

# botMania – A Cookbook On How To

Most computer games require the user to control the player in a virtual environment with the use of some input device like the mouse or the keyboard. In botMania, you are required to write a program that does it. So one can call botMania a programmer's game. The players in the virtual environment are called bots. Your program can make the bot move, scan the arena for enemies, throw bombs, send messages to its team members, etc. Typically two such programs are made to fight against each other.

There are two teams with three bots each, each controlled by a different program. The objective is to capture the flag of the opposition team while protecting your bot from the enemy. For each bot , one programme in C or C++ is required to be written.

## **botMania- A basic Overview:**

- The application is essentially a simulation server similar to microsoft herbert and linux AI bots, with as much as double the functionality and many times the excitement. Your bots can fire, scan for enemies, send messages, all through a piece of code that you write
- The very idea of such an IDE for bot programming strikes the thought of its deployment at a large scale such as various programming portals like topcoder, the well renowned organization that encourages development of algorithmic, logical and innovation skills in the mind of the programmer through collective knowledge sharing.
- botMania aims at establishing a network in which people will compete against each other, fire, send messages, but all through code!
- When the network grows large, such as other bigger programming networks, activities can even be hosted on the international level on the botMania server which is platform independent and based on totally open source technologies, and hence is very near to the common people.
- The essential aim of botMania is to bring out “raw intelligence” in a human being. It is superior than methods that only test the algorithmic skills of a person because along with the code, the person's strategic development can also be tested.
- The crux of the matter is that it is the “one of its kind” simulation server which has entirely built afresh without any use of pre-existing code whatsoever. Near about 5.5kloc of code was written by team botMania specifically for the purpose of proposing a better atmosphere for the larger intelligence development of the users of the application.

## **Minimum hardware and software required for starting the game**

Minimum System Requirements:

- 256 MB minimum, 512 recommended
- Processor, Pentium 4 or equivalent

Minimum Software Requirements:

- JDK 1.6
- MySQL (Preferably a recent version), WAMP Server for windows
- [Dev-cpp \(windows\)](#) or g++ (available with linux distros)
- Operating System: Linux (various distros) and Microsoft Windows (XP/Vista/Windows 7)

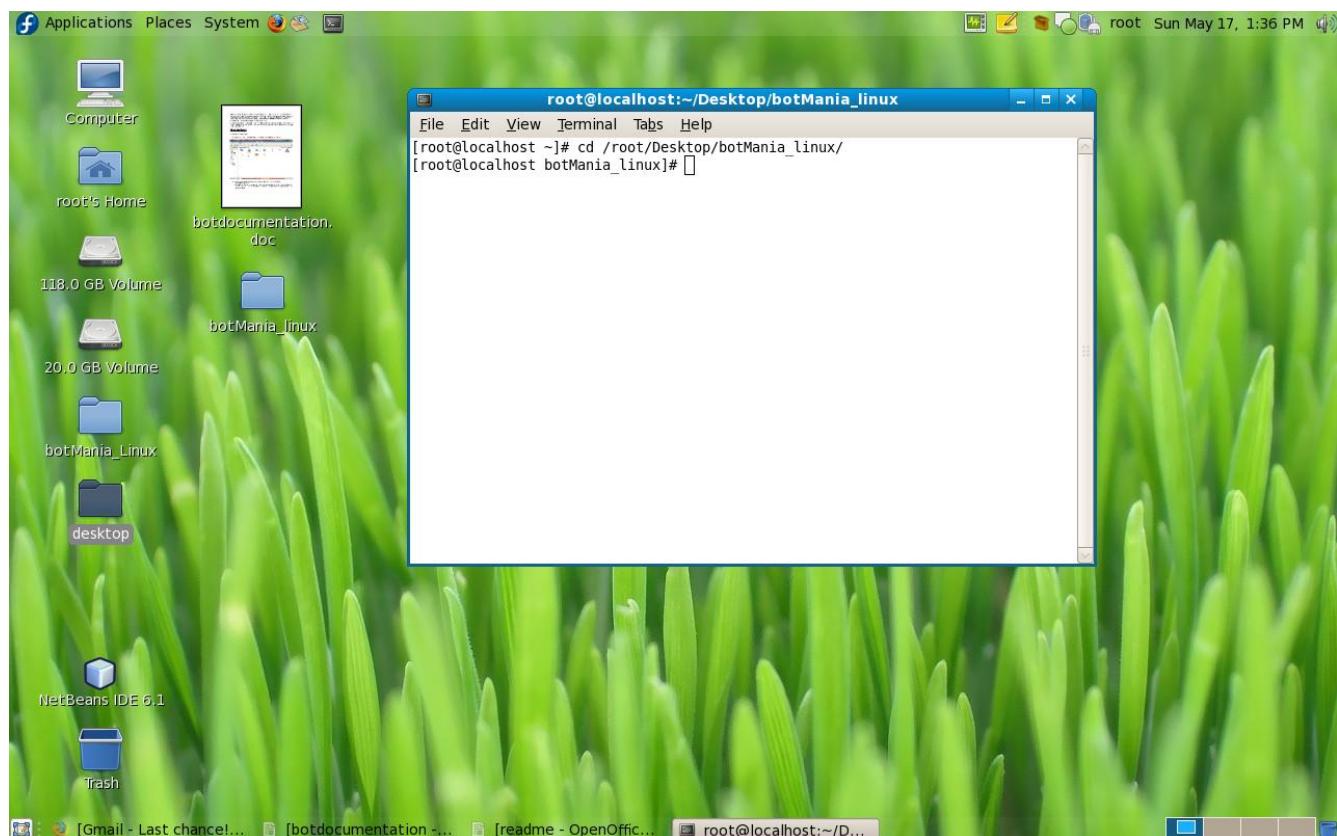
## **Initial setup required for starting the game**

- Make sure that jdk 1.6 or above is installed and running in your system. In case it is not, kindly download the freely available software from sun website.
- Make sure that mysql is installed and running in your system. In case it is not confirmed, You can check the status of MySql server by # **service mysqld status**  
In case it is stopped,start it using # **service mysqld start**  
In case it is not, kindly download the freely available software from sun website.
- In case you are using windows be sure to use MySql service provider like the WAMP server
- Setup the database as shown in the subsequent stages. Be sure that you know the root password to create new database.

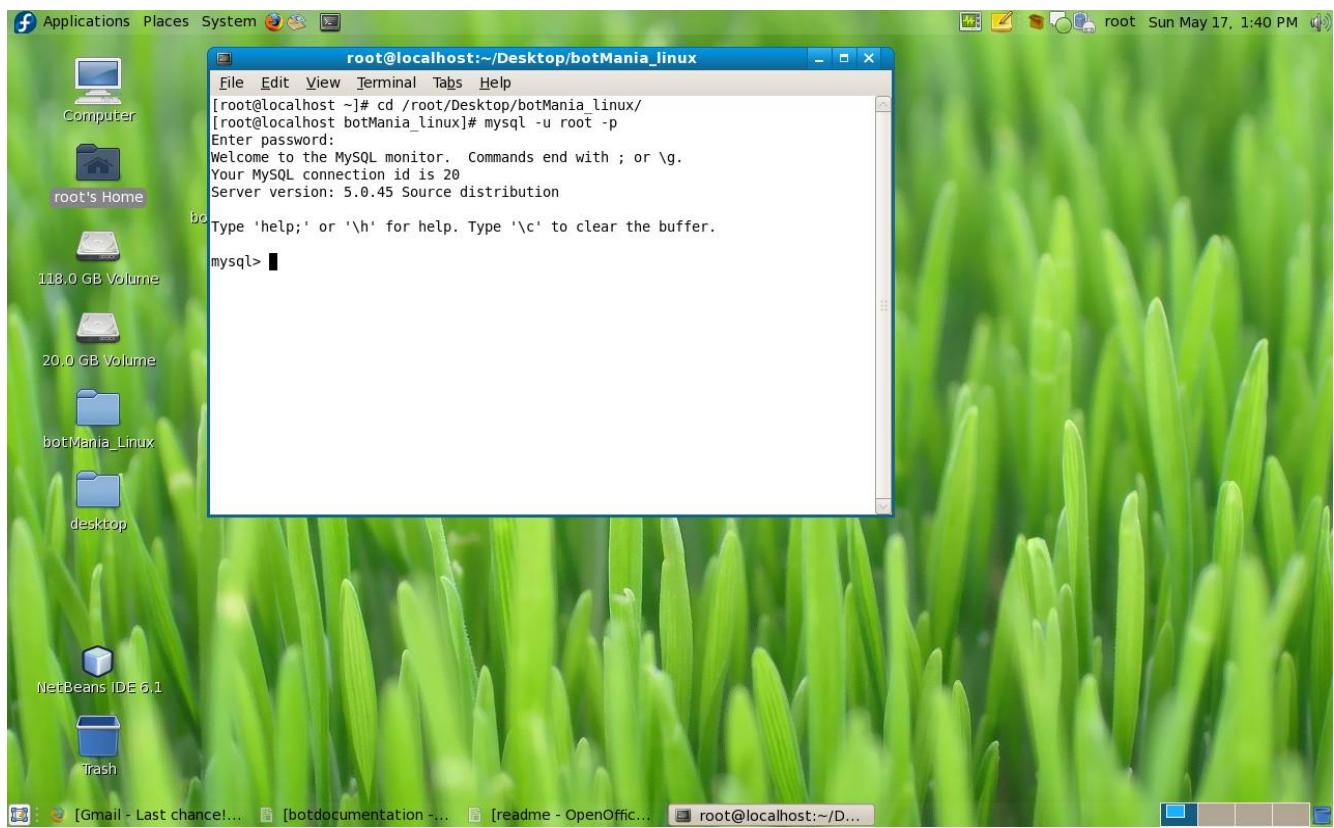
## Setting up MySql for botMania

The setup of windows can be done directly from the wamp server or command line by following similar basic steps as described below so we will not discuss it. We will discuss the configuration on linux.

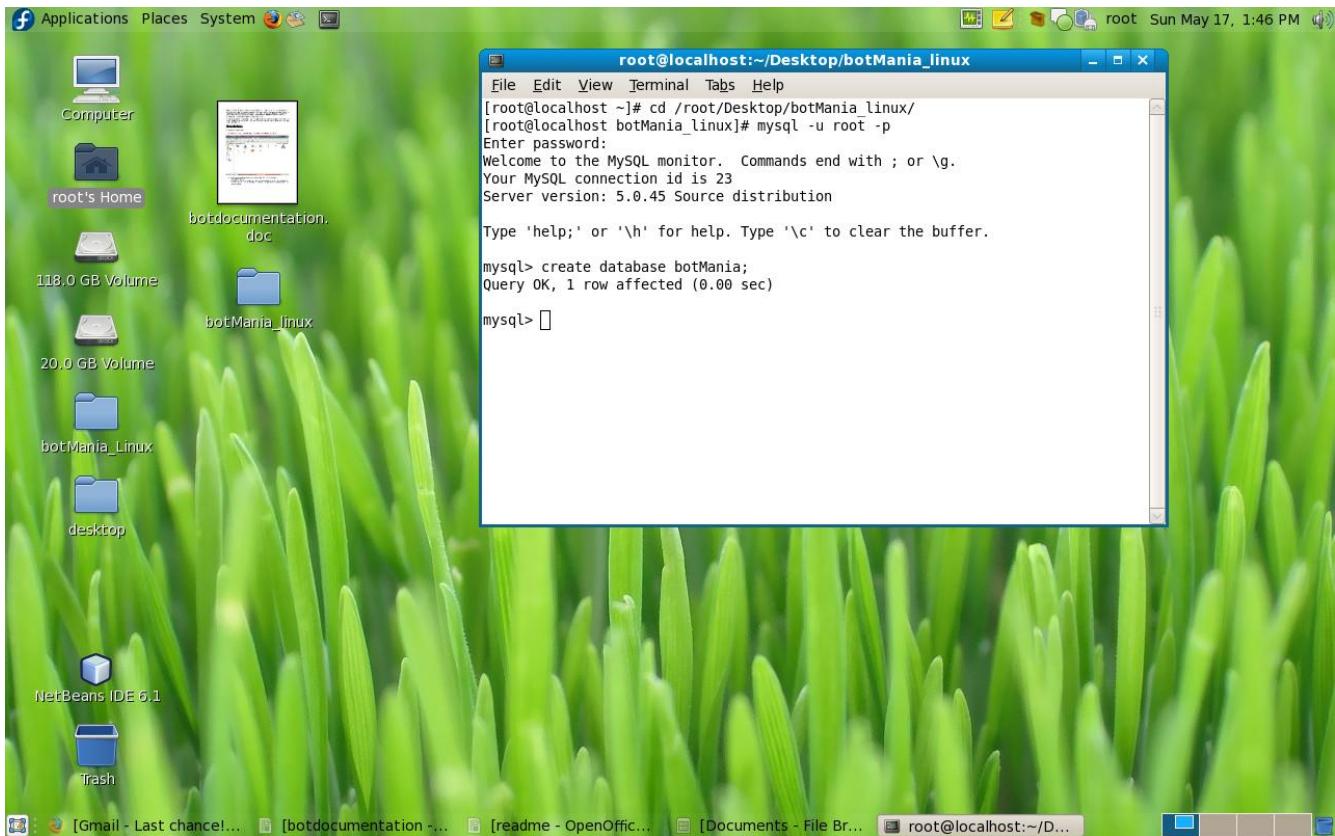
- Use the cd command to reach the location of botMania.sql in the provided package. In our case, the provided package is botMania\_linux and is placed at desktop. So we do the following in the terminal:



