1. INTRODUCTION

Today the world of entertainment is conquered by mobile games. Whenever and wherever possible we find people bending and craning their necks to win in their gaming world. Reasons vary. Be it to kill time or to pass time, we invariably depend on games. Many researchers found out that games increase mental flexibility. They also play a major role in enhancing the IQ levels of an individual.We can give any message through games effectively When players play against the clock, it will increase their mental ability and it will enable them to think and make quick and logical decisions. Mobile games can be categorized into many types:

1.1 Action

The most varied category, action games can be shooting galleries, old-school arcade games or fast moving fighters. All require sharp reflexes. A good example would be the intense World War II shooter Siberian Strike.

1.2 Adventure

Adventure games are often a blend of reflex testing and puzzle solving. The pace is a couple degrees slower than an action game. Glyder is one of the better examples of adventure games.

1.3 Card

One of the most popular genres, card titles include solitaire, poker and other titles. There are plenty on the cell phone, but there are some stinkers out there. Be sure to check out my Best Card Games before you buy.

1.4 RPGs

Role playing games, or RPGs, are complex, involving journeys. They have heavy storylines, diverse characters and hours of play. Many mobiles don’t have the tech power or the memory space to handle RPGs, so you’ll find them almost exclusively on the more

high-end smartphones Zenonia is a popular RPG.

1.5 Sports

Sports covers real-life activities like basketball and baseball, as well as more unusual titles. While they can be complex, most mobile sports games focus on one particular part of the experience. For instance, the super casual Paper Toss is strictly about making hoops.

1.6 Strategy

An emphasis on forethought and planning, strategy games are about taking turns on a battlefield.

The game we have developed falls under the category of sports. Here in this game, the user navigates through the complex labyrinth and reaches his destination, which will take him to the higher level. The virus strewn along the path add further complexity to the game. The navigating experience will be exciting and thrilling got the user.

2. SYSTEM ANALYSIS

2.1 Problem definition

The existing games are for the sake of entertainment.As they are not giving any message which is helpful to people.

2.2 Proposed System

In the proposed game a ball is moving from a starting point to the destination(end point)while moving through obstacles(virus) whenever the ball touches the virus he will fail the game . so that we should be careful while going from one point to another point and also level by level the difficulty of the game gets increased which increases the users concentration. The game uses a non-visible component called the orientation sensors,which helps the ball to navigate in the direction the user tilts.

2.2.1 Orientation sensor

An orientation sensor is a non-visible component that reports the following three values, in degrees.

Roll: 0 degree when the device is level, increasing to 90 degrees as the device is tilted up onto its left side, and decreasing to −90 degrees when the device is tilted up onto its right side.

Pitch: 0 degree when the device is level, increasing to 90 degrees as the device is tilted so its top is pointing down, then decreasing to 0 degree as it gets turned over. Similarly, as the device is tilted so its bottom points down, pitch decreases to −90 degrees, then increases to 0 degree as it gets turned all the way over.

Azimuth: 0 degree when the top of the device is pointing north, 90 degrees when it is pointing east, 180 degrees when it is pointing south, 270 degrees when it is pointing west, etc.

2.2.2 Proposed System Advantages

- The proposed game will educate people about the present situation(pandemic situation)
- The proposed game has many gates and obstacles along the path which makes it difficult for the user and increases his concentration.
- The game occupies a minimal size of 2Mb

3. SYSTEM DESIGN AND IMPLEMENTATION

3.1 Requirements

3.1.1 Hardware Requirements

We can use this application in android mobiles.

3.1.1 Configuration of a system was

3.1.1.1 REDMI ANDROID MOBILE:

Android version : 9.0 pie

Installed memory : 2GB

ROM : 16GB

CPU : Quad core 1.2GHz

3.1.1.2 SAMSUNG ANDROID MOBILE:

Android version :9.0 pie

Installed memory : 2GB

ROM : 16GB

CPU : Quad core 1.2GHz

3.1.1.3 MICROMAX ANDROID MOBILE:

Android

Version:9.0 pie

ROM : 16GB

CPU : Quad core 1.2GHz

3.1.1.4 LENOVO ANDROID MOBILE:

Android version : 9.0 pie

Installed memory : 2GB

ROM : 16GB

CPU : Quad core 1.2GHz

3.1.2 Software Requirements

Operating System : Android

Web Application : MIT App Inventor

Database : Tiny DB

Web Server : Google App Engine

System Development Kit : AppInventor_Setup_v_2_2

3.2 Design using UML Diagrams

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules. An UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

3.2.1 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message. Figure 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.

Fig. 3.1: Sequence Diagram

3.2.2 Usecase Diagram

Use cases are used during the requirements elicitation and analysis to represent the functionality of the system. Figure 3.3 shows that the system consists of use cases like open quiz, signup, sign in, view topic, select topic, open questions, maintain database, prepare questions, quit quiz, save answers. The actors are participant and admin.

3.2.2.1 Design using UML Diagrams:

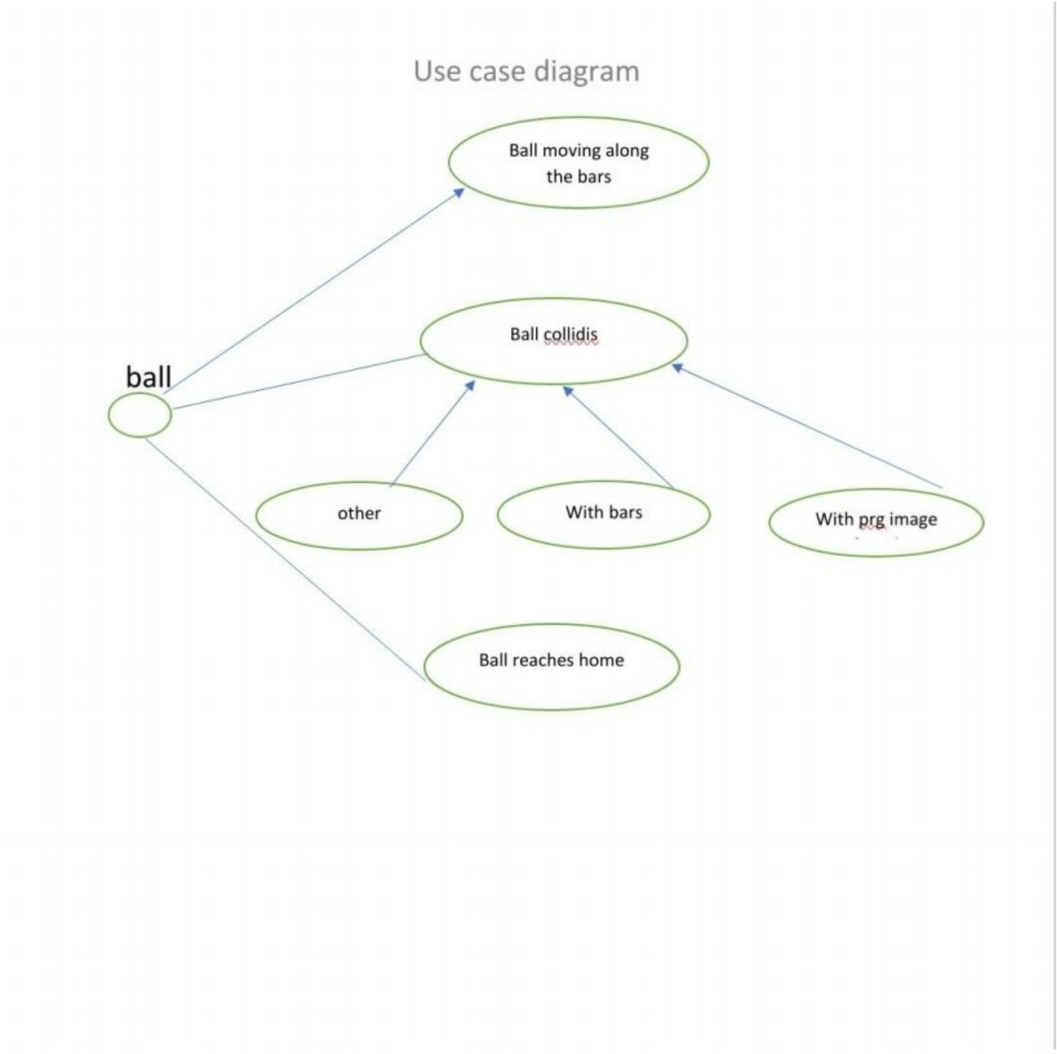


Fig. 3.2: Usecase Diagram

3.2.3 Class Diagram

A class diagram represents the set of attributes, operations and their collaborations. Figure shows the association relationship between quiz administrator, participant, score database, topic, and quiz database.

