Momo Software

Context Aware User Interface Application

USER MANUAL

Burak Kerim AKKUŞ

Ender BULUT

Hüseyin Can DOĞAN

1. How to Install

All the sources and the applications of our project is developed using Java, therefore no installers or preprocesses are necessary to work with the project. However, you need to have a Java Run Time Environment with JRE 1.6 all higher and Android Software Development Kit with support to version 2.3.3 or higher. The previous versions of both remain untested. Preferably an Integrated development environment such as Eclipse can be used but it is not a necessity. You can use the command prompt or terminal but Eclipse will reduce the time an effort quite a lot. Therefore, the following steps are explained assuming Eclipse is used with Android Development Tools (ADT) plugin. The source files and libraries are provided with the installation packet. There are three applications you need to have to run to completely use the main program:

Sensor Simulator: A Java application provided to simulate the accelerometer with many other sensors

Sensor Simulator Settings: An Android application provides connection between the Android device emulator and Sensor Manager

The Android Application: An Android application that is our main program

1.1. Sensor Simulator

This is an application provided by OpenIntents.org to simulate the sensor events for an Android emulator such as accelerometer, orientation, temperature, GPS etc. The project directory with libraries and source files is included with a "Read Me" file of the application. To examine the codes and others, you can import the project to an IDE or open the files with an editor. This is a standalone Java application so just run it. No preprocesses are required.

1.2. Sensor simulator Settings

This application is also provided by OpenIntents.org. It is an Android application that you need to run on the device you want to connect to the Sensor Simulator. Open the project directory with an IDE, and then choose a target device to run the application.

1.3. Android Application

This is main program we developed. It is also an Android application so just import the project on the IDE and run it with the same target device you run the Sensor Simulator Settings.

2. How to Run

2.1. Sensor Simulator

This program provides a simulator for most of the sensor an Android device could have. However we only use accelerometer. When you first open the application, take a look at the left side There you can see log area to display the changes in the sensors. Moreover, the IP addresses you need to use to connect the device to the simulator are also displayed there. There are two different ways to change the accelerometer values. One is to move or rotate the phone image that place on left top of the window. The second is to move the sliders on top next to the phone image. The changes should be reflected directly to the log area and the device emulator.

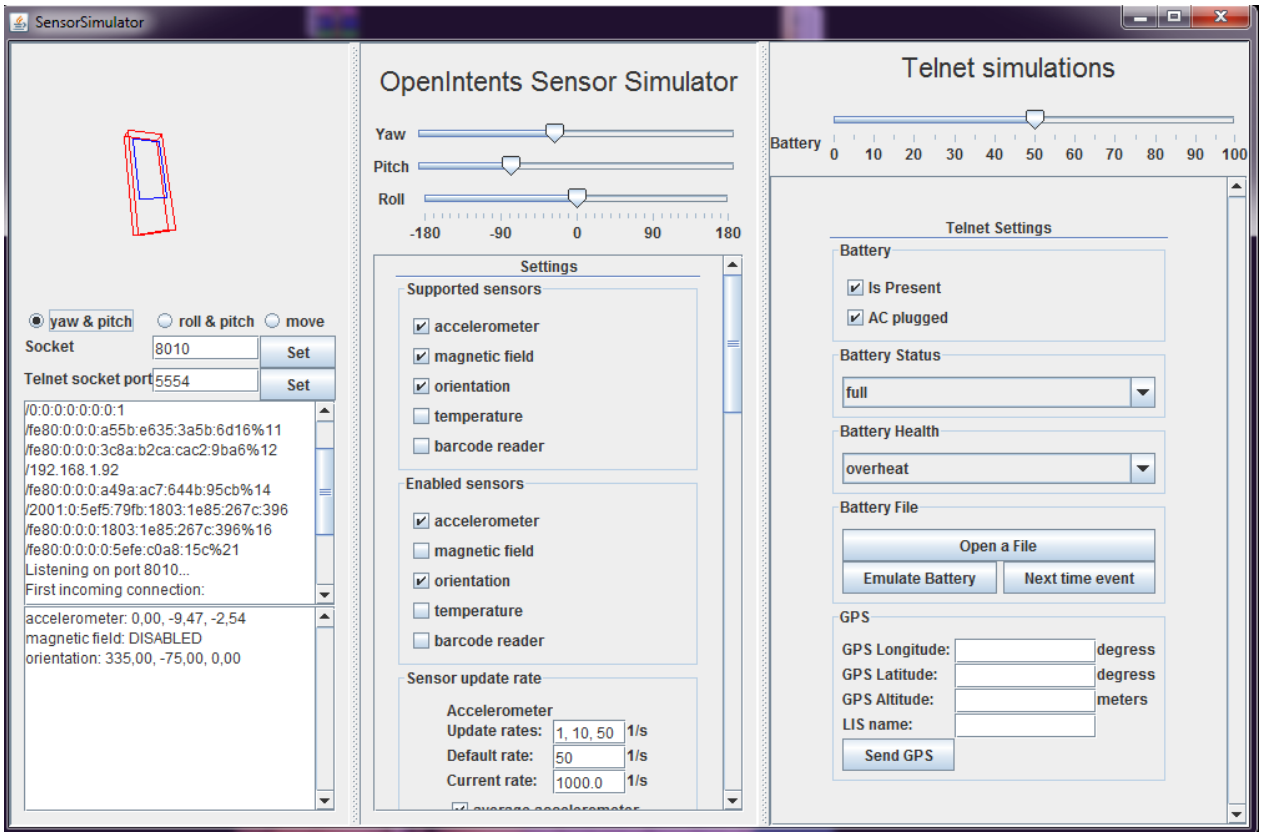

Figure 1: Sensor Simulator Window

2.2. Sensor Simulator Settings

This program sets the connection between Sensor Simulator and Android emulator. There are two tabs in this application one is for set up, the other is for testing. After starting the application, you need the set the IP and the port number given by the simulator. They can be found on the left side of the simulator window as mentioned previously.

