public class KeyCodeExample : MonoBehaviour

{

    void Update()

    {

        if (Input.GetKeyDown(KeyCode.Space))

            Debug.Log("Space key was pressed.");

        if (Input.GetKeyUp(KeyCode.Space))

            Debug.Log("Space key was released.");

    }

}
```

```
using UnityEngine;
using UnityEngine.VR;

public class UpdateEyeAnchors : MonoBehaviour
{
    GameObject[] eyes = new GameObject[2];
```

```

string[] eyeAnchorNames = { "LeftEyeAnchor", "RightEyeAnchor" };

void Update()
{
    for (int i = 0; i < 2; ++i)
    {
        // If the eye anchor is no longer a child of us, don't use it
        if (eyes[i] != null && eyes[i].transform.parent != transform)
        {
            eyes[i] = null;
        }

        // If we don't have an eye anchor, try to find one or create one
        if (eyes[i] == null)
        {
            Transform t = transform.Find(eyeAnchorNames[i]);
            if (t)
                eyes[i] = t.gameObject;

            if (eyes[i] == null)
            {
                eyes[i] = new GameObject(eyeAnchorNames[i]);
                eyes[i].transform.parent = gameObject.transform;
            }
        }

        // Update the eye transform
        eyes[i].transform.localPosition = InputTracking.GetLocalPosition((VRNode)i);
        eyes[i].transform.localRotation = InputTracking.GetLocalRotation((VRNode)i);
    }
}

```

Chapter 5: Project Test and Evaluation

5.1 Project Test and Evaluation

5.1.1 Introduction

This document proposes the plan used for testing the application “Virtual Room Decor”. The test plan proposed will ensure that the game will be working in a manner which was intended.

The purpose of this document is to describe the tests that will be conducted on the implementation of “Virtual Room Decor”. The aim is to check that the modules (system features) such as

- Displaying the graphical features of the app
- Connectivity configuration between devices
- Execution of main menu
- Communication between the devices
- Accurate virtual movements
- Room design elements performance

are performing the required operations. The user should be able to experience a 3D virtual environment error-free.

This document further includes the plan, approach, test items and the test deliverables. The pass/fail criteria are also defined for the items.

5.1.2 Test Items

The test items selected for testing include the following

- Performance
- Interface
- User control

5.1.2.1 Features to be Tested

The features of our game include the functionality mentioned in the use cases. Following features are to be tested keeping in view the test items and system features aforementioned

- Connection between the devices
- Main menu navigation
- Room elements and UI
- User interactions with the virtual environment
- Interaction of 3D elements in the room
- Saving and loading the state of the room
- Screenshot capturing and saving to device

5.1.3 Approach

Functional Testing will focus on each use case that is included in the version currently being worked on. Testing will mainly consist of execution of test cases written to address the gap identified. It will focus on inputs, outputs and system changes due to the actions.

The testing strategy for Virtual Room Decor will be alpha testing. Our project is in modules so we will start the testing phase by testing the modules separately and then step by step integrating modules to test them with each other i.e. integration testing and then the complete application is tested as a whole in system testing. Black Box testing technique will be used for testing functionality of each module.

5.1.4 Pass/Fail Criteria for Test Items

The criteria are as follows

- The pre-conditions are met
- Inputs are carried out as specified
- Test case will pass if it produces the desired output for a specified input

- Test will fail otherwise

5.1.5 Test Suspension Criteria

Testing procedure will be suspended whenever a defect is found that restricts further testing. A corrective measure will be applied depending upon the criticality of the defect and testing will be resumed.

Efforts have been made to remove all and every chance of failure but there are certain unpredictable factors such as connectivity issues, corrupt input data, or system failure that may lead to some issues. Error handling is applied more deeply to cover all these issues but unforeseen circumstances may happen.

5.2 Test Deliverables

5.2.1 Testing tasks

- Develop Test Cases.
- Execute tests based on the test cases developed.
- Report defects during tests if any.
- Manage the changes made after testing.

5.2.2 Test cases

5.2.2.1 Displaying The Main menu

Table 5-1 Displaying The Main menu

Test Case ID	TC 1
Test Procedure	The application is launched in the VR Display mode
Testing Technique	Black Box Testing

used	
Preconditions	The application has successfully launched
Expected output	User is able to see a main menu, that can be navigated
Actual output	Main menu is displayed
Status	PASS

5.2.2.2 Connectivity Configuration Between Devices

Table 5-2 Connectivity Configuration Between Devices

Test Case ID	TC 2
Test Procedure	App is launched on the phone and is connected to the Gear VR headset. Controller is connected to the phone
Testing Technique used	Black Box Testing
Preconditions	Controller is connected to the phone Application is launched
Input values	

Expected output	Connection has been established between all the required devices
Actual output	Connection has been established between all the required devices
Status	PASS

5.2.2.3 Main Menu Navigation

Table 5-3 Main Menu Navigation

Test Case ID	TC 3
Test Procedure	Check all the UI elements displayed, to see if functioning correctly.
Testing Technique used	Black Box Testing
Preconditions	Devices are connected. Application is launched. Main menu scene is displayed
Input values	Selection with controller movement and button trigger in the Main menu scene
Expected output	UI Panel prompts and scene changes
Actual output	All UI panel prompts functioning as developed.
Status	PASS

5.2.2.4 Room Interactions

Table 5-4 Room Interactions

Test Case ID	TC 4
Test Procedure	User Interaction with Room UI and elements
Testing Technique used	Black Box Testing
Preconditions	User enter the room scene via main menu
Input values	Controller movements and button interaction with room UI and elements.
Expected output	Correct functioning of UI and iteration with room elements.
Actual output	The elements work perfectly but rotation is very sensitive. UI working perfectly fine.
Status	PASS

