Черкаський національний університет імені Богдана Хмельницького

Кафедра програмного забезпечення автоматизованих систем

КУРСОВА РОБОТА

з дисципліни «Програмування та алгоритмічні мови»

НА ТЕМУ «Розробка гри «Pong»

Студента 1 курсу, групи КС-19

напряму підготовки «Програмна інженерія»

спеціальності «Інженерія програмного забезпечення»

Даценка Дениса Владиславовича

Керівник старший викладач Гребенович Ю.Є.

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії _____ Гребенович Ю.Є.
(підпис) (прізвище та ініціали)

_____ Онищенко Б.О.
(підпис) (прізвище та ініціали)

_____ Порубльов І.М.
(підпис) (прізвище та ініціали)

м. Черкаси – 2020 рік

Зміст

Вступ 3

Розділ 1. Мета роботи. Історичний бекграунд гри, її суть, модифікації та способи реалізації 4

1.1. Опис гри та її історичний розвиток 4

1.2. Огляд можливостей для створення інтерфейсу і вибір IDE 5

1.3. Огляд алгоритмів моделювання руху у двовимірній графіці 6

1.4. Висновки до першого розділу 6

Розділ 2. Проектування коду 7

2.1. Алгоритм руху платформи-комп’ютера 7

2.2. Регулювання напрямку кульки 7

2.3. Метод Collision 8

2.4. Відбиття від горизонтальних границь ігрового поля 8

2.5. М'ячик вийшов за межі ігрового поля 9

2.6. Висновки до другого розділу 9

Розділ 3. Реалізація програмного продукту 10

3.1. Інтерфейс користувача 10

3.2. Елементи управління грою 10

3.3. Рух м’ячика по ігровому полю 11

3.4. Висновки до третього розділу 15

Висновки 16

Список використаних джерел 17

Додаток А. Блок-схема алгоритму руху платформи-комп’ютера 18

Додаток Б. Блок-схема до методу Collision() 19

Додаток В. Блок-схема методу BallLeftField() 20

Вступ

У людства завжди була потреба якось себе розважити і у всіх поколінь були власні методи це зробити. Розвиток комп'ютерних технологій і «технологічний бум» власне і дав старт індустрії комп'ютерних ігор, яка не втрачає релевантності вже не перший десяток років. Ще одним поштовхом до активного розвитку ігрової індустрії став бестселер компанії Atari 1970-х років – «Pong» – гра, розробка якої почалася в далекому 1958 році Ральфом Байєром, який хотів здійснити свою задумку створити просту відеогру, в яку людина зможе зіграти на своєму телевізорі.

Сьогодні її вважають найвідомішою і найуспішнішою аркадою у світі. Успіх проекту оцінюється у 78,5 млрд доларів США. Саме цей продукт свого часу популяризував ігрову індустрію, а сьогодні завдяки йому, вона є більшою за усю світову кіноіндустрію, про що 60 років тому не можна було й навіть уявити.

Метою роботи є ознайомлення та вивчення алгоритмів руху у двовимірній графіці та розробка комп'ютерної гри «Pong» у розважальних цілях.

Постановка задач курсової роботи:

1. Проаналізувати методи реалізації руху у двовимірній графіці.
2. Побудувати блок-схеми створених алгоритмів.
3. Реалізувати у вигляді програмного продукту сформовані методи.
4. Зробити висновки щодо можливостей і актуальності створеного програмного забезпечення.

Розділ 1. Мета роботи. Історичний бекграунд гри, її суть, модифікації та способи реалізації

1.1. Опис гри та її історичний розвиток

“Pong” – це одна з перших комп’ютерних ігор у жанрі «аркада», яка імітує настільний теніс використовуючи просту двовимірну графіку. Саме вона стала першою найбільш успішною відеогрою і дала старт розвитку індустрії комп’ютерних ігор. Гравець контролює одну з платформ відбиваючи кульку, траєкторія якої змінюється залежно від кута зіткнення. Коли ж гравець або комп’ютер не встигають відбити кульку, то до рахунку суперника додається одне очко. Переможцем стає той гравець, котрий першим набере 10 очок.

