НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський Політехнічний Інститут імені Ігоря Сікорського»

Кафедра технічної кібернетики

Ігрова програма «Pong»

Курсова робота

3 дисципліни «Основи програмування»

Керівник Виконавець

Ас. Тимофєєва Ю. С. ст. Балась Костянтин Олександрович

зал. книжка № ІТ-8204

«Допущений до захисту» гр. ІТ-82

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(особистий підпис керівника) (особистий підпис виконавця)

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2019р. «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2019р.

Захищений з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оцінка)

Київ – 2019

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський Політехнічний Інститут імені Ігоря Сікорського»

Кафедра технічної кібернетики  
Дисципліна «Основи програмування»

Курс перший Група ІТ-82 Семестр другий

ЗАВДАННЯ

на курсову роботу студента

Балася Костянтина Олександровича

1. Тема роботи: Ігрова програма «Pong»

2. Строк здачі студентом закінченої роботи: 29.05.2019

3. Вихідні дані до роботи: мова програмування C#, середовище розробки Visual Studio 2017, операційна система Windows 10

4. Зміст пояснювальної записки (перелік питань, які підлягають розробці): реалізація можливості зміни складності гри, програмна реалізація алгоритму руху автоматизованого суперника, реалізація можливості отримати бонуси під час гри, реалізація привабливого і зрозумілого для користувача інтерфейсу.

5. Перелік графічного матеріалу: додатки

6. Дата видачі завдання: 06.05.2019

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Назва етапів виконання курсової роботи | Строк виконання етапів роботи | Підписи або примітки |
| 1. | Отримання та узгодження теми курсової роботи |  |  |
| 2. | Огляд існуючих рішень з тематики роботи |  |  |
| 3. | Розробка моделі та структури програми |  |  |
| 4. | Вибір алгоритмів розв'язання задачі |  |  |
| 5. | Розробка діалогового інтерфейсу програми |  |  |
| 6. | Кодування програми |  |  |
| 7. | Налагодження та перевірка програми |  |  |
| 8. | Оформлення пояснювальної записки |  |  |
| 9. | Захист курсової роботи |  |  |
| 10. |  |  |  |
| 11. |  |  |  |
| 12. |  |  |  |
| 13. |  |  |  |
| 14. |  |  |  |
| 15. |  |  |  |
| 16. |  |  |  |
| 17. |  |  |  |
| 18. |  |  |  |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_ Балась К. О. Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_ Тимофєєва Ю. С.

(підпис) (підпис)

АНОТАЦІЯ

В курсовій роботі розглядається питання створення ігрової програми «Pong». В ній реалізований простий та зрозумілий інтерфейс на основі текстових символів. Гра має три рівні складності. Також, розроблено систему корисних та шкідливих для користувача бонусів.

Програма реалізована мовою програмування високого рівня C# в середовищі Microsoft Visual Studio 2017 з використанням інтерфейсу програмування додатків Console Application. Під час реалізації було використано знання, набуті під час вивчення курсу основ програмування.

ЗМІСТ

ВСТУП5

1 ПОСТАНОВКА ЗАДАЧІ6

2 МАТЕМАТИЧНА МОДЕЛЬ ТА СТРУКТУРА ПРОГРАМИ 7

2.1 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ………………….................7