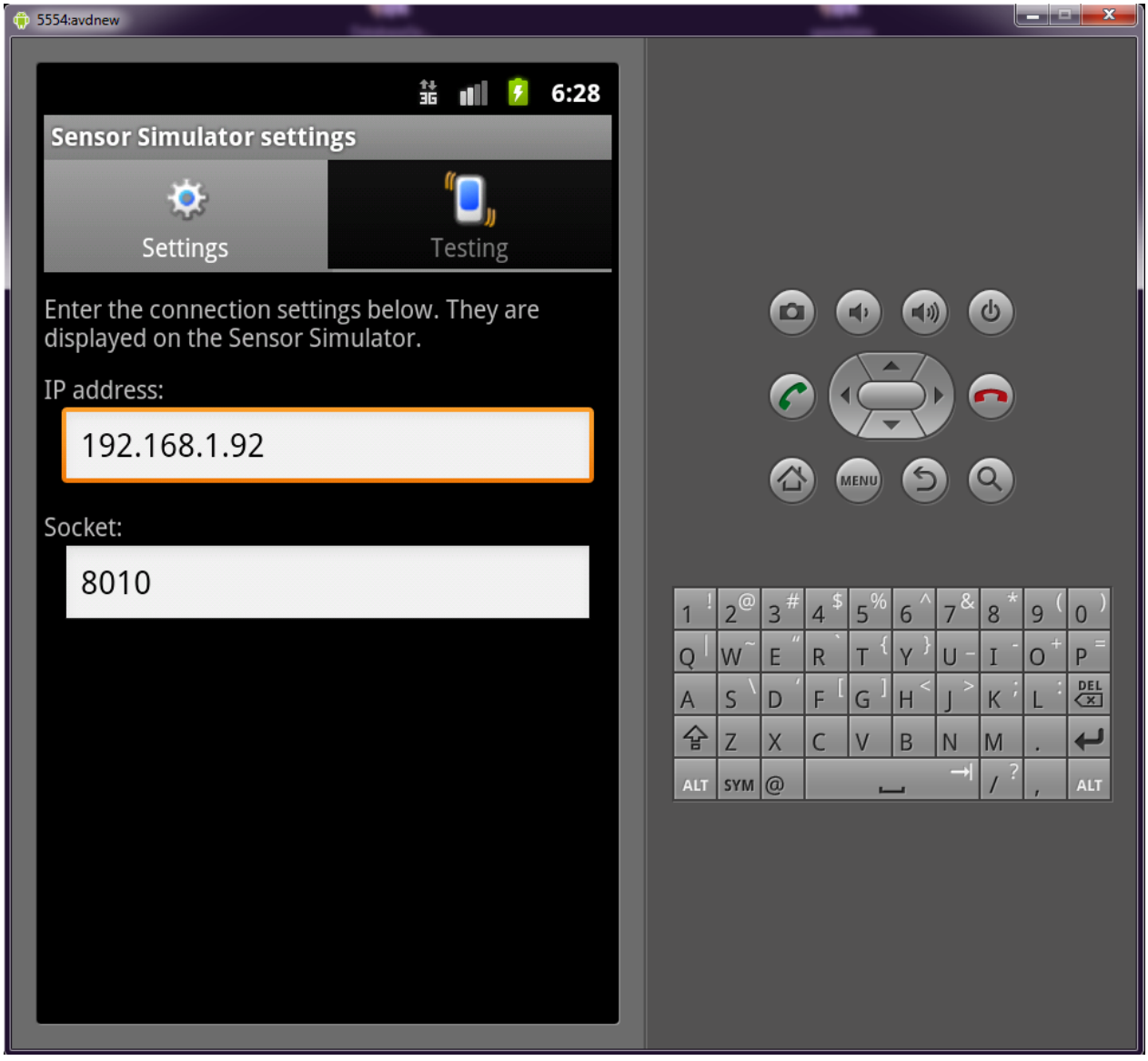

Figure 2: Sensor Simulator Settings setup tab

After setting up the connection you should see the same values of accelerometer, orientation and magnetic field. Just check the accelerometer. You can also set the update rate but you do not need to, the default is fine.