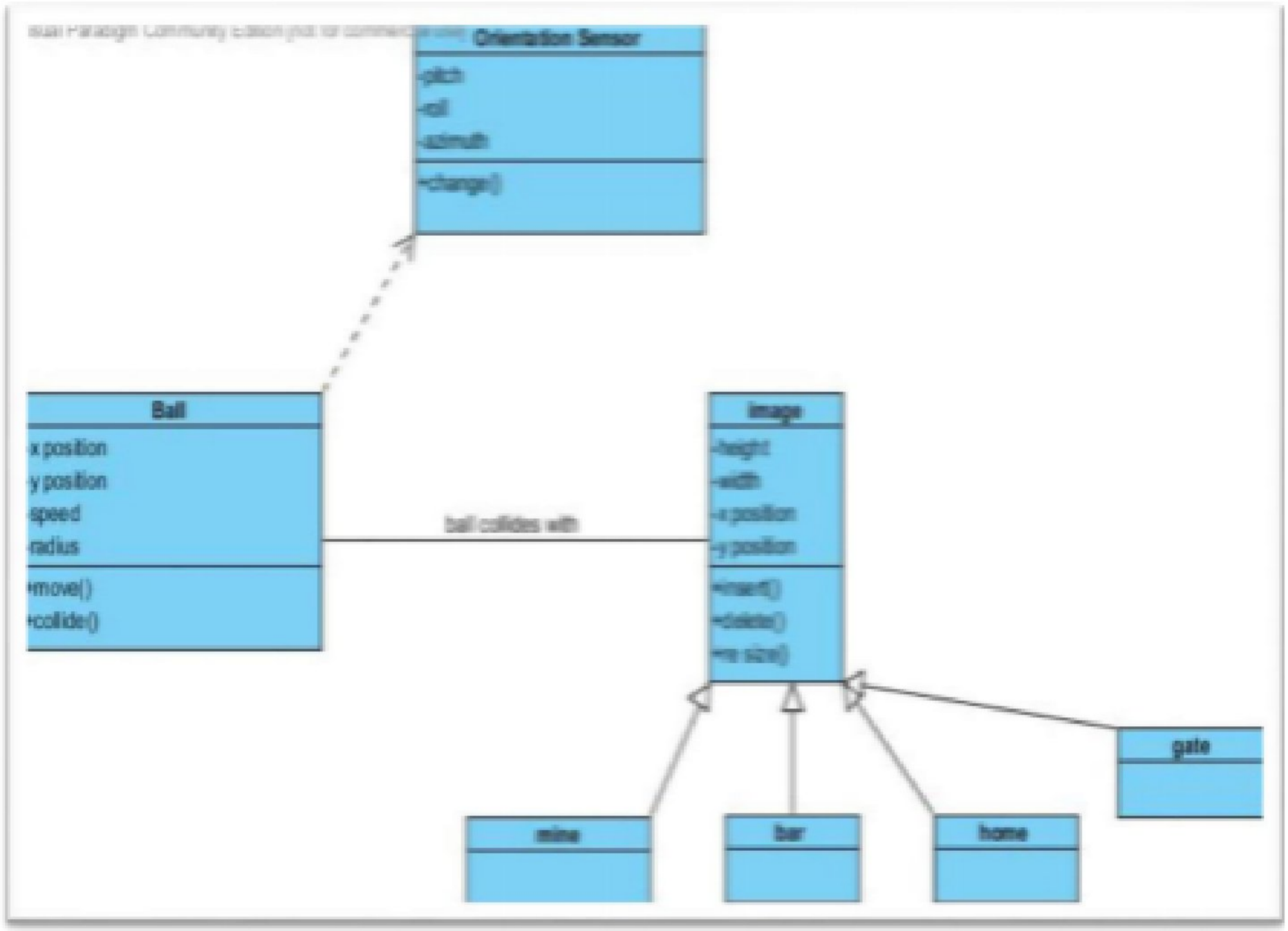


Fig. 3.3: Class Diagram

3.2.4 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Figure shows the flow from one activity to another activity.

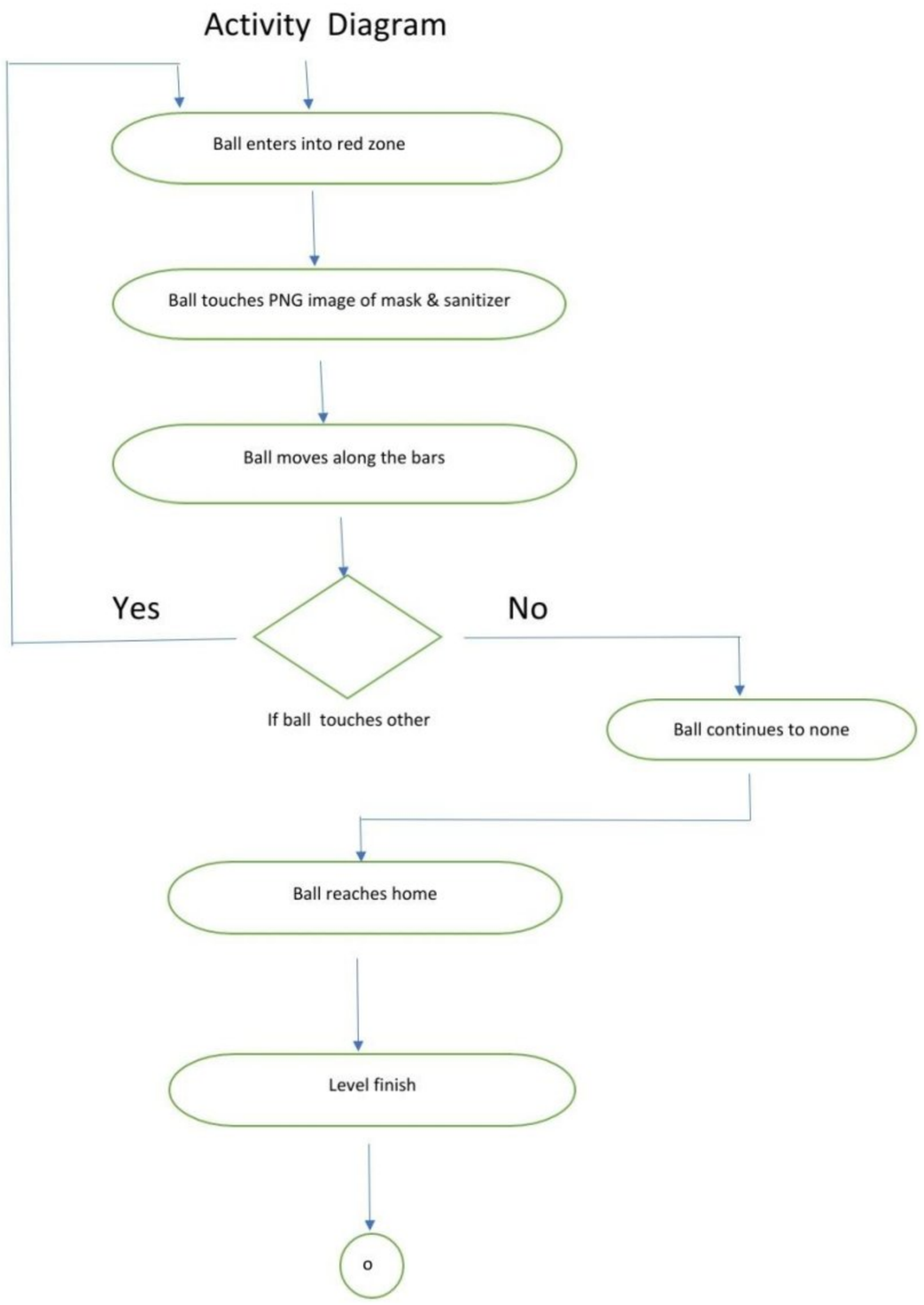


Fig. 3.4. Activity Diagram

3.3 Project Implementation Details

User Interface Components

The components which are interacted with system are called user interface components.

· Canvas

· Screen

3.3.1 CANVAS

Canvas components can detect user taps and can change their boolean state in response. A canvas[1] component raises an event when the user taps it. There are many properties affecting its appearance that can be set in the Designer or Blocks Editor.

3.3.1.1 Properties

Background Color

Color for canvas background.

Height

Canvas height (y-size).

Width

Canvas width (x-size).

Events

Click()

User touch and released canvas.

Got Focus()

Canvas became the focused component.

Lost Focus()

Canvas stopped being the focused component.

3.3.2 SCREEN

Top-level component containing all other components in the program

3.3.2.1 Properties

Information about the screen. It appears when "About this Application" is selected from the system menu. Use it to tell users about your app. In multiple screen apps, each screen has its own About Screen info.

Align Horizontal

A number that encodes how contents of the screen are aligned horizontally. The choices are: 1 = left aligned, 2 = horizontally centered, 3 = right aligned.

Align Vertical

A number that encodes how the contents of the arrangement are aligned vertically. The choices are: 1 = aligned at the top, 2 = vertical ly centered, 3 = aligned at the bottom. Vertical alignment has no effect if the screen is scrollable.

Background Color

This is the display name of the installed application in the phone. If the AppName is blank, it will be set to the name of the project when the project is built.

Scrollable

When checked, there will be a vertical scrollbar on the screen, and the height of the application can exceed the physical height of the device. When unchecked, the application height is constrained to the height of the device.

3.3.3 Title

The caption for the form, which appears in the title bar

Version Code (designer only)

An integer value which must be incremented each time a new Android Application

Package File (APK) is created for the Google Play Store.

Version Name (designer only)

A string which can be changed to allow Google Play Store users to distinguish between different versions of the App.

