



REGULATIONS

Due date: 16.10.2009 Saturday 23:59

Late Submission: Late submission is **not** allowed.

Submission: Use the **COW** system to submit your homework file. The homework should be done and submitted **individually**. You will submit a single tar.gz file named “warmup.tar.gz”, which includes your source code file(s) for the assignment and a makefile. Make sure that your code can be compiled by a single “make” command at the shell prompt. The makefile should produce an executable named “warmup” which can be executed directly by typing “./warmup <args>” at the command line.



Evaluation: Your homework will be evaluated in a Linux environment. Before submission, you have to test your homework at one of the department’s computers using Linux to have a common platform.

Cheating: In case of cheating, all parts involved (source(s) and receiver(s)) get zero.

Newsgroup: You **must** follow the newsgroup for discussions and possible updates on a daily basis.

MAIN THEME – FOOTBALL FIELD

In this warm-up homework, you will implement a simple OpenGL program called FOOTBALL FIELD. This program displays a football field with players and allows some simple operations on the players.

GENERAL SPECIFICATIONS

- **Only** for this homework, a **prototype** is provided. You can refer to this prototype for appearance and functionality of the program you are supposed to build. If you think that there is anything unclear or doubtful in the prototype, please ask the related issue in the course newsgroup.

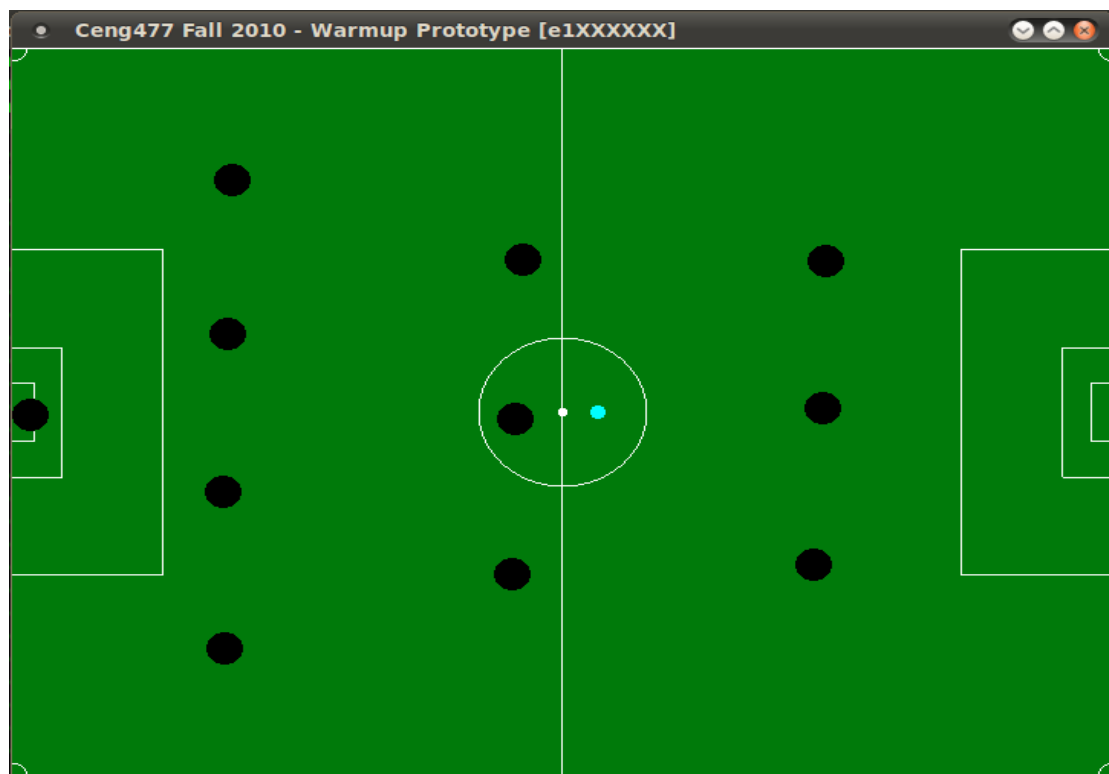
- The program runs with two command line arguments. The first one is the *width* and the second one is the *height* of the GLUT window that will be displayed on the screen. For instance, the following line creates a window of size 720x540

```
> ./warmup 720 540
```