5.2.2.5 Room Elements Interactions

Table 5-5 Room Elements Interactions

Test Case ID	TC 5
Test Procedure	User interaction with room elements such as furniture.

Testing Technique used	Black Box Testing
Preconditions	User enter the room scene via main menu
Input values	Controller movements and button interaction and elements.
Expected output	Movement , rotation and deletion of furniture objects (Room elements)
Actual output	Some elements move perfectly , some wobble , rotation is very sensitive. Overall everything looks fine.
Status	PASS

5.2.2.6 Saving the Room

Table 5-6 Saving the Room

Test Case ID	TC 6
Test Procedure	Use access the elements menu and hit save room button
Testing Technique used	Black Box Testing
Preconditions	User enter the room scene via main menu. Elements menu is displayed.
Input values	Controller movements and button interaction with elements menu.

Expected output	Room state is saved.
-----------------	----------------------

5.2.2.7 Taking Screenshot and saving to the device

Table 5-7 Taking Screenshot and saving to the device

Test Case ID	TC 7
Test Procedure	Designated key on controller is pressed to take screenshot.
Testing Technique used	Black Box Testing
Preconditions	User enter the room scene via main menu
Input values	Dedicated screenshot key is pressed.
Expected output	Screenshot is saved to the device

5.3 Environmental Needs

5.3.1 Hardware

- Supported android smartphone for Gear VR
- Gear VR Headset with Controller

5.3.2 Software

- Android operating system 6.0 (Marshmallow) or above.

5.4 Responsibilities, Staffing and Training Needs

5.4.1 Responsibilities

All developers of the project are responsible for the completion of all components testing and integration testing tasks.

5.4.2 Staffing and Training Needs

Basics knowledge of testing strategies and techniques is needed for the testing of the project.

Techniques such as Black Box testing, integration testing should be known to developers.

All the developers will be testing each other's work and will be actively participating in the development and testing of the project simultaneously.

Chapter 6: Future Work

VR and AR are slowly infiltrating our industries. We're going to see this technology influencing everything from designing the latest must-have sneakers, to visualizing buildings before a brick has been laid, to producing more and more immersive alternative reality experiences.

That means we're going to see a lot more people working in 3D, as more and more 3D assets and objects are required for these environments.

The challenge is that traditionally, the technical capabilities required to move to a 3D workflow are developed over a long period. There's a learning curve required to master specialized software, you need to think in polygons, edges, and vertices, and you have to have a lot of experience practicing before you can produce work to a high standard.

Many of the people involved in designing shoes, houses, or art works are still more comfortable working with physical substances, such as clay or paper, and less familiar with using technical computer software.

This is where VR can help bridge the gap. Using a VR headset and controllers makes the design process feel more hands on, more instinctive. Rather than using a mouse pointer and selecting commands from a menu, you can stretch, or pull, or cut, whatever's in front of you intuitively, much as you would in real life.

It's this easier, more natural process that means content creation in VR will be the way we create in the future.

It has the potential to make 3D far more accessible to people that found its technical barriers too challenging in the past.

Bibliography

1. Google Daydream
2. Virtual Reality Development
3. Unity 3D Development
4. Support Android Smart phones for Google Daydream
5. Support Android Smart phones for Gear VR
6. Gear VR

[1] <https://vr.google.com/daydream/>

[2] <https://www.udemy.com/build-virtual-reality-games-for-android-and-iphone/>

[3] <https://unity3d.com/learn/tutorials>

[4] <https://vr.google.com/daydream/smartphonevr/phones/>

[5] https://en.wikipedia.org/wiki/Samsung_Gear_VR

[6] <http://www.samsung.com/global/galaxy/gear-vr/>

Appendix A: Project Proposal

Virtual Room Decor

Extended Title: Nil
Brief Description of the Project/Thesis with Salient Space: <p>An android application, developed in Unity 3D, that runs in Virtual Reality (VR) environment. It shall provide the user a virtual room, where the user can decorate and furnish his room. Paint walls, paste wallpapers, add furniture to the room, add in doors and windows, change color and texture of all elements. Do all this virtually without moving a budge. Visualize their dream room. All of this functionality via simple drag-and-drop interface.</p>
Scope of Work: <p>The following are the basic <i>features</i> that will be provided to the users by the application.</p> <ul style="list-style-type: none">• Design rooms from scratch.• Select different room dimensions , grid based.• Select and edit ready-made designed rooms.• Place furniture objects, doors, windows and a lot of other fun stuff into your room.• Apply textures for walls, floor and furniture objects.• Interact with objects in your room: Throw a pillow onto the couch or lift your cupboard and place it somewhere else.• Take screenshots and photos of your room from every angle.• Save and load your rooms. <p><u>Extended Features</u></p> <ul style="list-style-type: none">• Import Custom 3D Models and Textures.
Academic Objective: <p>At the end of the project we'll be able to have command over</p> <ul style="list-style-type: none">• Unity 3D Engine• Advance C# Programming• 3D Modeling

- 3D Image Rendering
- Connecting and Programming Hardware APIs
- Create Application for Virtual Reality (VR)

Application /End Goal Objective:

The following objectives are expected of the project in the end:

- Make it a marketable product.
- Product can be used at home.
- Product to be polished for entertainment purpose.
- Users can virtually design their dream rooms.
- Users can visualize their room settings and rearrange and refurnish them to see how they look after the changes.
- Interior designers in training can virtually design and furnish rooms for training purpose.

Previous Work Done on The Subject:

No such work has been done in Virtual Reality in MCS.

Material Resources Required:

Software

- Unity 3D 5.3.4
- C#

Hardware

- Google Daydream
- Google Daydream supported Android Phone

No of Students Required: 3

ASC Kashan Sardar

NC Asad Ali Khan

Special Skills Required: Nil