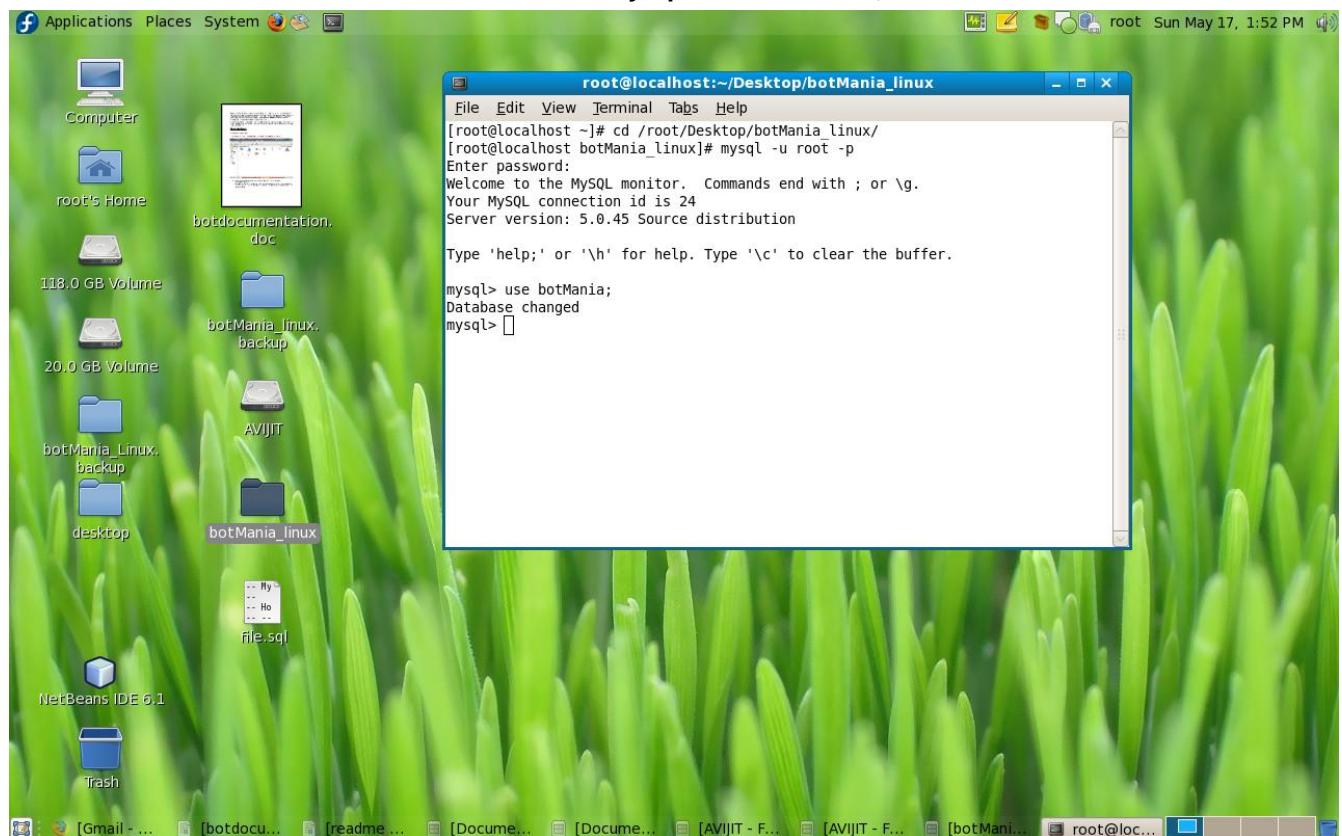
- Connect to mysql using the following command  
**# mysql -u root -p**  
**Enter password:** YourRootPasswordForMySql



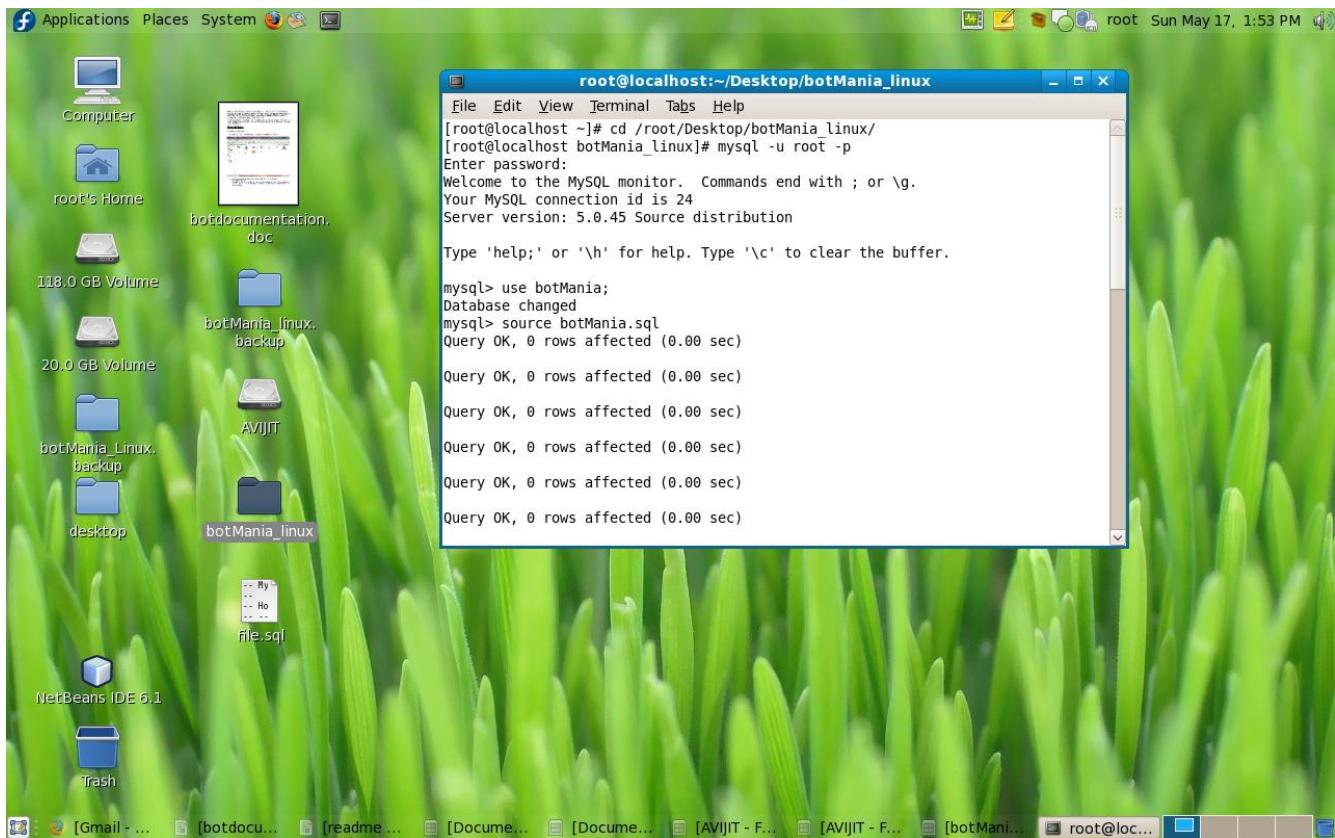
- Once you are logged in, create a database named '**botMania**'  
mysql> create database botMania;  
Query OK, 1 row affected (0.00 sec)



- Select the database botMania. For that do **mysql> use botMania;**



- Now execute the sql file. That is done simply by **mysql> source botMania.sql**

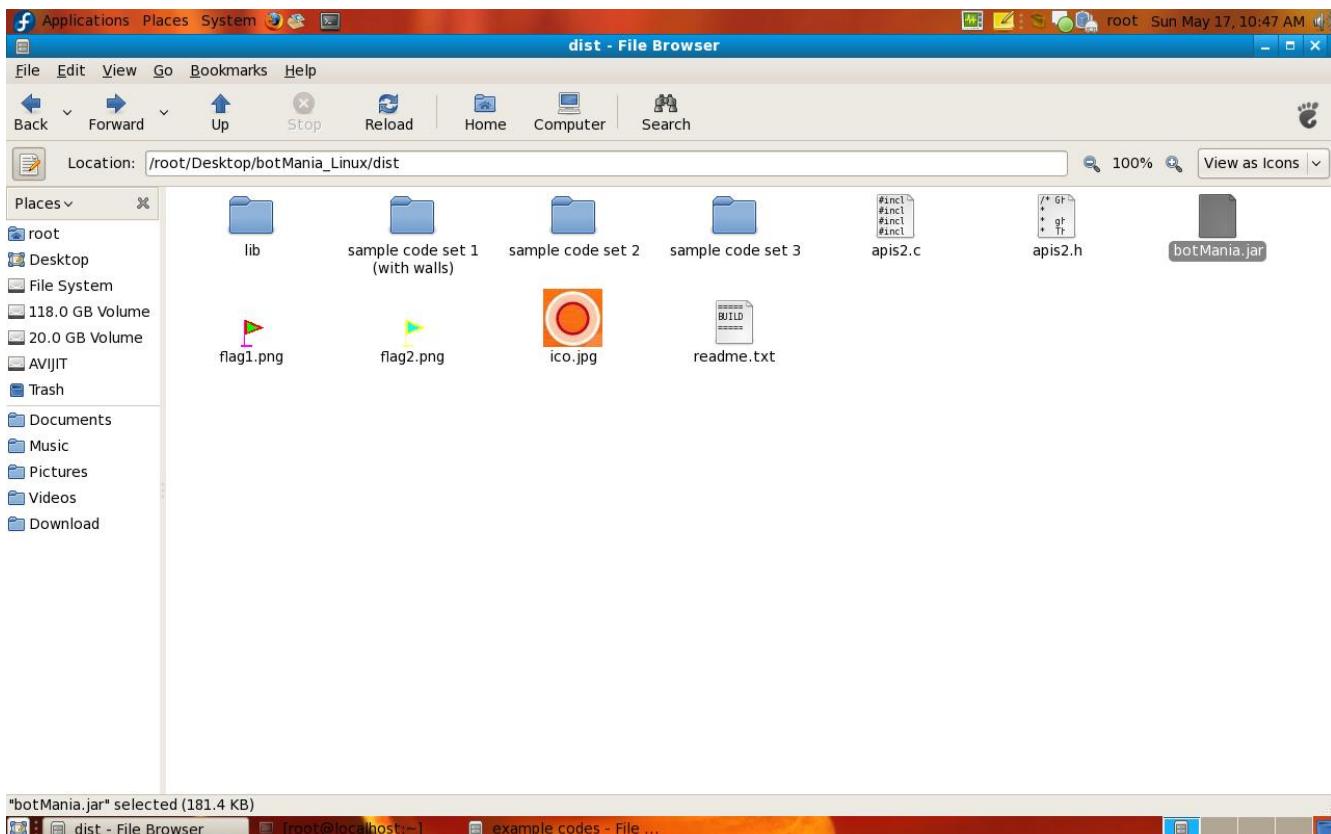


- You can check if the table is successfully created by **mysql>show tables;**
- If a table names '**scores**' is present, then the creation was successful else there may be some problems.
- Quit mysql by typing quit. **mysql>quit** or **Ctrl+d**

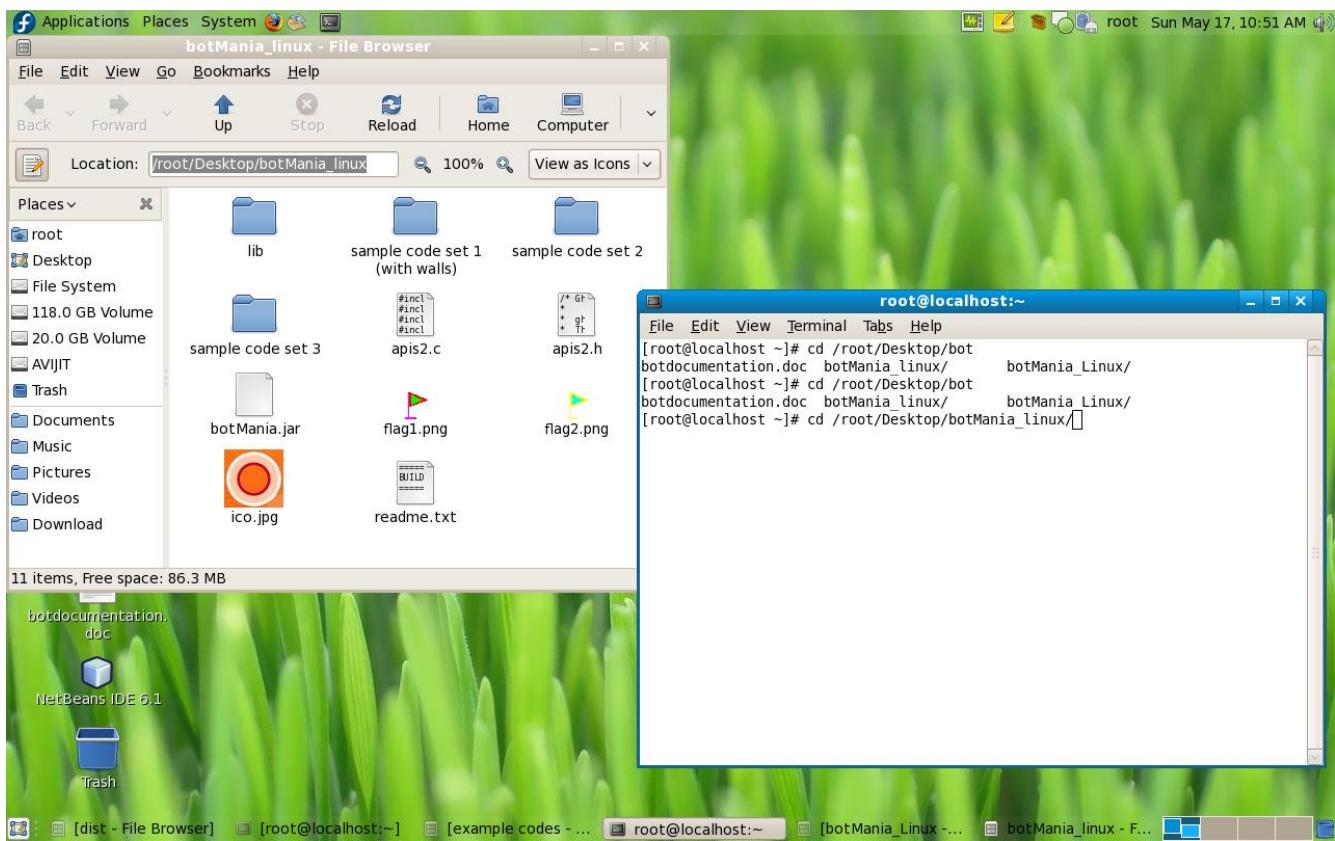
## Starting the Game :-

The game can be started by

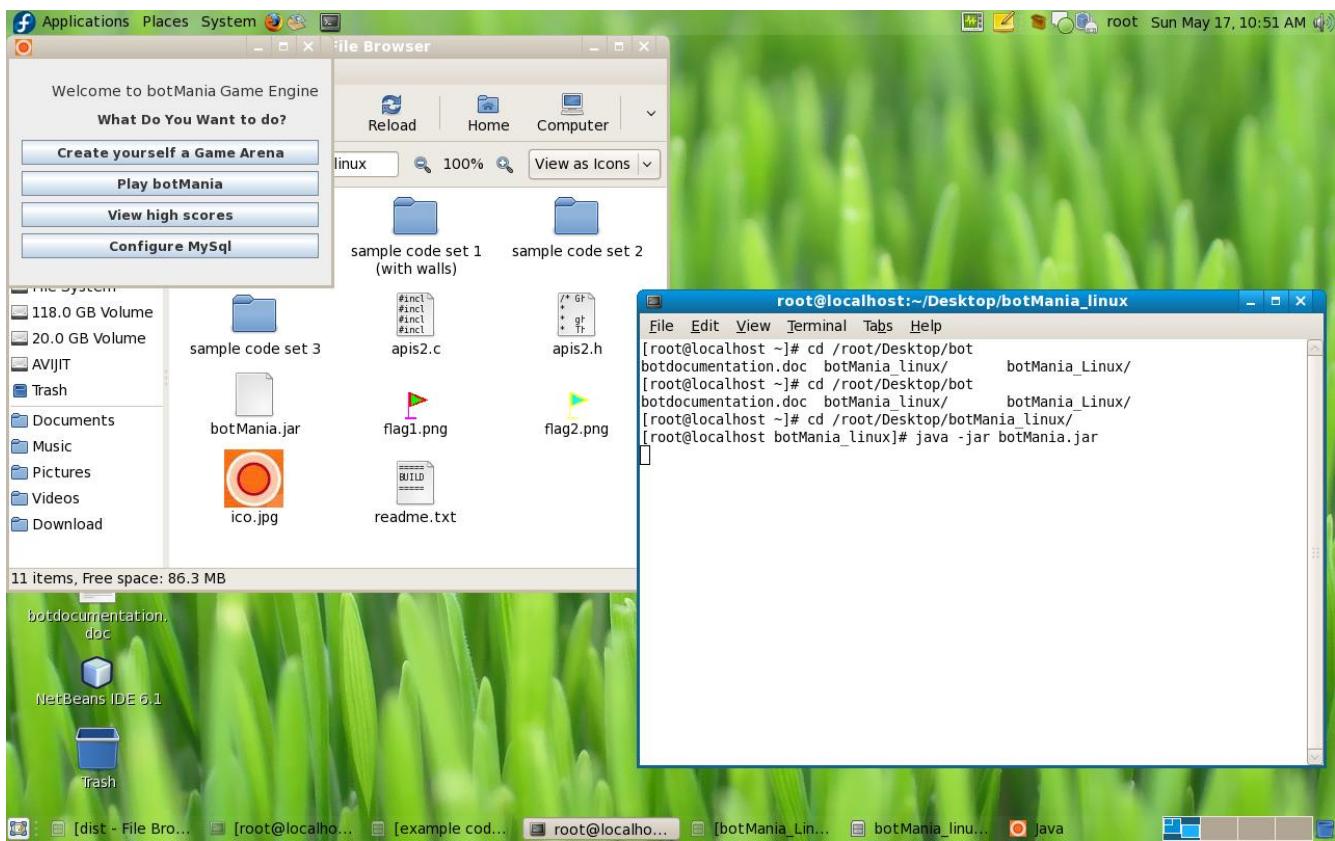
- double clicking on the botMania.jar (in windows, not suggested in linux) file



