Introduction to Programming

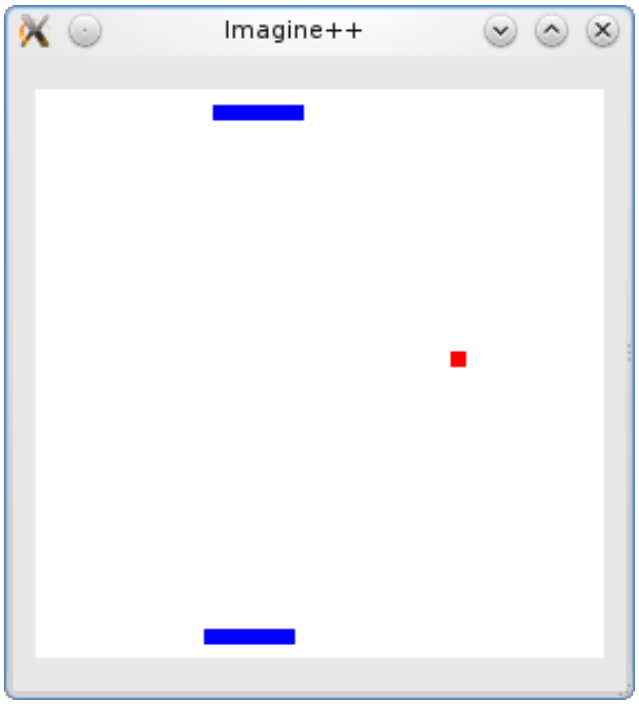
— Practical #2 —

1 Variables, loops, conditions, functions

1.1 First program with functions

1. *Get the example program :*
   Download archive `Tp2.zip` on webpage, extract the files and open the project by opening the `CMakeLists.txt` of `Tp2_Initial` from Qt Creator. Check project `Hop` whose source looks like :

```
1  #include <iostream>
2  using namespace std;
3
4  int plus(int a,int b) {
5      int c;
6      c=a+b;
7      return c;
8  }
9
10 void triple1(int a) {
11     a=a*3;
12 }
13
14 void triple2(int& a) {
15     a=a*3;
16 }
17
18 int main() {
19     int i,j=2,k;
20     i=3;
21     k=plus(i,j);
22     triple1(i);
23     triple2(i);
24     return 0;
25 }
```



Mini tennis...

2. *Debugger :*
   Run the program step by step and study how variables change. You must use the program in `Debug` mode.

1.2 First program with Imagine++

In this practical session and the following, we will use the graphics library of Imagine++ (see Appendix of lecture notes). It allows to manage easily the windowing system, drawings, and input/output of keyboard and mouse.

1. *Starter program :*
   Input your name in the placeholder of file `tennis.cpp`. You must do that at each practical. Study the code of project `Tennis` :

```
1  #include <Imagine/Graphics.h>
2  using namespace Imagine;
3  ...
4  int main() {
```

```
5       // Open window
6       openWindow(256,256);
7       // Position and speed of ball
8       int xb=128,
9           yb=20,
10          ub=2,
11          vb=3;
12      // Main loop
13      while(true) {
14          // Display ball
15          fillRect(xb-3,yb-3,7,7,RED);
16          // Temporisation
17          milliSleep(20);
18          // Erase ball
19          fillRect(xb-3,yb-3,7,7,WHITE);
20          // Rebound
21          if(xb+ub>253)
22              ub=-ub;
23          // update ball position
24          xb+=ub;
25          yb+=vb;
26      }
27      endGraphics();
28      return 0;
29  }
```

Do not care how function keyboard() works (it is a bit technical)

Build then execute the program.[1] What happens ?

2. *Online help.* At any moment, the key `F1` allows accessing the documentation of the keyword under cursor. Test this with keywords if, while and return.

   Useful key : F1 = Access documentation

3. *Understand how the program works :*
   Identify the main loop of the program. It proceeds so :
   (a) Display ball
   (b) Temporisation of a few milliseconds so that the ball does not move too fast
   (c) Erase ball
   (d) Handle rebounds by updating speed
   (e) Update ball coordinates.

   The line with instruction while *may* signal a warning. Can you understand why ? What is the point of the condition by instruction if ?

4. *Managing all rebounds :*
   Complete the program so that all rebounds are handled. For example, inverse the horizontal component `ub` of the speed vector when the balls is going to meet the left or right border of the window.

5. *Global variables :*
   Double the window height. Modify the ball size. This requires hanging the code at several locations. Instead of placing numerical variables "in the rock", it is better to define variables for them. To make it clear, use constant global variables. For that, insert right after the two include lines the following code :

   ```
   const int width = 256;    // Window width
   const int height = 256;   // Window height
   const int ball_size = 3; // Radius of ball
   ```

   and reformulate numeric values in the program with these variables. The keyword const indicates that the variables cannot be modified after their initalization. Try to add the line width=300; at the start of function main and note that it makes a compilation error.

1. Since the project has two executable programs, make sure the project `Tennis` is selected with the Project button on top of the start green triangle

6. *Usage of functions :*
   The ball is drawn twice in the loop, the first time in red and the second one in white to erase it. Here, drawing the ball requires a single line but could be much more if it had a more complex shape. Thus, to structure the program and make the code more readable while minimizing duplication, group the display of the ball and its erasure in a function DrawBall defined before the function main :

   ```
   void DrawBall(int x, int y, Color col) {
       ...
   }
   ```

   In the same manner, define a function
   void MoveBall(int &x, int &y, int &u, int &v)
   to handle rebounds and movement of the ball.

1.3 Tennis

We are going to make the program more fun by adding two rackets moving horizontally at top and bottom of the screen, and controlled by the keyboard.

1. *Display of rackets :*
   Add in function main variables xr1,yr1,xr2,yr2 dedicated to the position of both rackets. Then define a function DrawRacket taking example from DrawBall. Put the calls to these functions at appropriate locations in the loop.
2. *Keyboard handling :*
   Management of the keyboard is ready for use in function keyboard whose content we will not comment. This functions allows knowing directly if one of the keys of interest (s and d for the first raquet, k and l for the second) are pressed or not. This function, keyboard(int& sens1, int& sens2), returns in sens1 and sens2, the values 0, -1 or 1 (0 : no move, -1 : to the left, 1 : to the right).
3. *Racket motion :*
   Code the motion of a racket in a function
   void MoveRacket(int &x, int sens)
   then call this function in the main loop for each racket. Naturally, make sure the rackets do not move outside the window.
4. *Rebounds on rackets :*
   Take inspiration from the management of rebounds of the ball. Here, we have to check not only whether the ball will reach the top or bottom of the screen, but also if it is close enough in abscissa to the corresponding racket.
5. *Count and display score :*
   Modify the function MoveBall so as to keep track of the score of the players and display it in the terminal.
6. *For the ones that have finished :*
   During a rebound on the racket, modify the inclination of the trajectory of the ball as a function of the speed of the racket and of the location of the rebound.

You should have obtained a program looking like the one in the figure.