Свого часу гра була неодноразово перевидана різними компаніями та отримала безліч ремейків з новими ігровими елементами. Але майже через 50 років після створення будь-яку гру навряд чи можна назвати актуальною, і ця не є виключенням. Вона вже визнана своєрідним історичним спадком ігрової індустрії і з’явилися набагато цікавіші та візуально більш привабливі комп’ютерні ігри. Також гра іноді використовується для вивчення основ деяких мов програмування.

За ці роки гра отримала багато нових бачень авторства інших вчених та розробників програмного забезпечення, серед яких є версії для людей з вадами зору [1], версії з використанням систем захоплення рухів і ракеткою, яку можна взяти у власні руки та пограти проти комп’ютера [2] та багато інших.

Рис 1.1. Тестування віртуального пінг-понгу з використанням системи захоплення рухів на виставці SEK2007 у Сеулі, Корея. [2, с.1]

1.2. Огляд можливостей для створення інтерфейсу і вибір IDE

Для розробки програмного продукту було обрано мову програмування C#. Вона є відомою завдяки власній простоті й потужності, великому різноманіттю синтаксичних конструкцій, можливістю роботи з платформою .NET і є достатньо надійною.

Серед інтегрованих середовищ розробки вибір пав на Microsoft Visual Studio зі встановленим пакетом інструментів мови C#. Visual Studio дозволяє зручно синхронізуватися з системами контролю версій, таких як Git, автоматично створює стандартний код з рядом необхідних деталей, щоб розробнику не доводилося вводити його вручну, підтримує інтуїтивний стиль кодування та має російськомовну локалізацію. Цей ряд переваг робить дане IDE очевидним лідером серед своїх конкурентів.

Серед графічних бібліотек мною було обрано Windows Forms, яка спрощує доступ до елементів інтерфейсу. Не дивлячись на те, що вона є трохи застарілою відносно Windows Presentation Forms, її ефективності буде більше ніж достатньо для реалізації поставленого перед нами завдання. Windows Forms підтримує кросплатформенні додатки і прості проекти, які були написані з її використанням можна вільно перенести на іншу операційну систему, якщо на ній встановлена необхідна версія .NET Framework.

1.3. Огляд алгоритмів моделювання руху у двовимірній графіці

Рух примітивної двовимірної графіки зімітувати доволі нескладно. Варто лише розуміти принцип відео, яке по факту складається з певної кількості кадрів, які з’являються на екрані за одну секунду, імітуючи рух.

Рис 1.2. Два кадри, що йдуть один за одним, накладені один на одного з 50-відсотковою прозорістю. Можна спостерігати, як зображення ніби трохи зсунулося вправо через рух телевізійного оператора по колу навколо артистки.

Такий самий принцип використовується і у графіці. Певний елемент необхідно наділити таймером, який буде працювати, коли це потрібно, і зсувати зображення на невелику відстань, знищуючи попереднє. Цей алгоритм і буде використаний при реалізації руху кульки та «платформ» гравців для створення «ілюзії» руху.

1.4. Висновки до першого розділу

У цьому розділі була розглянута історія та концепт гри «Pong», головний алгоритм програмного продукту, а також можливості для його реалізації.

Розділ 2. Проектування коду

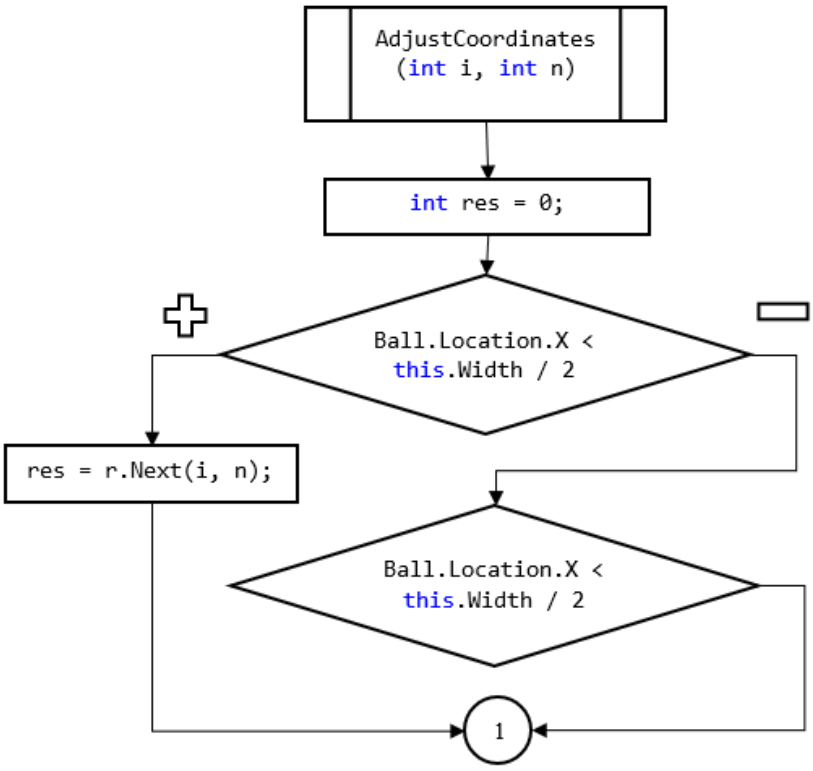
Хоча реалізація комп'ютерних методик і є основною метою, багато зусиль при розробці ігор спрямовується на створення основи, до якої слід її застосувати. Замість того, щоб зосередитися на розробці повністю нової гри, ми відтворюємо класичну гру, що дозволяє докласти більше зусиль до реалізації.