- By starting a command prompt or terminal (linux) and writing the command  
**java -jar botMania.jar**  
and pressing the enter key. Before you use the command be sure that you have navigated to the correct directory. You can use **cd path** command in windows and linux (as illustrated in the next pic) to do so.



and finally run # **java -jar botMania.jar** . A small window with some options will appear before you.

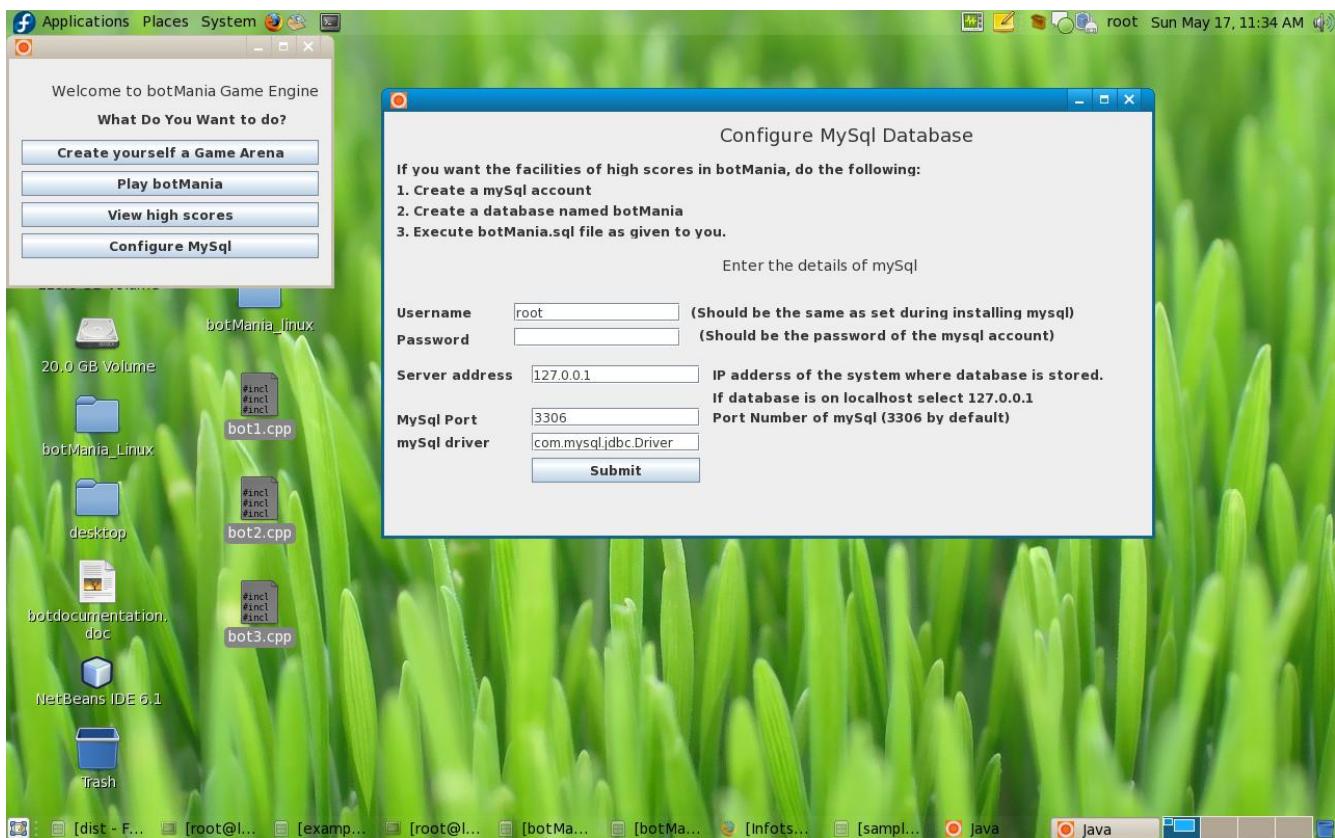


upon clicking the enter key one window opens which has two options :

1. Create yourself a Game Arena.
2. Play botMania.
3. View High Scores
4. Configure MySql

### Linking botMania to MySql(Important for first time configuration) :-

- The configuration is generally done only during the first run of botMania. But in case there are problems, the game engine can be reconfigured.
- The configuration must be done in case a connection to the database is needed. The connection to the database is needed for displaying user high scores
- If you need to configure, simply click on **Configure MySql** in the main menu
- The following window will be displayed.



**Username:** Enter the username of mysql user who can access the database botMania that was initially created. Preferably type in root.

**Password:** Enter the password of mysql user who can access the database botMania that was initially created. Preferably type in the root password.

**Server address:** Type in the address where the MySQL Database is located. On stand alone applications, choose 127.0.0.1

**MySQL Port:** Fill in the port number in which the MySQL service is running (mysqld in linux). It is generally 3306.

**MySQLDriver:** Fill in the driver that is to be used by java engine as MySql Driver.

Once you have completed the configuration, **click on Submit**. The information if filled correctly is stored else an error message reflecting the problem you might be facing is displayed.

### Create yourself a Game Arena :-----

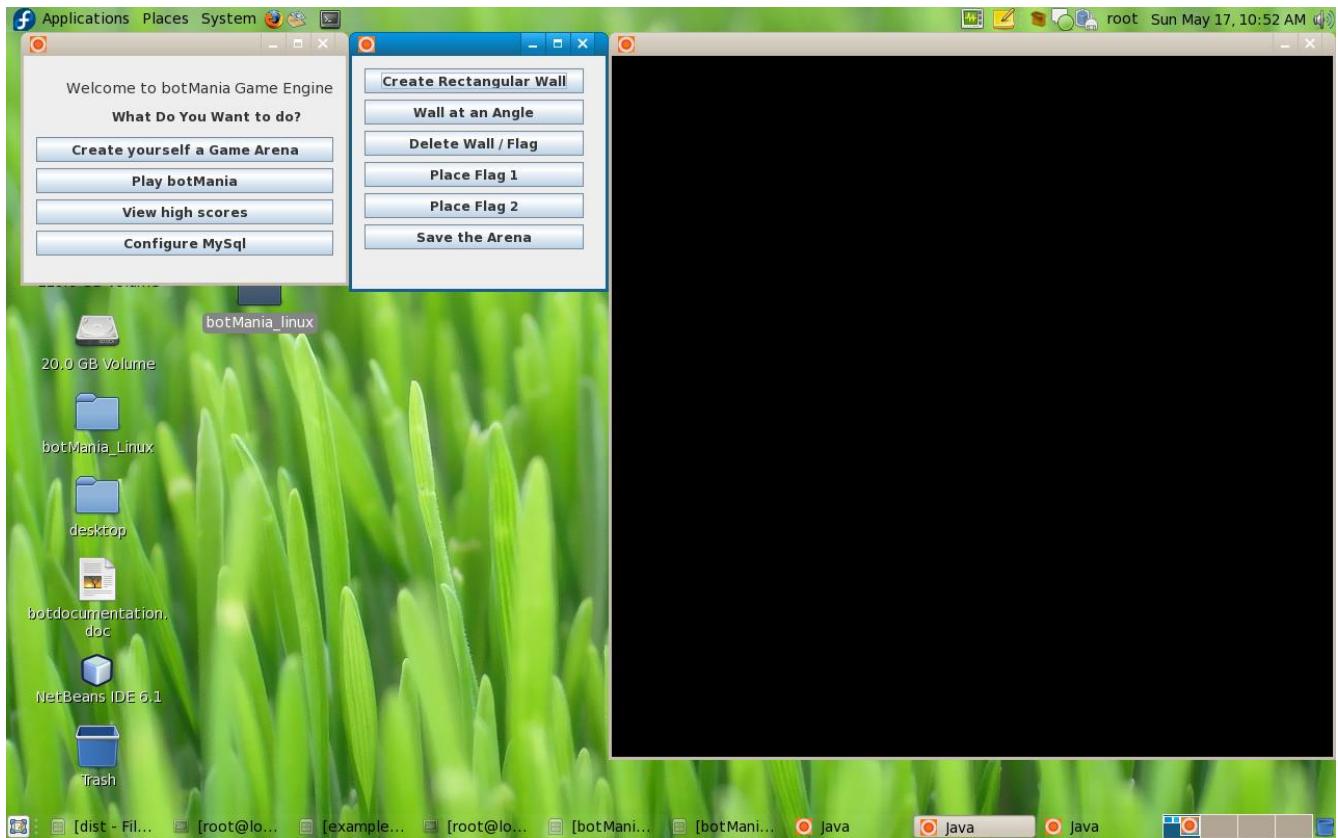
Why is a Game arena required?

A game arena essentially provides unknown and undiscovered challenges to the users. Following the philosophy of knowledge sharing, people across the globe can make tougher and more logical arenas just like maps of Age of empires. These arena help the contestants to improve upon their algorithmic skills by practicing and competing against the best.

For the purpose of providing a game arena, we provide a botMania arena builder. To enter the arena builder,

simply click on “Create yourself a game arena:” in the options that appear.

This option can be exercised if one wants to create oneself a game arena. Upon click this button two windows appear as shown below.



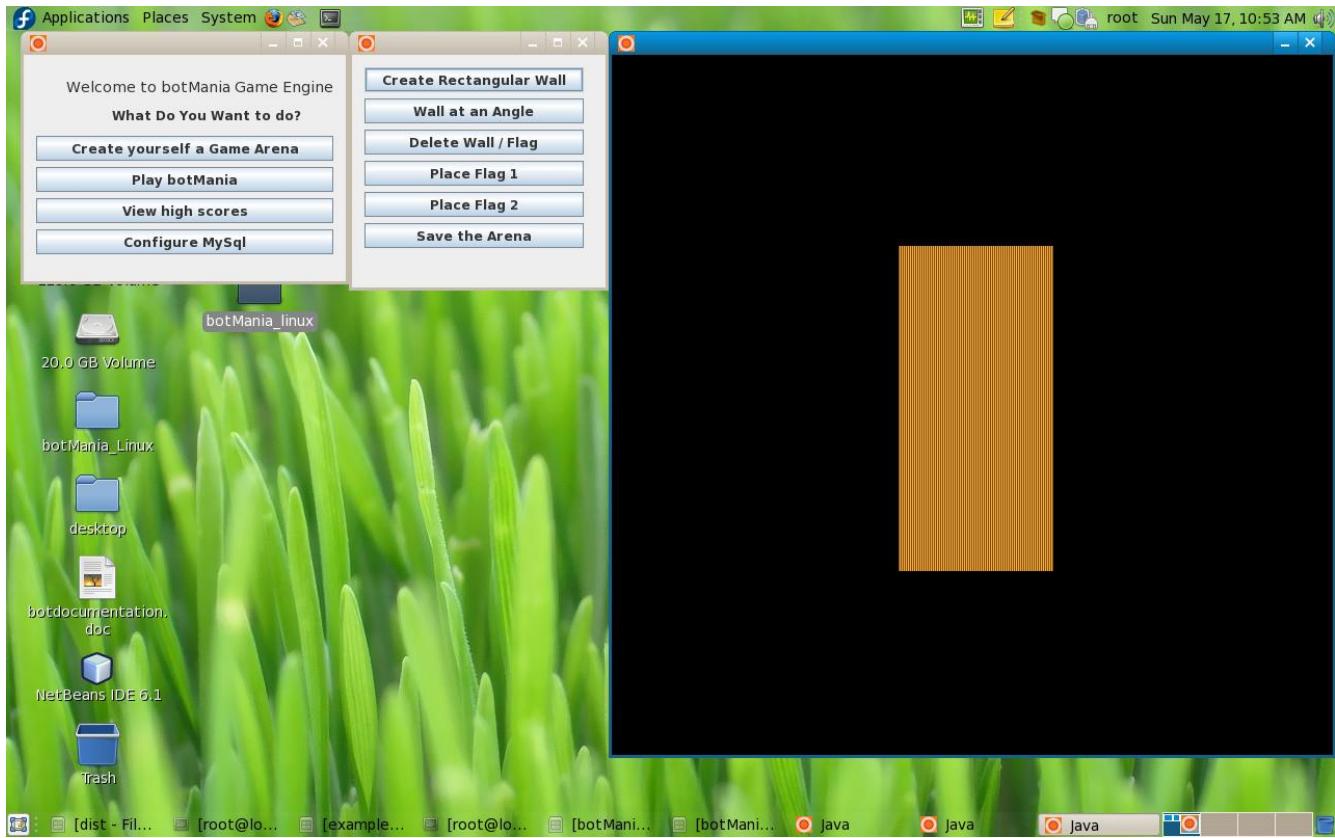
The first window has six options while the second one is a blank window which represents the blank arena. The game arena is built on the second window.

The first window has six buttons each having different functions.