2.1.1 ОНЛАЙН СЕРВІС PONG GAME…………………………..………………..7

2.1.2 ОНЛАЙН СЕРВІС PONG-2………………………………………………….7

2.2 ПІДСУМКОВИЙ АНАЛІЗ РОЗГЛЯНУТИХ РІШЕНЬ……………………....8

2.3 МАТЕМАТИЧНА МОДЕЛЬ РУХУ М’ЯЧА………………………………….8

2.4 МАТЕМАТИЧНА МОДЕЛЬ АЛГОРИТМУ СУПРОТИВНИКА…………...8

2.5 РЕЗУЛЬТАТИ АНАЛІЗУ………………………………………………………9

3 ПЕРЕЛІК І ПРИЗНАЧЕННЯ РЕЖИМІВ ТА СТРУКТУРА ДІАЛОГУ11

4 СТРУКТУРА ДАНИХ ТА РЕСУРСІВ ПРОГРАМИ13

5 ОПИС ПРОГРАМИ15

5.1 МОЖЛИВОСТІ ПРОГРАМИ………………………………………………...16

5.2 ОПИС РОБОТИ ПРОГРАМИ……………………………………..…………16

6 ІНСТРУКЦІЯ ПРОГРАМІСТА19

6.1 УМОВИ ПРОЕКТУВАННЯ………………………………………………….19

6.2 КЕРІВНИЦТВО РОЗРОБНИКА ПРОГРАМИ………………………………19

6.3 РЕКОМЕНДАЦІЇ ЩОДО ПОДАЛЬШОЇ РОЗРОБКИ……………………...20

7 КЕРІВНИЦТВО КОРИСТУВАЧА21

7.1 УМОВИ ЗАСТОСУВАННЯ ПРОГРАМИ……………………………...……21

7.2 ХАРАКТЕРИСТИКА ПРОГРАМИ…………………………………………..21

7.3 ПОСЛІДОВНІСТЬ ДІЙ КОРИСТУВАННЯ ПРОГРАМОЮ……………….22

7.4 МОЖЛИВІ ПРОБЛЕМИ ПРОГРАМИ……………………………………….22

ВИСНОВКИ24

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ25

ДОДАТОК А26

ДОДАТОК Б48

ВСТУП

Останнім часом досить популярним стало ностальгувати за іграми минулого століття. Одною з найпопулярніших ігор була гра «Pong», тому розробка програми, яка дозволяє користувачу грати у теніс з ботом - актуальна задача для програмістів.

«Pong» — відеогра для аркадних ігрових автоматів, яка була розроблена японською компанією Taito Corporation і випущена компанією Atari 29 листопада 1972 року. «Pong» є найпростішим симулятором настільного тенісу. Невеликий квадратик позначає пінг-понговий м'ячик, що рухається по екрану за лінійною траєкторією. Якщо він ударяється об межу ігрового поля, або об одну з ракеток, то його траєкторія змінюється відповідно до кута зіткнення.

Ігровий процес полягає в тому, що гравці пересувають свої ракетки вертикально для захисту власних воріт – вертикальних бортів ігрового поля. Гравець отримує один бал, якщо йому вдається відправити м'ячик за лінію ракетки опонента.

В оригінальний «Pong» грає або одна людина проти комп'ютерного опонента, або вдвох, коли кожен гравець контролює свою ракетку.

Метою розробки є програмна реалізація гри «Pong» задля тренування і демонстрації навичок роботи з технологією Console Application.

Використані програмні засоби: Visual Studio 2017 (середовище розробки).

1. ПОСТАНОВКА ЗАДАЧІ

Завданням даної курсової роботи є розробка гри «Pong» з можливістю використання її користувачами з розважальною та розвиваючою здібності реакції ціллю.

Під час виконання курсової роботи планується створити програму, яка має виконувати такі задачі:

* реалізація можливості зміни складності гри;
* програмна реалізація алгоритму руху автоматизованого суперника;
* реалізація можливості отримати бонуси під час гри;
* реалізація привабливого і зрозумілого для користувача інтерфейсу.

1. МАТЕМАТИЧНА МОДЕЛЬ ТА СТРУКТУРА ПРОГРАМИ

2.1 Аналіз існуючих програмних рішень

В ході аналізу існуючих рішень, було виявлено безкоштовні аналоги реалізації програми у вигляді онлайн сервісів та готових програм в мережі Інтернет.

Оскільки велика кількість подібного контенту знаходиться у відкритому доступі в мережi Iнтернет, програма матиме замалий попит. Тому, аналізуючи програми такого типу, треба звернути увагу на такі критерії:

* зручність і естетичність інтерфейсу;
* особливі можливості і додаткові функції;
* гнучкість у системних вимогах;
* простота у користуванні.

