

```

1  #include "raylib.h"
2  #include <iostream>
3
4  using namespace std;
5
6  // Couleur
7  Color Vert = Color{38, 185, 154, 255};
8  Color Vert_noir = Color{20, 160, 133, 255};
9  Color Vert_claire = Color{129, 204, 184, 255};
10 Color Jaune = Color{243, 213, 91, 255};
11
12 int player_score    = 0;
13 int cpu_score       = 0;
14
15 class Ball{
16 public:
17     float x, y;
18     int vitesseX, vitesseY;
19     int rayon;
20
21     void Draw()
22     {
23         DrawCircle(x, y, rayon, Jaune);
24     }
25
26     void Update()
27     {
28         x += vitesseX;
29         y += vitesseY;
30
31         if( y + rayon >= GetScreenHeight() || y - rayon <= 0 )
32         {
33             vitesseY *= -1;
34         }
35
36         if( x + rayon >= GetScreenWidth() )
37         {
38             cpu_score++;
39             Reinitialise();
40         }
41         if( x - rayon <= 0 )
42         {
43             player_score++;
44             Reinitialise();
45         }
46     }
47
48     void Reinitialise()
49     {
50         x = GetScreenWidth() / 2;
51         y = GetScreenHeight() / 2;
52
53         int choix_vitesse[2] = {-1, 1};
54         vitesseX *= choix_vitesse[GetRandomValue(0, 1)];
55         vitesseY *= choix_vitesse[GetRandomValue(0, 1)];
56     }
57 };
58
59 class Paddle{
60 protected:
61     void LimiteMouvement()
62     {
63         if( y <= 0 ){
64             y = 0;
65         }
66         if ( y + longueur >= GetScreenHeight() )

```

```

67         {
68             y = GetScreenHeight() - longueur;
69         }
70     }
71
72 public:
73     float x, y;
74     float largeur, longueur;
75     float vitesseY;
76
77     void Draw()
78     {
79         DrawRectangleRounded(Rectangle{x, y, largeur, longueur}, 0.8, 0, WHITE);
80     }
81     void Update()
82     {
83         if(IsKeyDown(KEY_UP))
84         {
85             y = y - vitesseY;
86         }
87         if(IsKeyDown(KEY_DOWN))
88         {
89             y = y + vitesseY;
90         }
91         LimiteMouvement();
92     }
93 };
94
95
96 class CpuPaddle: public Paddle{
97 public:
98     void Update(float ballY)
99     {
100         if(y + longueur/2 > ballY)
101         {
102             y = y - vitesseY;
103         }
104         if(y + longueur/2 <= ballY)
105         {
106             y = y + vitesseY;
107         }
108         LimiteMouvement();
109     }
110
111 };
112
113
114 Ball        ball;
115 Paddle      player;
116 CpuPaddle   cpu;
117
118 int main()
119 {
120     const int ecranX = 1200;
121     const int ecranY = 600;
122
123     // Balle
124     ball.rayon      = 10;
125     ball.x          = ecranX / 2;
126     ball.y          = ecranY / 2;
127     ball.vitesseX   = 7;
128     ball.vitesseY   = 7;
129
130     // Barre
131     player.largeur   = 25;
132     player.longueur  = 120;

```

```

133     player.x          = ecranX - player.largeur - 10;
134     player.y          = ecranY / 2 - player.longueur / 2;
135     player.vitesseY    = 6;
136
137     // CPU
138     cpu.longueur        = 120;
139     cpu.largeur         = 25;
140     cpu.x              = 10;
141     cpu.y              = ecranY / 2 - cpu.longueur / 2;
142     cpu.vitesseY       = 6;
143
144     //      cout << "Start the game" << endl;
145
146     SetTargetFPS(60);
147
148     InitWindow(ecranX, ecranY, "Jeu de Pong [Pong - Game]");
149
150     while(WindowShouldClose() == false)
151     {
152         // Gestion d'evenement
153
154
155         // Mise a jours des variables
156         ball.Update();
157         player.Update();
158         cpu.Update(ball.y);
159
160         // Collision
161         if( CheckCollisionCircleRec(Vector2{ball.x, ball.y}, ball.rayon, Rectangle{
162     player.x, player.y, player.largeur, player.longueur}))
163         {
164             ball.vitesseX *= -1;
165         }
166
167         if( CheckCollisionCircleRec(Vector2{ball.x, ball.y}, ball.rayon, Rectangle{
168     cpu.x, cpu.y, cpu.largeur, cpu.longueur}))
169         {
170             ball.vitesseX *= -1;
171         }
172         // Dessin
173         BeginDrawing();
174
175         // Clean
176         ClearBackground(Vert_noir);
177
178         // Header
179
180         DrawRectangle(ecranX/2, 0, ecranX/2, ecranY, Vert);
181
182         DrawCircle(ecranX/2, ecranY/2, 150, Vert_claire);
183         // Ligne
184         DrawLine(ecranX/2, 0, ecranX/2, ecranY, WHITE);
185
186         // Cercle
187         ball.Draw();
188         // Rectangle
189         cpu.Draw();
190         player.Draw();
191
192         // Afficher du texte
193         DrawText(TextFormat("%i", cpu_score), ecranX/4 - 20, 20, 80, WHITE);
194         DrawText(TextFormat("%i", player_score), 3 * ecranX/4 - 20, 20, 80,
195     WHITE);
196
197         EndDrawing();
198     }

```

```
196
197     CloseWindow();
198
199     return 0;
200 }
```