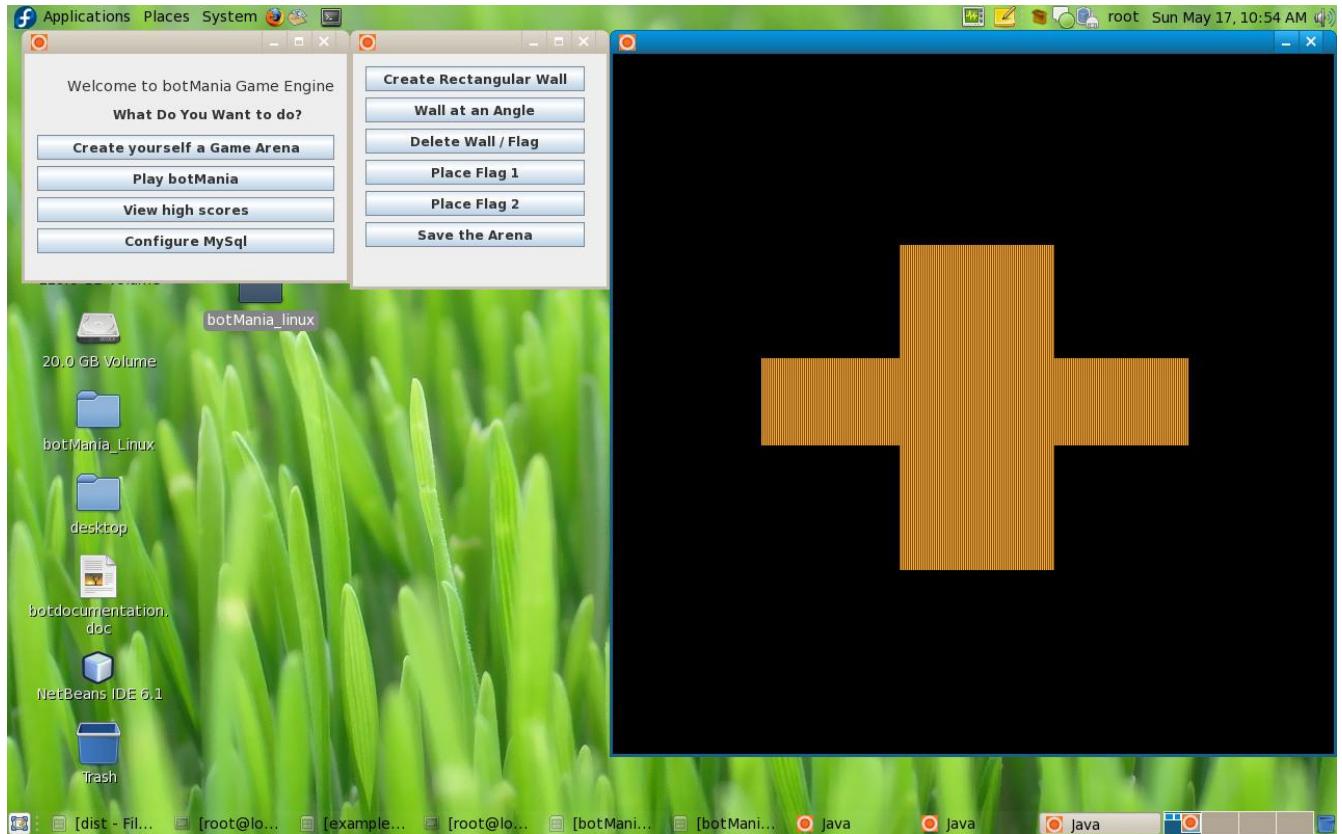
#### Create Rectangular Wall :----

This option is used for creating rectangular walls sing the botMania arena builder arena .For using this option just click on the '**create rectangular wall**' option in the first window. Hereafter you can go on drawing walls on the game window. Simply press and release the mouse at some points that lie within an arena and a solid wall would be build in the area.

Suppose the mouse was pressed at  $(x_1, y_1)$  and released at  $(x_2, y_2)$  then the rectangular wall is built with  **$x_1, y_1$  and  $x_2, y_2$  as its diagonal points**. As illustrated in the figure below,



and further with another rectangular wall, we have the game arena which looks like:

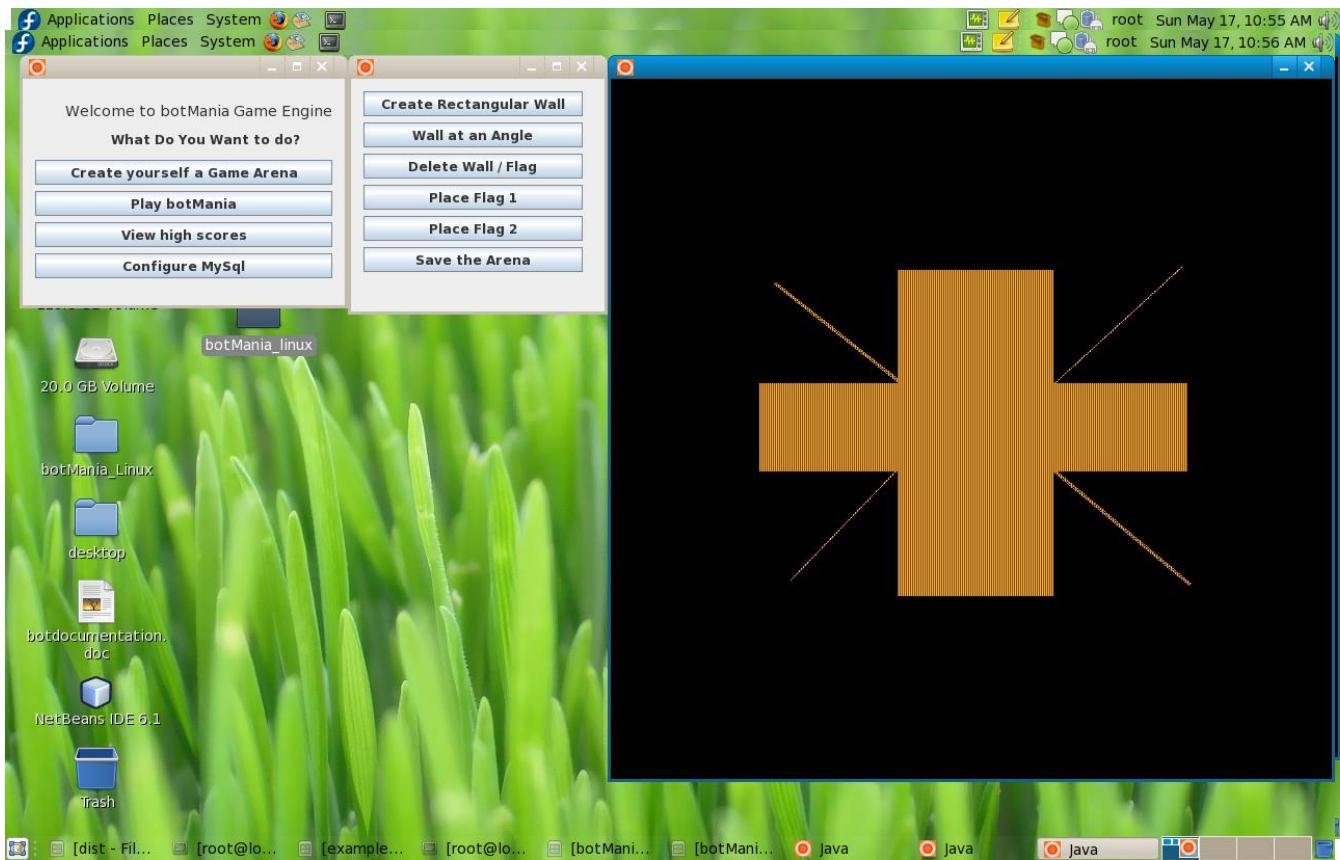


**Wall at an Angle :---** This option can be exercised to make walls inclined at some angle. For using this option just click on the wall at an angle option in the first window. After then you can go on making walls at an angle on the game window.

Simply **press and release the mouse** at some points that lie within an arena and a wall would be built along the line joining the points.

Suppose the mouse was pressed at  $(x_1, y_1)$  and released at  $(x_2, y_2)$  then the rectangular wall is built with  $x_1, y_1$  and  $x_2, y_2$  as its diagonal points. As illustrated in the figure below,

Sample Images of rectangular walls are given below.



And finally repeating it 4 times, we have the arena looking like:

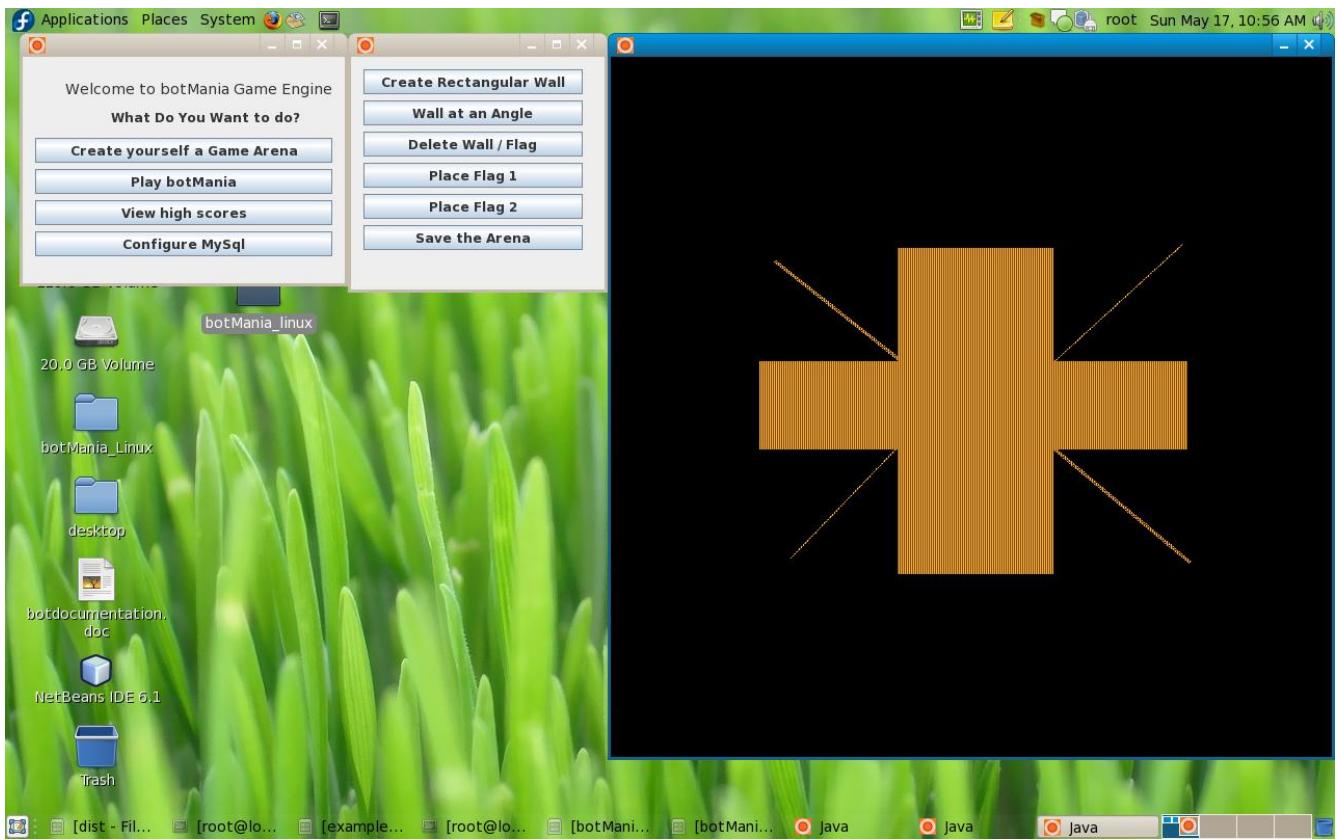
#### **Delete Wall/flag :----**

The next option in the fray is 'Delete Wall/flag' . This option can be used in order to delete walls and/or flags made in the game window. In order to use this option just click on this option and then go on erasing the flag and/or wall you want to remove from the game window. All you have to do is press and release the mouse at appropriate distances.

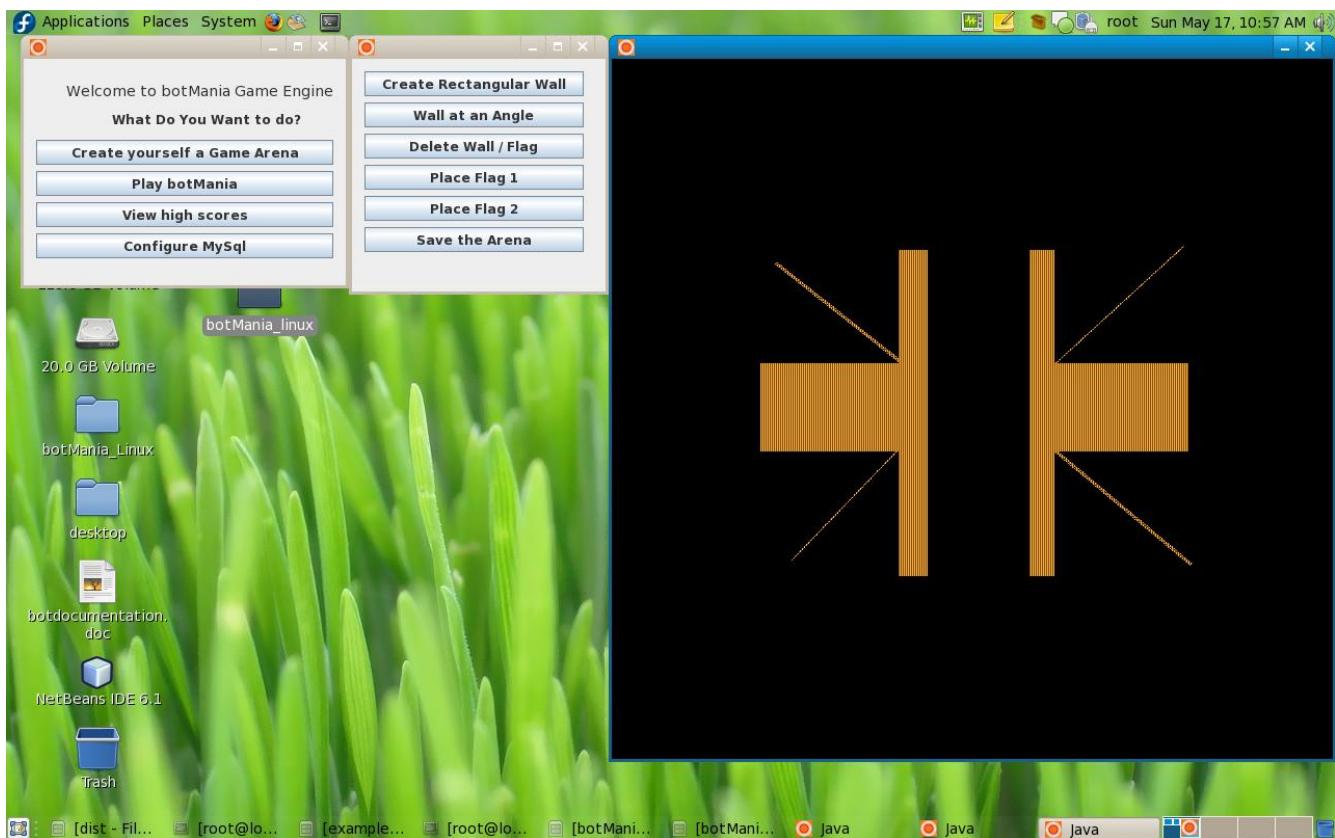
**Note that if mouse is pressed at (x1,y1) and released at (x2,y2) everything within the rectangle with (x1,y1) to (x2,y2) as their diagonal points is deleted.**

In case a flag needs to be deleted, please remove it completely from the game arena by selecting appropriate distances.