3.3.4 Designer

Figure 3.5 shows App building begins in the Designer. Here you create the user interface, or the "look and feel" of the app. You also add the components needed to receive input from the user, as well as the components needed to display output or information to the user. The Designer also is where you specify which nonvisible components the app will use, such as the dialer, GPS, or SMS. Notice that because we are in Designer, the Designer button in is slightly grayed out in the top-right corner of the screen. This button, along with the one next to it, labeled Blocks, indicates which editor you are using.

The left side of the screen features the Palette \ which, as the name implies, is the space for all the creation tools

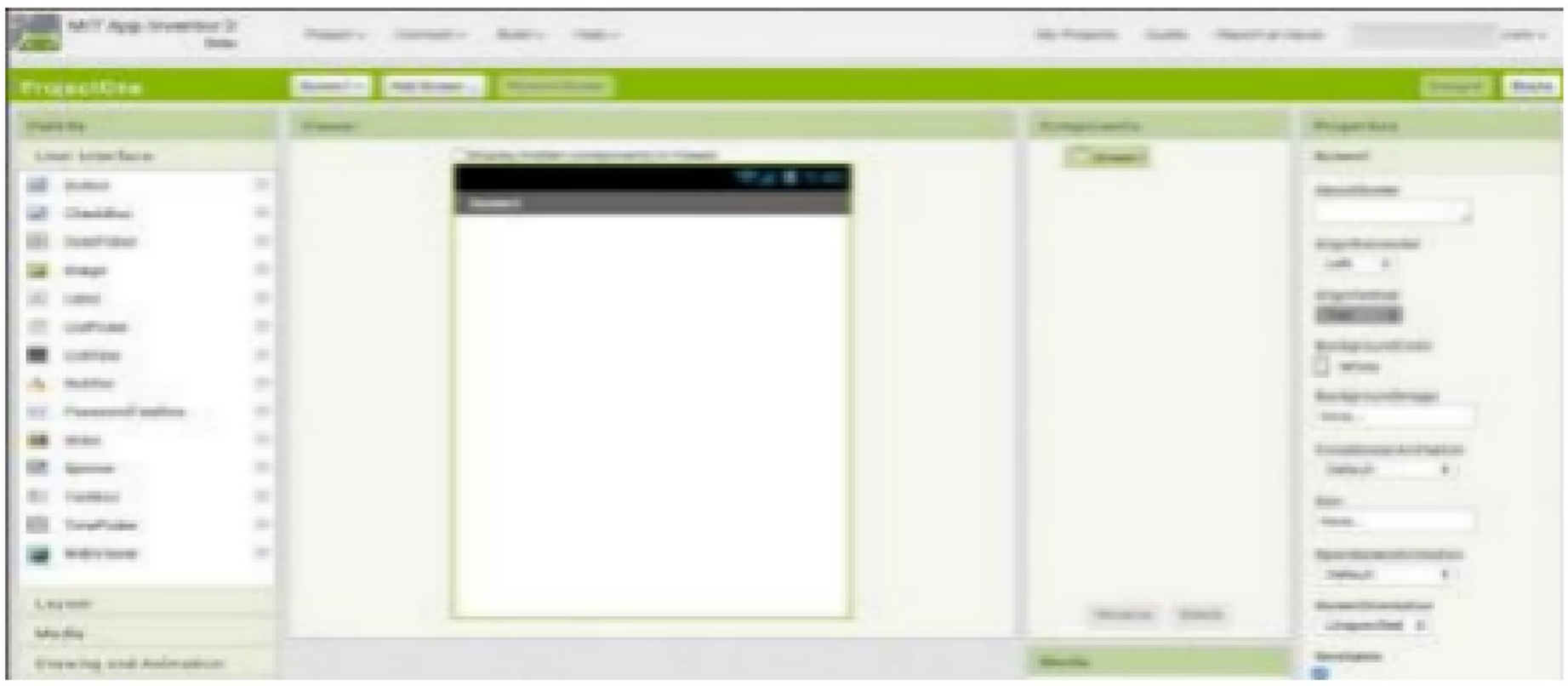


Fig. 3.5: The App Inventor Designer screen.

3.3.5 Blocks Editor

Figure 3.6 shows Blocks Editor is where you will be programming an app's

behavior here you will add the commands that do the work of the app. As just noted, you access it from the Blocks button at the top right.

MIT App Inventor uses the metaphor of drawers containing puzzle pieces for programming. Each item in the Blocks palette under Built-in is considered a drawer. The drawers contain the puzzle-looking pieces. The programming is accomplished by connecting the puzzle-looking pieces. Despite its seeming simplicity, App Inventor has many powerful capabilities that enable the user to build complex applications.

To better understand what programming an app entails, it is useful to understand what is going on inside an application.

Fig 3.6: More specific programming takes place in the Blocks Editor.

3.3.6 The AI2 Companion App

Figure 3.7 shows App Inventor has a useful tool for continuously seeing your app in real time on an Android device during each step of the development process. You can find the MIT AI2 Companion app in the Google Play Store by performing a search for "MIT AI2 Companion[2]."

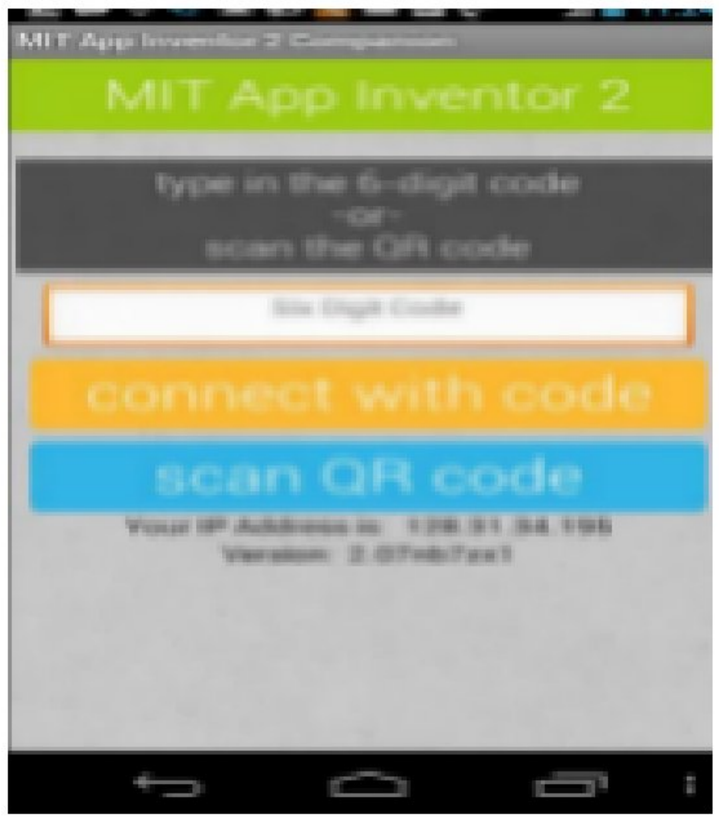


Fig 3.7: The AI2 Companion Android app.

When you are building your app, the computer and Android device must be connected to the same wireless network (the desktop machine might have a wired connection). To connect your app to your device in App Inventor on your computer, click AI Companion on the Connect tab as show in figure 3.8.

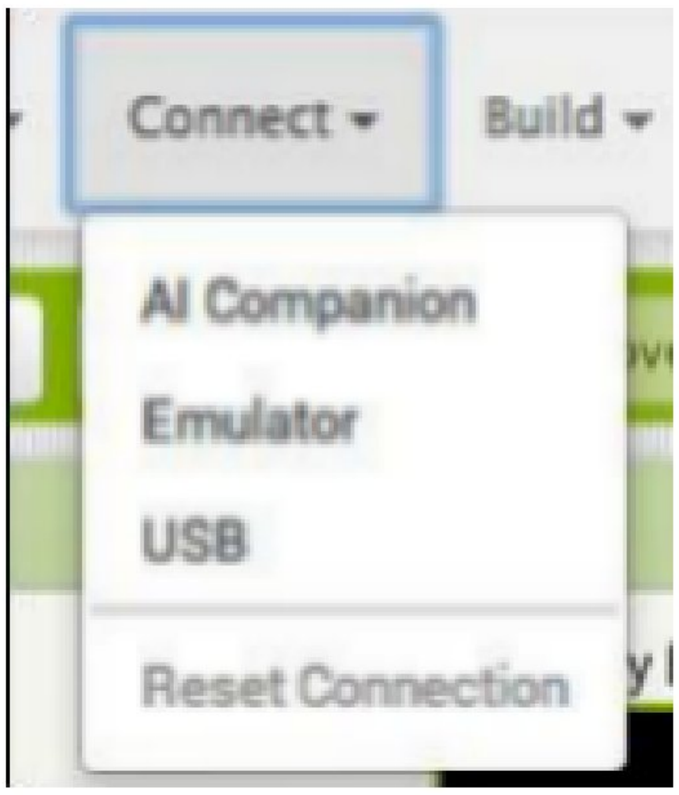


Fig 3.8: Connecting to the Companion app.

You can then type in a six-digit code or scan the QR code with your device, using the App Inventor app doing so brings up a live view of your app. As you add elements to it with the MIT App Inventor software[4], those changes are reflected in real time on your device as show in figure 3.9.

Fig 3.9: Connecting to the Companion app.

