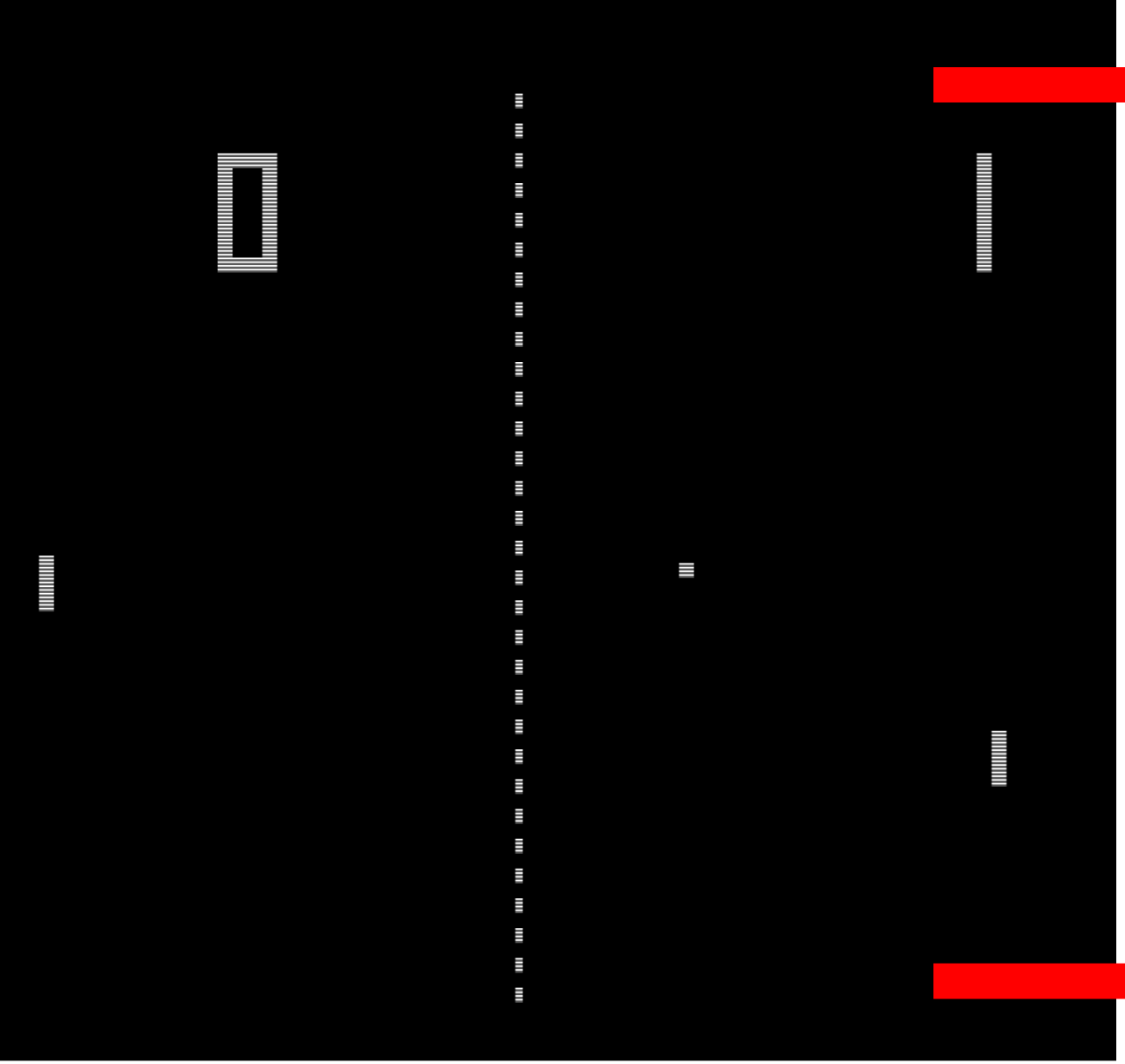
Bohdan Khmelnytsky National University of Cherkasy
Academic School of Computer Engineering, Intelligent and Control Systems
Software Of Automated Systems Department

DEVELOPMENT OF THE "PONG" GAME

Done by
Denys Datsenko

Mentored by
Julia Grebenovich



Objective of the work and the main tasks

An objective of the work was to become acquainted and to learn the algorithms of movement in two-dimensional graphics and to create a computer game “Pong” for amusement.

