جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

Software Development for Android Devices

ENCS 539

Project #3

Adventure Game

Student Name: Mohammad Kabajah

ID: 1110600

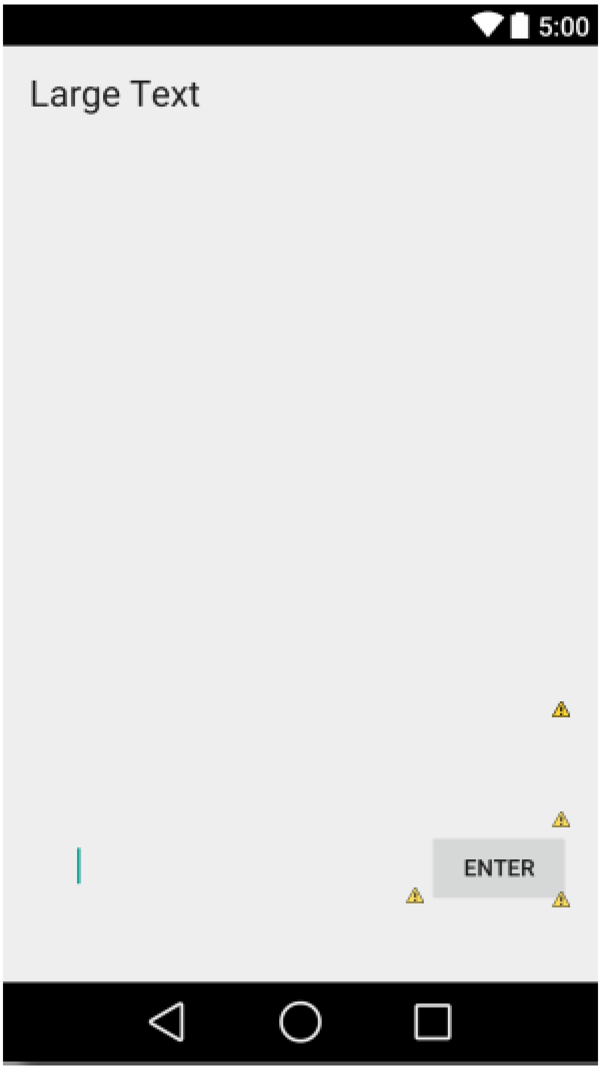
Instructor's Name: Stephen Taylor

Section #:1

Date: 21/10/2014

Java and XML codes

The XML file is included in the folder and here is a picture of the layout



And all Java files is included in the folder and here is all the codes that I modified or write

In the adventure class I create a method called read Line that takes input Stream and return String and here is the code

```
String readLine(InputStream fr){
            String s = "";
            try {
                  char c = (char) fr.read();
                  while(c!='\n'){
                  s+=c;
                  c =(char) fr.read();
                  }
            } catch (IOException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }

            return s;
      }
```

I used input Stream because I was talking with “Ibrahem Abu Farha” about the project and he told me that we need to use input stream to read the file.

I also modified the setup method to take input stream parameter instead of the file name, and created a method called used to make sure that the setup is completed and start printing the description of the game, and here is the code of the ready method.

```
	String ready(){
		if (!allSetUp) throw new Error("oops! not set up!");
		String s = speak(1);
		current = Room.getRoomById(1);
		s += "\n" + current.longDescription;
		return s;
	}
```

Also modified the method play to take Command and return String and inside the method I started to verify that the two word commands is correct but didn’t complete it its now checking if there is tow verbs together and told the user that. And here is the code.

```
	String play(Command  cmd){
		String s = "";
		Command c = cmd;
		if((c.first.id>2000 && c.first.id<3000) && (c.second!=null ) && 
(c.second.id >2000 && c.second.id<3000))
			s = speak(61);
		else if((c.first.id>2000 && c.first.id<3000)&& (c.second!=null 
) && (c.second.id >1000 && c.second.id<2000)){
			s = vact(c);
			}
		if(c.first!=null)
			s += "\n" +act(c);
		
		return s;
	}
```

In the act method I just modified it to return a string and create another method called vact that is used when the command is verb with item like Take Lamp and here is the code of vact.

```
	 private String vact(Command c) {
		String s = "";
		if(haveItem(Item.getItemName(c.second.id))==false){
		    Item.takeItem(c.second.id);
		    additem(c.second.id);
		    s = speak(54);
		}
			else  s = speak(24);
			
			
		return s ;
	       }
```

Also create two more methods to add the items that we carry and to check what item we are carrying and here is the code of the two methods.

```
boolean haveItem(String s){
            if(carryingIndex==-1){
                for (int i = 0 ; i < carrying.length;i++)
                   carrying[i]=new String();
               carryingIndex++;
            }
            for (int i = 0 ; i < carrying.length;i++){
                if(carrying[i].equals(s))
                    return true;
            }
            return false;
        }
        void additem(int id){
           String name = Item.getItemName(id);
           if(carryingIndex==-1)
           {
               for (int i = 0 ; i < carrying.length;i++)
                   carrying[i]=new String();
               carryingIndex++;
           }
           else {
               carrying[carryingIndex++]=name;
           }
        }
```

Also modified the speak method to return String so when it was used to return the string back to play method and play method is called in the main activity so the string is returned to the main activity so we can print it to the text view in the main activity.

I also added new table to the to the dat file this table contain the number of the room in the first number and each number value near this room is the item number that is found in this room to be added to the room object that each room have these items, and the file is also included in the folder.

In the Command class I modified it and just keep the two constructers, and the commands is being created in the main activity and sent to the play method in Adventure class.

The Item class I wrote it to add the items by their ID and name and get them and mark if taken or not and if they have more than one case like the Grate where it could be open or closed.

In the main activity I opened the dat file and run the setup to the game to be ready to play and added On Click Listener to the button which work as the following get the command check if the written command is valid or not the if valid send it to the paly method in Adventure class.

In the Message and Messages classes I modified the speak methods to return String so it could return it to the main activity to be printed on the screen instead of System.out.println(); that was used.

In the classes Portal and Word there is nothing modified from the files you sent us.

In the Room class I modified the two methods the describelong() and mentionItems(), what I made was to make each method return the String instead of print it so it can be printed from the main activity and here is the code of the two methods.

```
public String describeLong() {
            String s = longDescription;
            s += mentionItems();
            return s;
      }
```

```
private String mentionItems() {
            String s= "";
            if(itemIds != null)
            for (int i=0; i<itemIds.length; i++){
                  if (itemIds[i] != 0){
                        Word w = Word.getWordById(itemIds[i]);
                        s +=
Item.getItemMsg(Item.getItemById(itemIds[i]).id);
                        s+="\n";
                  }
            }
            return s;

      }
```

I also modified the itemIds array that contain the items and added the method called addItems to add the item id to the array for each room and use it in the mention items and here is the code of the method.

```
        public void addItems(int id){

            if(itemIds==null){
                itemIds= new int[1];
                itemIds[0]=0;
            }
            for(int i = 0;i<=itemIds.length;i++){
                if(i>itemIds.length-1){
                    int [] t = new int[2*itemIds.length];
                    System.arraycopy(itemIds,0,t,0,itemIds.length);
                    itemIds = t;
                }

                if(itemIds[i]==0){
                    itemIds[i]=id;
                    return;
                }
                else  if(itemIds[i]!=0)
                    continue;
            }
        }
```