2.1. Алгоритм руху платформи-комп’ютера

Даний метод скаладається з двох if-statement'ів, перший з яких обмежуватиме рух платформи по рамкам форми, а другий постійно слідкуватиме за рухом м’ячика. (див. Додаток А).

2.2. Регулювання напрямку кульки

У екземплярі класу Random, який є винесеним окремо генеруються випадкові числа. В залежності від позиції м’ячика (вище чи нижче середини ширини форми), платформа відбиватиме м’ячик у протилежну сторону надаючи значення змінним, що контролюють напрямки горизонтального та вертикального руху м’ячика.



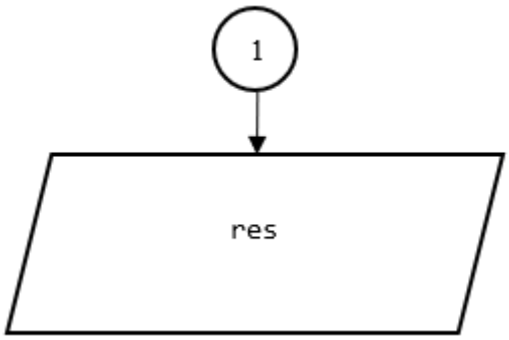


Рис 2.1. Блок-схема до алгоритму регулювання напрямку кульки

2.3. Метод Collision

Collision – з англ. зіткнення.

Даний метод має викликатися при зіткненні м’ячика з однією з платформ і перевіряє, від якої п’яти рівних частин, на які вона поділена, він відбився, регулюючи напрямок руху. (див. Додаток Б).

2.4. Відбиття від горизонтальних границь ігрового поля

При відбитті м’ячика від верхньої чи нижньої границі ігрового поля, він змінює свій напрямок на протилежний. Тому коли значення вертикального вектора приймає значення менше за 0 (де 0 – це верхня границя ігрового вікна) відносно осі oy або більше за ширину вікна, значення змінної, що контролює вертикальний напрямок руху треба змінити на таке ж, але зі знаком «мінус».