The main tasks of the work were to analyze the methods of movement implementation and to implement the actual game in a type of software format which would be easy to use.

Ralph Henry Baer, the inventor of the game

Game concept and its history

"Pong" is one of the earliest arcade video games and the very first sports arcade video game. It is a table tennis sports game featuring simple two-dimensional graphics. Pong was one of the first video games to reach mainstream popularity. This simple game features two paddles and a ball. The goal is to defeat your opponent by being the first one to gain 10 points, a player gets a point once the opponent misses a ball.

Windows Forms

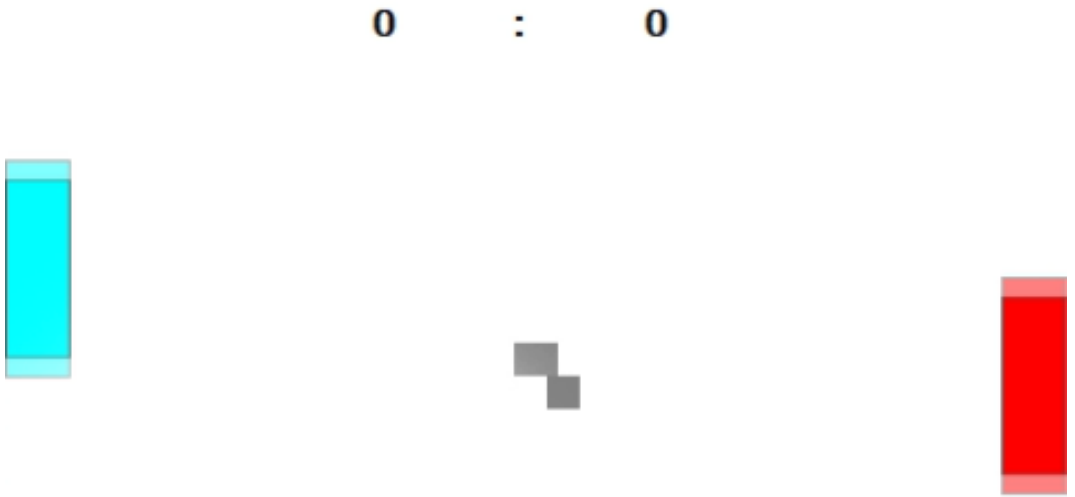
Visual Studio

— DEVELOPMENT OF THE “PONG” GAME

algorithms

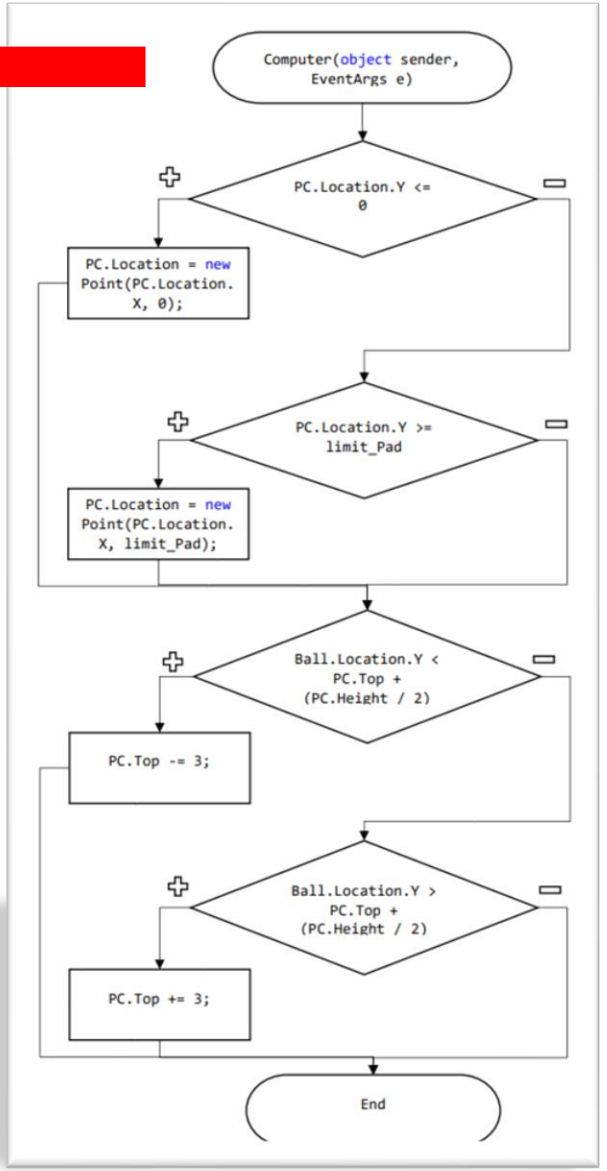
Imitation of movement

Each of the paddles and a ball are controlled by timers which run very quickly creating a new element on the screen and deleting the previous one. This algorithm is used to imitate the movement.