My Game Description

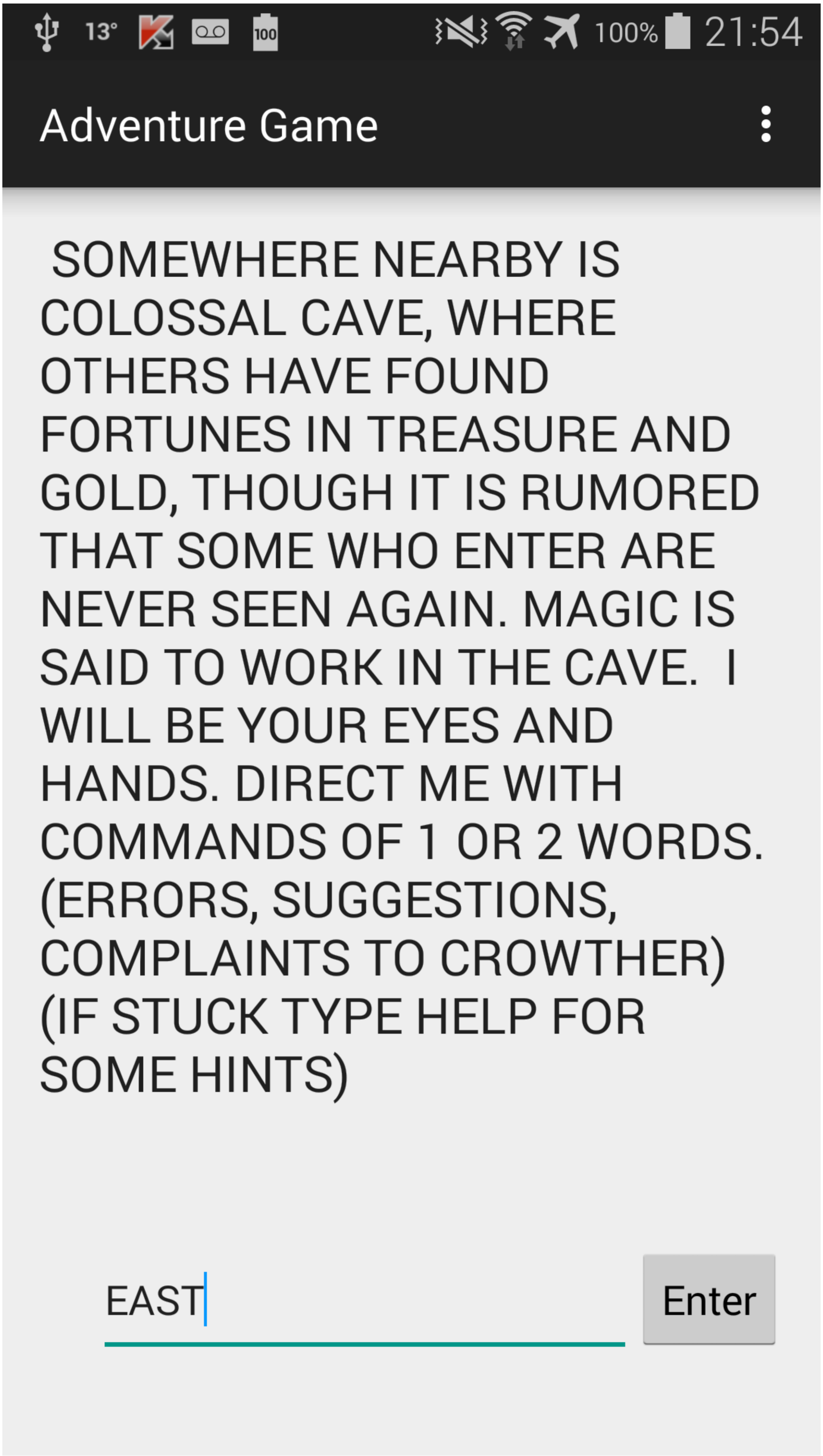
The game is almost the same as the original game that was released in the 1970's, you start near the building you enter the building and take the items you find there then you go and explore the other places in the game, you go to the forest and so on.

My Game Implementation

The implementation of the game is some kind like the original game you enter the command in the text field and then press/touch/click the button and then the command will be executed and the output will be printed on the screen in the textview.

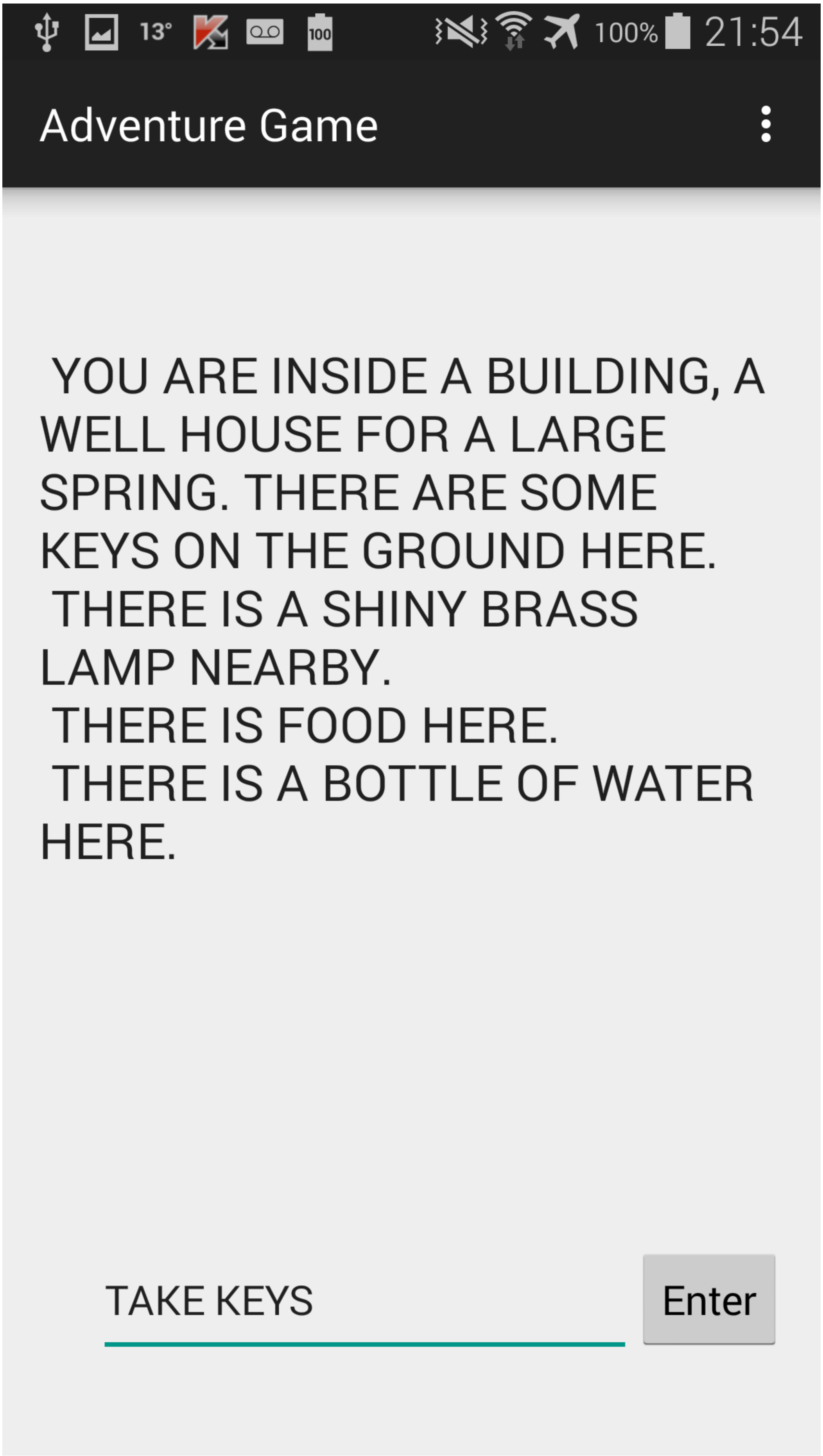
Screen Shots

Some screen shots of the game running on my phone



SOMEWHERE NEARBY IS COLOSSAL CAVE, WHERE OTHERS HAVE FOUND FORTUNES IN TREASURE AND GOLD, THOUGH IT IS RUMORED THAT SOME WHO ENTER ARE NEVER SEEN AGAIN. MAGIC IS SAID TO WORK IN THE CAVE. I WILL BE YOUR EYES AND HANDS. DIRECT ME WITH COMMANDS OF 1 OR 2 WORDS. (ERRORS, SUGGESTIONS, COMPLAINTS TO CROWTHER) (IF STUCK TYPE HELP FOR SOME HINTS)