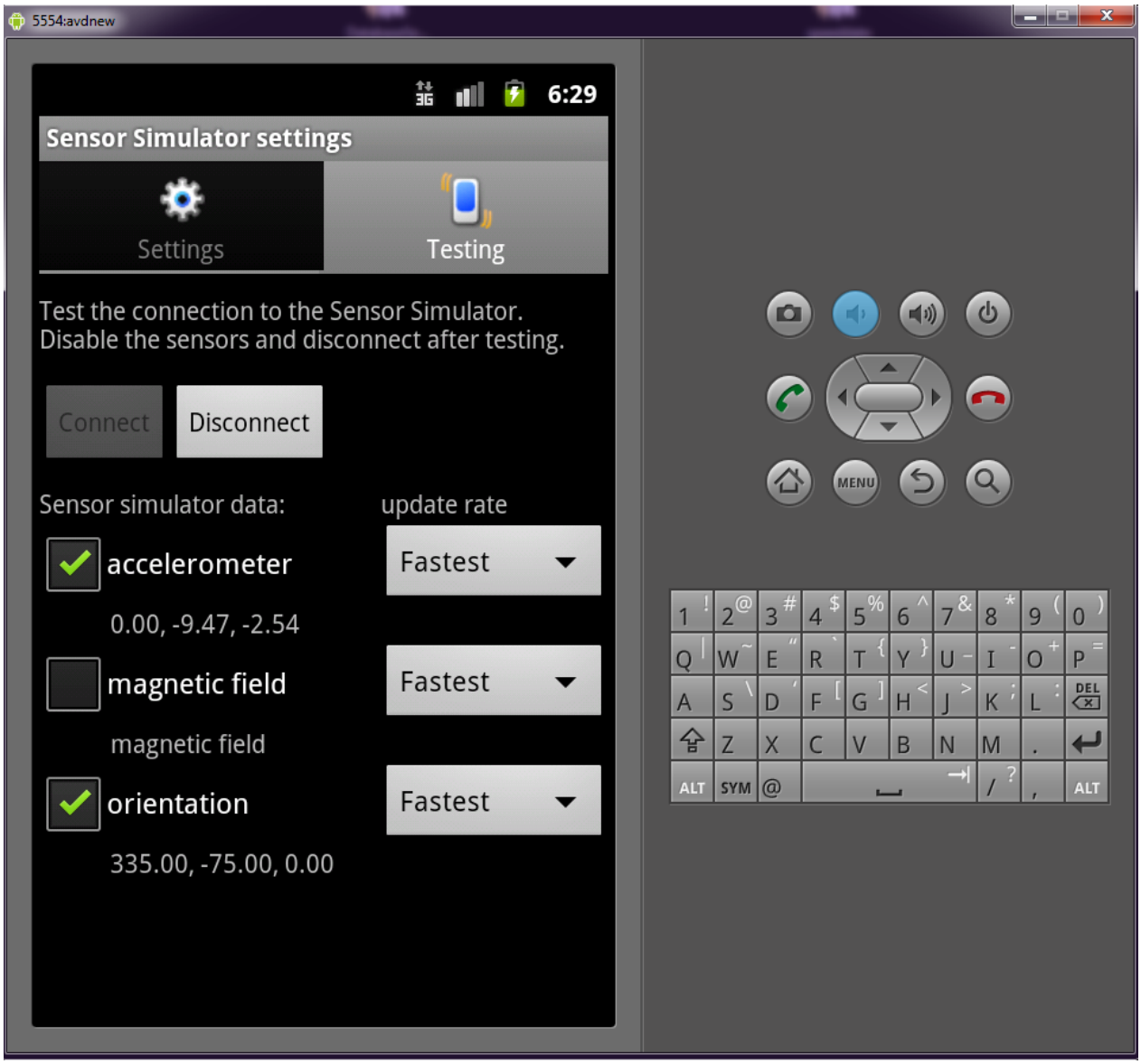

Figure 3: Sensor Simulator Settings testing and adjustment tab

2.3. Android Application

On first run take a look at the main window. The map covers the most of the area and it is also the main interaction part. There is a clock placed on top of the screen. The light simulator button is placed on the top right corner. Moreover, there are two buttons on the bottom to access the missions and messages screens. If this is the first time you run the main program, start by using the settings opened by pressing the menu button and selecting the "Settings" item.

2.3.1. Settings

This window provides a basic control over the application. It has three main items: calibration, sensitivity and map type**.**

2.3.1.1. Calibration

This is provided to adjust the device to movements of the user. The device needs to obtain the user specific variables on first run to provide a better interface. After selecting the calibration from the settings window, just follow the instructions. Firstly stand still until the progress bar is full then start walking to complete the calibration. When it is finished you should see the baselines that are measured from your movements.

2.3.1.2. Sensitivity

This item sets a sensitivity value to add as a cofactor to compute the context around the user. Just select the value you prefer if you are using a real device. This item is provided for use in a real device, therefore you do not need to use it for the device emulator.

2.3.1.3. Map Type

You can choose whether to display the territories around the departments or not. Just select the item and choose the map type you prefer. You should see the map as desired on the main window when you exit the settings.