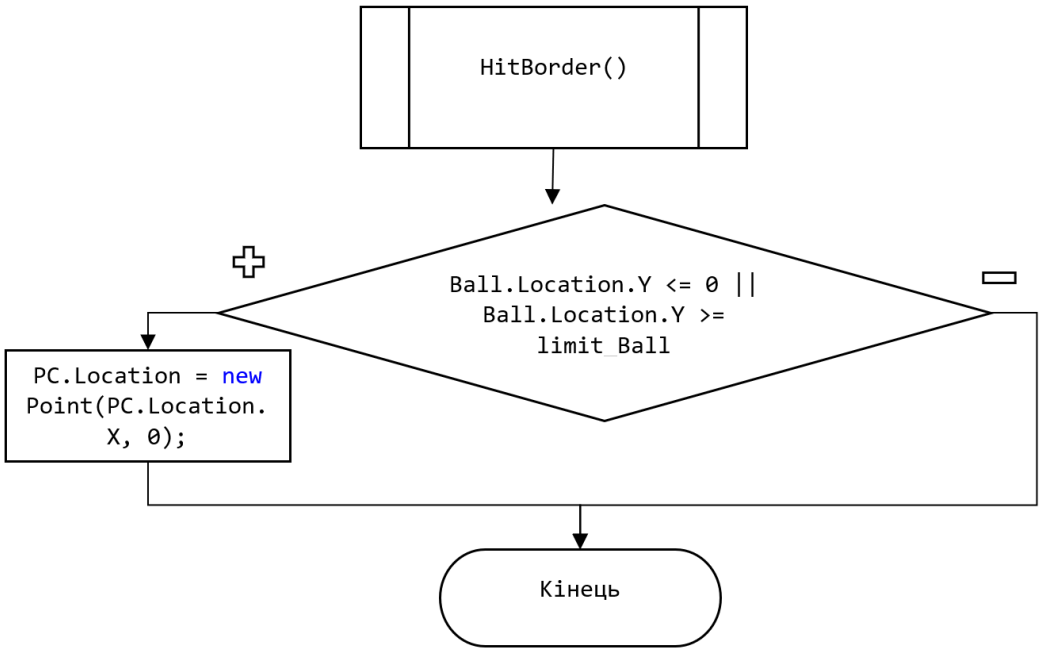


Рис 2.2. Блок-схема до алгоритму, шо не дає кульці вилетіти за межі форми і відбиває її від границь ігрового вікна

2.5. М'ячик вийшов за межі ігрового поля

Щоразу, коли якийсь з гравців не встигає відбити м’ячик, перевіряється, чи не набрав хтось із них 10 очок для початку нової гри і визначення переможця у цій. Якщо дана умова виконується, то викликається метод завершення гри, який анулює значення змінних з результатами, ставить платформи користувача та комп’ютера на вихідні позиції та чекає на натиснення кнопки початку гри користувачем. Якщо ж дана умова не була пройдена, то перевіряється, в якій частині поля знаходиться м’ячик після того, як один з гравців не встиг його відбити. Якщо у лівій відносно середини довжини форми, то комп’ютеру нараховується одне очко, якщо у правій – користувачу. М’ячик повертається на вихідну позицію – у центр поля.(див. Додаток В).

2.6. Висновки до другого розділу

У другому розділі курсової робити були продемонстровані основні методи програми, які відповідають за елементи управління (алгоритм руху платформи-комп’ютера) та методи, що відповідають за руху кульки по ігровому полю. Ці методи можна назвати основними у контексті отриманого завдання.

Розділ 3. Реалізація програмного продукту

3.1. Інтерфейс користувача

Користувацький інтерфейс фактично складається з двох прямокутних платформ, самої кульки і кнопки «Start Game», яка починає гру. До того, як користувач не натисне кнопку «Start Game», він не зможе почати гру.