EAST



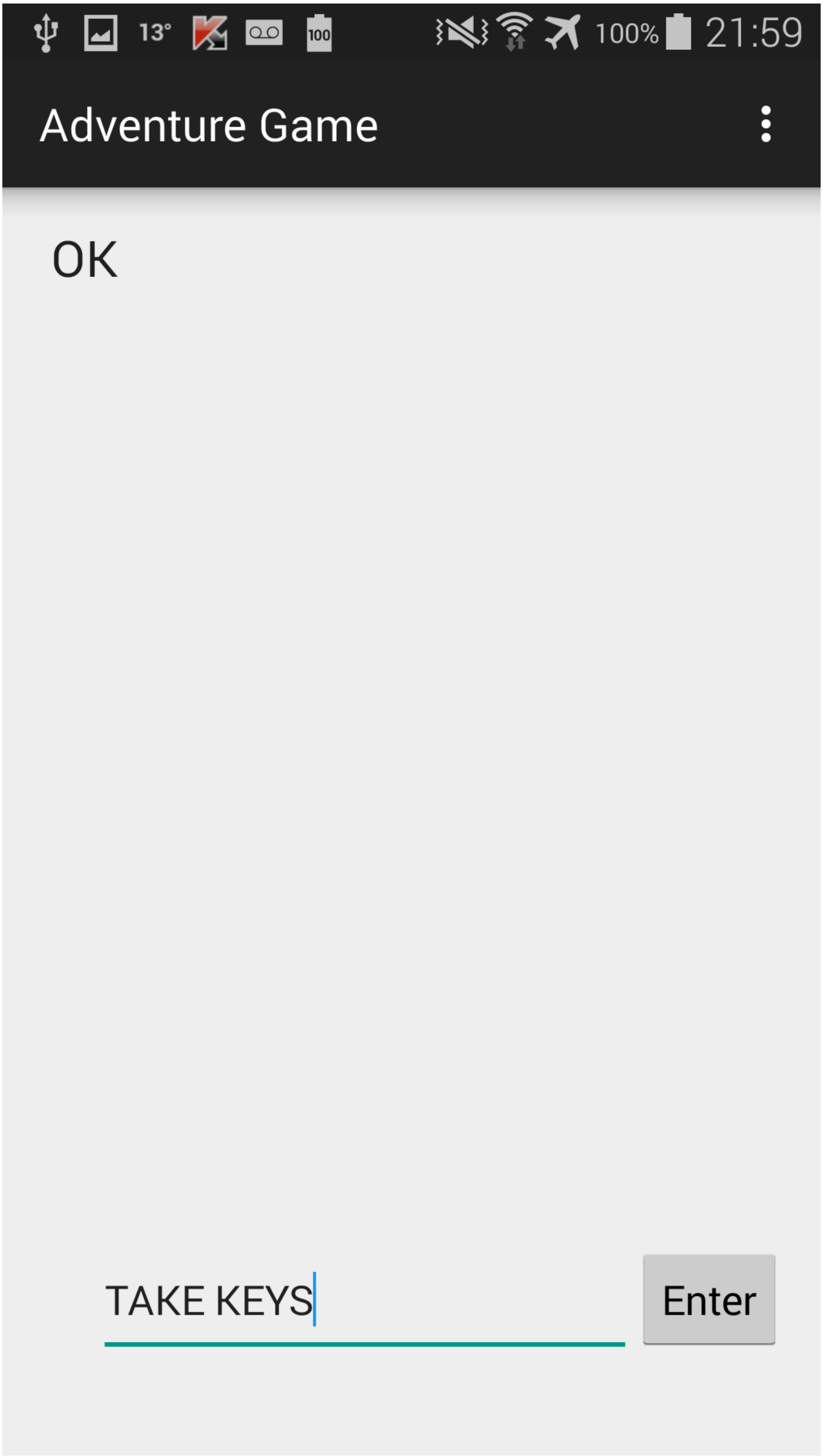
YOU ARE INSIDE A BUILDING, A WELL HOUSE FOR A LARGE SPRING. THERE ARE SOME KEYS ON THE GROUND HERE.

THERE IS A SHINY BRASS LAMP NEARBY.

THERE IS FOOD HERE.

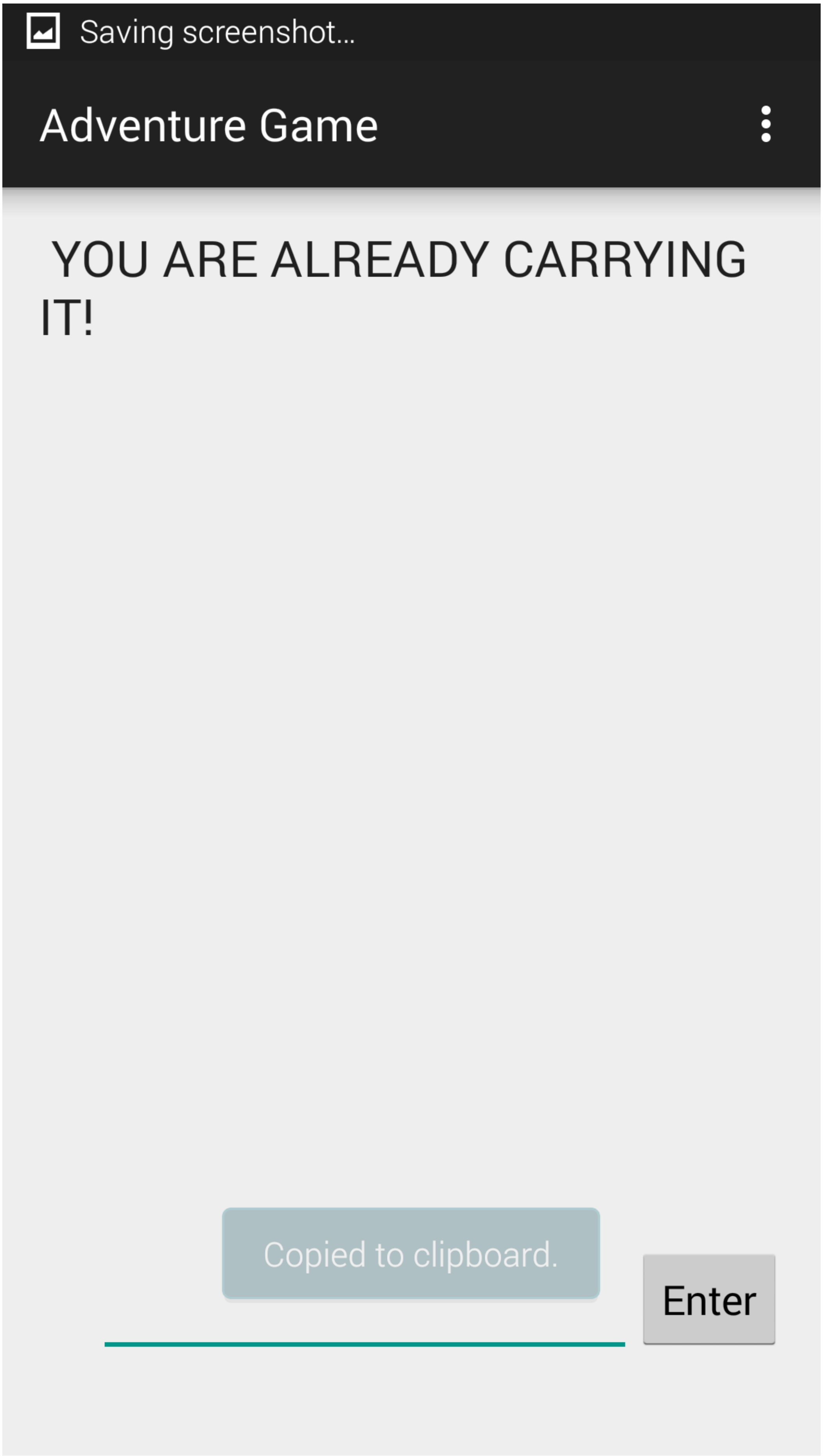
THERE IS A BOTTLE OF WATER HERE.