3.3.7 The Android Emulator

As another option, App Inventor has an Android emulator[5]that puts a simulated Android screen on the computer desktop. This enables you to view the app's progression if you do not have an Android device. It also is useful for anyone using App Inventor in a classroom environment.

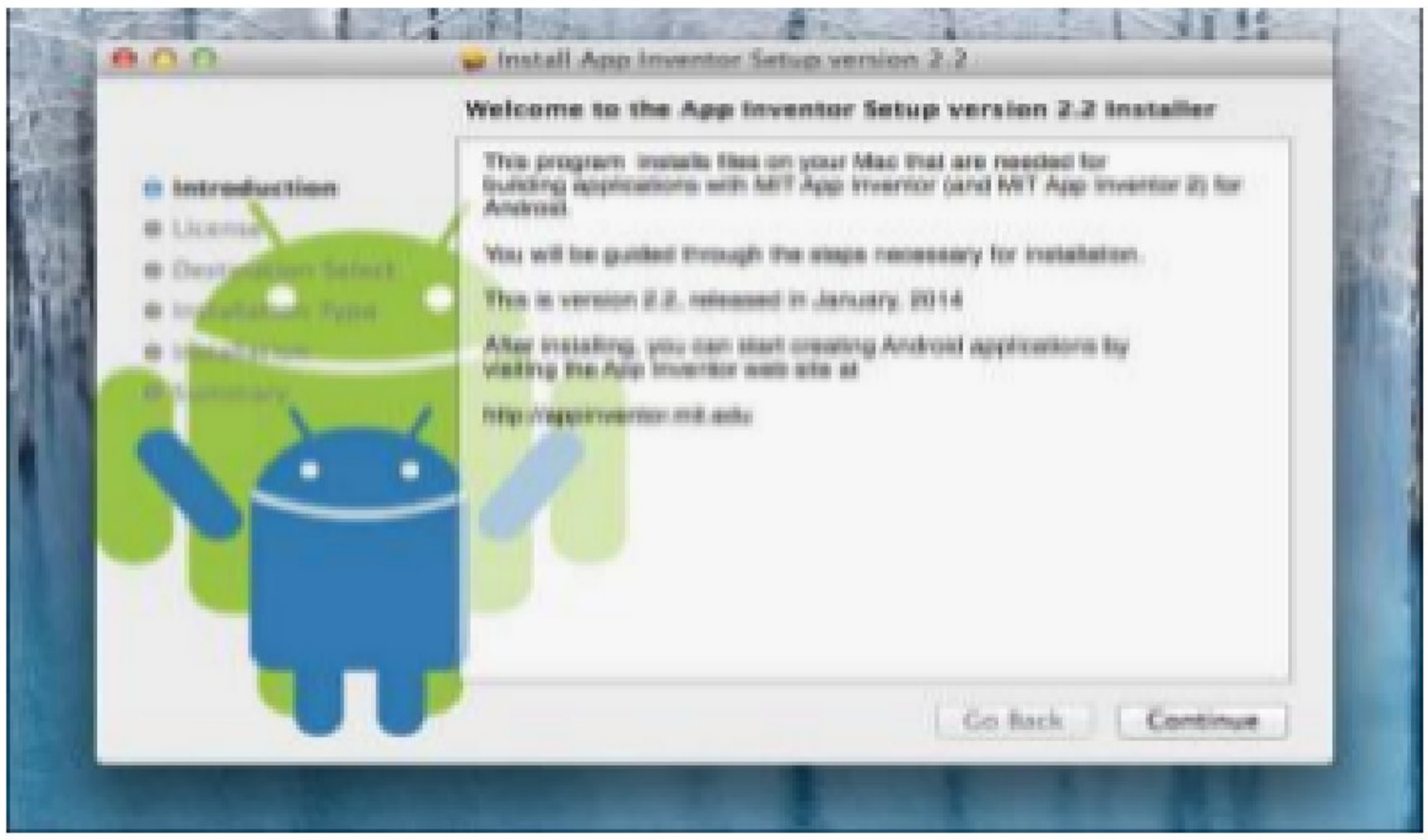


Fig 3.10: App inventor setup installation process.

Figure 3.10 shows an installer package is available for Windows or Mac. Choose the proper platform from the App Inventor site and download it to your computer. The emulator download package. The emulator can be downloaded to a Mac or a Windows

PC.

3.3.7.1 USB Connection to Android Device

Another option is to connect your device to the computer with a USB cable. This method provides the benefit of seeing the app on your Android device just as if you were using the MIT App Inventor Emulator application. This option also does not require a wireless network connection.

Fig 3.11: The Android emulator.

First, you need to install the App Inventor setup software to your Mac or Windows PC. Many Android devices also require the installation of driver software as show in figure 3.11. Your device might require other changes to the device's settings. Bringing your app to market involves what is known as **packaging**. What this is just a process whereby your app is assembled into the Android Package format with an .apk extension

that is at once machine readable and easily and widely distributable.

3.3.8 Versioning

Proper versioning is an important step if you wish to distribute your app commercially. This is done on the Design page via **Screen's VersionCode** and **VersionName** properties. **VersionCode** is an integer value and should be incremented with every new major or minor version of your app. The **VersionName** can be anything you like, however it traditionally includes the name and a decimal number with the whole number representing the major version, and the fractional part representing the number of any minor revision.

3.3.9 Sharing

In order to share your app with other Android users we first create the .apk file by clicking the **Package for Phone** button in the **Design** window.

4. TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner [5]. There are various types of test. Each test type addresses a specific testing requirement.

4.1 Testing Objectives

Testing is a process of executing a program with the intent of finding an error. A good test case design is one that has probability of finding an as yet undiscovered error. If testing is conducted successfully it will cover errors in the software.

4.1.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. A unit testing

is a piece of code written by a developer that executes a specific functionality in the code which is tested [3]. The percentage of code which is tested by unit tests is typically called test coverage. A unit test targets a small unit of code e.g., a method or a class (local tests).

4.1.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent.

4.1.3 System Testing

System testing ensures that the entire integrated software system meets requirements and is based on descriptions and flows. Emphasizing per-driven process like and integration points.