2.1.1 Онлайн сервіс PongGame

«PongGame» - одна з найкращих реплік відомої гри (за пошуковою видачею Google). Переваги: подібність до оригінальної гри, велика кількість режимів складності. Недоліки: необхідність встановлення Adobe Flash Player для можливості грати.

2.1.2 Онлайн сервіс Pong-2

«Pong-2» - найкращий онлайн аналог даної гри. Переваги: простий та приємний інтерфейс, гра написана на JavaScript, що дозволяє їй бути мультиплатформовою; багато настоюваних параметрів гри. Недоліки: на великій швидкості м’яч іноді відхиляється від передбачуваної траєкторії, відсутні бонуси.

2.2 Підсумковій аналіз розглянутих рішень

На основі переваг та недоліків цих двох реалізацій гри було вирішено акцентувати процес розробки програми на алгоритмах руху м’яча та дизайні гри, а також реалізувати систему ігрових бонусів. Конкуренція, що є очевидною у цій ніші, потребує серйозного підходу до усього циклу розробки програми.

2.3 Математична модель алгоритму руху м’яча

Найважливішою математичною частиною алгоритму є процедура руху м’ячика із відбиванням від стінок. Ця процедура реалізована на основі логічних змінних та їх перевірки із кожним оновленням положення м’яча наступним чином:

Крок 1. Вибрати поточну клітинку;

Крок 2. Порівняти клітинку з параметрами стінки, що знаходиться на шляху;

Крок 3. Розрахувати наступну позицію м’яча відповідно до кута падіння;

Крок 4. Перейти до наступної клітинки;

Крок 5. Повторювати кроки 2-4 доки гра не завершиться.

Переваги алгоритму:

* проста реалізація;
* висока швидкість роботи;

Недоліки алгоритму:

* можливі порушення у роботі.

2.4 Математична модель алгоритму супротивника

Комп’ютерний супротивник – керівник другої платформи, що знаходиться на ігровому полі навпроти платформи користувача. Цей супротивник, фактично, є запрограмованим алгоритмом, ефективність якого має залежати від вибраного користувачем рівня складності.

Під час дослідження аналогів гри та матеріалів у відкритому доступі, не було знайдено достатньо точного опису такого алгоритму, тому даний алгоритм – розробка автора.

За результатами детального аналізу рухів супротивника під час гри в аналоги даної гри, було помічено, що супротивник рухається синхронно із кулькою (вгору та вниз), а також – що він починає рух, коли кулька починає рух у його сторону.

Отже, було вирішено зробити наступним чином: на легкому рівні складності алгоритм рухається лише коли кулька летить у бік суперника та пропускає проходження основного циклу випадковим чином. На нормальному рівні складності алгоритм не пропускає клітинок, а на складному – повністю повторює рухи м’яча, навіть коли той рухається у бік платформи користувача.

2.5 Результати аналізу

Блок схема – поширений тип схем (графічних моделей), які описують алгоритми та програми, в яких окремі кроки зображують у вигляді блоків різної форми, з’єднаних між собою лініями.

Схема програми – алгоритм роботи програми, поданий у вигляді блок-схеми. Вона допомагає краще зрозуміти роботу програми та зв’язок між усіма функціями проекту.

На основі аналізу існуючих програмних рішень з теми даної роботи було розроблено блок-схему роботи програми.



Рисунок 2.1 – Блок-схема роботи програми

1. ПЕРЕЛІК І ПРИЗНАЧЕННЯ РЕЖИМІВ ТА СТРУКТУРА ДІАЛОГУ

Перелік режимів роботи програми: користувач може обрати складність гри. Зі збільшенням складності збільшується швидкість м’яча та «інтелект» суперника.