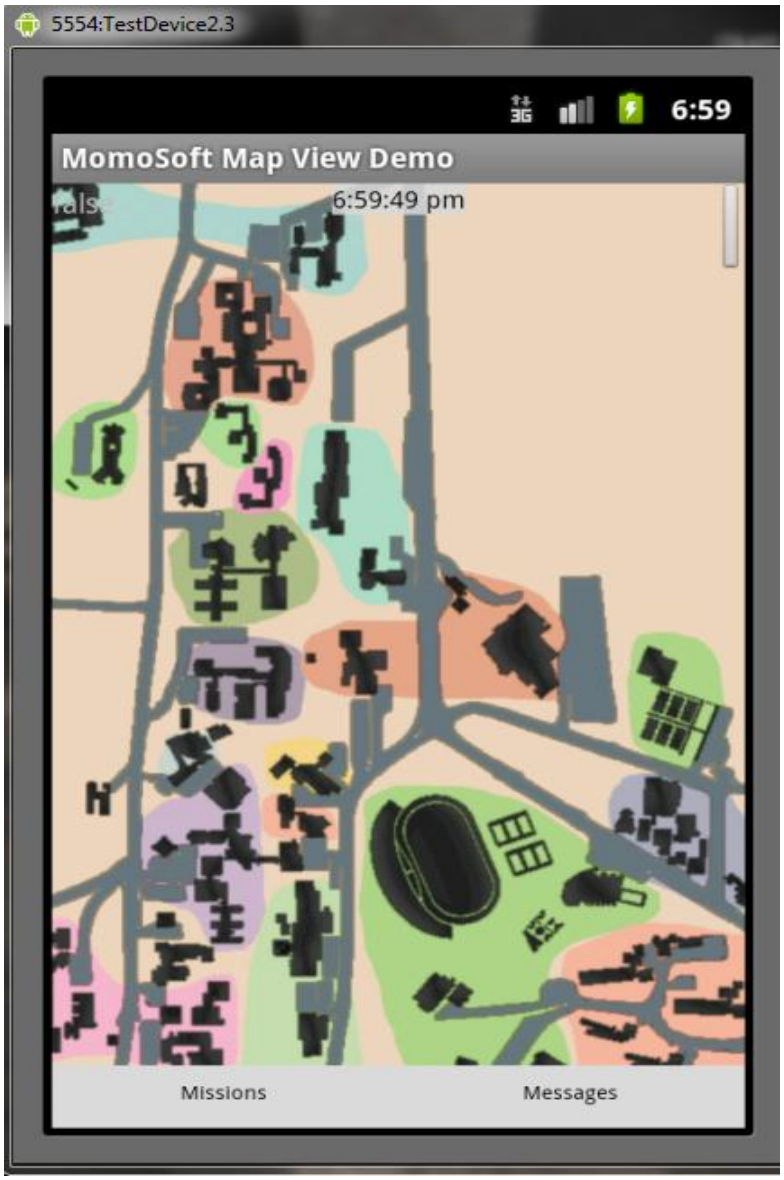

Figure 4: Territories are shown on the map

2.3.2. Reset

Reset option sets the map to its default zoom level and center of view. To use this option open the menu and choose reset.

2.3.3. Main Window

In the product there are some buttons and small icons and it is easy to understand how to use it.

2.3.3.1. Missions and Messages

First of all, we start with two buttons at the bottom part namely; Missions and Messages. In our scenario, there are many missions that were assigned to or many messages that were sent to the user. The user can easily access missions and messages by clicking these two buttons. After pressing, he/she can see the list of related items. Furthermore, at the top of our main screen there is a clock that enables the user to follow the time.

2.3.3.1. Map

Secondly, the most important subject about our main screen is the map and its usage. Firstly, we have three maps and the user can use any of them according to the light condition of environment. These three maps are suitable for environments with light, low light and dark respectively. The user can change the type of map just by clicking the small button at top-right of the map. Moreover on the each map, there are some icons that were drawn by different colors. One of the colors represents the items (building, forest etc.) and the other one represents the teams on the map. The items and teams are read from xml files and they have different characteristics. (name, size, priority etc.) In the map the user can see the icons as a circle and find the characteristics of them at the below part of icons. For example in the first map, the teams are represented as green circles and the items are represented as red circles. When the user changes the type of the map to adapt light condition, he/she can see the teams in yellow color. We changed the colors in order to enable the users to see the map more easily. In the final map, the teams are represented as blue circles just for the same reason. In the initial states of maps, the name and size info about the teams are shown below the related circles. However the only info that can be found about the items is the name of them. When the user becomes active, the info that is shown on the map will be changed.

2.3.4. Context

There are two main concepts for context information to adapt the graphical user interface in this application.

2.3.4.1. Accelerometer

Sensor simulator application is used in order to observe the changes on the screen due to change of acceleration parameter. This simulator is connected with our application so that user can change the results of acceleration values by the red rectangular prism can be seen on the left top of Figure 5. User can choose a mode from "*yaw&pitch*", "*roll&pitch*" and "*move*" modes. After mode selection, user can yaw, roll or move the object so that mobile device can understand the acceleration changes and user state.