TAKE KEYS

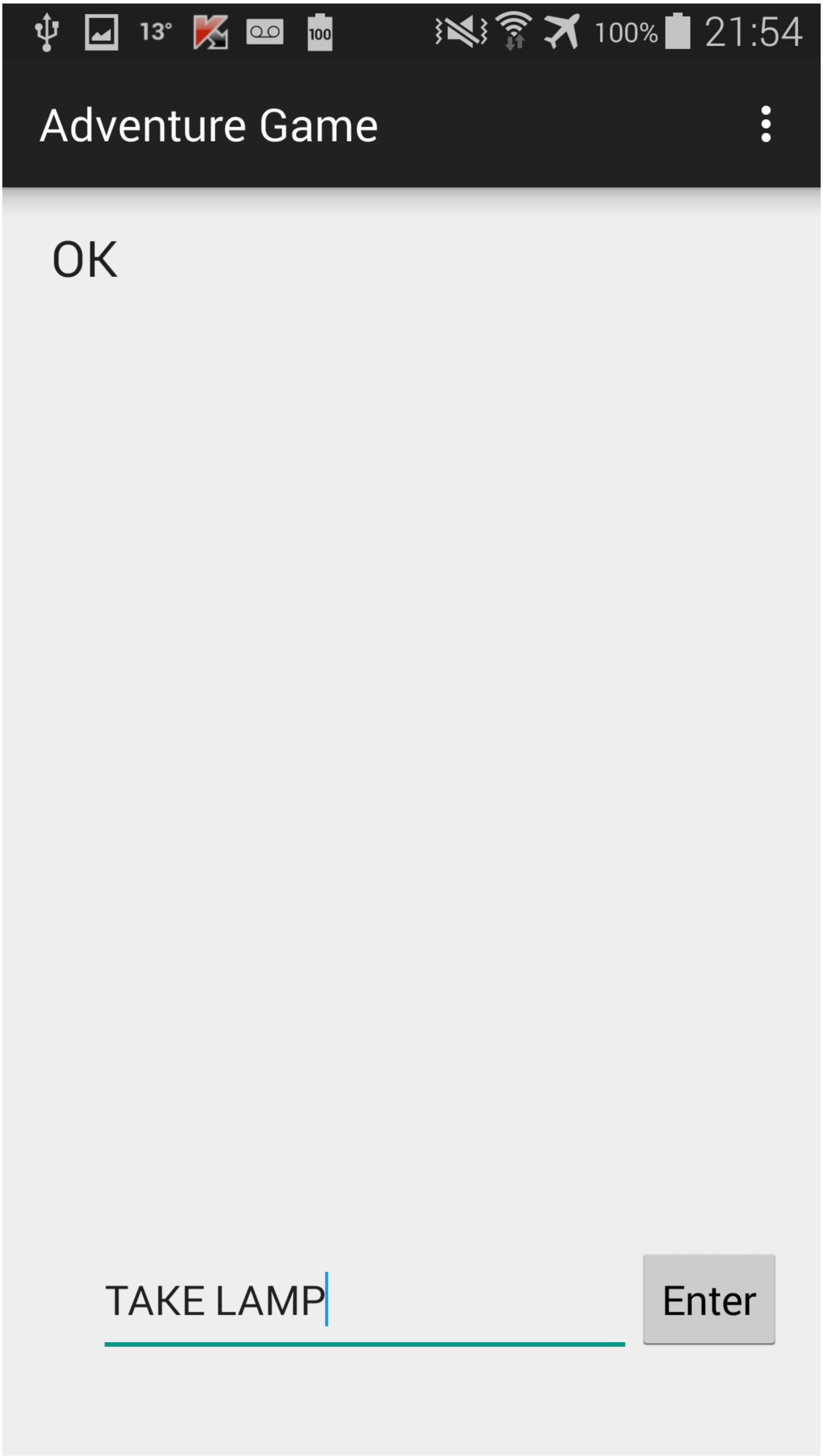


OK

TAKE KEYS

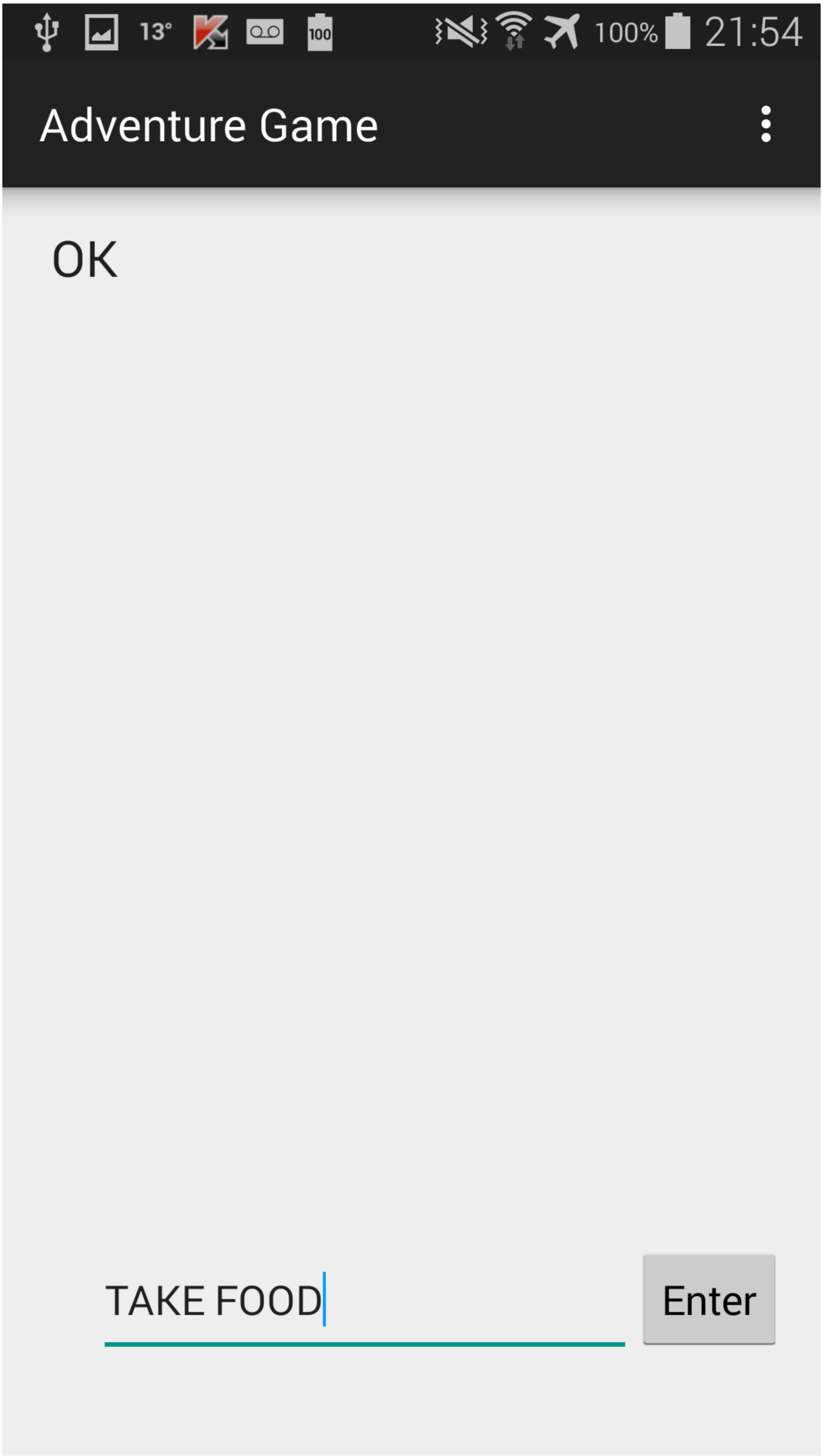


YOU ARE ALREADY CARRYING IT!