4.1.4 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

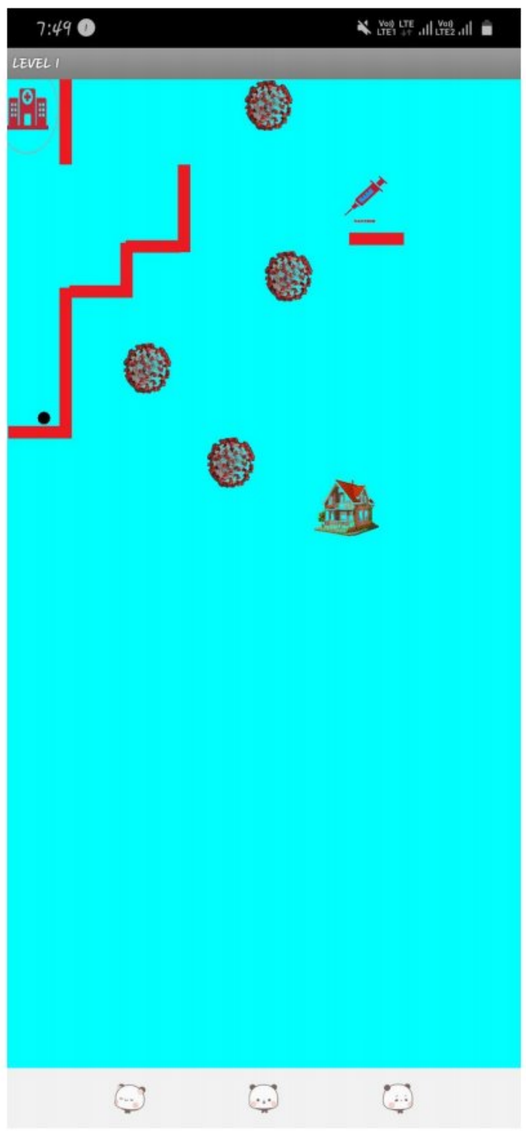


Fig4.1 Ball when collects mask,sanitizer

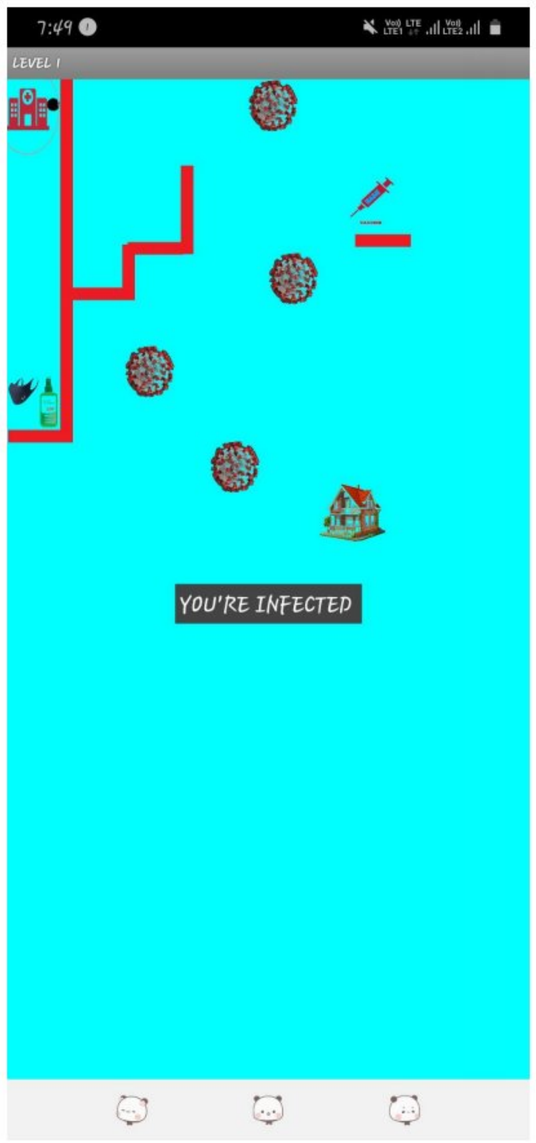


Fig4.2 Ball touches virus

Fig4.3 ball get infected

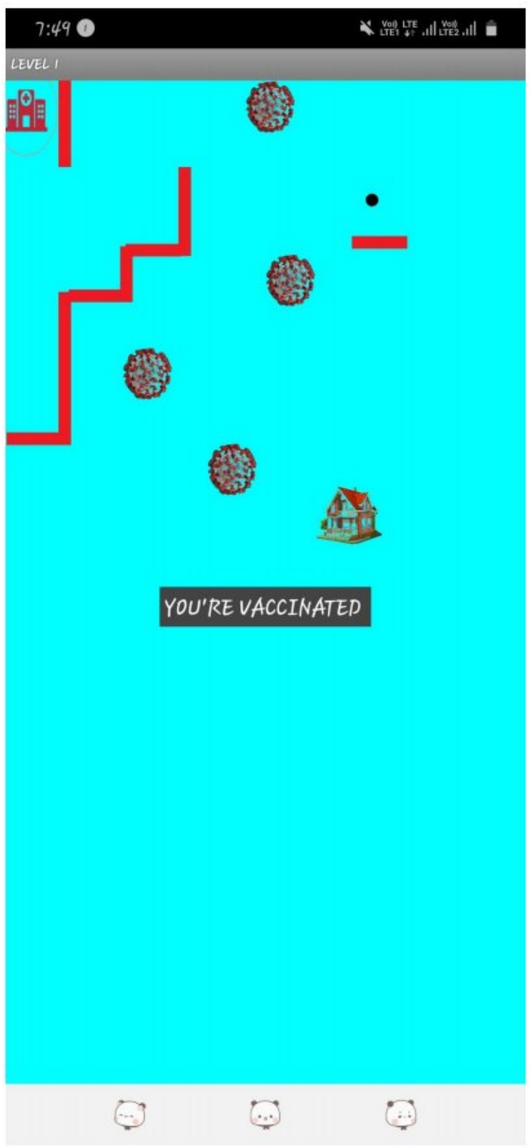


fig4.4ball touches vaccine

fig4.1:shows when ball touches mask and sanitizer the gate will open in level1.

Fig4.2:when ball touches virus it will print a message that you're infected in level1.

Fig4.3:after the message you're infected it will print a message that you're moving to quarantine in level1.

Fig4.4:when ball touches vaccine it will give the message you're vaccinated in level1.

