

HACETTEPE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
BBM204 PROGRAMMING LAB.
ASSIGNMENT #2

Subject: Data Structures
Submission Deadline: March 12, 2015
Programming Language: C
Advisor: Abdurrahman BAYRAK

1 Overview

In this experiment you are expected to write a ping-pong game. The program's description will be read from an input file (`Input.txt`) and program's results should be written to the output file (`Result.txt`).

The ping-pong game will be played via the commands given in the input file. In Figure 1, a visualization is given for a sample game. As can be seen, there are two vertical bars (blue bars) which represent the players which are positioned outside the ball field. There is also a ball moving in the field. In the figure, several balls of different colors are illustrated to explain the game constraints.

You will implement the game in C language.

2 Specifications

The game begins with the specified initial state stating the ball's initial position and the game settings. The game area spans from (Row=0, Column=0) to (Row=24, Column=79). In addition, two players are positioned in (Row=0) and (Row=24) outside the ball field, respectively. The size of the player bars should always be taken as 5 vertical boxes.

There are also some factors affecting the game's state. First of all, the ping-pong game has two modes. If the game is in the first mode (MODE I), the ball reflects when it hits the green and yellow walls or blue bars. If the game is in the second mode (MODE II), the ball passes the related green wall and comes out from the other parallel green wall when it hits one of the green

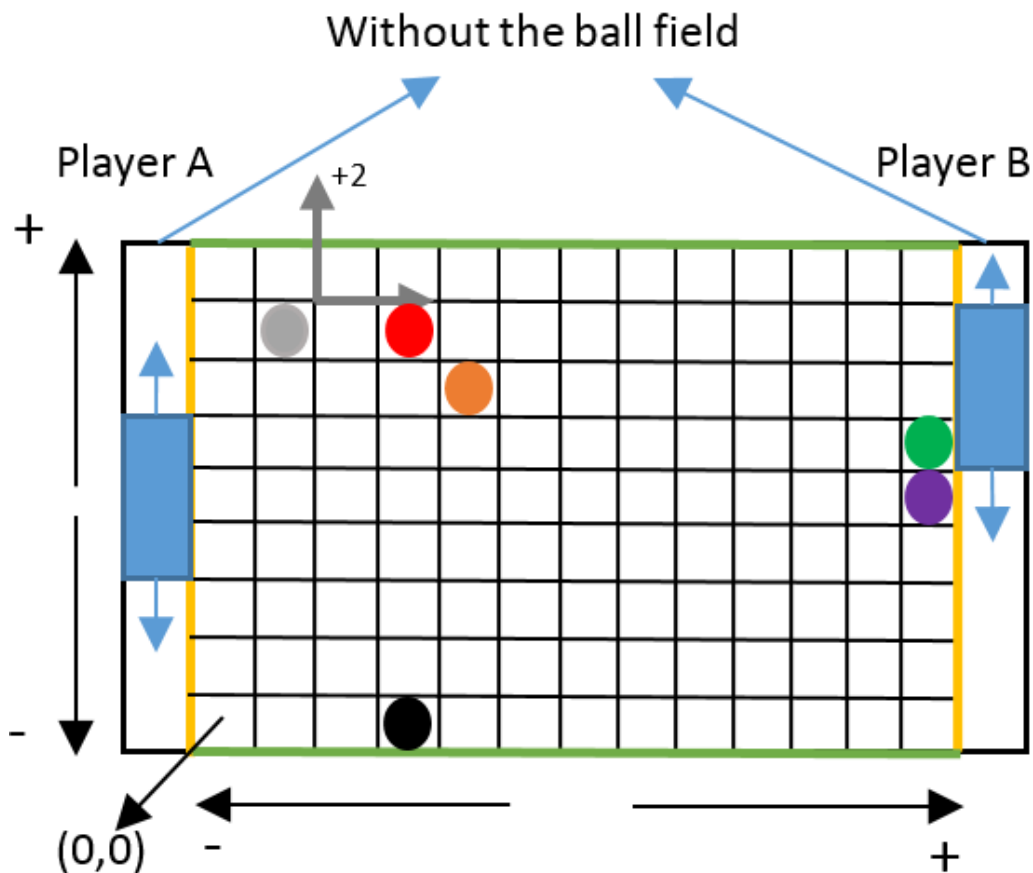


Figure 1: A sample visualization of the game.

walls. The ball always reflects from yellow walls and blue bars whether the game is in *MODE I* and *MODE II*. Note that the ball is always in motion. For instance, in Figure 1, if the speed of the gray ball is currently (2,2) (the gray arrow), its next position is demonstrated by the red ball for *MODE I*, and by the black ball for *MODE II*. At any time, the speed of the ball can be changed for example from (2, 2) to (8,-8). However, the ball will always move with degrees of 45. The sign of speed of the ball shows direction of the ball. *Priority for all the movements in the game belongs to the player bars.*

When one of the player misses the ball and ball hits his side of the yellow wall, then the other player gathers one point. For example, while Player A gathers one point for the purple ball, he/she can not gather any points for green ball. During the game, with a specific command, the game may return

to a previous state. When returned, all the old game information should be loaded, including the scores, the players' positions, the ball's position and its speed, and the game continues accordingly.