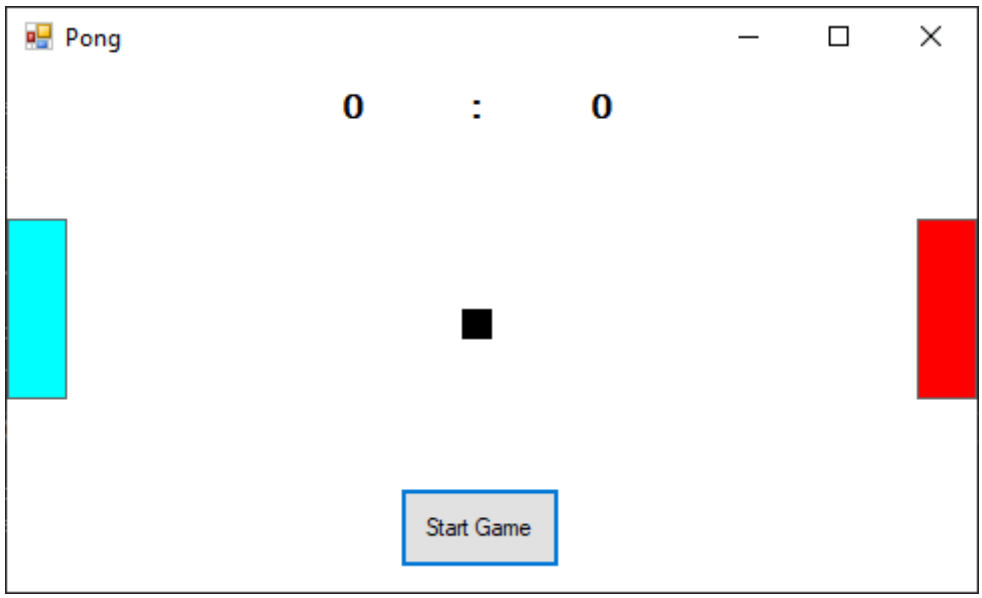


Рис 3.1. Скріншот ігрового вікна

3.2. Елементи управління грою

Спочатку задамо глобальні змінні. Їх складають постійні цілочисельні змінні, де limit_Pad – нижня границя для руху платформи гравця, щоб вона не виходила за межі вікна; limit_Ball – нижня границя для м’ячика перед відбиванням від стінки; звичайні цілочисельні змінні computer_won та player_won, в яких зберігається рахунок гравців; speed_Top та speed_Left, які керують напрямком та рухом кульки; булеві змінні up і down, які керуватимуть платформою користувача та game, яка не дає управляти платформою до початку гри; екземпляр r класу Random.

Спочатку була створена модель руху платформи користувача. Для реалізації цього завдання мені знадобилося два методи:

- Pressed – визначає, чи натиснута клавіша W/↑ або S/↑ за допомогою активації таймера, який пересуває платформу на 3 одиниці вгору/вниз;
- Released – визначає, коли була відпущена кнопка і зупиняє таймер.

Модель руху платформи, якою управляє комп'ютер є достатньо простою. Вона управляється таймером і складається з двох операторів if. Один з них обмежує його рух у границях форми, а інший завжди слідує за кулькою незалежно від того, чи вона нижче, чи вище середини платформи.

3.3. Рух м'ячика по ігровому полю

Перейдемо до алгоритму руху кульки. Спочатку необхідно задати значення напрямку руху кульки. Після старту кулька рухається до платформи користувача керуючись лише значенням speed_Left паралельно горизонтальним границям вікна. Для цього створено два методи: один задає значення змінних speed_Top та speed_Left, а інший змушує кульку рухатися по ігровому полю:

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

У методі NewPoint() описується план дій, коли одному з гравців не вдалося відбити кульку. Коли м'ячик виходить за одну з вертикальних меж, необхідно встановити його на місце по центру вікна і подача кульки у новому раунді надається гравцю, який її не відбив:

Наступним, що необхідно зробити є встановлення вертикальних та горизонтальних границь для м’ячика. Вертикально: коли кулька торкається верхньої границі форми, то вона відбивається у протилежному напрямку. Тому, ми маємо домножити значення змінної speed_Top на -1 і тоді напрямок кульки буде змінюватися незалежно від того, верхня це чи нижня границя форми:

```
private void NewPoint(int n)
{
    Ball.Location = new Point(x, y);
    speed_Top = 0;
    speed_Left = n;
}
```

Горизонтально ж роботи трохи більше. Основний метод BallLeftField включає у себе множину менших процедур, де спочатку перевіряється, чи не набрав один з гравців необхідну кількість очок, а потім перевіряючи координати м’ячика звертається до методів, які повернуть м’ячик до центра вікна, нададуть подачу м’ячика гравцю, який перед цим не встиг його відбити (NewPoint()) і, відповідно, додасть до рахунку одне очко (PlayerWon() або ComputerWon()):

```
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();
    }
}
```

Однією з найважливіших складових руху м’ячика є визначення місця, де він зіткнувся з платформою. Для цього платформи були поділені на 5 рівних

частин і записані в окремі методи, де перевіряються нижче вказані умови на істину чи хибу:

- Upper – місце зіткнення кульки з платформою вище або дорівнює різниці вершини платформи та висоти кульки і нижче або дорівнює сумі вершини платформи та висоти кульки;
- High – місце зіткнення кульки з платформою вище суми вершини платформи та висоти кульки і нижче або дорівнює сумі вершини платформи та двох висот кульки;
- Middle – місце зіткнення кульки з платформою вище суми вершини платформи та двох висот кульки і нижче або дорівнює сумі вершини платформи та трьох висот кульки;
- Low – місце зіткнення кульки з платформою вище суми вершини платформи та трьох висот кульки і нижче або дорівнює сумі вершини платформи та чотирьох висот кульки;
- Bot – місце зіткнення кульки з платформою вище суми вершини платформи та чотирьох висот кульки і нижче або дорівнює сумі вершини платформи та висоти кульки.