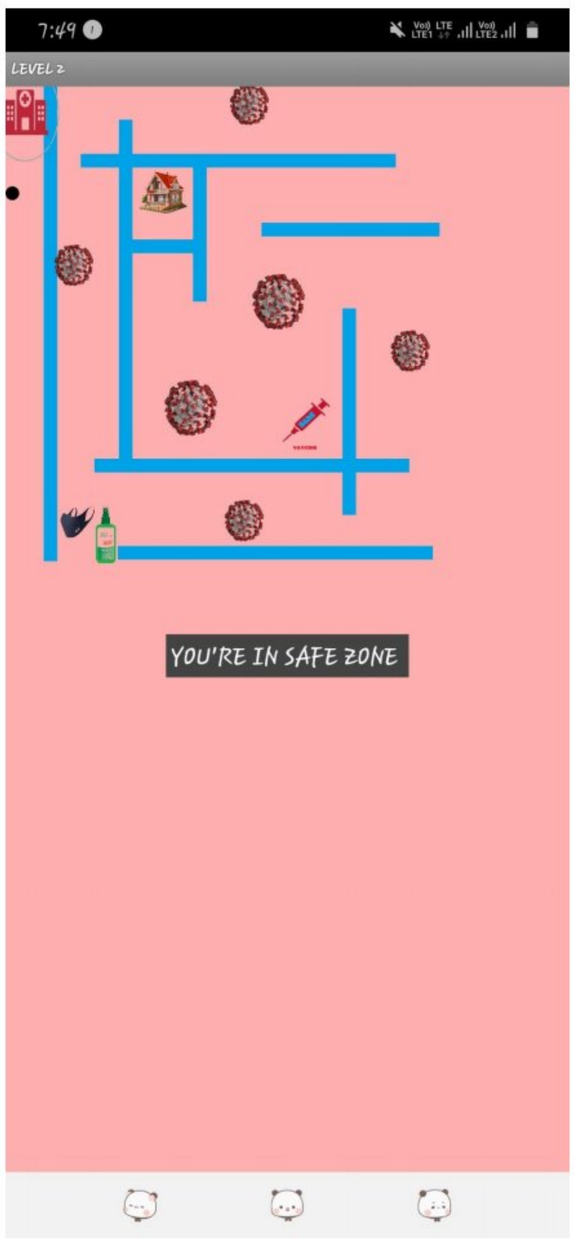


Fig4.5 completion of level 1

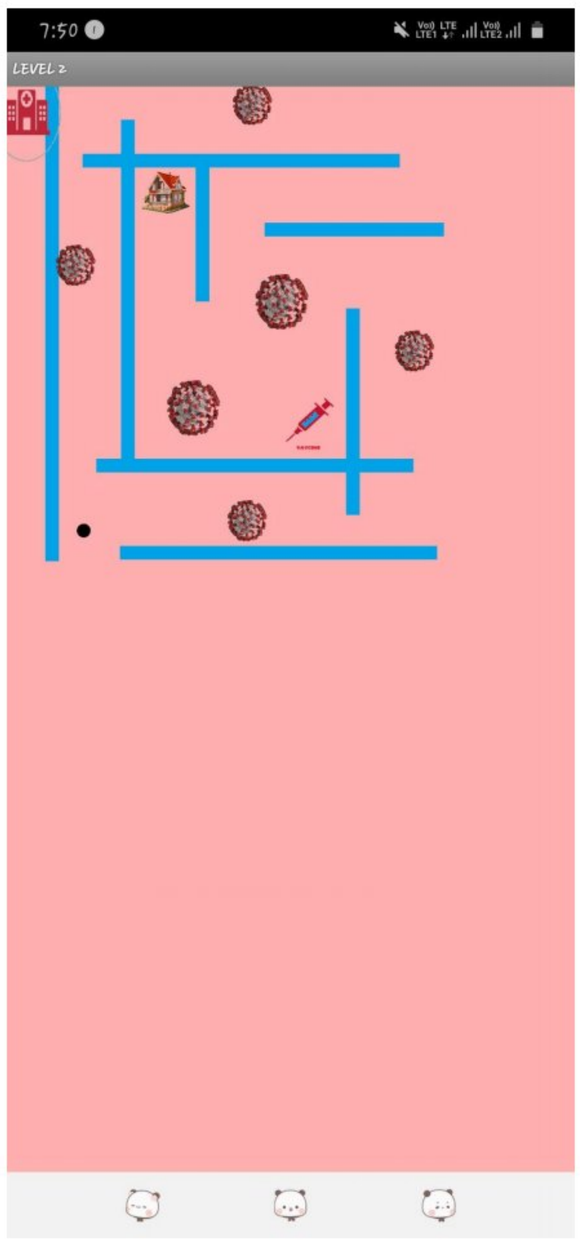


Fig4.6ball collects mask and sanitizer

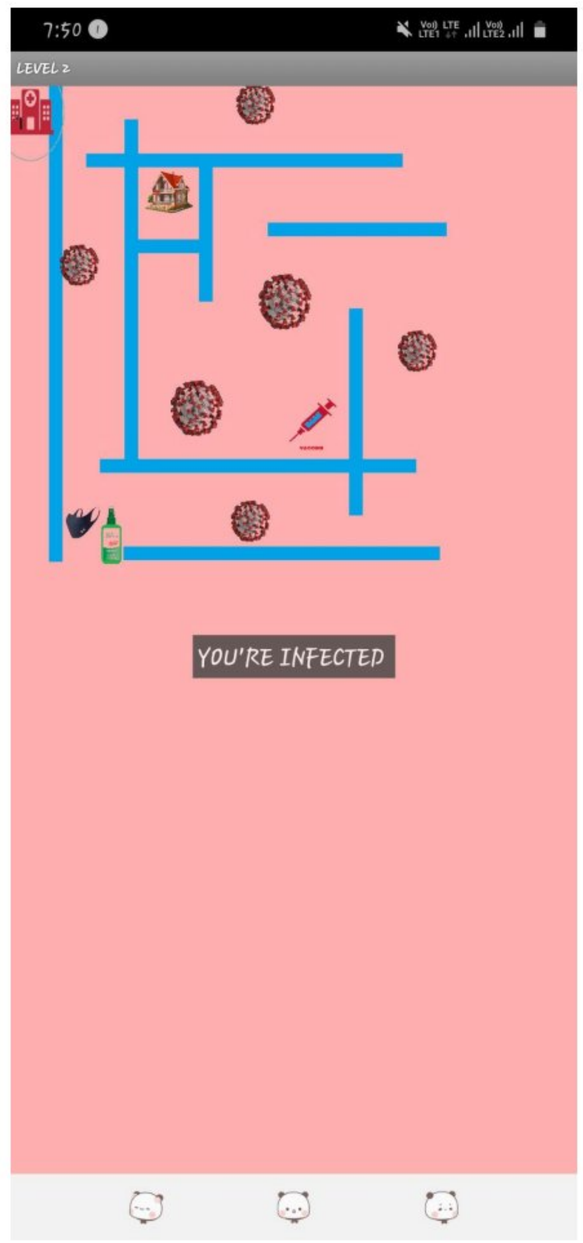


fig4.7ball touches virus

Fig. 4.3: ball gets infected

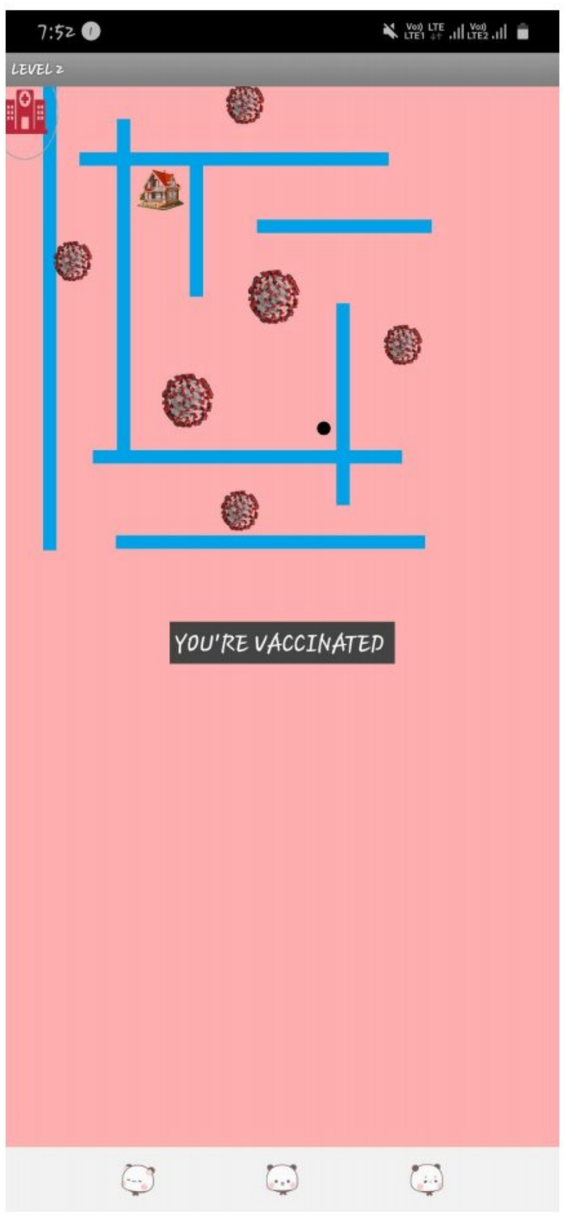


Fig. 4.9: Ball touches vaccine

Fig4.5:after completion of level 1 I will

give the message you're in safe zone.

Fig4.6:when ball touches mask and

sanitizer the gate will open

Fig4.7:when ball touches virus it will print

a message you're infected in level 2.

Fig4.8:after ball gets infected I will give the

message you're moving to quarantine in

level 2.

Fig4.9:when ball collect vaccine I will give

the message you're vaccinated in level 2.