To summarize,

- Your program must be able to work on the two modes.
- Your program must be able to change the ball's speed dynamically.
- Your program must return to a previous state.

3 Implementation Details

3.1 Input

When your program starts, it has to read an initialization data from the input data file (`Input.txt`). This file includes initial position information of player bars and ball and bars movement information. Each line in the input file denotes the working time (t), and at each time step there are certain commands that can be executed. Input file's format is as follows:

```
set A_pos,B_pos,Ball_P_X,Ball_P_Y,Ball_speed_X,Ball_speed_Y,Mode
```

`set` command loads all the initial information about the game settings and constitutes the first line of the input file. `set` command includes the following parameters:

`A_Pos` is the first player's initial Y position in the player area. You will locate the first player's bar to the given coordinate. In the game, the player bars' size will be taken as 5 vertical boxes. `A_pos` must be written as lowest box number of vertical boxes in player area.

`B_Pos` is the second player's initial Y position in the player area. You will locate the second player's bar to the given coordinate. Again, the player bars' size will be taken as 5 vertical boxes. `B_pos` must be written as lowest box number of vertical boxes in player area.

`Ball_P_X` is ball's X position.

`Ball_P_Y` is ball's Y position. You will locate the ball to the (`Ball_P_X`, `Ball_P_Y`) coordinate.

`Ball_speed_X` is the speed of the ball at X axis. If the value is positive, ball's X position will increase. If the speed is negative, ball's X position will decrease. If the speed value is zero, ball's X position will not change.

`Ball_speed_Y` is the speed of the ball at Y axis. If the value is positive, ball's Y position will increase. If the speed is negative, ball's Y position will decrease. If the speed value is zero, ball's Y position will not change.

`Mode` is the running mode of the game as explained above.

```
move PlayerA_Move,PlayerB_Move
```

`move` command moves the player A and the player B. It includes the following parameters.

`PlayerA_Move` is the change in the value of the first player's Y position. If the value is positive, bar's Y position will increase. If it is negative, bar's Y position will decrease. If it is zero, bar's Y position will not change.

`PlayerB_Move` is the change in the value of the second player's Y position. If the value is positive, bar's Y position will increase. If it is negative, bar's Y position will decrease. If it is zero, bar's Y position will not change.

```
changemode Mode
```

`changemode` command changes the game's mode. The value of the `Mode` could be either 1 or 2, denoting the game's mode.

```
changespeed Ball_speed_X,Ball_speed_Y
```

`changespeed` command changes the ball's speed. It includes the following parameters:

`Ball_speed_X` is the speed of the ball at X axis.

`Ball_speed_Y` is the speed of the ball at Y axis.

```
goback score
```

`goback` command returns the game old scores. The value of the `score` represents the previous state that the game is returned to. For example,

`goback 4` means to return the game to the state denoting the four scores before. It includes given below parameter.

A sample input file is shown in Figure 2.

time	input file
0	set 3,5,4,6,5,5,1
1	move 0,0
2	move 0,0
3	changemode 2
4	move -1,0
5	move -1,0
6	move 0,0
7	changespeed 10,-10
8	move 0,0
9	move 1,0
10	move 0,0
11	move 0,0
12	goback 1
13	move 2,0
14	move 0,0
15	move 0,-1
16	move 0,0
17	move 1,0

Figure 2: A sample input file.

3.2 Output

When the commands of input file (`Input.txt`) are read and processed successfully, the program must provide the related information about the final state of the game to the output file (`Result.txt`). Output file's format is as given below:

<code>A_pos,B_pos,Ball_P_X,Ball_P_Y,Score_A,Score_B</code>
--

Here, the output line includes the following information:

A_Pos is the first player's final Y position in the player area.
B_Pos is the second player's final Y position in the player area.
Ball_P_X is ball's final X position.
Ball_P_Y is ball's final Y position.
Score_A is the first player's score.
Score_B is the first player's score.

The output for the sample input given in Figure 2 is provided in Figure 3.

$$\frac{\text{output line}}{4,4,1,1,1,2}$$

Figure 3: The output for the sample input given in Figure 2.

4 What to Hand In

You are required to submit all your code files in the given format below:

```
[Student_id]  
studentid_assignment2.c
```

Submit it to <http://www.dersler.cs.hacettepe.edu.tr>.

5 Notes

- Give explanation lines in your code(necessary details).
- Save all your work until the experiment is graded.
- Your submission must be an original, individual work. Downloaded or modified source codes will be considered as cheating. Also the students who share their codes will be considered as the same.

- During this experiment session, to communicate with your advisor or with your friends, use BBM202-204 piazza group as far as possible. Ask your questions through this piazza group. Note also that when you use any communication channel like news groups, your messages must be a little polite. Your messages should have a minimum courtesy that you already have in advisor's office. If you need to see the advisor, respect the office hours. (Tuesday 15:30-17:00, Wednesday 09:00-10:30).