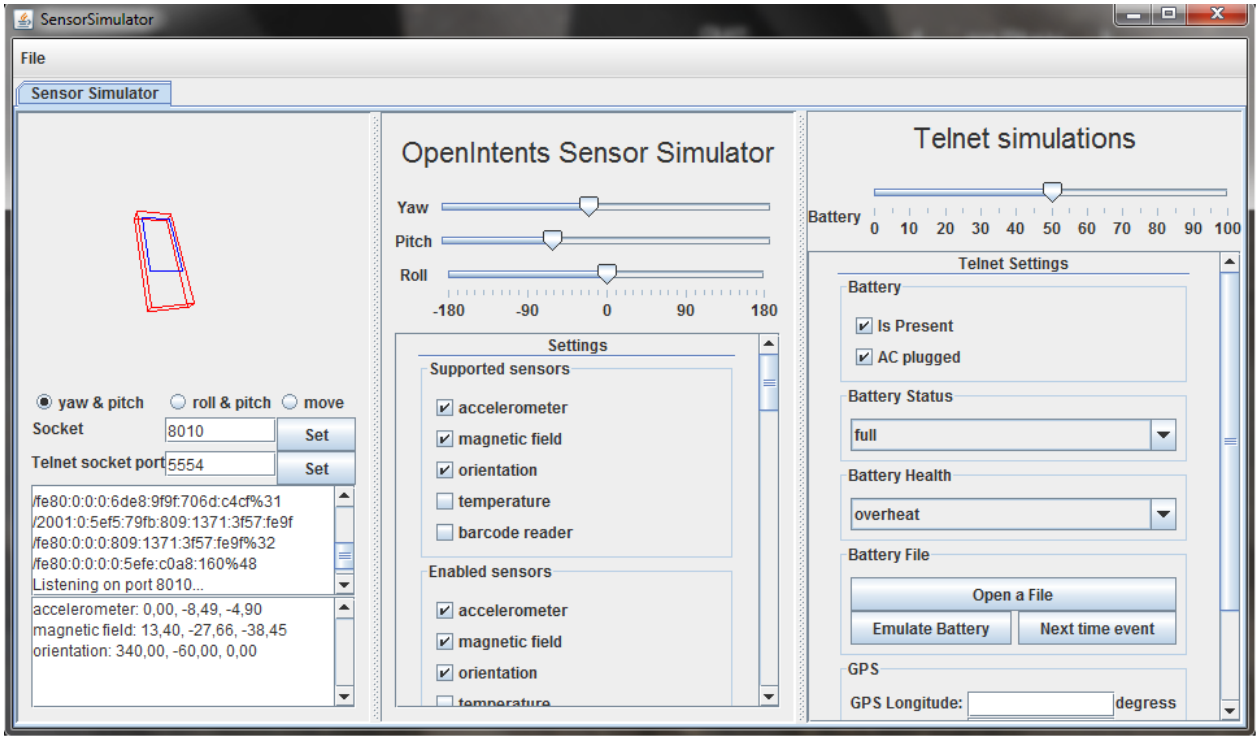

Figure 5: Sensor Simulator

There are three motion state called *standing, walking* and *running* mode for user state. These modes can be seen differently in Figure 6, Figure 7, and Figure 8. Three states are specified in order to prevent unnecessary changes in user interface.

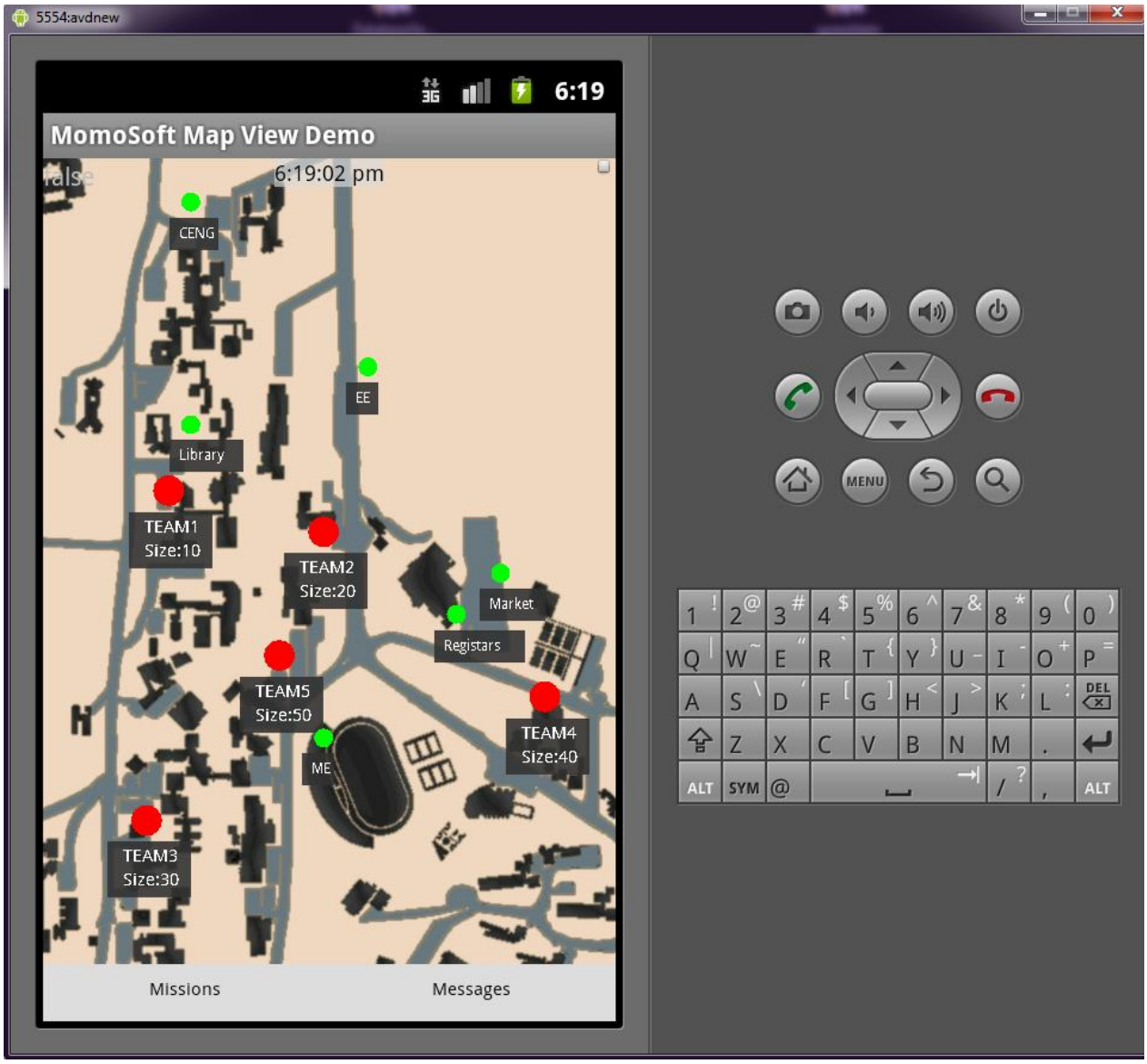

Figure 6: Standing Mode

The red and green items, its information boxes missions-messages buttons and clock in “Walking Mode” are bigger than in “Standing Mode”. (Figure 7)