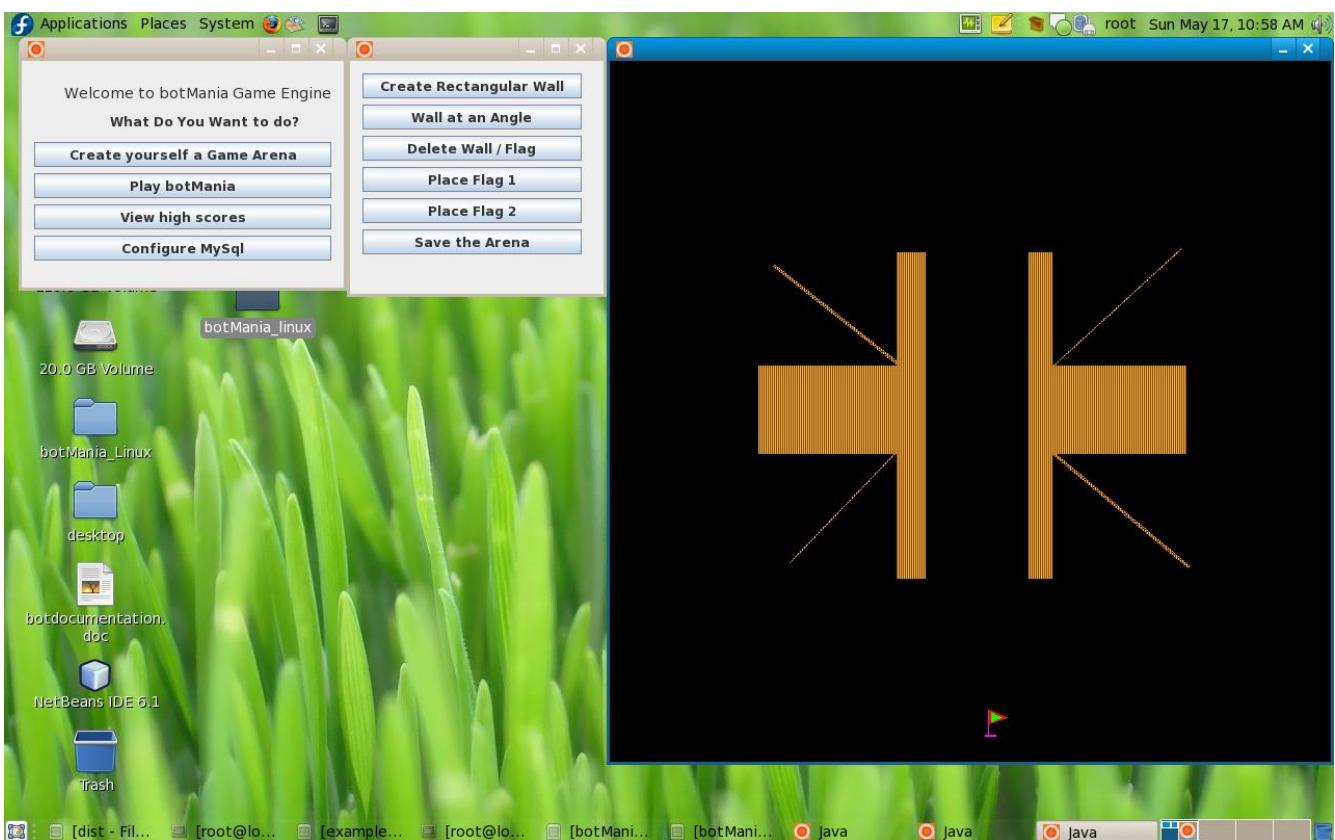
Before deleting the Wall/flag the arena looks like as below :-----



After deleting the Wall/flag the arena looks like as below :-----

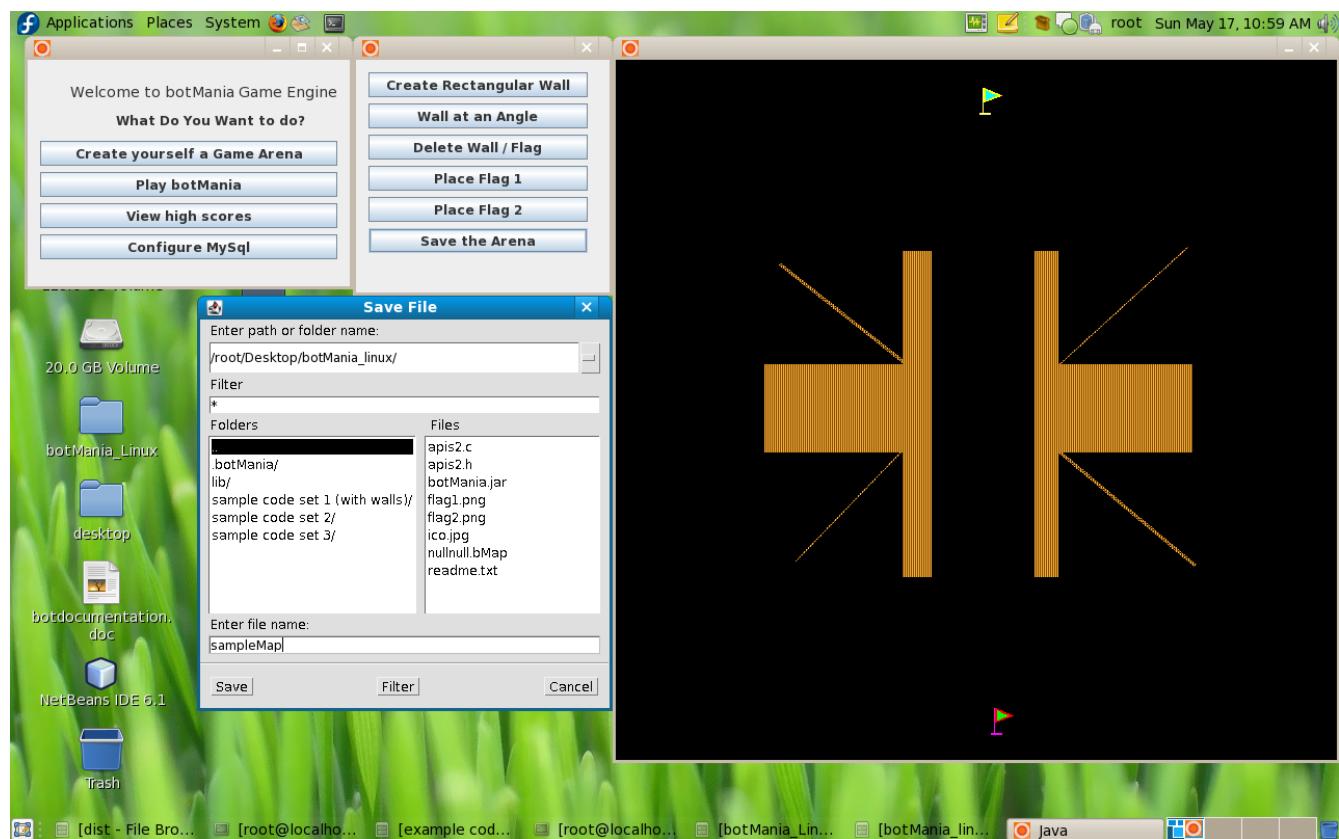


**Place Flag 1** ---- This option can be in order to place flag 1 on in the game arena. Flag 1 is owned by the team 1. This flag can be put anywhere in the game arena. Simply click the mouse at that point in the arena.



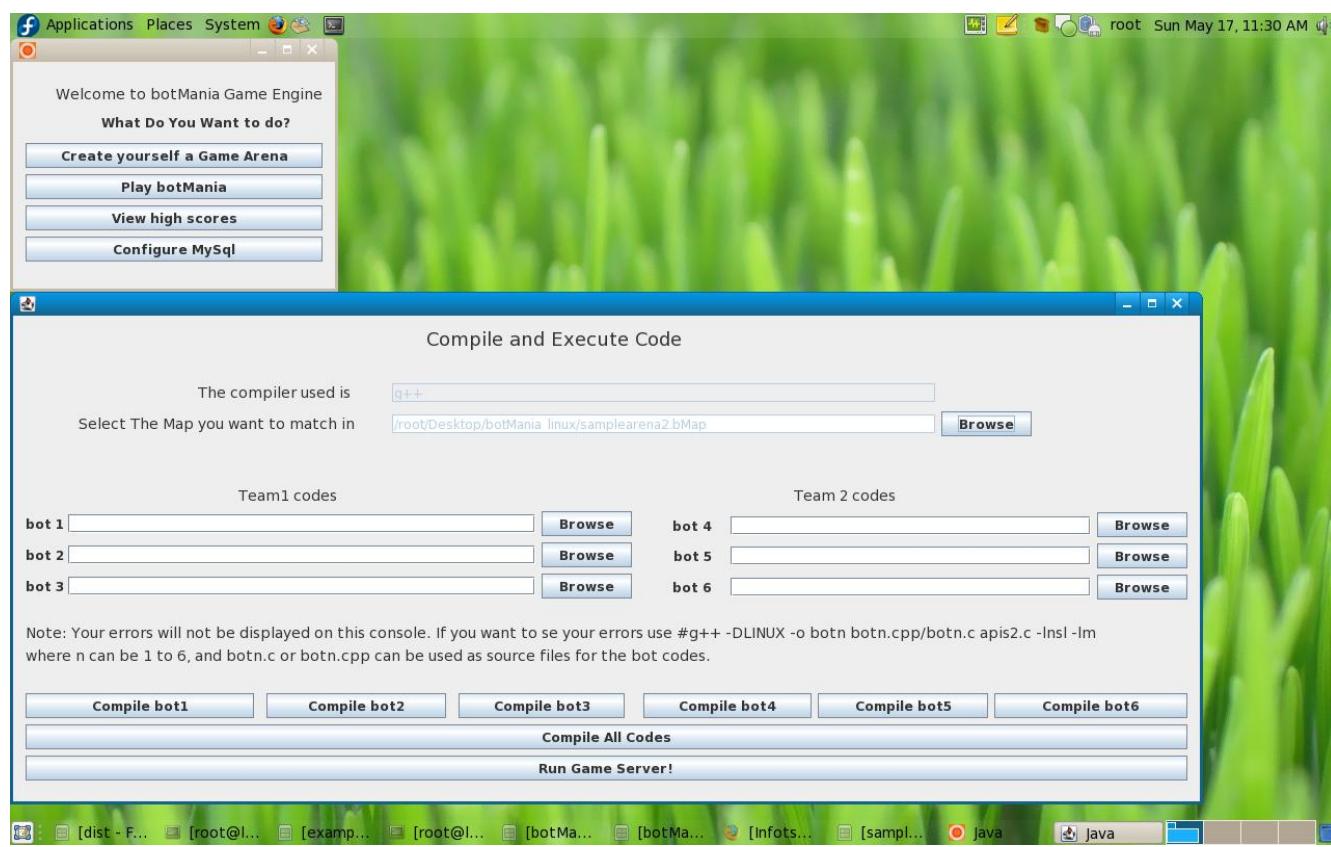
**Place Flag 2** ---- Like the previous option this option can be exercised in order to put flag 2 i.e. The flag owned by the team 2 in the game arena. This is similar to placing of flag 1 described above.

**Save The Arena** ----- The last option in the fray of the options is the 'save the arena' option. This option can be used to save the final setup of the game which you want to use in playing the game. For exercising this option just click on it and a new window will appear. Just type the name which you want to give to the arena. No extension needs to be given. Suppose you want to save the arena with the desired sample map . You just have to type the initially entered file name in the 'Enter file name' box and the arena will be saved with an extension ".bMap"



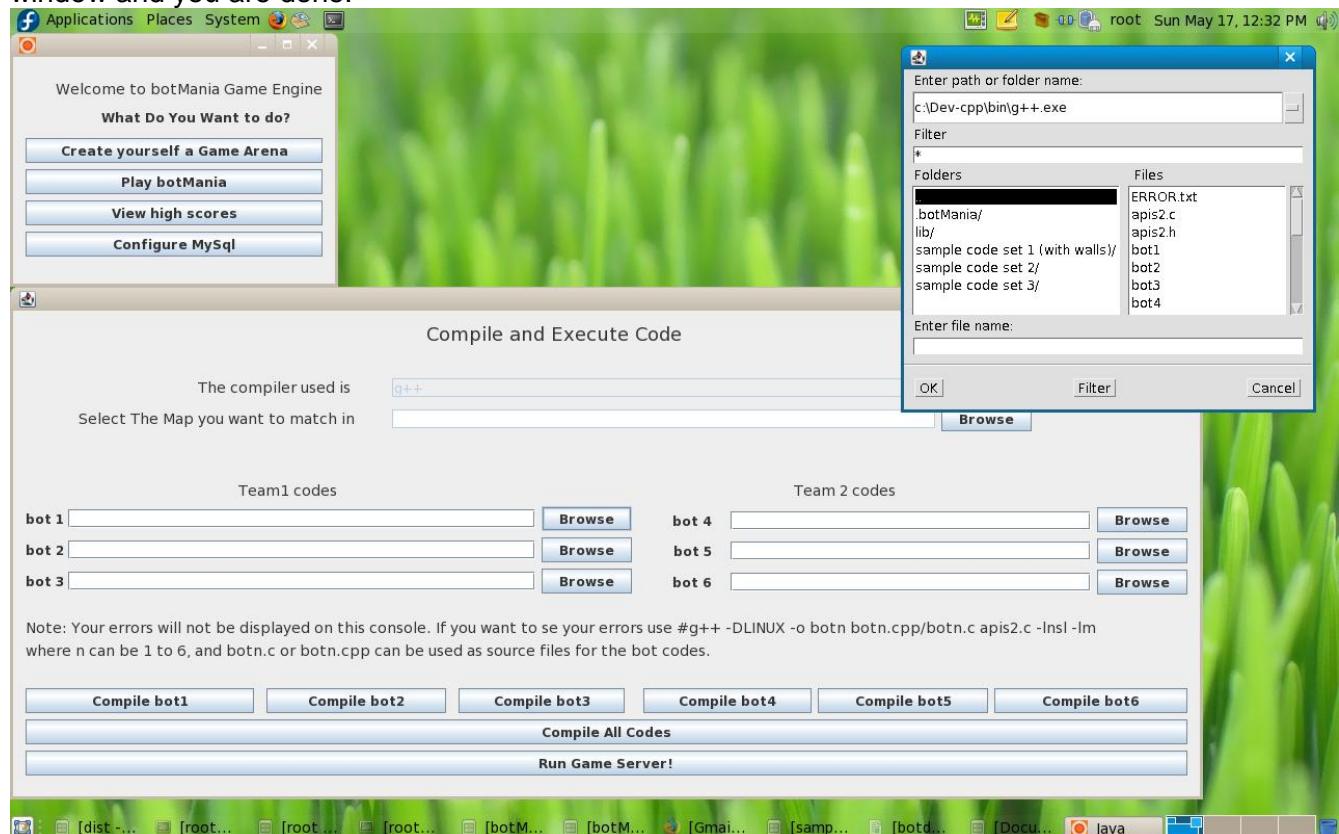
Here you can also browse the folder where you want to save the game. Then after just click on the save button on lower leftmost corner of the window and the game will be saved.