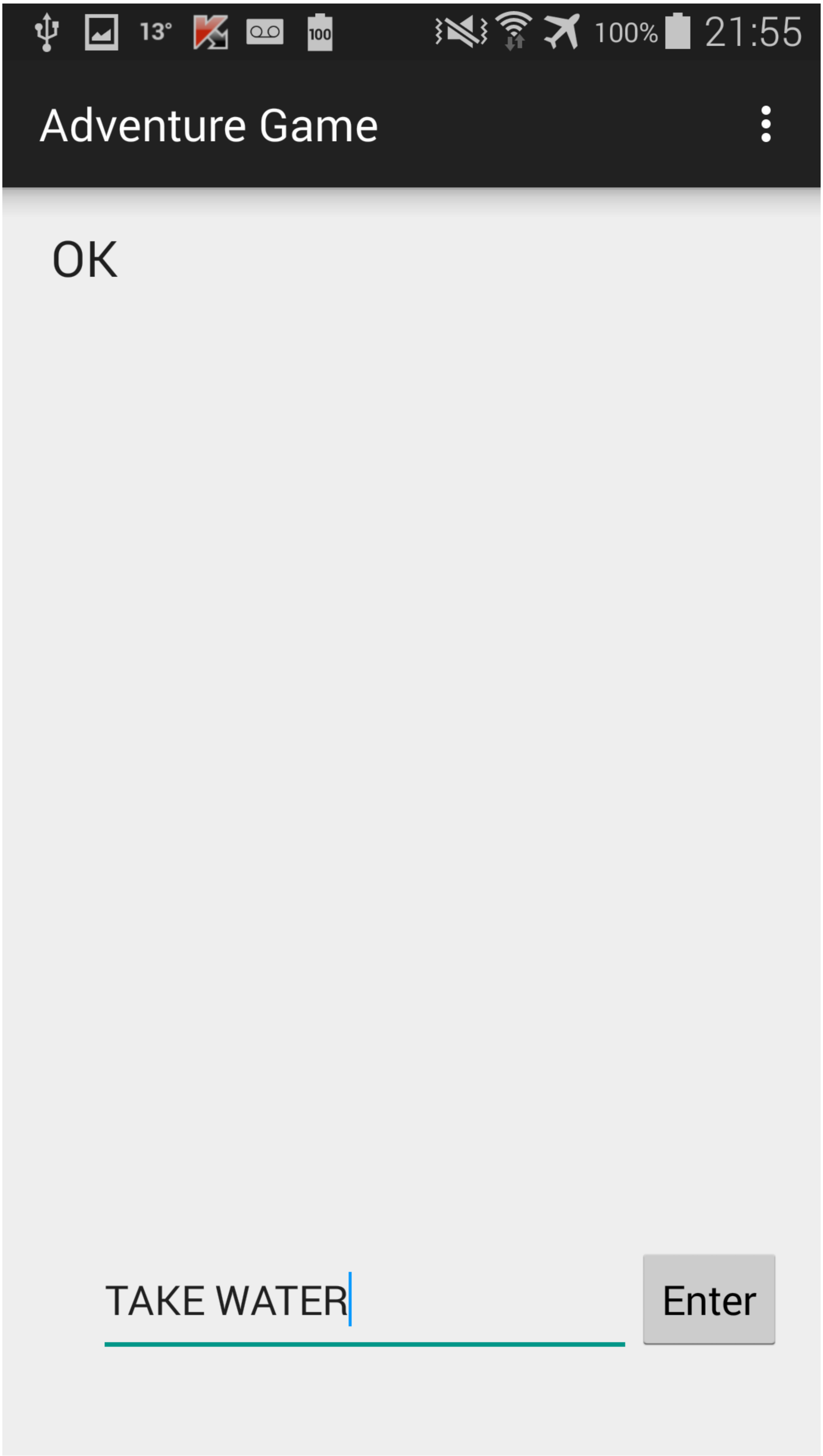
OK

TAKE LAMP



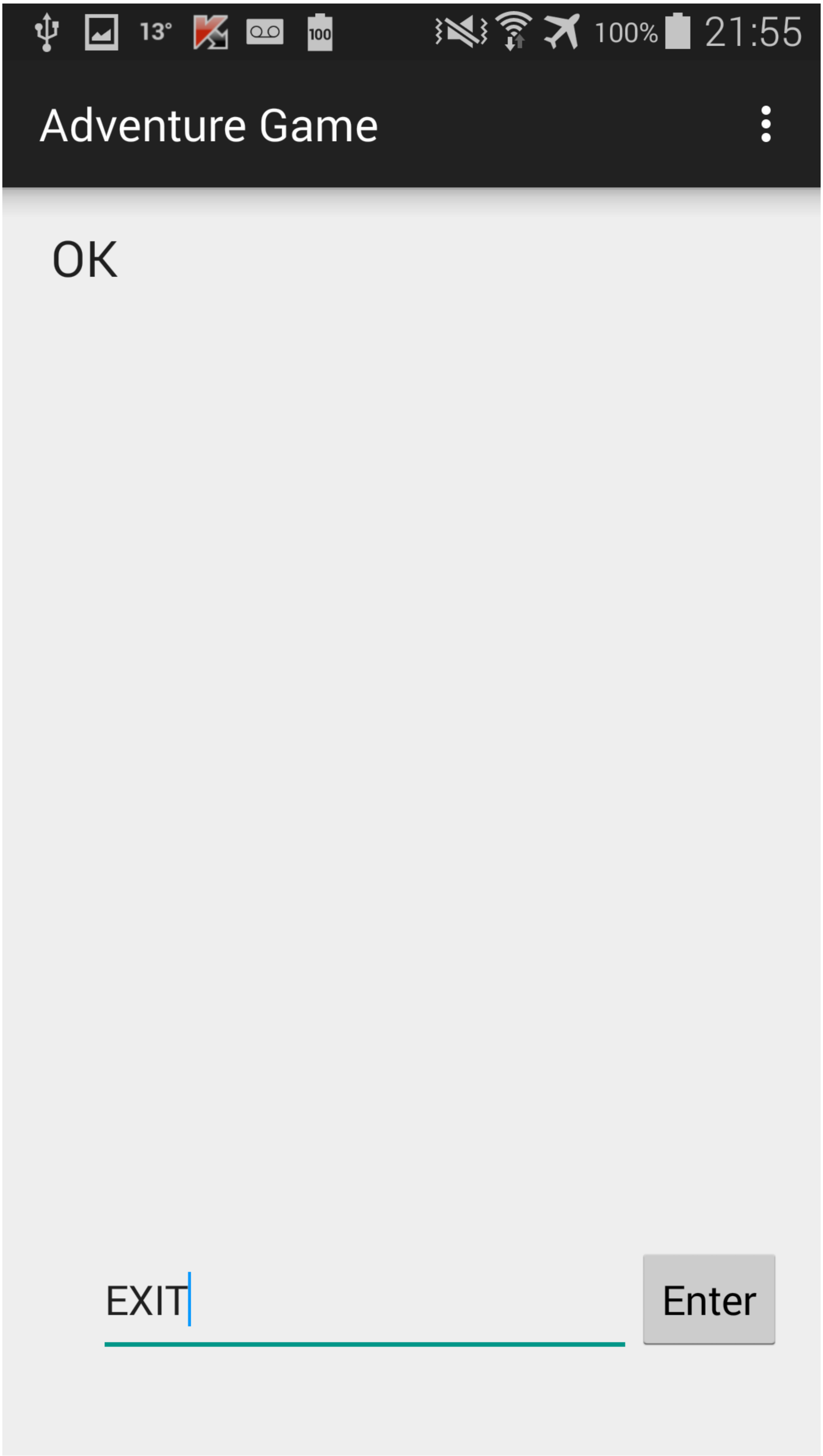
OK

TAKE FOOD

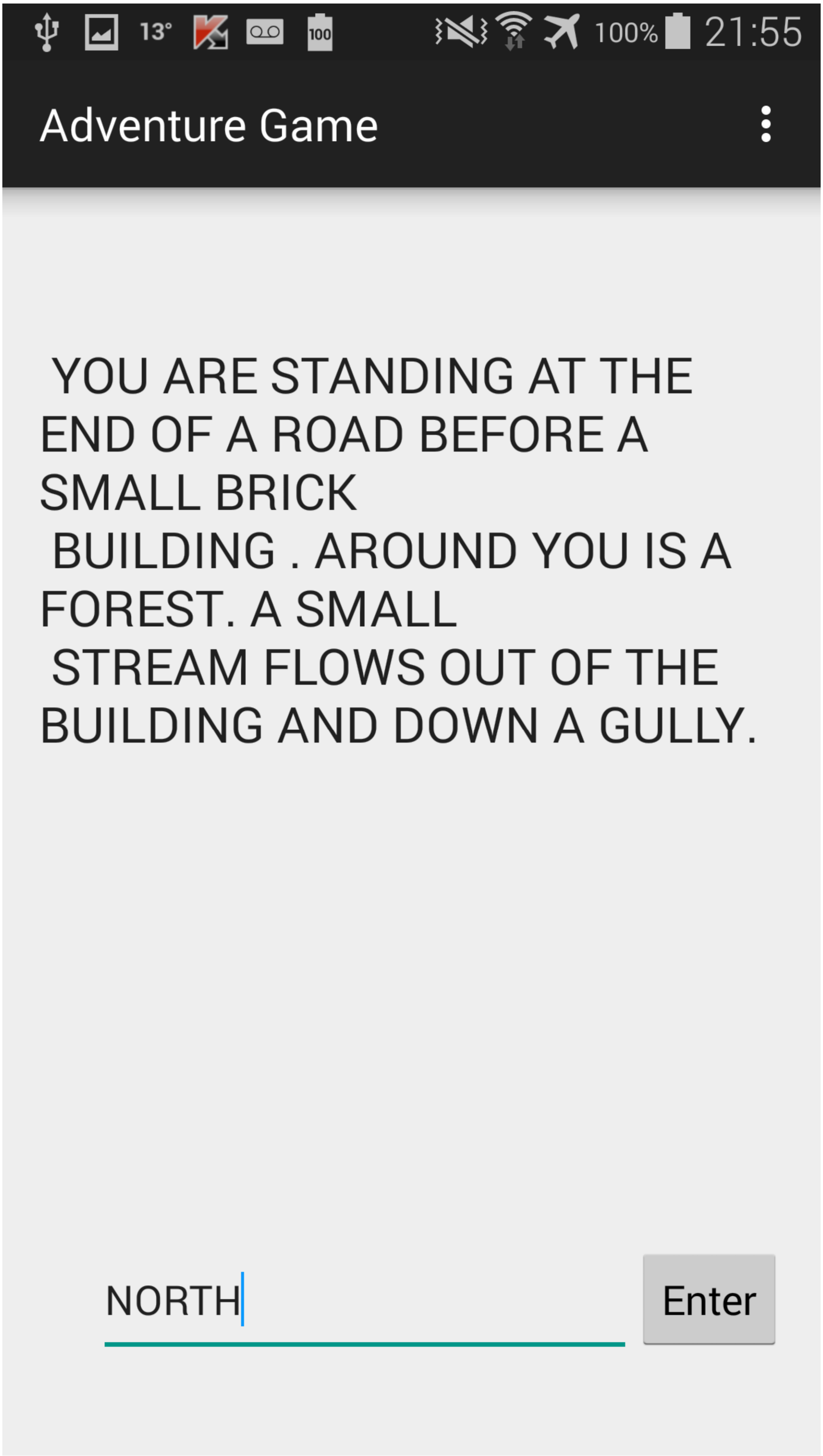


OK

TAKE WATER



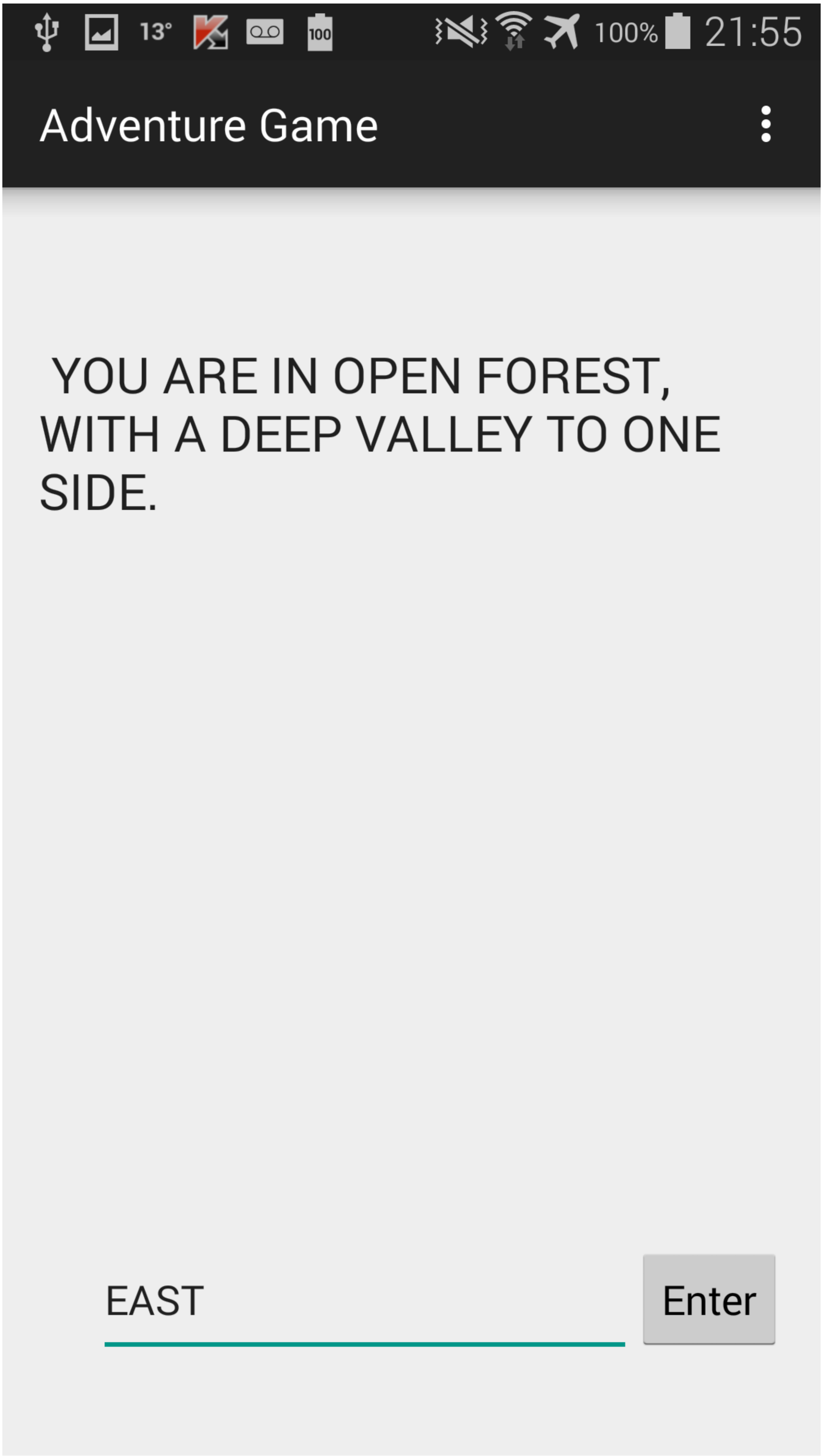
OK



Adventure Game

YOU ARE STANDING AT THE END OF A ROAD BEFORE A SMALL BRICK
BUILDING . AROUND YOU IS A FOREST. A SMALL
STREAM FLOWS OUT OF THE BUILDING AND DOWN A GULLY.