Two sequential frames taken from the demo of the program drawn on top of each other with 50% transparency

Flow chart describing the mentioned algorithm



PC paddle

Algorithm of the movement of computer-controlled paddle is pretty simple as well. It consists of two **if-statements**. One of them will not let the paddle leave the form borders and the second one always follows the ball depending on its location.

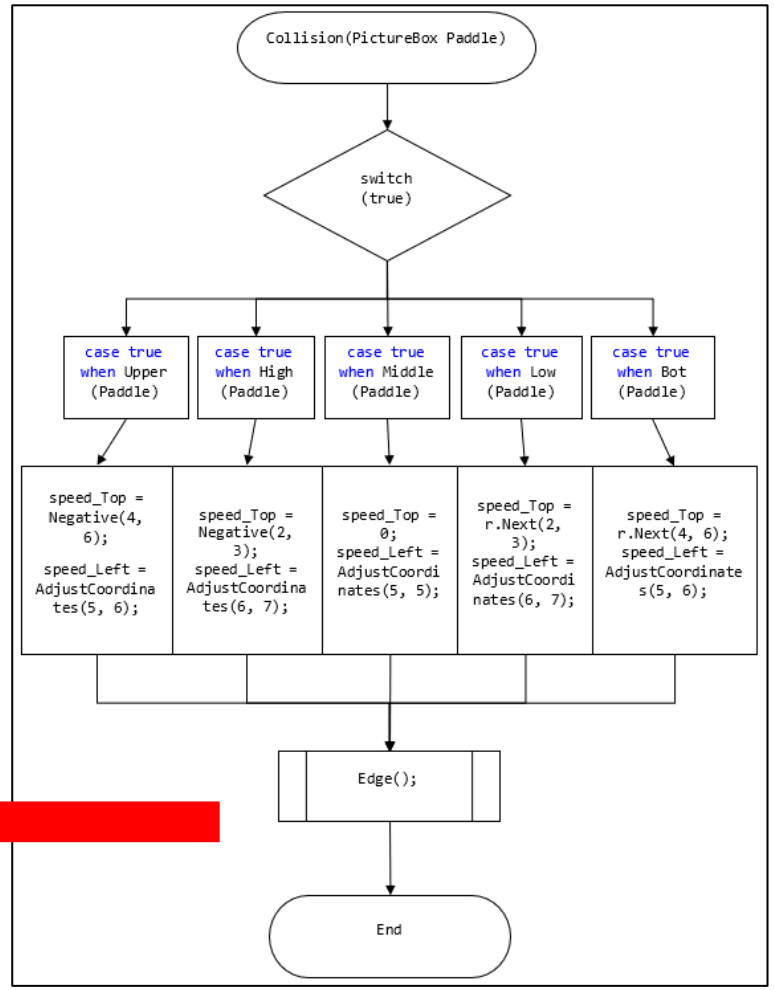
Movement of the ball

The movement of the ball is controlled by two variables which can take positive or negative numbers and which change with each collision with either horizontal borders of the form or the paddles.

```
private void StartValues()
{
    speed_Top = 0;
    speed_Left = -5;
}
ссылка: 1
private void BallMoves()
{
    Ball.Top += speed_Top;
    Ball.Left += speed_Left;
}
```