- You can initialize OpenGL using the following statements:

```
glViewport(0, 0, width, height);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(0, width, 0, height, -10, 10);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
```

- [Hint] With the above initialization code, OpenGL's coordinate system and GLUT's coordinate system do not match. The origin of the GLUT's coordinate system is at the top left corner of the window whereas the origin of the OpenGL's coordinate system is at the bottom left corner of the window. You can adjust *y* values that come from GLUT's callback functions: $y = \text{<window height>} - y$
- It is guaranteed that the main window will not be resized.
- Remember that to use the keyboard events defined below, your main window must be the currently active window.
- Logging is a critical issue that you have to consider for a graphics program. For this warm-up homework, details are given in the “**Logging**” section.
- The title of the main window that you have created with GLUT should contain your student id.



- **Football Field:** You will display a football field on the screen. The above screen-shot from the prototype shows what the field looks like:
- You can refer to the following figure for dimensions of the field and the lines. These values are in meters and should be scaled according to GLUT window dimensions given as command line arguments to the program. You can use the width of the field for scaling. For instance, when run with the arguments `720 540`, the scale ratio will be $720/120=6$ and the center circle radius will be $9.15*6=55$. Note that, these values are given for your convenience; you are not obliged to use them. Just try to make your field seem like a real football field. This is also valid for colors. You are free to choose your own colors but please do not exploit this freedom. After all there is no point making the grass red.



FUNCTIONALITY

- **Objects (Players and Ball):**
 - Players will be displayed as small circles with a radius of 12 pixels.
 - Users should be able to change the position of each player by clicking on the left mouse button and dragging.
 - Users should be able to place a player into the field. After pressing the key '**P**', a new player is placed on the field when the user clicks the left button.
 - Users should be able to delete a player from the field. After pressing the key '**D**', the user clicks on a player to remove it from the field.
 - There will be a single ball on the field displayed as a circle of radius 5 pixels.

- The ball cannot be moved with a mouse-drag operation but pressing the key '**B**' can change the position of the ball. If there is no ball in the field, this action places a ball on the field where the left mouse button is clicked. On the other hand, if there is already a ball in the field, this action only changes the position of the ball.
- **Football Formations:**
 - In association football, the formation describes how the players in a team are positioned on the field.
 - Your program should be able to generate the following formations: 3-5-2, 4-4-3 and 4-4-2.
 - You should use GLUT's context menu to place these options. Clicking on the right mouse button should open up a menu that contains formations.
 - The exact positions of the players are not very important. For instance, if the user clicks on the 3-5-2 menu item, it is acceptable to display a formation that seems like a 3-5-2 formation. However, please make sure your formations scale well when the program runs with different parameters.
- **File Operations:**
 - **Reset (F1):** Clears the field, removing all the players and the ball.
 - **Save (F2):** Saves the location of the players and the ball into the text file named "field.txt".
 - **Load (F3):** Loads the players and the ball from the file, "field.txt".
 - **"field.txt"** has the following format:
 - Each line contains a coordinate in 2D: x, y (x and y are integers)
 - The first line specifies the location of the ball. Note that there is only a single ball in the field.
 - The rest of the lines contain the locations of the players.

LOGGING

In order to debug more efficiently, you should better print **event logs** to the standard output of your program. In order to have a little exercise, in this warm-up homework, you have to print the following to the standard output:

- When the field is cleared, saved or loaded:
 - "Field cleared!"
 - "Field saved!"
 - "Field loaded!"
- When a player/ball is added to the field:
 - "Placing player"
 - "Player placed: [<x coordinate as integer> <y coordinate as integer>]"
 - "Placing ball"
 - "Ball location: [<x coordinate as integer> <y coordinate as integer>]"

- When a player is deleted from the field:
 “Deleting player”
 “Player deleted”
- When a player is started/ended dragging:
 “Drag player started”
 “Drag player ended”

These logs are not very strict. You can log extra information as long as it does not clutter up the terminal.