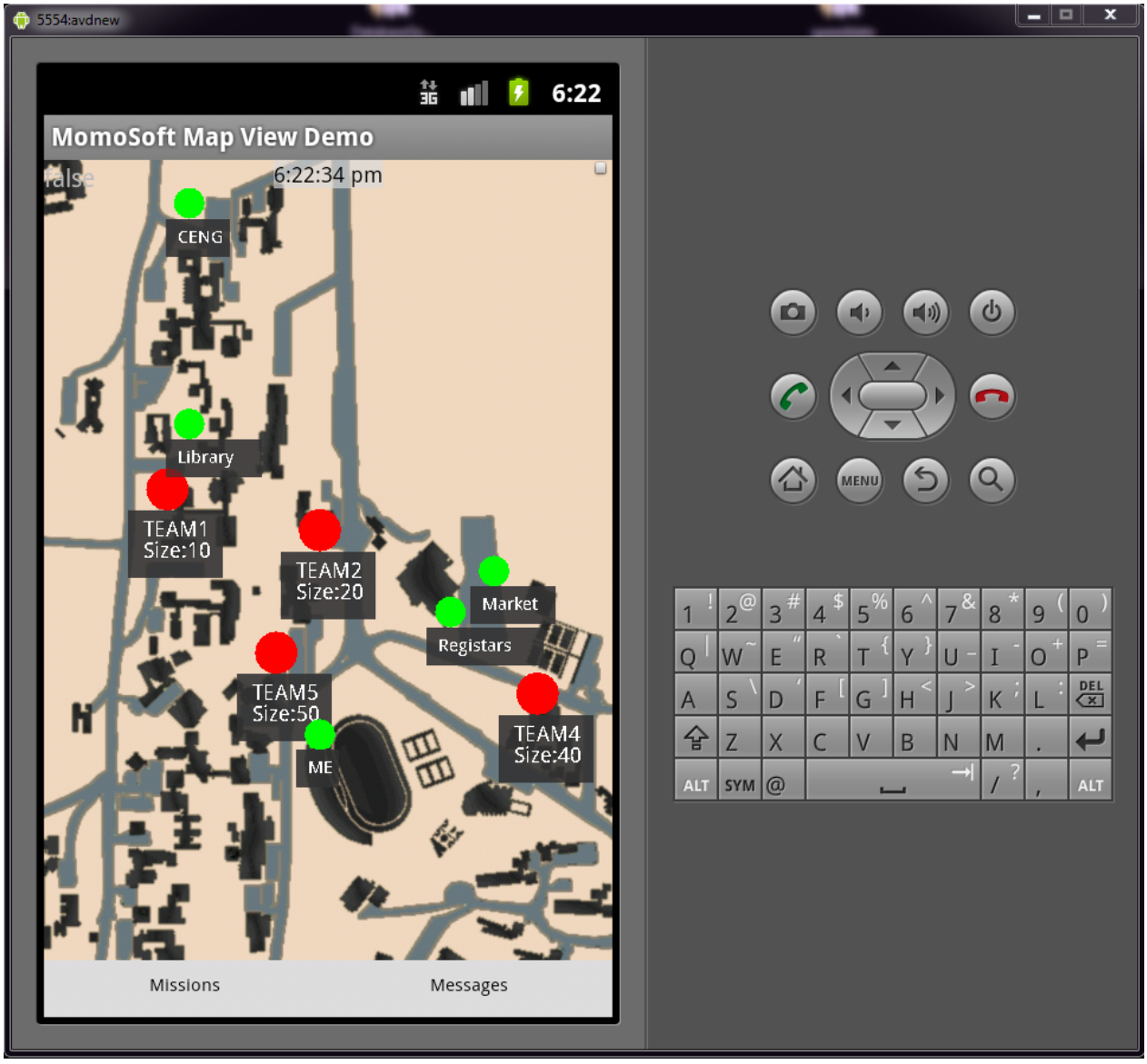

Figure 7: Walking Mode

The red and green items, its information boxes missions-messages buttons and clock in “Running Mode” are bigger than in “Walking Mode”. Different from the previous change of state, there are only important items specified in terms of the priority values of them in the screen. (Figure 8)