Також ми маємо знати, які координати передавати м’ячику в залежності від того, з платформою якого з гравців він зіткнувся. Спочатку напишемо функцію, яка повертатиме від’ємне значення випадкового числа. Потім в залежності від позиції м’ячика (вище чи нижче ширини форми), платформа буде відбивати м'ячик у зворотню сторону надаючи змінній speed_Left від’ємне чи додатнє значення:

```
private int Negative(int i, int n)
{
    int myval = r.Next(i, n) * -1;
    return myval;
}
```

```
private int AdjustCoordinates(int i, int n)
{
    int res = 0;

    if (Ball.Location.X < this.Width / 2)
    {
        res = r.Next(i, n);
    }
    else if (Ball.Location.X > this.Width / 2)
    {
        res = Negative(i, n);
    }
    return res;
}
```

У методі Collision() перевіряються булеві результати методів Upper, High, Middle, Low та Bot на «правдивість». Правильним буде лише один з них, без виключень. В залежності від результату м'ячику передаються координати руху і він направляється у ту чи іншу сторону.

Також під час тестування гри я помітив, що коли м'ячик вже мав би й покинути поле, але він відбивається від країв платформи, що суперечить логіці й концепту гри. Тому у наступному методі такий випадок виключається і м'ячик все одно покине ігрове поле:

```
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;
        }
    }
}
```

Після того, як усі функції та процедури вже готові, можна перейти до головного таймеру, який керує рухом м'ячика. Спочатку перевіряється, чи відбився з платформою користувача і задає відповідні координати для руху в

іншу сторону, а потім те ж саме, але вже для платформи, якою управляє комп’ютер. Після цього перевіряється чи зіткнувся м’ячик з горизонтальними краями форми і змінює значення змінної speed_Top або ж, якщо м’ячик покинув межі форми, ставить його на центр і подає його до певного гравця.

І заключною частиною цього розділу є кнопка «Start Game», натискання якої, як не дивно, запускає гру, ініціює певні змінні, змінює статус змінної game на true, що дозволяє почати гру запускає таймери і приховує саму кнопку з форми.

3.4. Висновки до третього розділу

Отже, у цьому розділі була детально описана й розглянута реалізація основних і допоміжних функцій програмного продукту та елементи інтерфейсу користувача. Програма може вважатися такою, що задовольняє поставлене завдання.

Висновки

Виходячи із завдання курсового проекту, вивчивши предметну область і виконавши аналіз поставленого завдання, була розроблена гра «Pong».

Дотримуючись постановки задачі, було спроектовано сам додаток.

Було проведено:

- аналіз концепту гри, її модифікацій, існуючих рішень;
- обирання і обґрунтування вибору алгоритмів;
- проектування відповідного графічного інтерфейсу;
- розробка та описання блок-схем до основних методів у лістингу програми;
- описання схем алгоритмів ключових її функцій;
- описання ключових методів.

Програмний продукт, отриманий в результаті виконання курсової роботи може служити у розважальних цілях і відповідає висунутим до нього вимогам, і, як наслідок, його можна використовувати за поставленим завданням.