**Play botmania :---** This option is used to start playing the game after creating the game arena. If this option is clicked without creating the game arena then the default game arena is used . Upon clicking this button a new window will appear containing several options.These options include selecting the g++.exe file(necessary only for windows) , selecting the map which we want to be used as the game arena. We can make the map ourselves as described above and select the

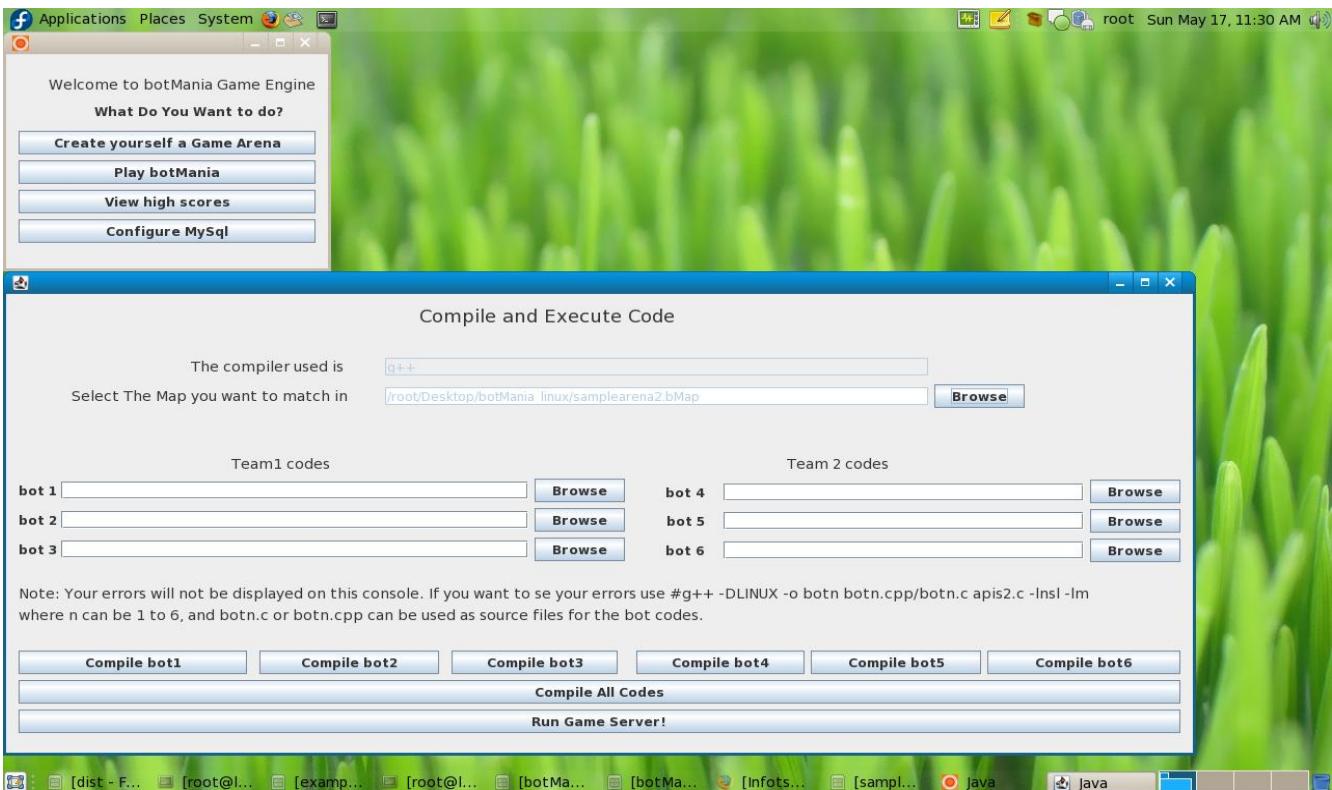


The various option listed in this window are as follows :----

**Locate the g++.exe file (in Dev-cpp\bin):--** This option should be exercised only if the game is being played in windows. There is no need to use this option in Linux. The g++.exe file is used to compile the codes written for the bots. Upon clicking this option a separate window appears ,where the desired file can be browsed. After browsing the file just click on the OK button on the lower leftmost part of the window and you are done.



**Select the map you want to match in : --** This option can be used in order to use one of the ready - made or self -made maps . If someone does not want to make his own game arena , he can use this option. Upon clicking this option as usual a separate window appears where the desired map can be browsed. At last just click on the OK button and you are done. In case no map is selected, the game will run with a blank arena with only two flags and without any walls.



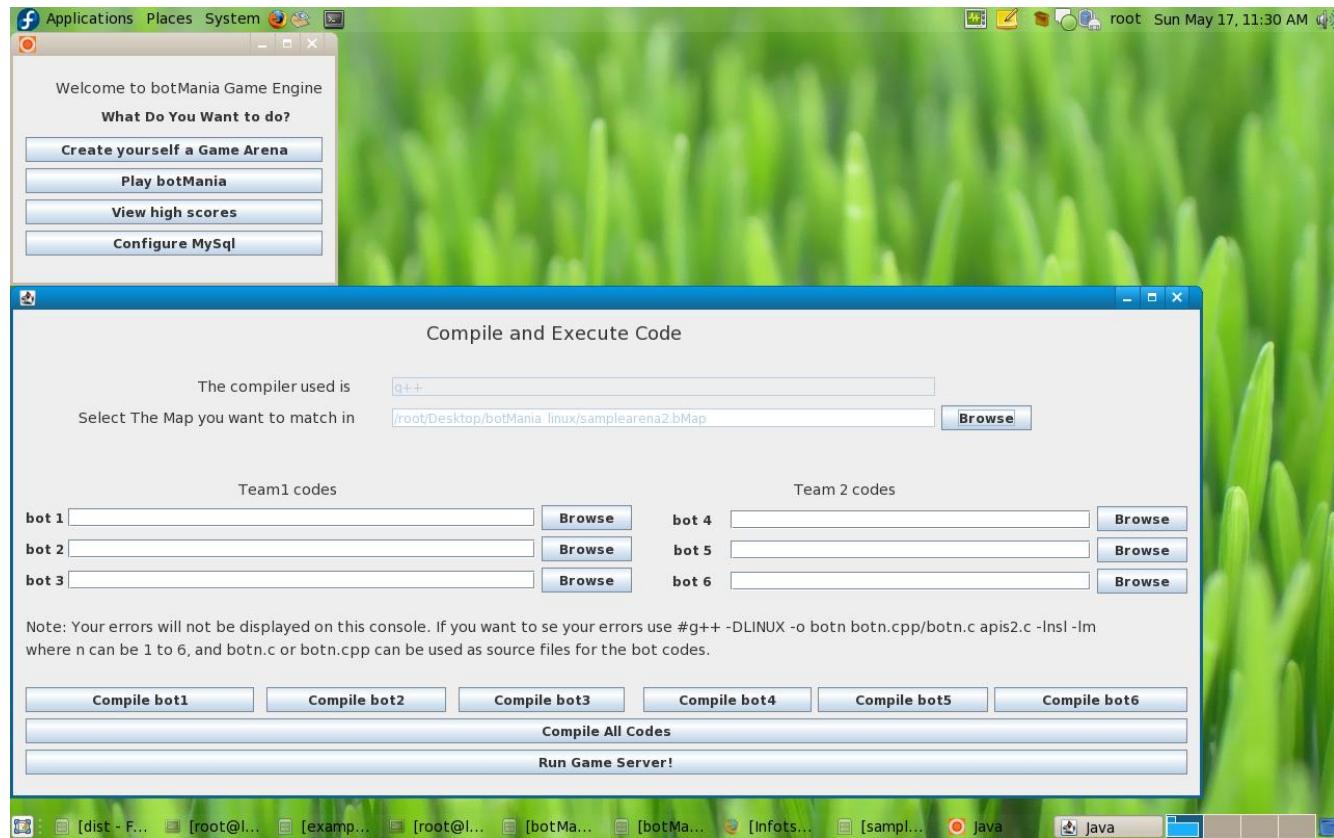
**Writing the Codes for the Bots:**---- The codes for running the bots can be written in C/C++ . These codes use some specific APIs in order to perform some specific task:---

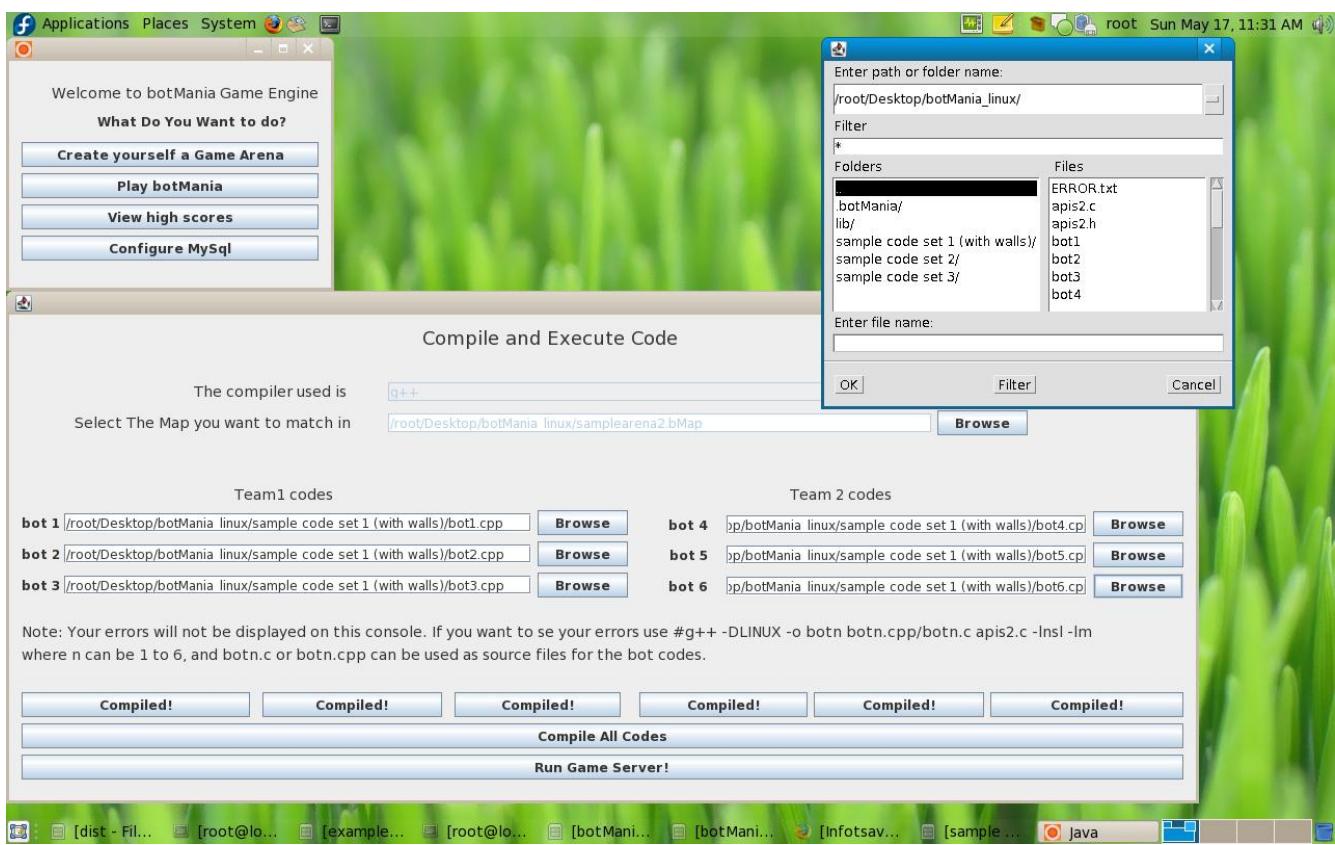
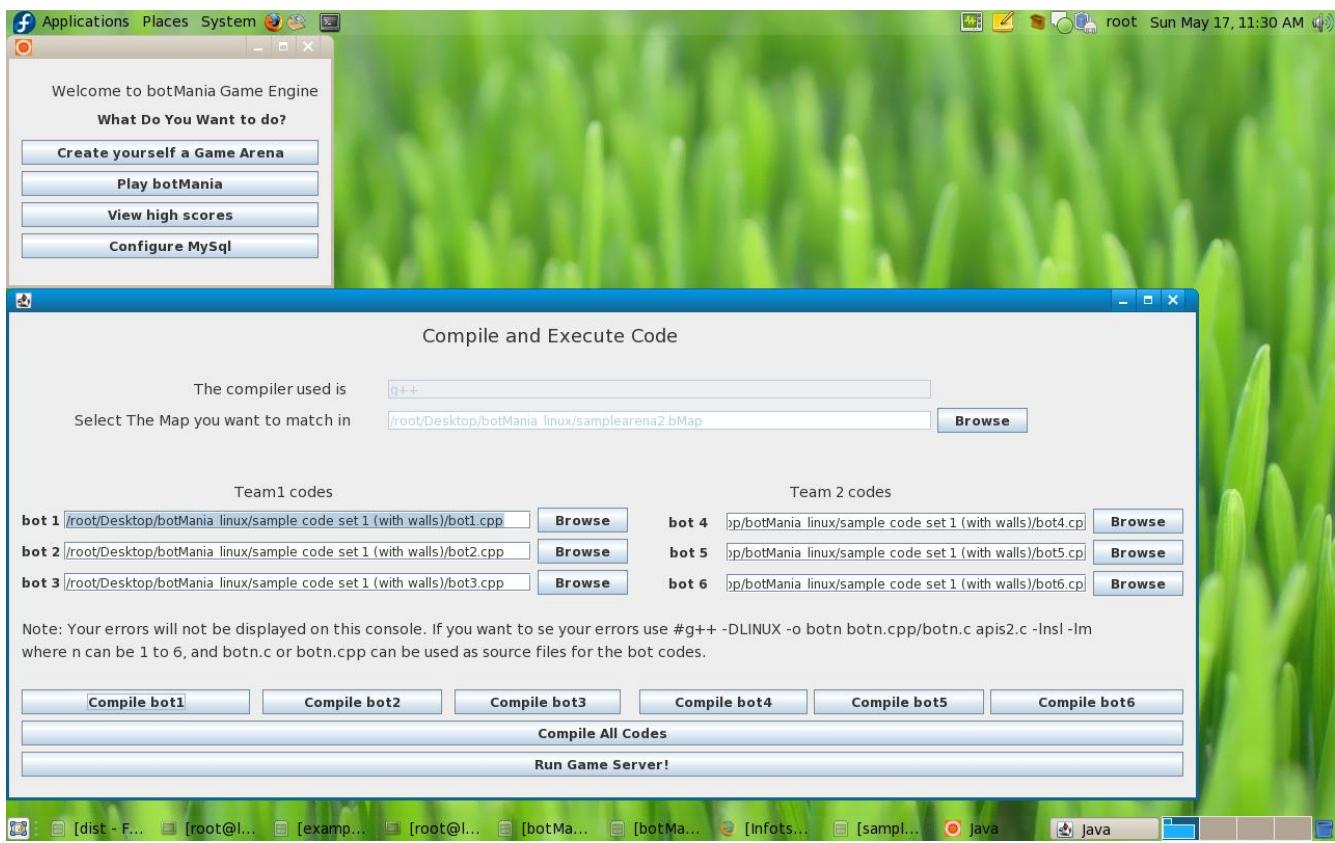
- int init(char \*name) : : This function is used for initializing the bots.
- void move(double direction, int ismoving) : This function is used for moving the bots in a certain direction.
- int statusinfo(int \*enemyx, int \*enemyy , int \*myx, int \*myy, int \*ismoving, int \*health, char \*mymessage) : : This function is used for getting various information related to the bots e.g. its coordinate ,whether it is moving or not etc.
- void fire(int range, int type) : This function fires a laser or a bomb in the direction of movement of the bot and range.
- void getmyids(int \*teamid, int \*playerid) : This function returns the teamid and playerid of the invoking bot.
- void enemyflaginfo(int \*x,int \*y,int \*istaken, int \*teamid, int \*playerid) : This function is used to get information about the enemy flag co-ordinates X and Y, and the player-id and team-id of the bot who has it.
- void myflaginfo(int \*x,int \*y,int \*istaken, int \*teamid, int \*playerid) : This function is used to get information about the own flag co-ordinates X and Y, and the player-id and team-id of the bot who has it.
- void flagdc(int \*x,int \*y) : When the flag has been collected, it has to be deposited at a particular coordinate reflected by the x and y of the above API.