Initialization of the variables which control the ball

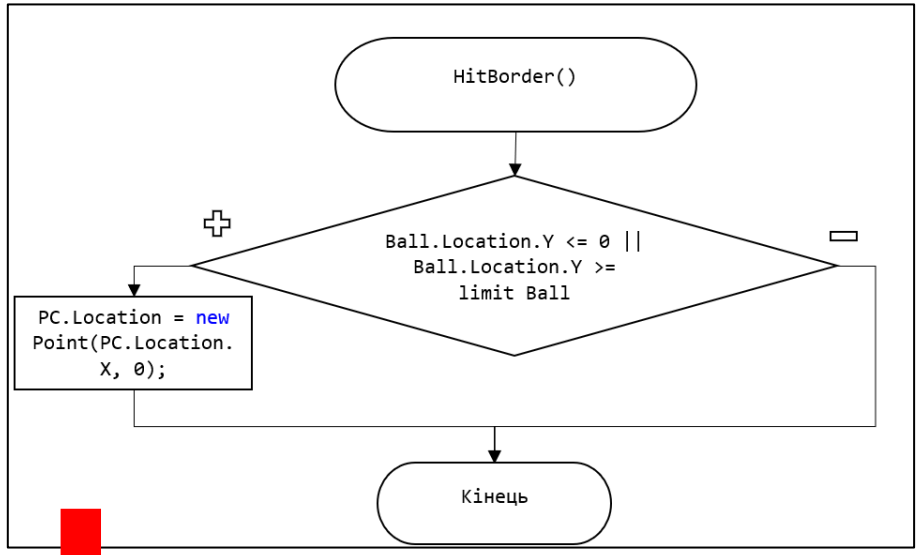
Collision of the ball with paddles



```
private bool Upper(PictureBox Pad)
{
    return Ball.Location.Y >= Pad.Top - Ball.Height && Ball.Location.Y <= Pad.Top + Ball.Height;
}

private bool High(PictureBox Pad)
{
    return Ball.Location.Y > Pad.Top + Ball.Height && Ball.Location.Y <= Pad.Top + 2 * Ball.Height;
}
ссылка: 1
private bool Middle(PictureBox Pad)
{
    return Ball.Location.Y > Pad.Top + 2 * Ball.Height && Ball.Location.Y <= Pad.Top + 3 * Ball.Height;
}
ссылка: 1
private bool Low(PictureBox Pad)
{
    return Ball.Location.Y > Pad.Top + 3 * Ball.Height && Ball.Location.Y <= Pad.Top + 4 * Ball.Height;
}
ссылка: 1
private bool Bot(PictureBox Pad)
{
    return Ball.Location.Y > Pad.Top + 4 * Ball.Height && Ball.Location.Y <= Pad.Bottom + Ball.Height;
}
```

Each of the paddles is divided into 5 identical parts, represented as a bool method and called in a “Collision” subprogram which detects the exact numbers for the variables controlling the ball and assigns them.



***HitBorder()* subprogram which controls the direction of the ball while hitting one of the horizontal borders of the game field**

```
private void Edge()
{
    if (Ball.Location.X < this.Width / 2)
    {
        if (Ball.Location.X < 0 + Ball.Height / 3)
        {
            speed_Left *= -1;
        }
    }
    else if (Ball.Location.X > this.Width / 2)
    {
        if (Ball.Location.X > PC.Location.X + (Ball.Width / 3))
        {
            speed_Left *= -1;
        }
    }
}
```

***Edge()* method deals with not letting the horizontal borders of the paddles to hit the ball**

```
private void BallLeftField()
{
    if (player_won == 10 || computer_won == 10)
    {
        EndGame();
    }

    if (Ball.Location.X < 0)
    {
        NewPoint(5);
        ComputerWon();
    }
    else if (Ball.Location.X > this.ClientSize.Width)
    {
        NewPoint(-5);
        PlayerWon();
    }
}
```

When the ball leaves the playing field, it is checked whether one of the players have gained 10 points. If this statement is true, the variables where the score is stored change their value to 0, each of the paddles and a ball are coming back to their start positions and the “Start Game” button is being shown. If it is false, then one of the players gets one point depending on the ball’s position on the playing field (is it on the left or the right part of it) and the ball comes back on its starting position which is center of the field.

Conclusions

This course work was dedicated to creating a software product for entertaining and learning the basics of movement algorithms in simple video games like this one. This project may be useful for other people who have started learning programming. It is really easy to use and simple. In the future this work may be improved by adding more graphic elements, the possibility of playing with 2, 3 or 4 players, adding the parameters which will control the speed of the paddles and the ball or their appearance and the possibility to create a ranking for different players playing on the same device.

THANK YOU FOR YOUR ATTENTION!

Any questions?