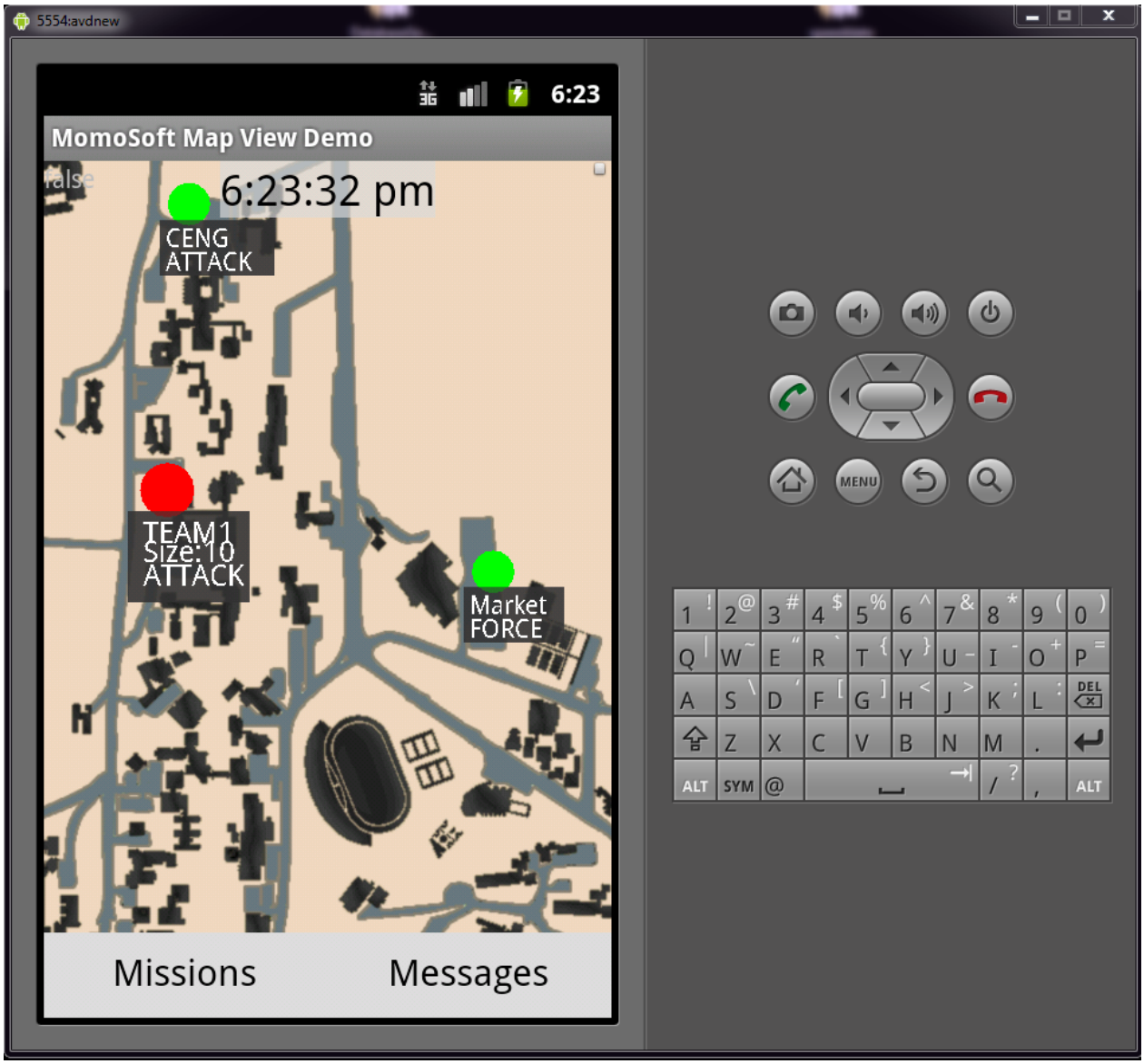

Figure 8: Running Mode

2.3.4.2. Light

Android has no light sensor in order to understand the intensity of light in user's environment and so this application provides three light modes that is optional to use for users. A user can select a light mode by using the small button in the top-right of the screen. "Day Time Mode" can be seen in the previous figures. The "Dark Mode" for user friendly interface can be seen in Figure 9.