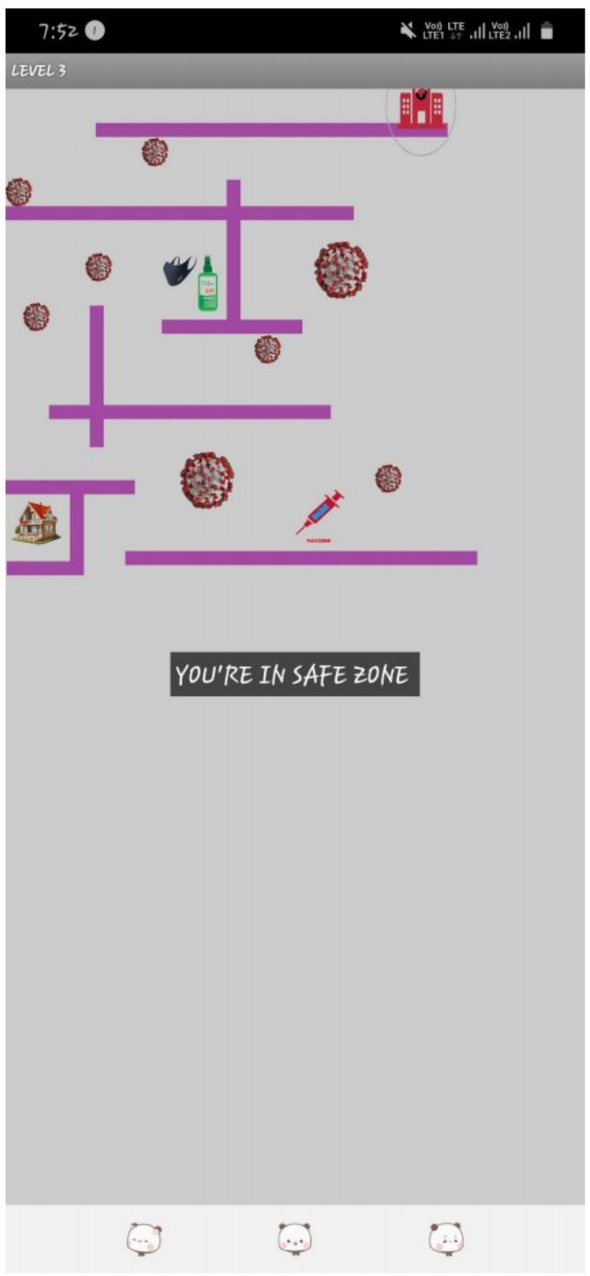


Fig4.10 completion of level2

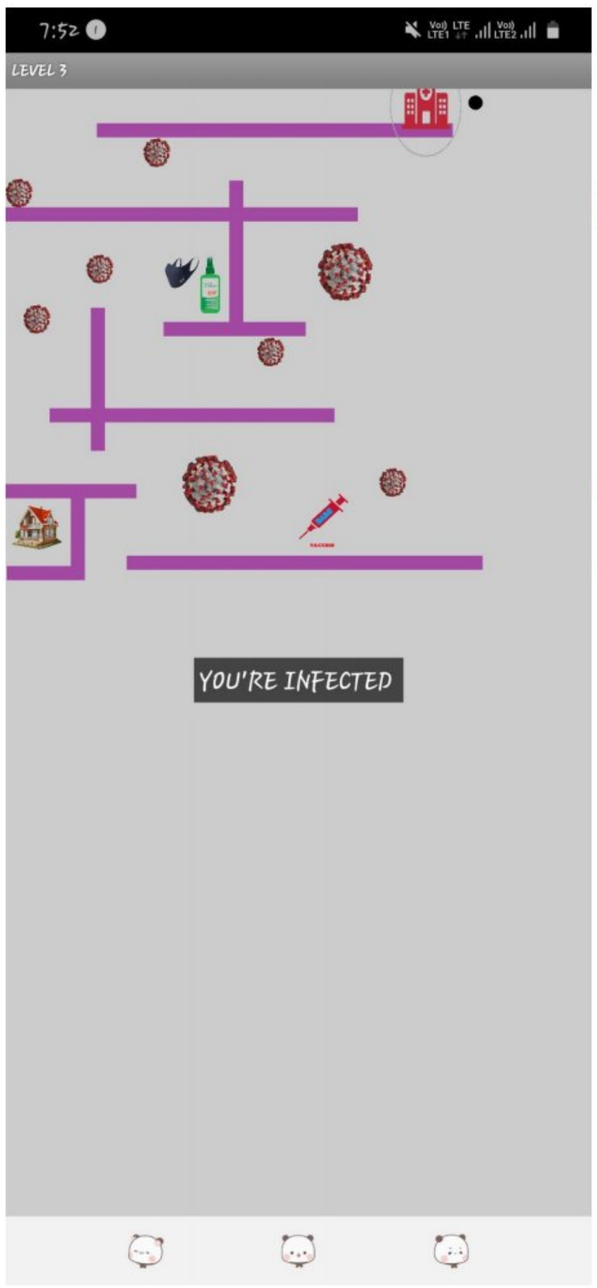


fig4.11ball touches virus

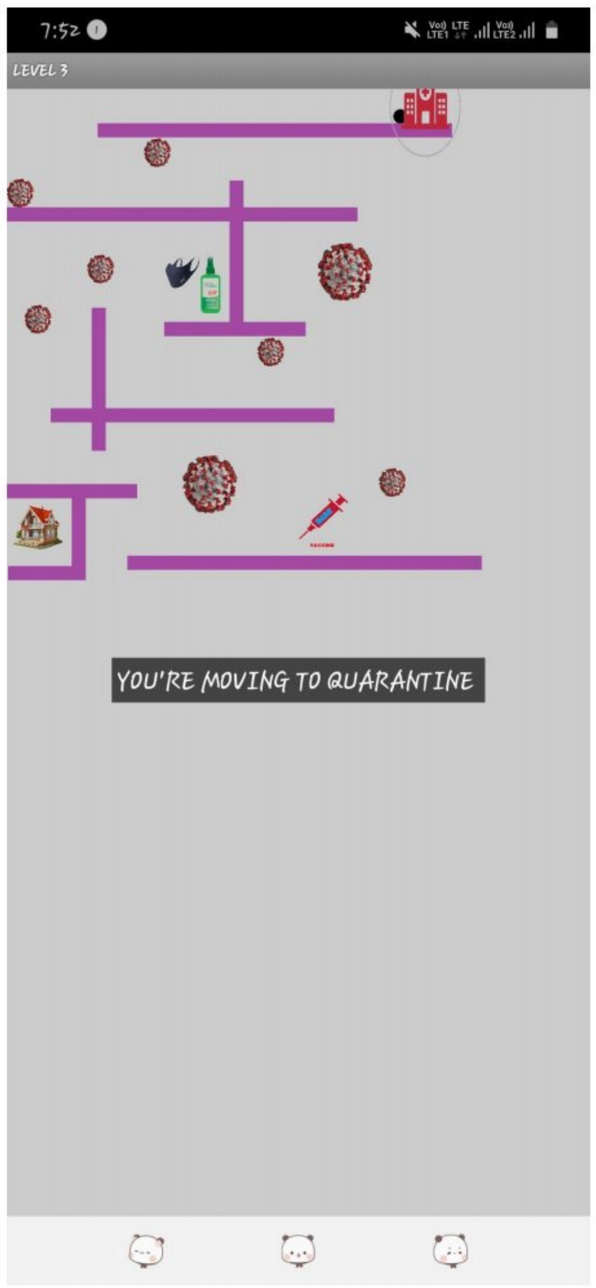
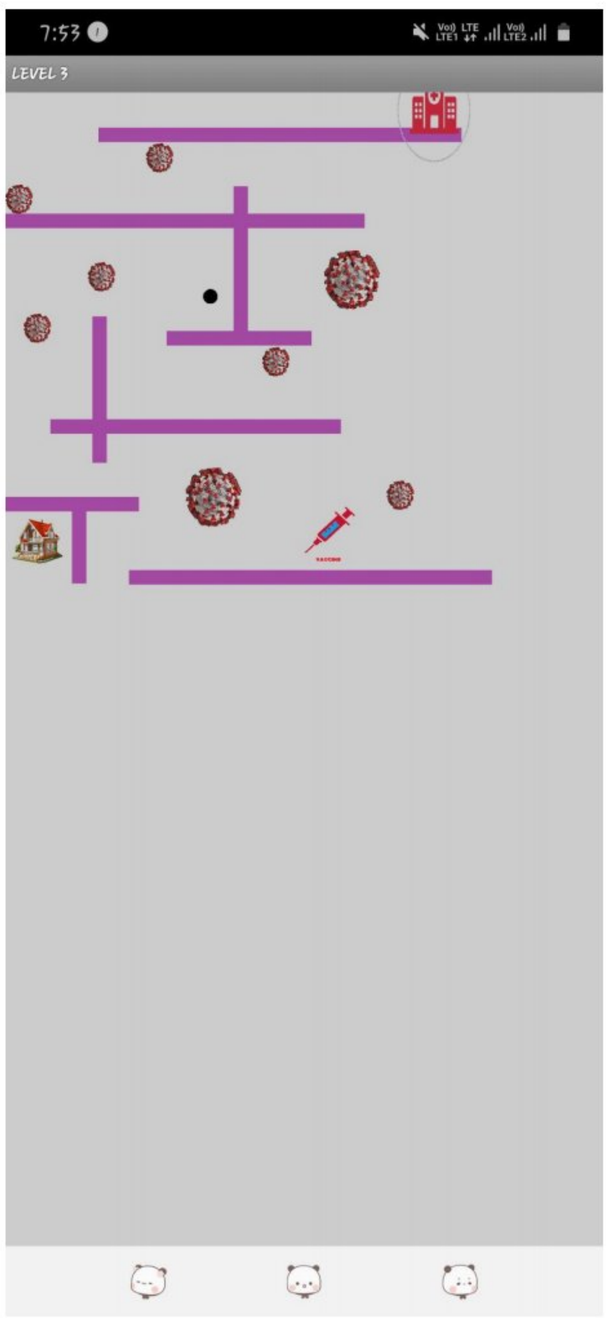


fig4.12ball get infected



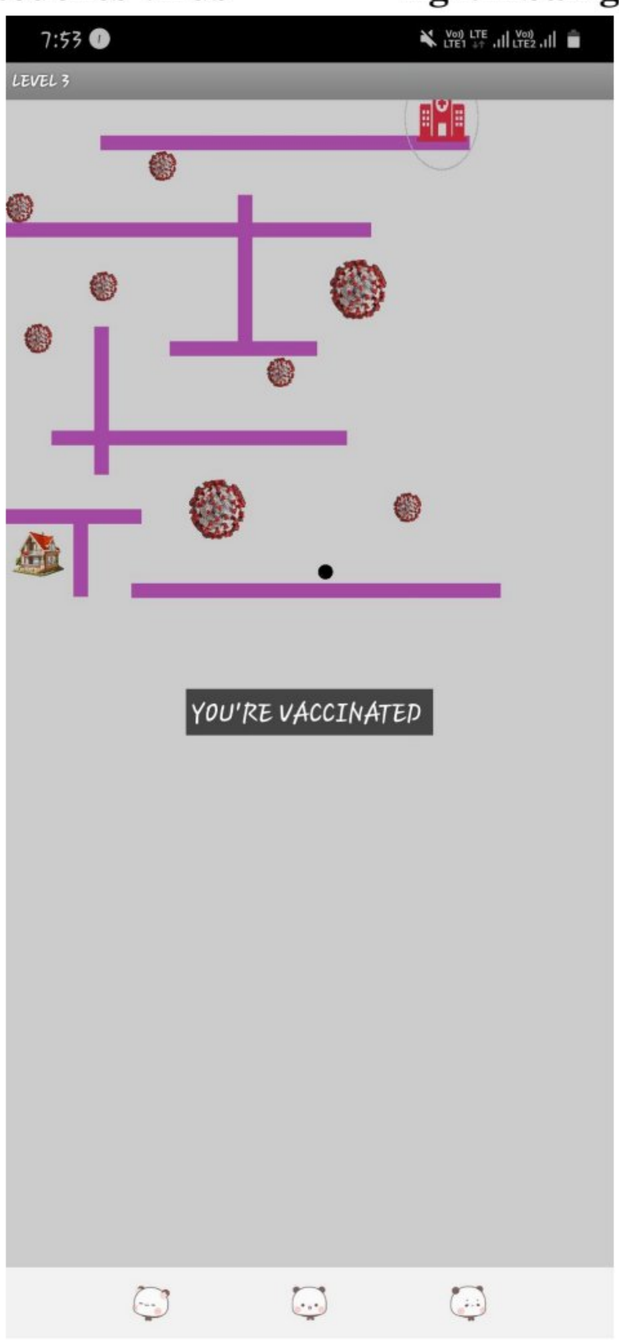


Fig4.13ball collects mask and sanitizer **fig4.14ball touches vaccine**

Fig4.10:after completion of level 2 I will

give the message you're in safe zone.

Fig4.11:when ball touches virus I will give

the message you're infected in level 3

Fig4.12:after ball gets infected I will give

the message you're moving to quarantine

in level 3.

Fig4.13:when ball touches mask and

sanitizer in level 3 the gate will open .

Fig4.14:when ball collects vaccine I will

give the message you're vaccinated.

5. RESULTS

Level-1

This is the initial level we are moving from starting point to end point in less infected region.The difficulty of the level is less.