Перелік елементів форми головного меню програми:

* «Play». Кнопка, яка відкриває ігрове поле програми.
* «Exit». Кнопка, яка закриває програму.
* «Mode». Кнопка, яка відкриває поле вибору складності гри.

Перелік елементів меню ігрової форми програми:

* Панель дозволяє графічно зобразити рахунок, ігрові платформи та м’яч
* Кнопки «↑», «↓» дозволяють здійснювати переміщення платформи.

Перелік елементів форми виходу з програми:

* «Exit». Кнопка, яка закриває програму.

У головному меню програми у першу чергу користувачу буде запропоновано розпочати гру або змінити складність. Можна завершити гру, вибравши кнопку «Exit». Кнопка «Play» розпочинає гру. З’являється ігрове поле, гравці. Під час гри, користувач може керувати платформою, якою він пересувається по полю, за допомогою кнопок «↑», «↓». Гра автоматично перейде до головного меню, коли користувач переможе або програє.

Під час гри працюють обробники виключень, а саме – методи, що не дозволяють м’ячу та платформам вийти за межі ігрового поля.

Ігровий рахунок показаний під ігровим полем. Гра продовжується до 10 очок. Перший гравець, що набрав 10 очок, виграє.

Окремо слід описати алгоритм комп’ютеризованого суперника. Залежно від складності, він пересувається вгору або вниз разом із рухом м’яча вгору або вниз відповідно, роблячи паузу кожні один, два та три ітерації основного циклу програми. Така поведінка відповідає складному, нормальному та легкому рівням. Також, із вірогідністю п’ятнадцять відсотків платформа суперника може випадковим чином рухатися на одну або дві клітинки вгору або вниз. За умови, що вона при цьому не вийде за межі поля.

Також під час гри з’являються бонуси – прямокутні ділянки поля, позначені відповідним чином. Якщо м’яч летить від платформи користувача і потрапляє у цю зону, платформа користувача набуває одного з трьох кольорів випадковим чином: жовто-блакитний, червоно-чорний або чорний. Останній колір є антибонусом, оскільки користувач перестає бачити свою чорну платформу на чорному фоні.

Бонуси скидаються після того, як одна зі сторін отримує очко.

1. СТРУКТУРА ДАНИХ ТА РЕСУРСІВ ПРОГРАМИ

Для роботи програми були розроблені методи Menu, Game, Skin, SetConsole, Greeting, PrintAtPosition.

Метод Greeting виводить різнокольорове привітання «Welcome. Ping-pong» при запуску гри.

Метод Menu – основний метод гри, що має параметр selected, який зберігає вибрану кнопку.

Метод SetConsole встановлює вікно консолі з певними параметрами.

Метод PrintAtPosition ставить курсор у задане положення та виводить заданий символ у консоль.

Метод Skin змінює зовнішній вигляд платформи користувача випадковим чином, коли той підбирає бонус.

Метод Game відповідає за процеси гри, рух м’яча та нарахування очок.

Також, було введено досить багато змінних, що використовуються для описання позицій меж поля, м’яча та ігрових платформ:

Змінні BallХ та BallY зберігають позицію м’яча на ігровому полі.

Змінні PrevBallХ та PrevBallY зберігають попередню позицію м’яча. Це необхідно для того, щоб видаляти символ, що позначає м’яч, з консолі при пересуванні на наступну позицію.

Логічні змінні balldirectionup та balldirectionright зберігають вектор руху м’яча по полю.

Змінні FirstPlayerХ та FirstPlayerY зберігають позицію платформи корисувача.

Змінні SecondPlayerХ та SecondPlayerY зберігають позицію платформи комп’ютеризованого суперника.

Змінні firstplayerscore та secondplayerscore зберігають рахунок користувача та суперника відповідно.

Змінна gamemode зберігає складність гри, за замовченням – нормальний.

Статичні змінні ConsoleWidth = 120, ConsoleHeight = 33 та LowerBarrierY = 24 зберігають величини, необхідні для відрисовки ігрового вікна та поля. Остання змінна зберігає координати нижнього бар’єра ігрового поля.

