

```

#include <allegro.h>
#include <cstdlib>
#include <time.h>

int ball_x = 320;
int ball_y = 240;

int ball_tempX = 320;
int ball_tempY = 240;

int p1_x = 20;
int p1_y = 210;

int p1_tempX = 20;
int p1_tempY = 210;

int p2_x = 620;
int p2_y = 210;

int p2_tempX = 620;
int p2_tempY = 210;

time_t secs; //The seconds on the system clock will be stored here
//this will be used as the seed for srand()

int dir; //This will keep track of the circles direction
//1= up and left, 2 = down and left, 3 = up and right, 4 = down
and right

BITMAP *buffer; //This will be our temporary bitmap for double buffering

void moveBall(){

    ball_tempX = ball_x;
    ball_tempY = ball_y;

    if (dir == 1 && ball_x > 5 && ball_y > 5){

        if( ball_x == p1_x + 15 && ball_y >= p1_y && ball_y <= p1_y + 60){
            dir = rand()% 2 + 3;
        }else{
            --ball_x;
            --ball_y;
        }
    }

    } else if (dir == 2 && ball_x > 5 && ball_y < 475){

        if( ball_x == p1_x + 15 && ball_y >= p1_y && ball_y <= p1_y + 60){
            dir = rand()% 2 + 3;
        }else{
            --ball_x;
            ++ball_y;
        }
    }
}

```

```

} else if (dir == 3 && ball_x < 635 && ball_y > 5){

    if( ball_x + 5 == p2_x && ball_y >= p2_y && ball_y <= p2_y + 60){
        dir = rand()% 2 + 1;
    }else{
        ++ball_x;
        --ball_y;
    }

} else if (dir == 4 && ball_x < 635 && ball_y < 475){

    if( ball_x + 5 == p2_x && ball_y >= p2_y && ball_y <= p2_y + 60){
        dir = rand()% 2 + 1;
    }else{
        ++ball_x;
        ++ball_y;
    }

} else {

    if (dir == 1 || dir == 3)    ++dir;
    else if (dir == 2 || dir == 4)    --dir;

}

acquire_screen();
circlefill ( buffer, ball_tempX, ball_tempY, 5, makecol( 0, 0, 0));
circlefill ( buffer, ball_x, ball_y, 5, makecol( 128, 255, 0));
draw_sprite( screen, buffer, 0, 0);
release_screen();

rest(5);

}

void p1Move(){

    p1_tempY = p1_y;

    if( key[KEY_W] && p1_y > 0){

        --p1_y;

    } else if( key[KEY_S] && p1_y < 420){

        ++p1_y;

    }

    acquire_screen();
    rectfill( buffer, p1_tempX, p1_tempY, p1_tempX + 10, p1_tempY + 60,
makecol ( 0, 0, 0));
    rectfill( buffer, p1_x, p1_y, p1_x + 10, p1_y + 60, makecol ( 0, 0,
255));
    release_screen();
}

```

```

}

void p2Move() {

    p2_tempY = p2_y;

    if( key[KEY_UP] && p2_y > 0){

        --p2_y;

    } else if( key[KEY_DOWN] && p2_y < 420){

        ++p2_y;

    }

    acquire_screen();
    rectfill( buffer, p2_tempX, p2_tempY, p2_tempX + 10, p2_tempY + 60,
makecol ( 0, 0, 0));
    rectfill( buffer, p2_x, p2_y, p2_x + 10, p2_y + 60, makecol ( 0, 0,
255));
    release_screen();

}

void startNew(){

    clear_keybuf();
    readkey();
    clear_to_color( buffer, makecol( 0, 0, 0));
    ball_x = 320;
    ball_y = 240;

    p1_x = 20;
    p1_y = 210;

    p2_x = 620;
    p2_y = 210;

}

void checkWin(){

    if ( ball_x < p1_x){
        textout_ex( screen, font, "Player 2 Wins!", 320, 240, makecol( 255,
0, 0), makecol( 0, 0, 0));
        startNew();
    } else if ( ball_x > p2_x){
        textout_ex( screen, font, "Player 1 Wins!", 320, 240, makecol( 255,
0, 0), makecol( 0, 0, 0));
        startNew();
    }

}

}

```

```

void setupGame(){

    acquire_screen();
    rectfill( buffer, p1_x, p1_y, p1_x + 10, p1_y + 60, makecol ( 0, 0,
255));
    rectfill( buffer, p2_x, p2_y, p2_x + 10, p2_y + 60, makecol ( 0, 0,
255));
    circlefill ( buffer, ball_x, ball_y, 5, makecol( 128, 255, 0));
    draw_sprite( screen, buffer, 0, 0);
    release_screen();

    time(&secs);
    srand( (unsigned int)secs);
    dir = rand() % 4 + 1;

}

int main(){

    allegro_init();
    install_keyboard();
    set_color_depth(16);
    set_gfx_mode( GFX_AUTODETECT, 640, 480, 0, 0);

    buffer = create_bitmap( 640, 480);

    setupGame();

    while( !key[KEY_ESC]){

        p1Move();
        p2Move();
        moveBall();
        checkWin();

    }

    return 0;

}
END_OF_MAIN();

```

Lets break this down. There is a new header file not covered in the lessons so far, time.h. This contains the functions used to get the current time from the system clock, this will be used to calculate the random numbers.

main() runs through the usuall setup for the game and then calls setupGame(). This function draws the screen and calls time() to get the current time. srand() is a function that changes the "seed" that rand() will use to make random numbers, the seed is the initial value rand() will begin with. In this case the seed is set to the current amount of seconds the system clock is displaying, this way there will always be a new seed everytime the program runs so there will never be a repetitive pattern.

dir = rand() % 4 + 1; will give dir a value between 1 and 4. rand() % x will return a random

number between 0 and  $x-1$ , in this case 0 to 3. Since we want a number from 1 to 4, we simply add 1 to the result.

Once the setup is complete, the main loop begins. `p1Move()` will move the paddle on the left side of the screen when W and S are pressed, and `p2Move()` will do the same for the other paddle when up and down are pressed. `moveBall()` checks to see if it is about to go off the screen, or hit a paddle and change its course accordingly.

`checkWin()` will see if the ball has passed a paddle and declare a winner if it has. `startNew()` will simply clear the screen and reset the values for the ball's and players' positions.