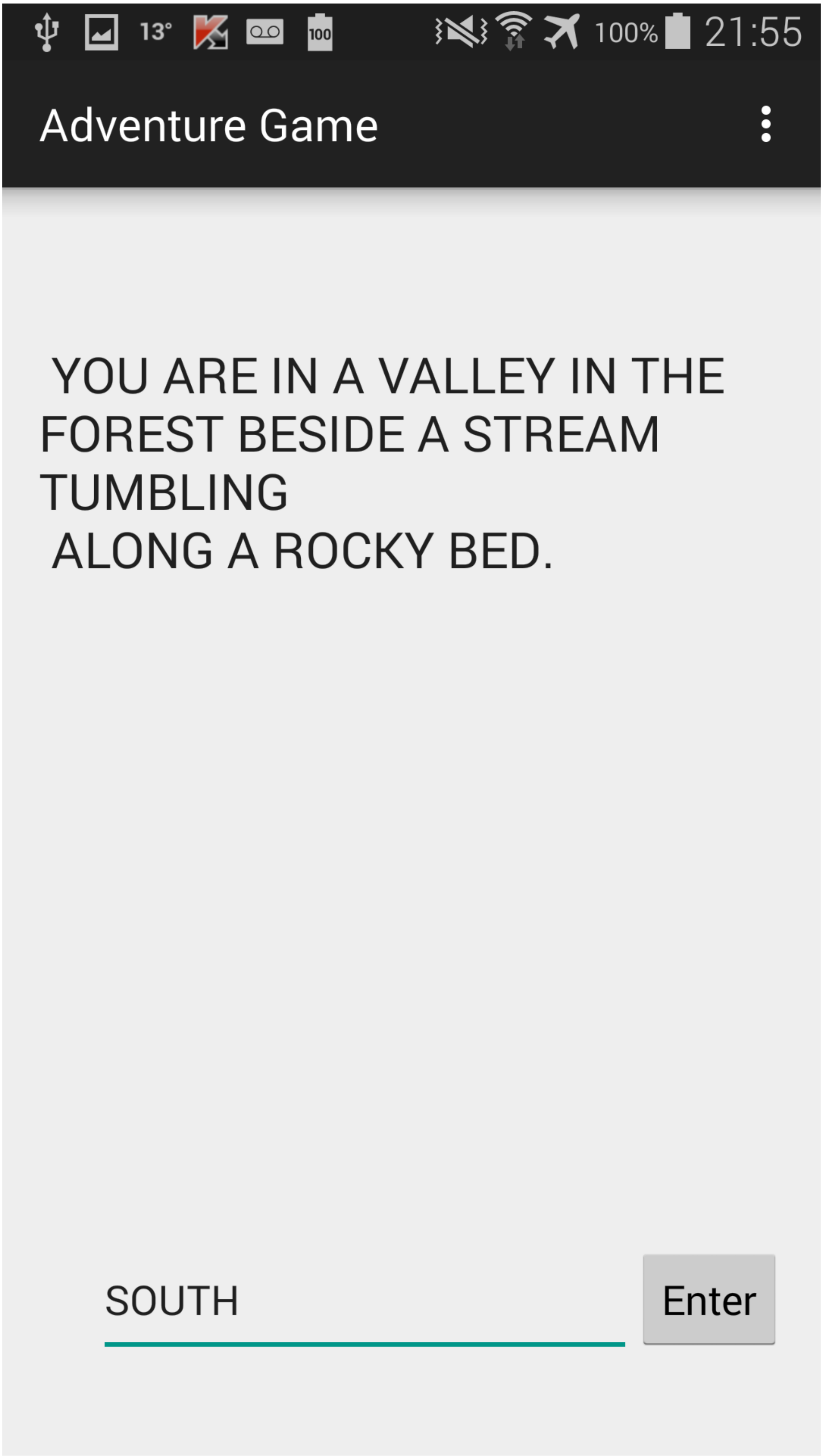
NORTH



Adventure Game

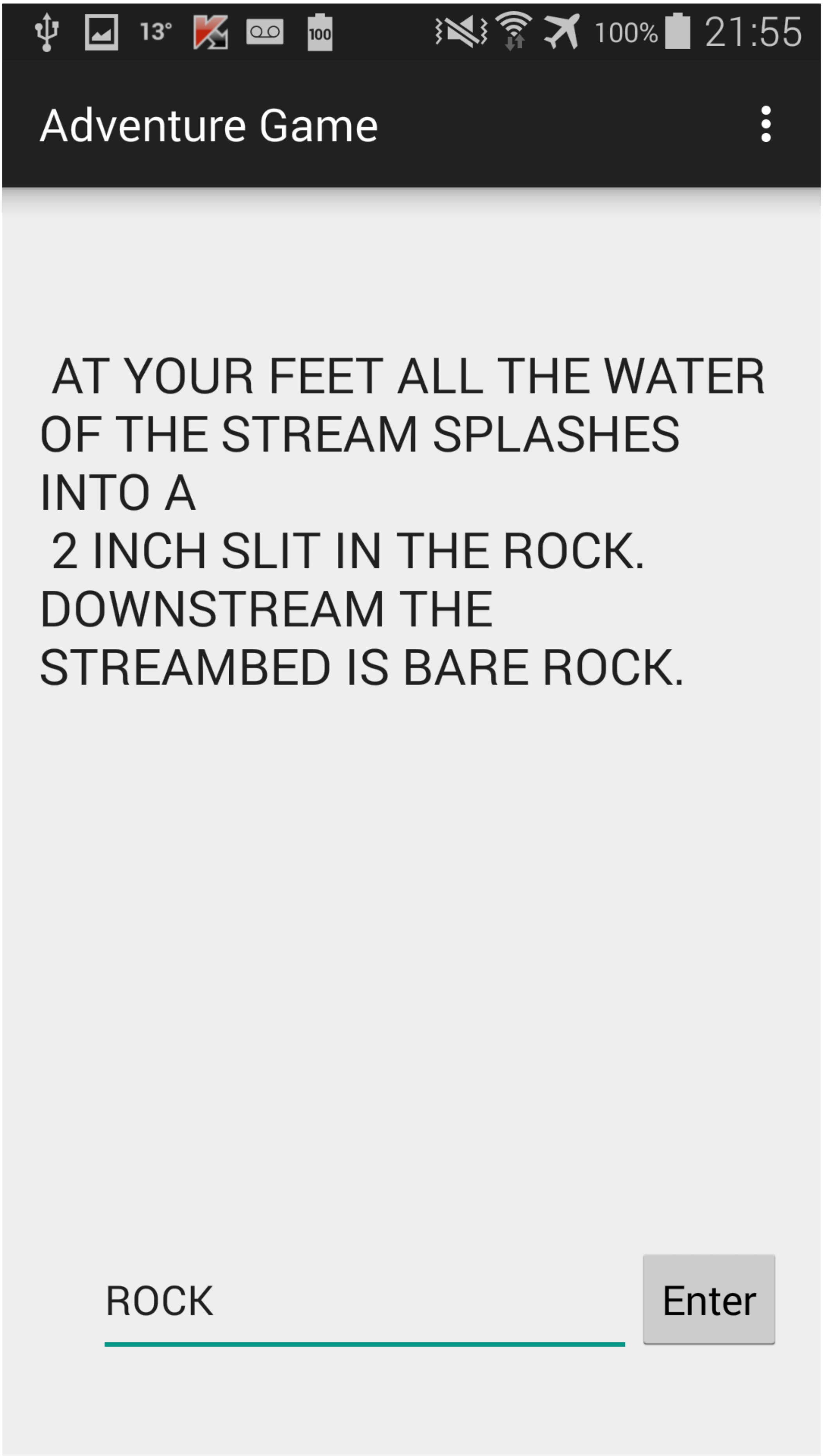
YOU ARE IN OPEN FOREST, WITH A DEEP VALLEY TO ONE SIDE.

EAST



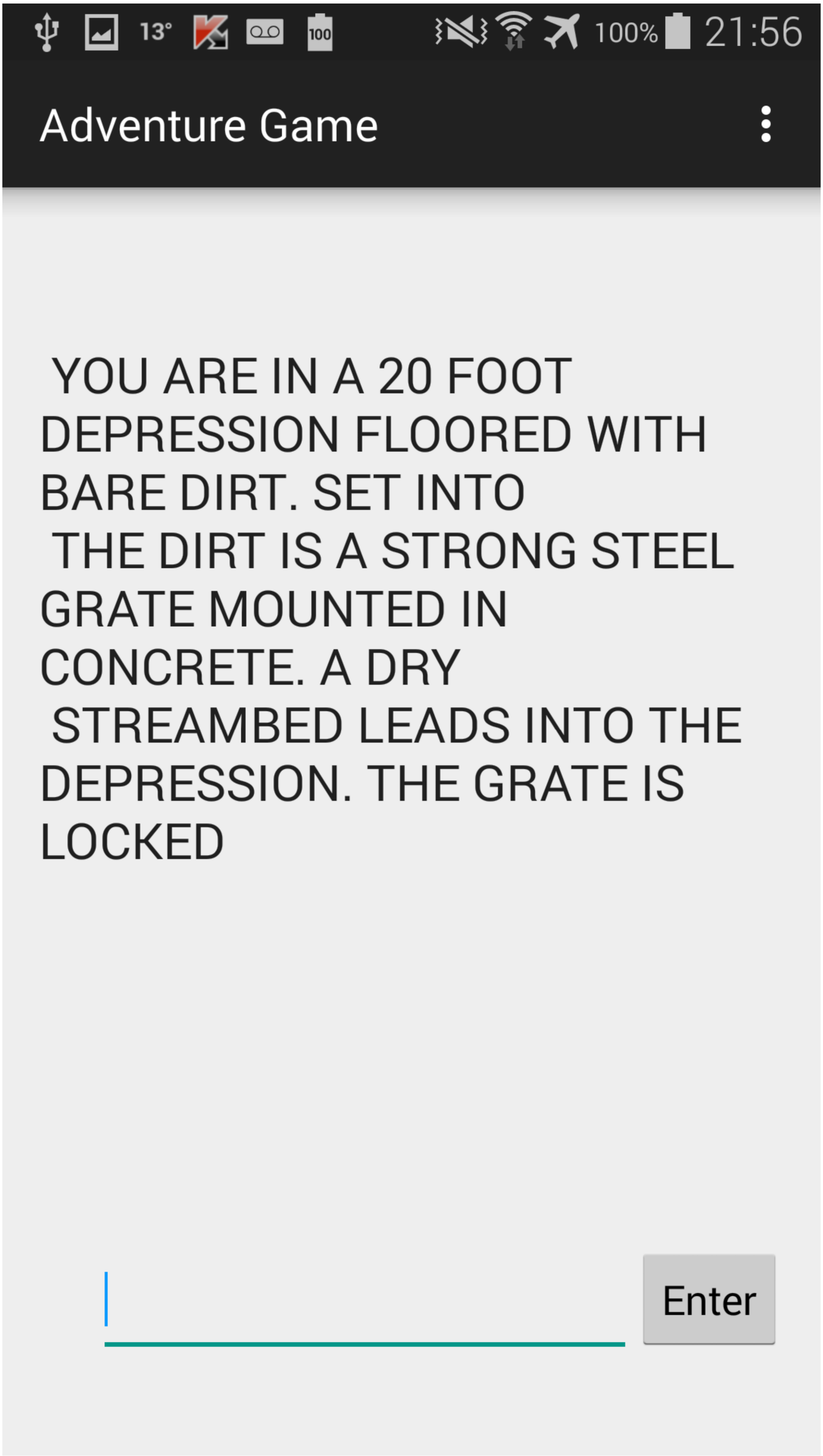
YOU ARE IN A VALLEY IN THE
FOREST BESIDE A STREAM
TUMBLING
ALONG A ROCKY BED.

SOUTH



AT YOUR FEET ALL THE WATER OF THE STREAM SPLASHES INTO A
2 INCH SLIT IN THE ROCK. DOWNSTREAM THE STREAMBED IS BARE ROCK.

ROCK



YOU ARE IN A 20 FOOT DEPRESSION FLOORED WITH BARE DIRT. SET INTO
THE DIRT IS A STRONG STEEL GRATE MOUNTED IN CONCRETE. A DRY
STREAMBED LEADS INTO THE DEPRESSION. THE GRATE IS LOCKED