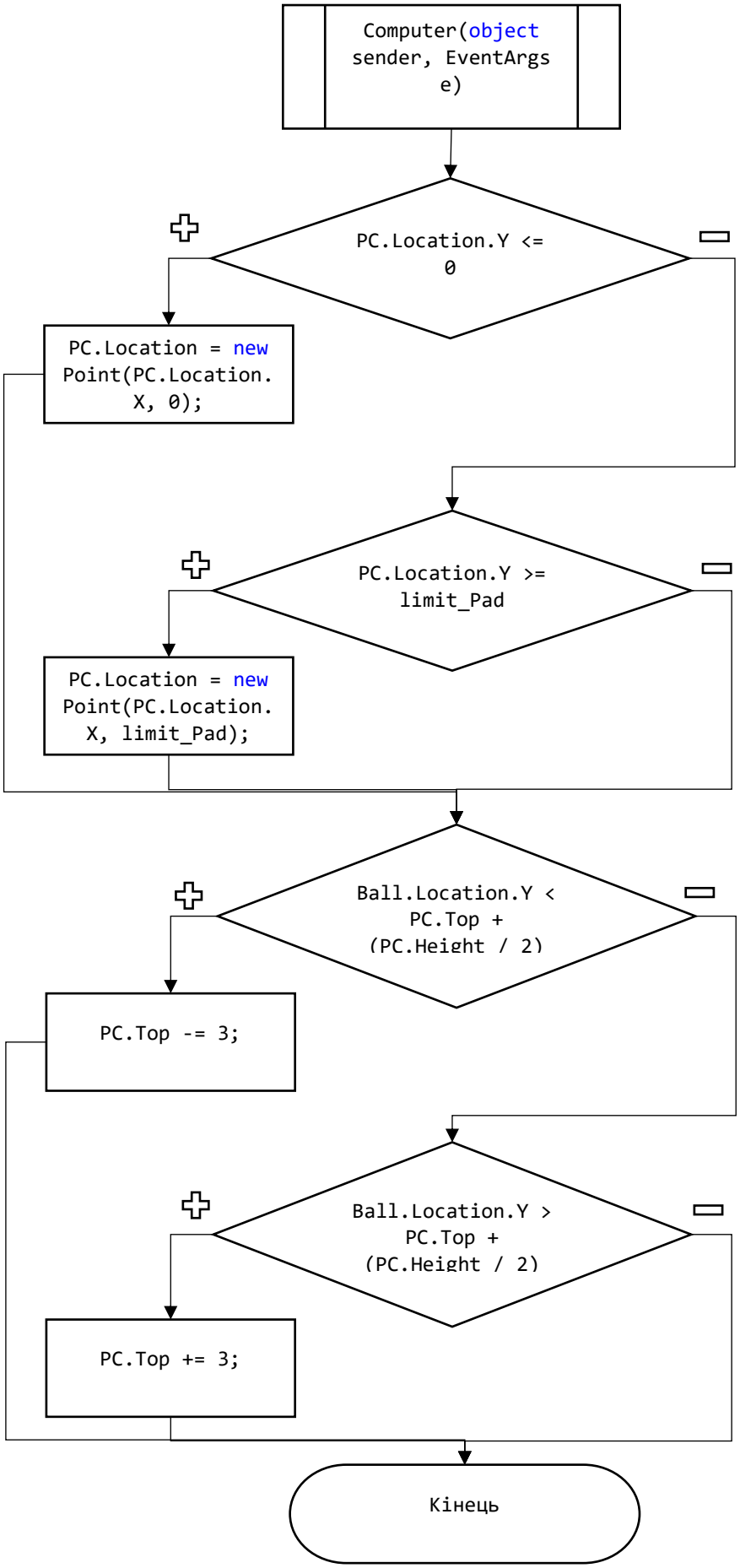
Подальші кроки з метою поліпшення додатку можна описати таким чином:

- додавання нових модифікацій гри;
- можливість переглянути рейтинг перемог та поразок;
- кастомізація користувацького середовища.