- int msgsend(int playerid ,char message[]) : This function sends a **message** (a string) to the bot with the particular player index.
- void scanwall(int \*dist) : This function is used to scan the nearest wall in the direction the player is moving.

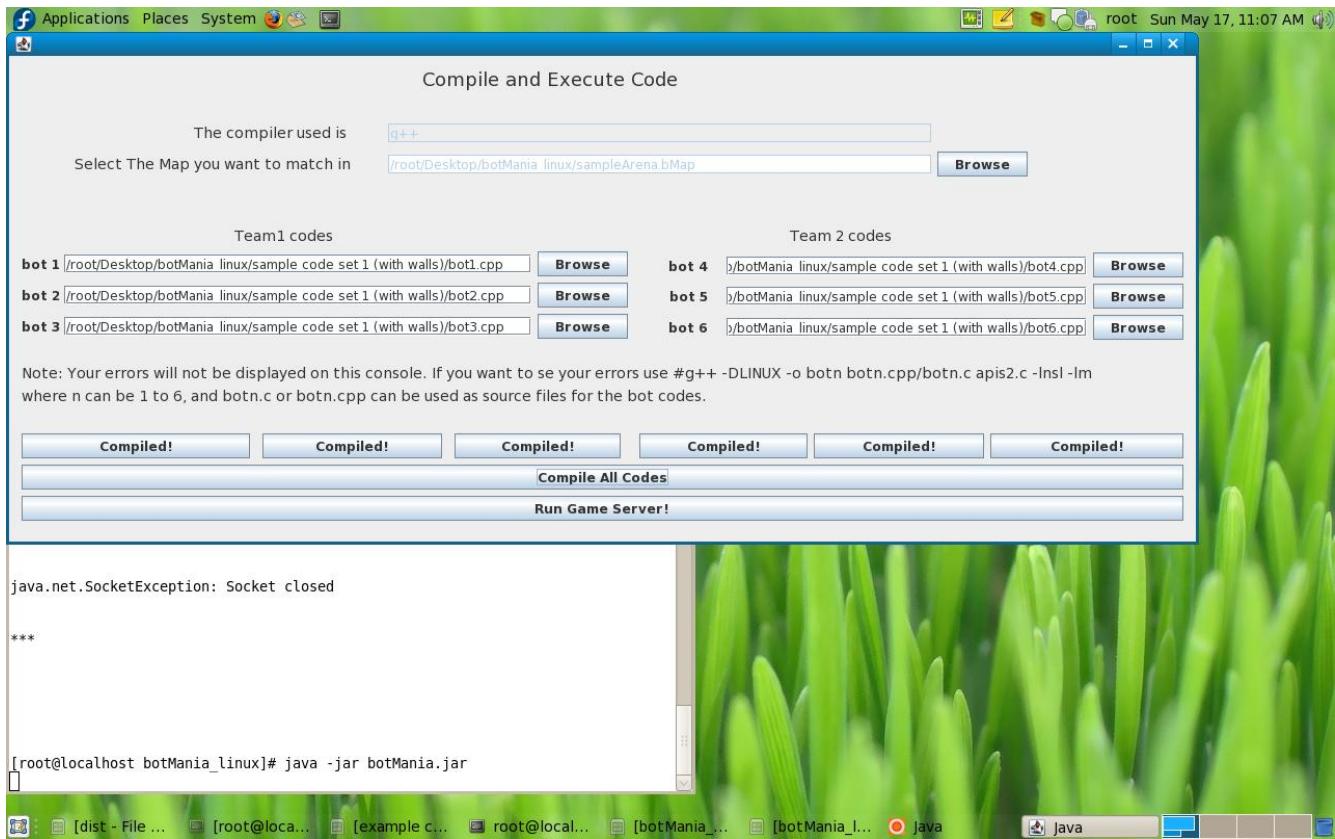
For further information on making the code please refer to the readme.doc file. You can also see the codes of the sample bots in order to get a broader view on making the code.

**Browsing the Codes Written for the Bots** :--- Just click on the browse button against the name of the bot for which you want to browse the code .Doing this will get you a separate window where you can browse the file After browsing just click on the OK button on the lower leftmost part of the browsing window. Do the same for all other bots.





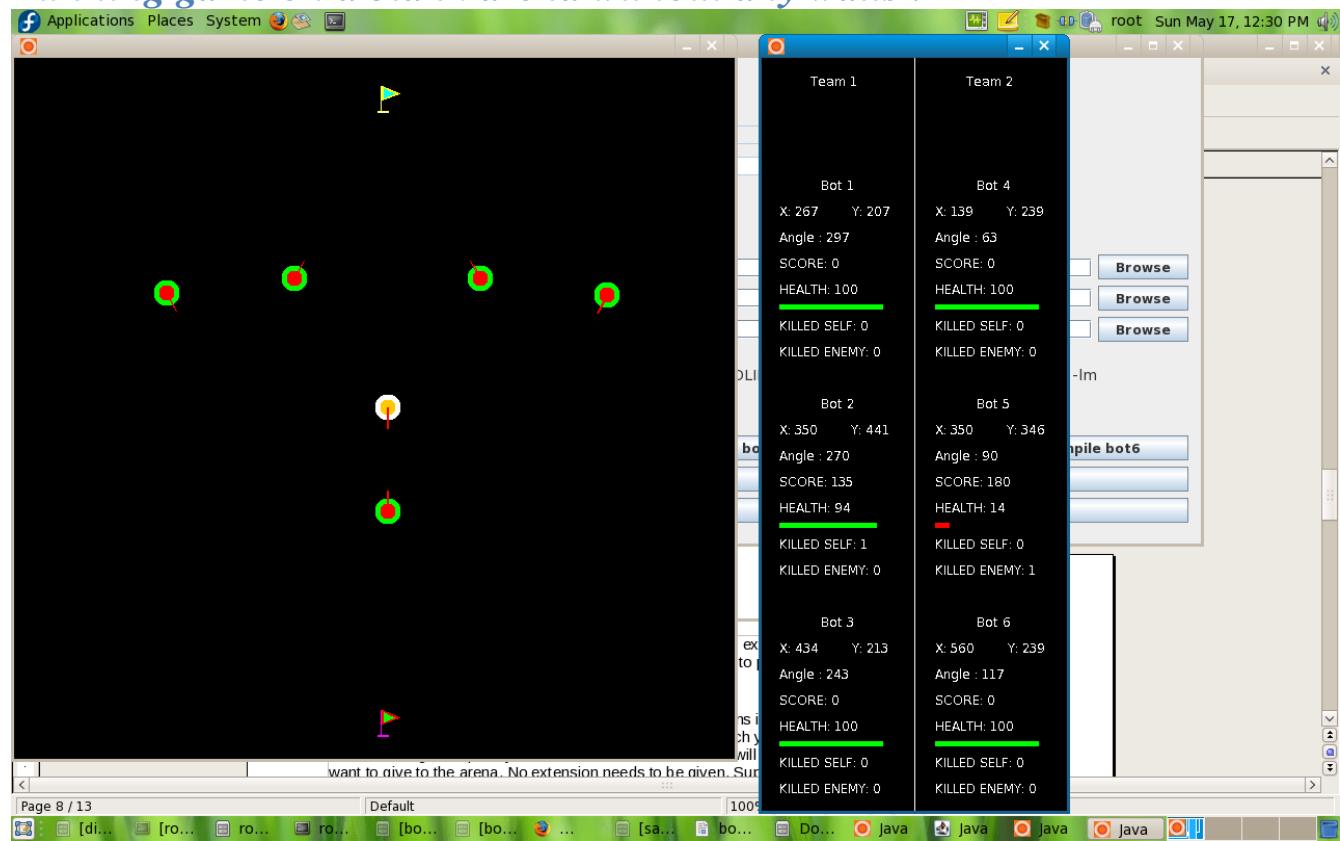
**Compiling the Codes:** ---After browsing the codes, the next step is to compile the codes. There are two ways to compile the codes . Either the codes can be compiled one by one by click on the compile bot-name option or by clicking on the 'compile all codes' option . Exercising this option will compile all codes in one go.



**Run Game Server!** :---- This option is to be exercised for running the game server. Upon clicking this option two separate windows will appear .One of the windows will be the game window itself wherein the bots will be fighting with the opponent bots . The other window will be the status window. This window will contain several information such as health of the individual bots, running individual scores of the bots. number of killed enemy by each bot,the inclination angle of each bot , the X and Y coordinate of each bot , number of times the bot itself has been killed etc.

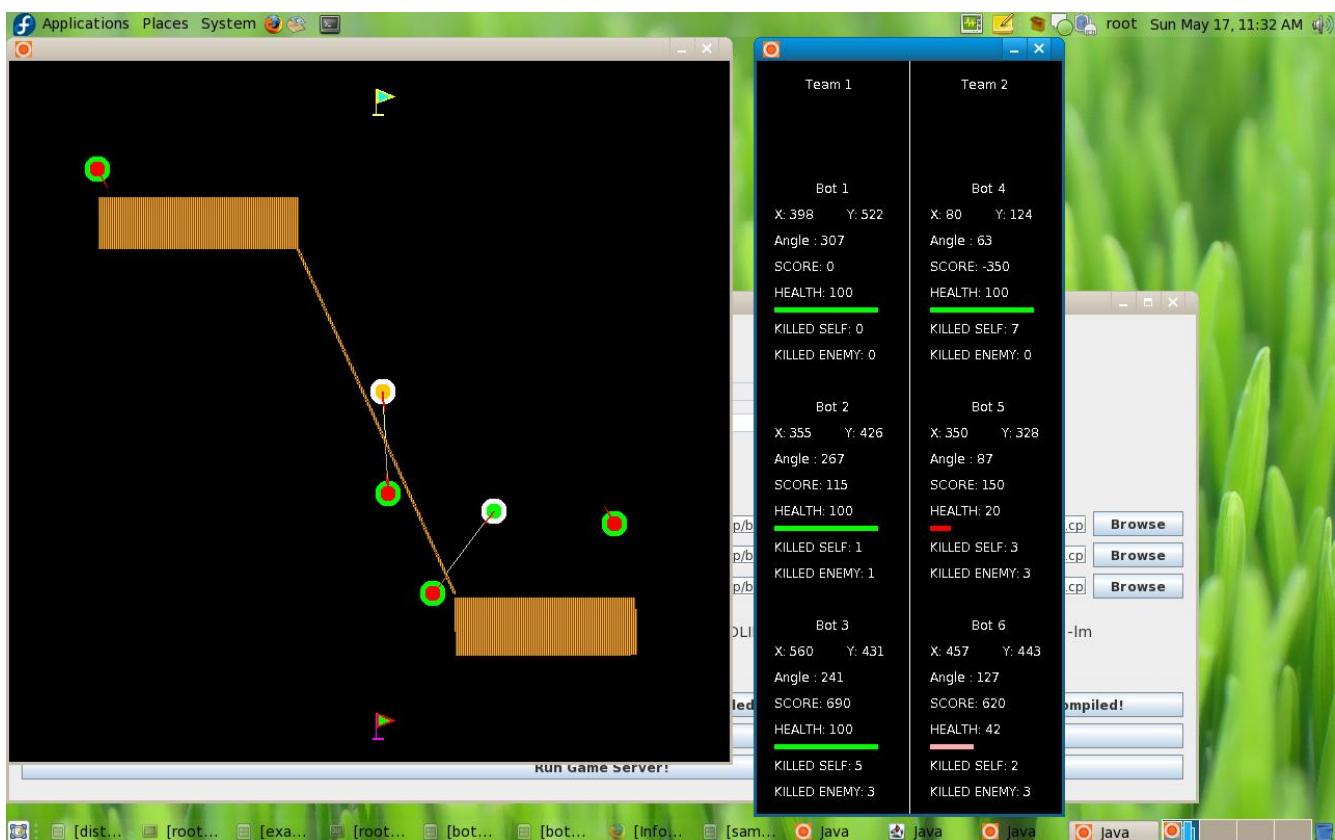
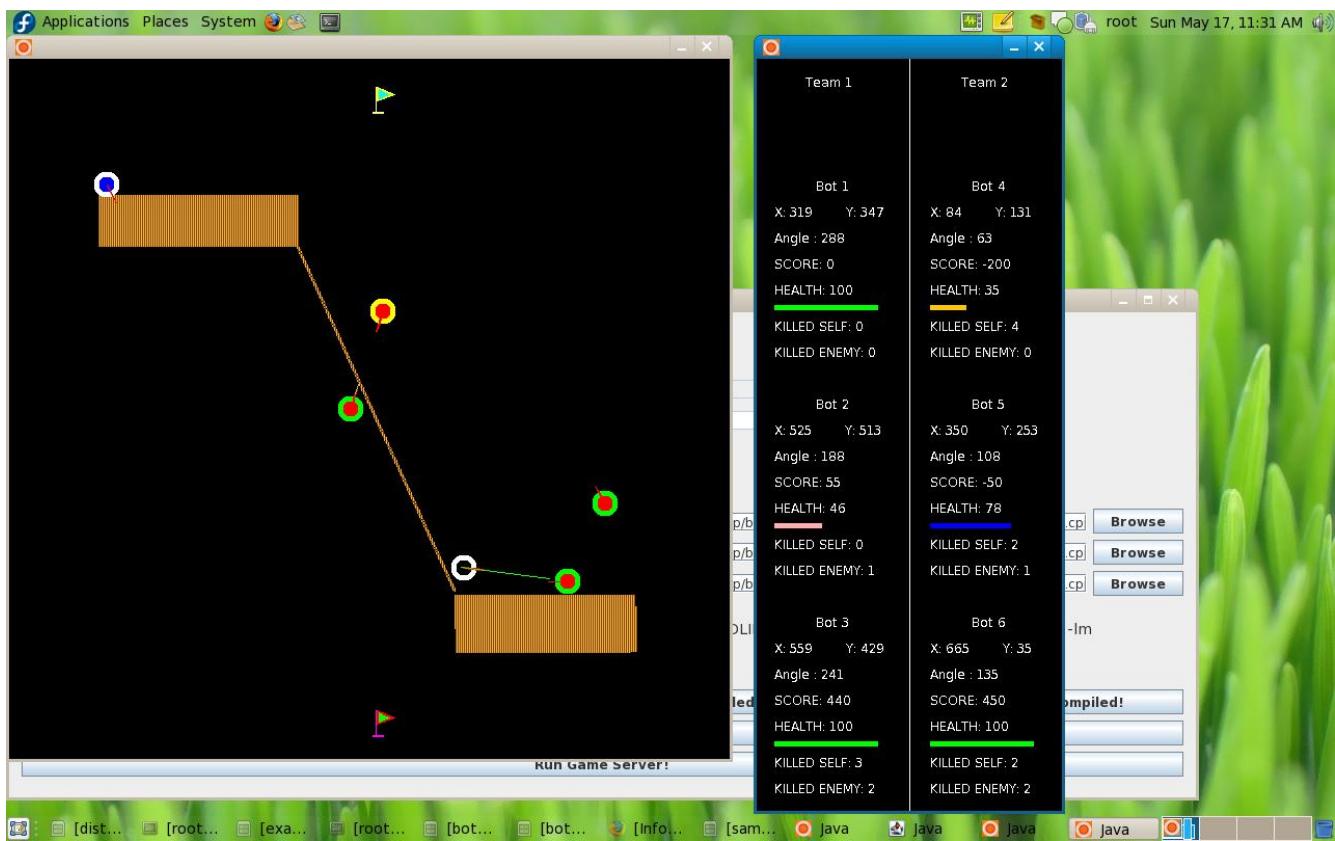
## Run Demos