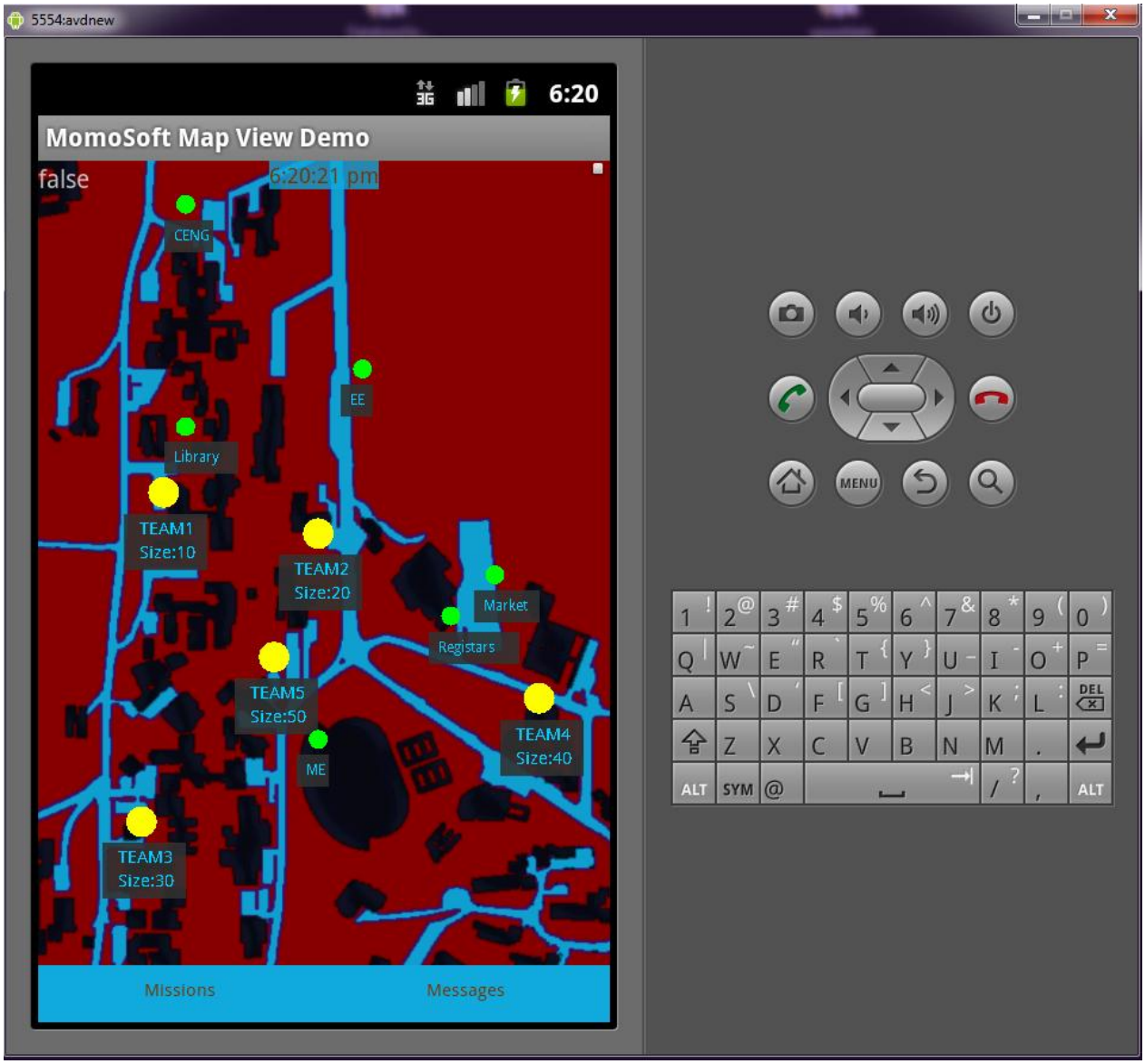

Figure 9: Dark Mode

The “Night Mode” for user friendly interface can be seen in Figure 10.

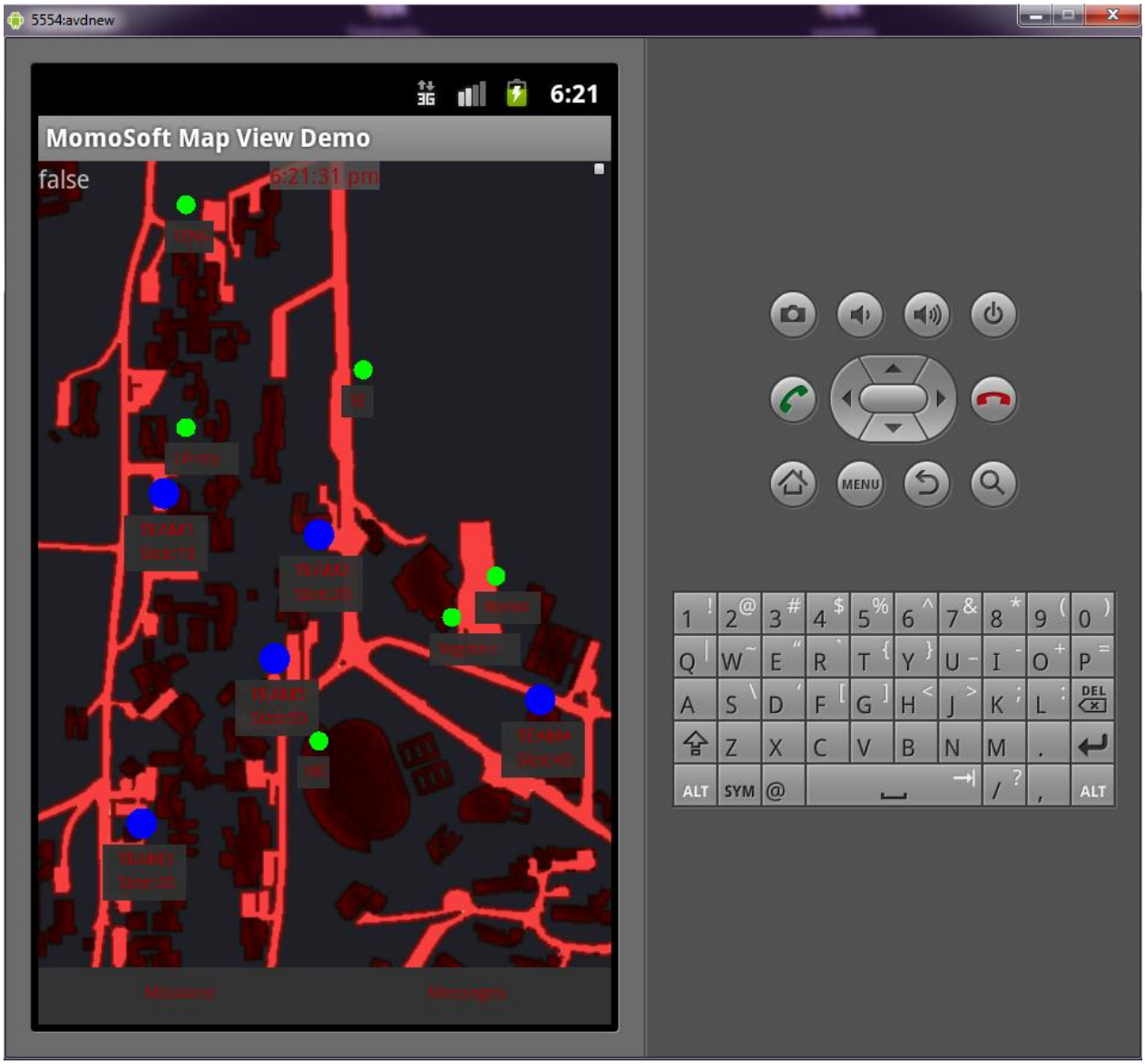

Figure 10: Night Mode