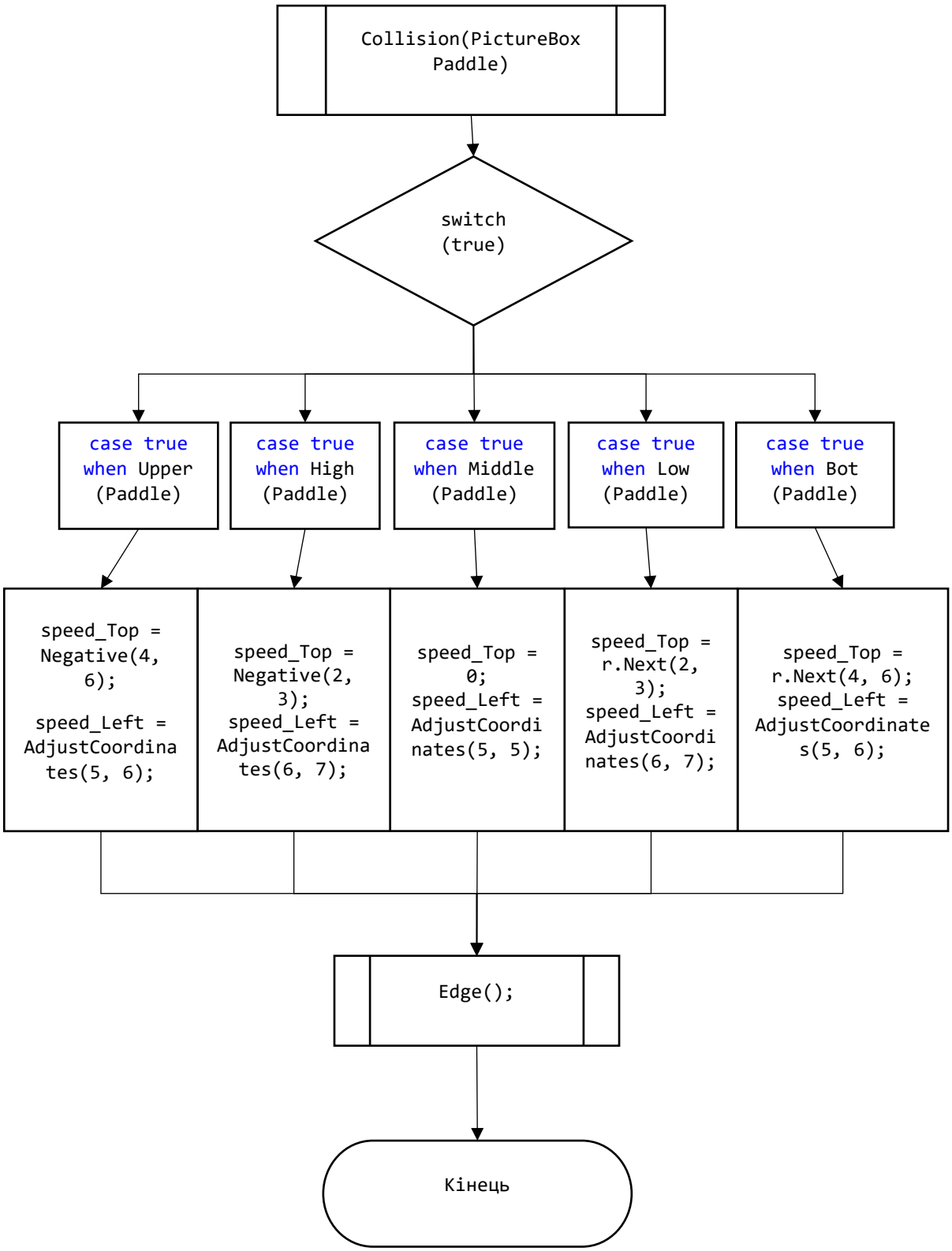
Список використаних джерел

1. Savidis Anthony. Universal Access in Ambient Intelligence Environments. 9th ERCIM Workshop on User Interfaces for All, Königswinter, Germany, September 27-28, 2006, Revised Papers [Текст] // Anthony Savidis, Apostolos Stamou, Constantine Stephanidis. – Springer-Verlag Berlin Heidelberg, 2007 – с. 405.
2. Young-Bum Kim. Multi-Player Virtual Ping-Pong Game [Електронний документ] // Young-Bum Kim, Seung-Hoon Han, Sun-Jeong Kim, Eun-Ju Kim, Chang-Geun Song, Режим доступу: https://core.ac.uk/download/pdf/194708676.pdf. Перевірено: 17.05.2020.
3. Гребенович Ю.Є. Методичні вказівки до виконання та оформлення курсової роботи з дисциплін «Основи програмування», «Програмування та алгоритмічні мови», «Алгоритмізація та програмування» для студентів, які навчаються за напрямами підготовки 050101 – «Комп'ютерні науки», 050103 – «Програмна інженерія» та 040303 – «Системний аналіз» усіх форм навчання [Текст] // Ю.Є. Гребенович, Онищенко Б.О., О.О. Супруненко. – Черкаси: Вид. від. ЧНУ імені Богдана Хмельницького, 2015. – 32 с.
4. Harris Ryan A. Pong: An Introduction to Implementing Computer Game Strategies [Електронний документ] // Ryan A. Harris, Jayesh B. Gorasia. Режим доступу: http://www.jgorasia.com/Files/Spring08/ICB/Gorasia_Harris.pdf. Перевірено: 17.05.2020.

Додаток А. Блок-схема алгоритму руху платформи-комп’ютера



Додаток Б. Блок-схема до методу Collision()



Додаток В. Блок-схема методу BallLeftField()