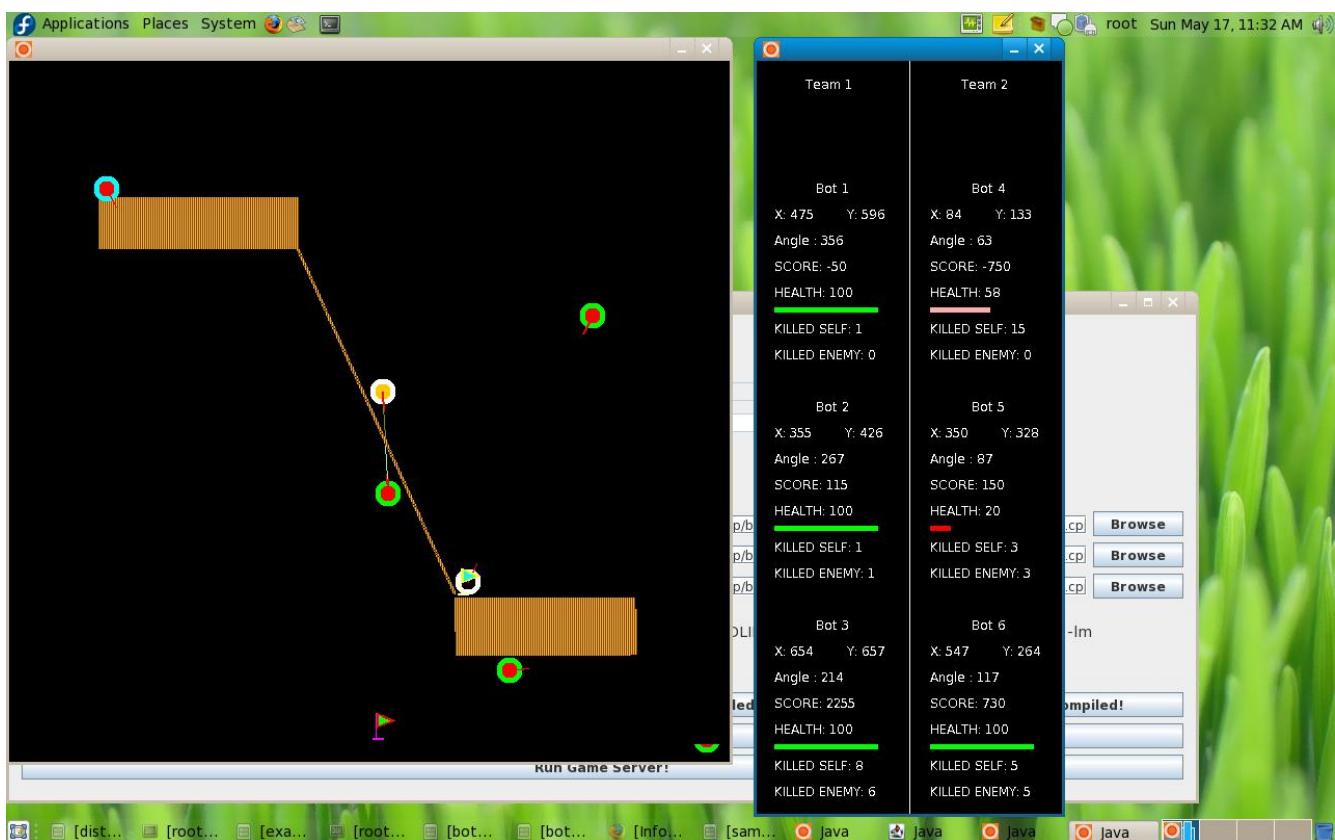
*Running game on a blank arena without any walls :----*





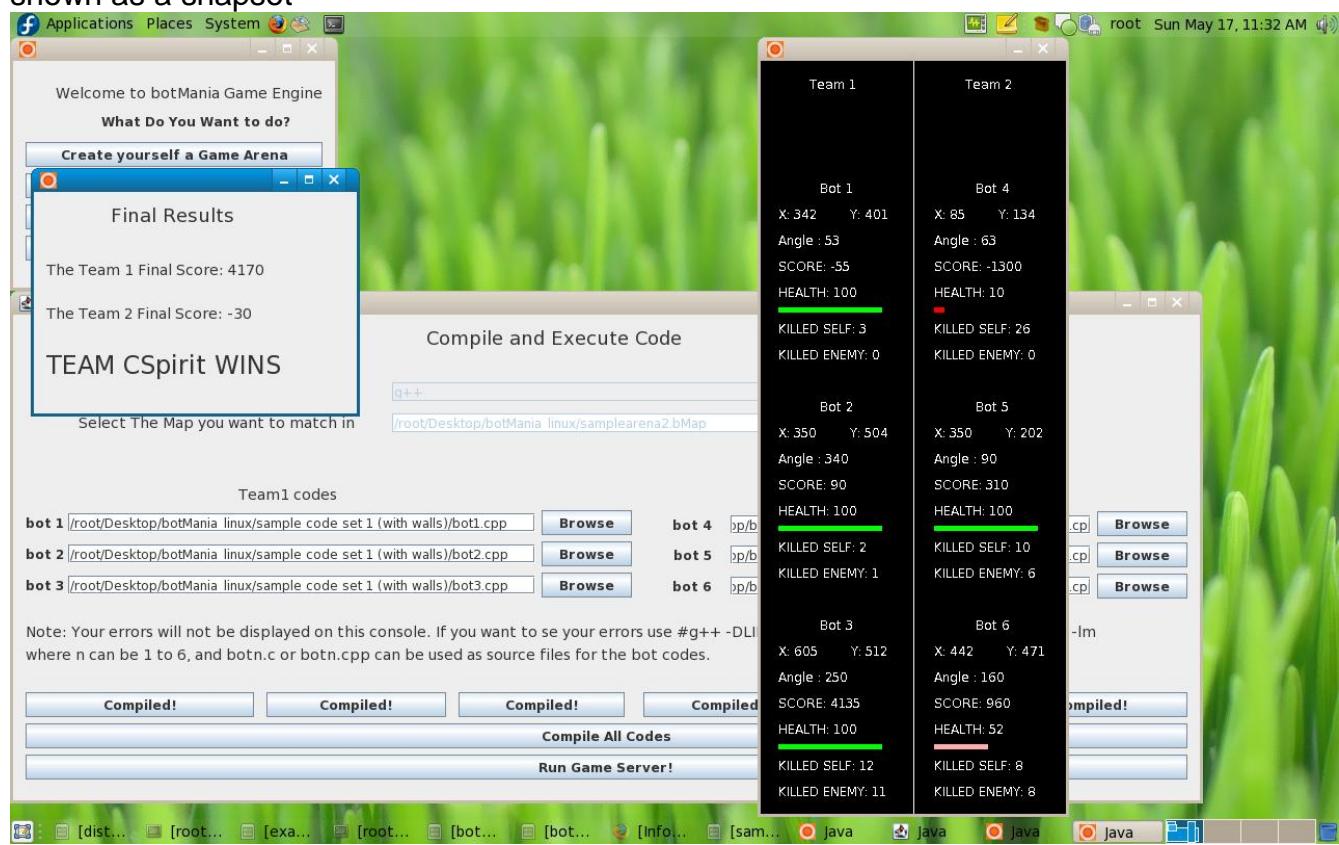
*Running game on a selected map (sampleArena2.bMap):----*





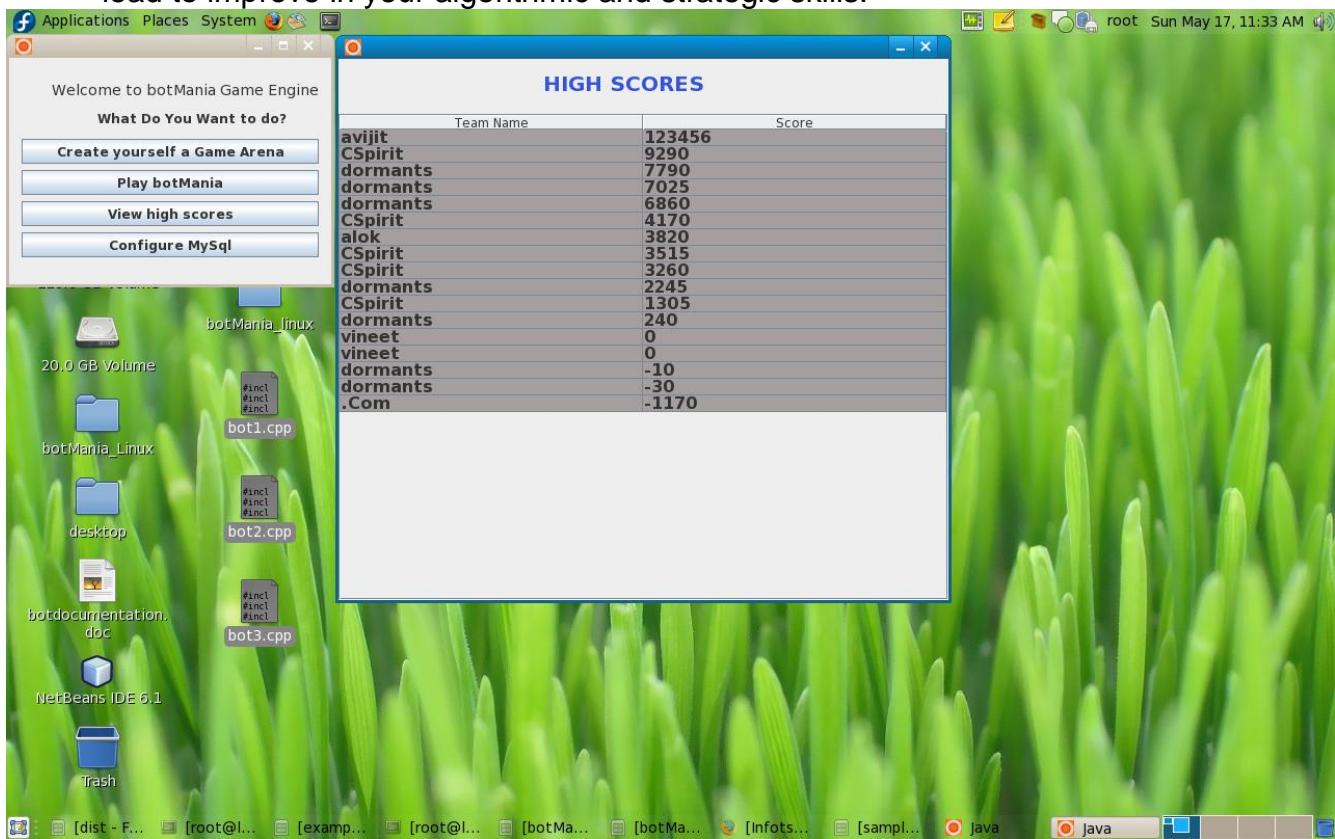
## Final Result of the Game: ----

Upon completion of the game a separate window appears containing the final scores obtained by each team and the final result of the game based on the scores of each team. This window also contains the name of the team which has won the match. The final score will be saved in the database if connection setting is provided as explained inthe above sections (Configuring mySql for botMania and Linking botMania with mySql)A demo of the final scoreboard is shown as a snapshot



### **3. View high scores:**

There is no fun without competition. If your MySQL is configured correctly, you can simply view highscores in the game sop far by various contestants. This in turn can lead to improve in your algorithmic and strategic skills.



### **4. Configuring MySQL** has been discussed before. It has to be done before all other steps before the game is ready to execute.

A detailed document on the various APIs used in botMania bot source code writing for users is available with this package as "**readMeToo.doc**". The document will act as a guide as to how you can program your bots can develop stratagy to beat your opponents.