5.6.3 Moving Rectangle

This example SwinGame code will move a rectangle back and forth across the screen.

Program Description	
Name	*Moving Rectangle*
Description	Displays a rectangle that is moved back and f… the screen.

Table 5.17: Description of the Moving Rectangle prograa

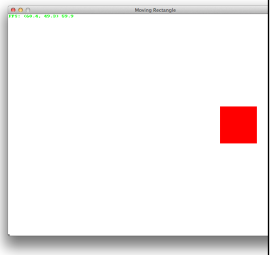

Figure 5.90: Example execution of the Moving Rectangle pro

C++

```
#include "SwinGame.h"

#define RECT_WIDTH 100
#define RECT_HEIGHT 100
#define MOVE_X 5

// Update the x position of the rectangle, by the specified am
void update_rect_position(int &x, int &dx)
{
    // Move x (passed in by reference)
    x += dx;

    // Check if it went of the screen
    if (x < 0)
    {
        // off the left of the screen
        dx = -dx;   // change movement direction
        x = 0;      // put it back on the screen
    }
    else if ((x + RECT_WIDTH) > screen_width())
    {
        // off the screen to the right
        dx = -dx;   // change movement direction
        x = (screen_width() - RECT_WIDTH); // put it back on th
    }
}

// Draw a rectangle moving across the screen
int main()
{
    int rect_x = 0;
    int rect_y = 250;
    int rect_x_move = MOVE_X;

    open_graphics_window("Moving Rectangle", 800, 600);
    load_default_colors();

    do
    {
        process_events();

        // Update the location of the rectangle
        update_rect_position(rect_x, rect_x_move);

        // Clear the screen, then draw the rectangle
        clear_screen(ColorWhite);
        fill_rectangle(ColorRed, rect_x, rect_y, RECT_WIDTH, RE
        draw_framerate(0,0);

        // Refresh the screen, keep it at 60fps
        refresh_screen(60);
    } while ( ! window_close_requested() );

    return 0;
}
```

Listing 5.32: C++ Moving Rect SwinGame code

Pascal

```
program MovingRect;
uses sgGraphics, sgUtils, sgInput, sgText;

const
  RECT_WIDTH = 100;
  RECT_HEIGHT = 100;
  MOVE_X = 5;

// Update the x position of the rectangle, by the specified amount
procedure UpdateRectPosition(var x, dx: Integer);
begin
    // Move x (passed in by reference)
    x += dx;

    // Check if it went of the screen
    if x < 0 then
    begin
        // off the left of the screen
        dx := -dx;   // change movement direction
        x := 0;      // put it back on the screen
    end
    else if (x + RECT_WIDTH) > ScreenWidth() then
    begin
        // off the screen to the right
        dx := -dx;   // change movement direction
        x := (ScreenWidth() - RECT_WIDTH); // put it back on the screen
    end;
end;

// Draw a rectangle moving across the screen
procedure Main();
var
  rectX: Integer = 0;
  rectY: Integer = 250;
  rectXMove: Integer = MOVE_X;
begin
    OpenGraphicsWindow('Moving Rectangle', 800, 600);

    repeat
        ProcessEvents();

        // Update the location of the rectangle
        UpdateRectPosition(rectX, rectXMove);

        // Clear the screen, then draw the rectangle
        ClearScreen(ColorWhite);
        FillRectangle(ColorRed, rectX, rectY, RECT_WIDTH, RECT_HEIGHT);
        DrawFramerate(0,0);

        // Refresh the screen, keep it at 60fps
        RefreshScreen(60);
    until WindowCloseRequested();
end;

begin
  Main();
end.
```

Listing 5.33: Pascal Moving Rect SwinGame code