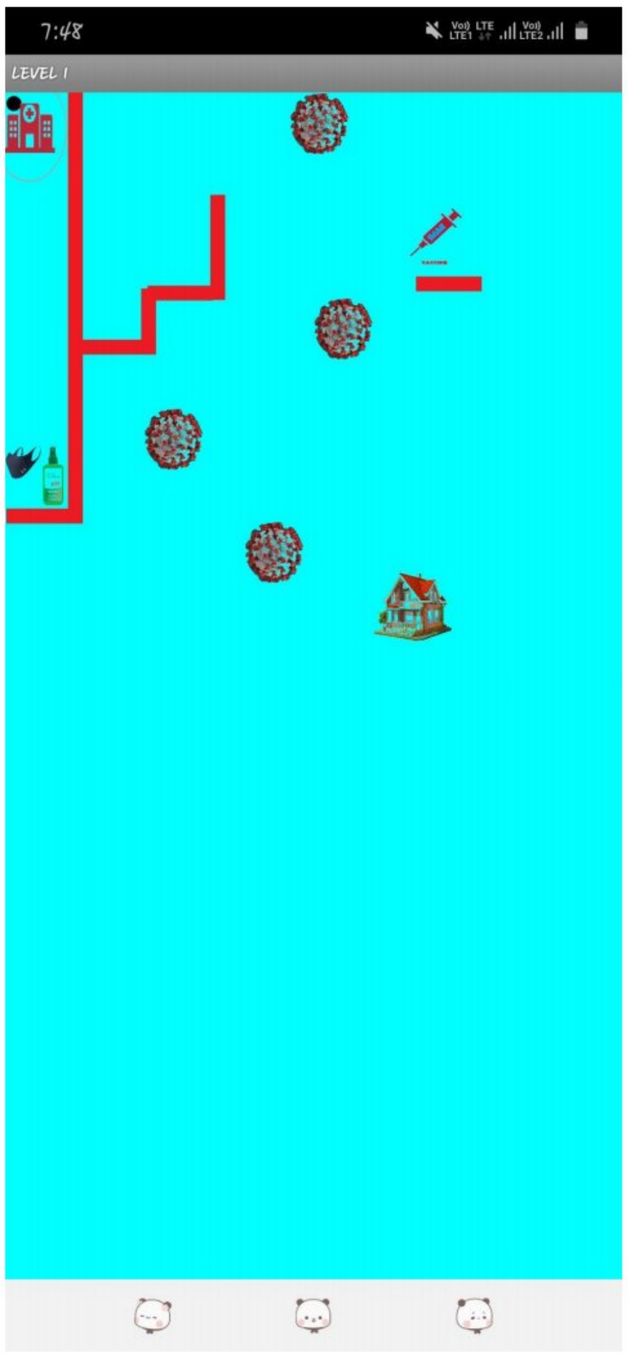


Fig. 5.1: Initial stage of level-1

Level-2

In this level we are moving through the area which is more infected compared to level 1.The difficulty of the level is more compared to level 1

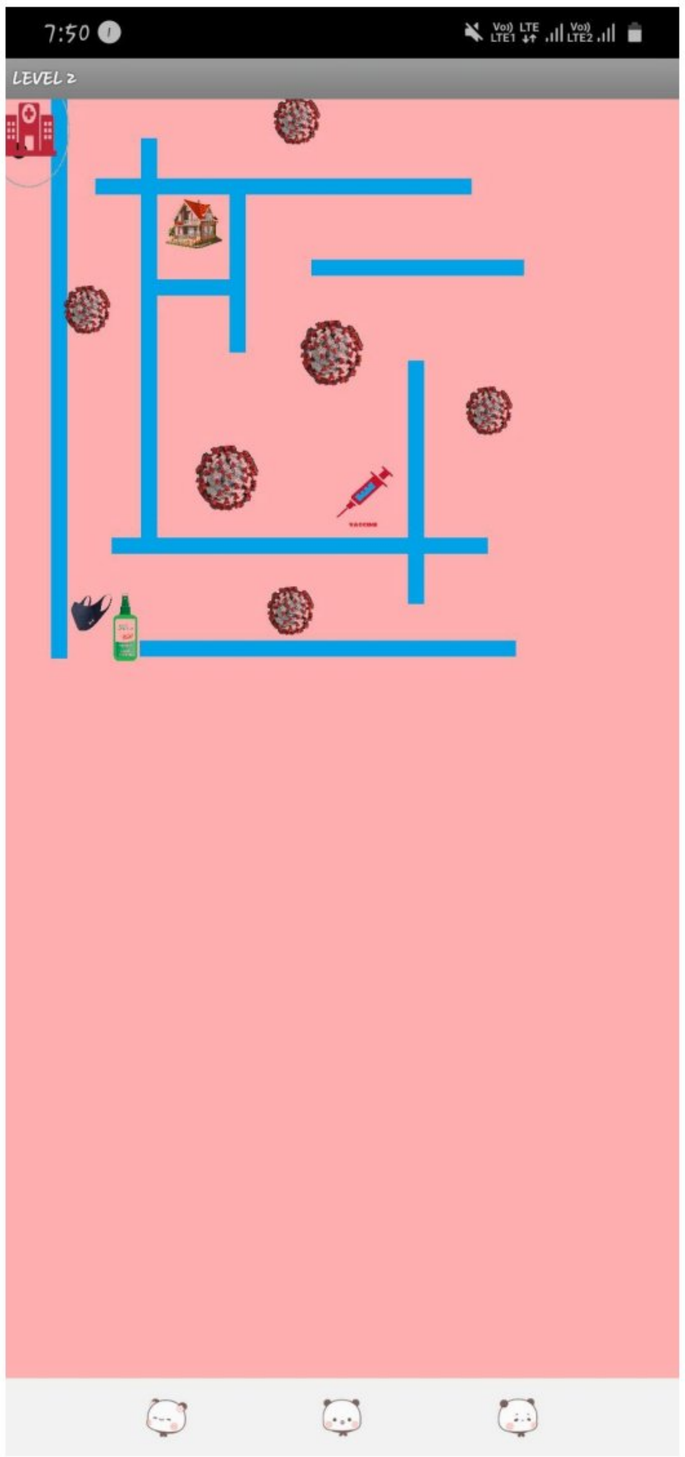


Fig. 5.2: Initial stage of level-2

Level 3

In this level we are moving through more infected areas compared to all levels.The difficulty of the level is more.But collecting vaccine is safe whenever we get infected by the virus.

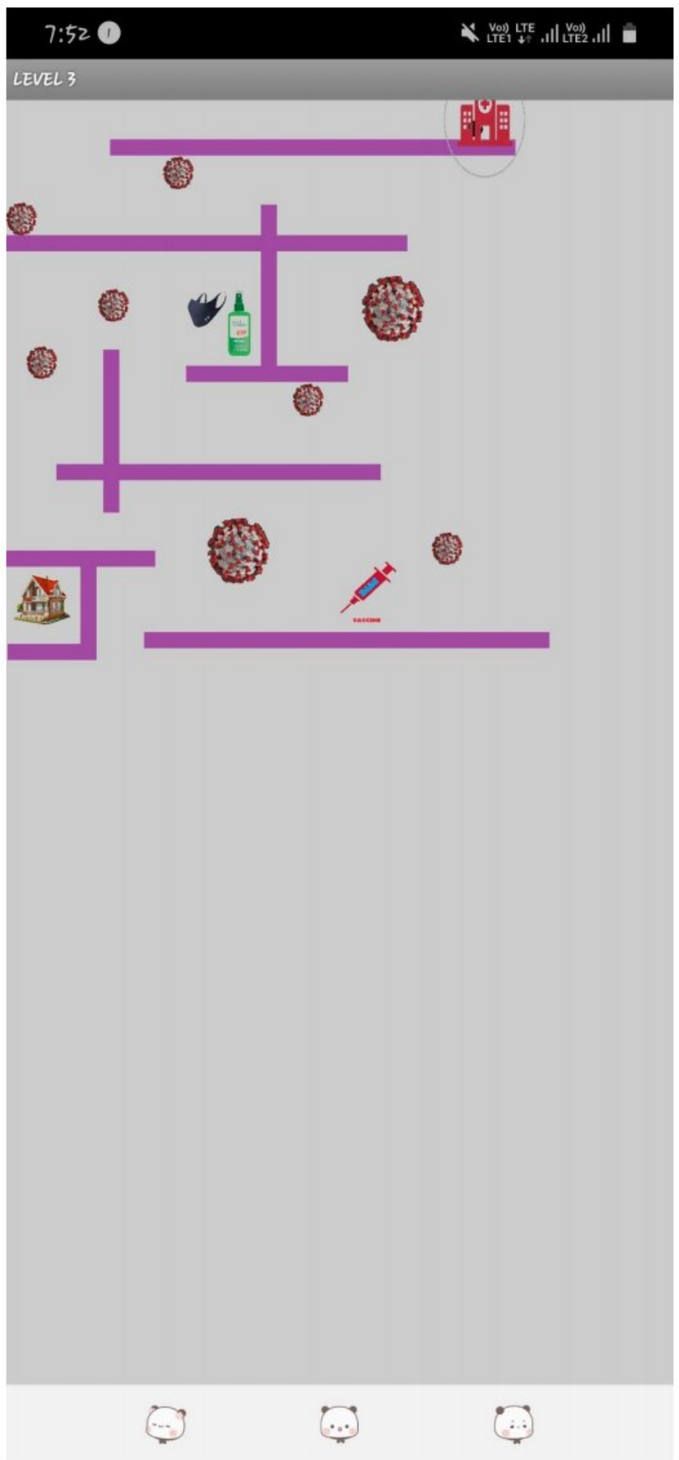


Fig. 5.3: Initial stage of level-3

CONCLUSION

We developed a survival fittest app that is much more exciting then existing games.The main reason for developing this game is to give a message then we should go out with mask and sanitizer .Getting infected by virus is a dangerous thing for life .When we get vaccinated we are safe then we cannot be affected by the virus.level by level the difficulty of the game will gets increased which increases the users concentration.