Також, було створено десять символьних масивів для зберігання цифр 0-9, шість символьних масивів для зберігання кнопок меню та два масиви для зберігання написів привітання.

Змінна rand є екземпляром системного Random(). Вона використовується для генерації псевдовипадкових чисел, що використовуються у процесі гри.

1. ОПИС ПРОГРАМИ

Програму було розроблено для приємного проведення часу за грою у одну з найпопулярніших консольних ігор. Ця гра – полегшений варіант класичної гри, бо не має сюжетного режиму, а режими складності полегшені. Маючи зрозумілий інтерфейс, незначні потреби в апаратних ресурсах, доступність і нескладний алгоритм гри, даний продукт може бути цікавим не тільки дорослим, а й дітям. Проект було розроблено здебільшого за допомогою функцій та класів, тож необхідно навести означення цих термінів.

Функція  у програмуванні – один з видів підпрограми. Особливість, що відрізняє її від іншого виду підпрограм – процедури – полягає в тому, що функція повертає значення, а її виклик може використатися в програмі як вираження.

Підпрограма – частина програми, яка реалізує певний алгоритм і дозволяє звернення до неї з різних частин загальної (головної) програми.

Підпрограма часто використовується для скорочення розмірів програм у тих завданнях, у процесі розв'язання яких необхідно виконати декілька разів однаковий алгоритм при різних значеннях параметрів. Оператори(команди), які реалізують відповідну підпрограму, записують один раз, а в необхідних місцях розміщують оператори передачі управління на цю підпрограму.

З погляду теорії систем, функція в програмуванні - окрема система (підсистема, підпрограма), на вхід якої надходять керуючі впливи у вигляді значень аргументів. На виході системи одержуємо результат виконання програми, що може бути як скалярною величиною, так і векторним значенням. По ходу виконання функції можуть виконуватися також деякі зміни в керованій системі, причому як оборотні, так і необоротні.

Клас в об'єктно-орієнтованому програмуванні — це спеціальна конструкція, яка використовується для групування пов'язаних змінних та функцій. При цьому, згідно з термінологією ООП, глобальні змінні класу (члени-змінні) називаються полями даних (також властивостями або атрибутами), а члени-функції називають методами класу. Створений та ініціалізований екземпляр класу називають об'єктом класу. На основі одного класу можна створити безліч об'єктів, що відрізнятимуться один від одного своїм станом (значеннями полів). Поля дозволяють вмістити дані про певний реальний об'єкт, а методи здійснювати обробку цих даних. При створенні ж на основі класу конкретного екземпляру, дані поля заповнюються конкретними даними. Обробкою цих даних можуть займатися відповідні методи.

5.1 Можливості програми

Програма має такі можливості:

* «швидкий» запуск (без встановлення);
* забезпечення максимального комфорту гри за допомогою привабливого інтерфейсу
* вибір рівня складності гри;
* отримання корисних та шкідливих бонусів під час гри;
* забезпечення максимальної відповідності даної гри оригінальній;

5.2 Опис роботи програми

Для роботи програми були розроблені методи Menu, Game, Skin, SetConsole, Greeting, PrintAtPosition.

Клас Program – основний клас програми, у якому реалізовано процес гри. Цей клас створюється за замовчуванням та реалізує метод Main(), який виконується у першу чергу та викликає усі інші методи програми. Також, клас Program реалізує метод SetInitialPoints (). Він відповідає за ініціалізацію початкового вікна програми, виставляє ширину, висоту вікна, скидає усі статичні змінні до величин за замовченням. У методі Main() реалізовано метод Menu(), у якому здійснюється увесь ігровий процес.

Метод Menu() містить змінні для зберігання вибраної кнопки меню, що перемикається натисненням кнопок «↑» та «↓». Також, метод викликає метод Game(), в якому відбувається процес гри, а також має підрозділ меню для вибору складності гри. Цей вибір записується у змінну gamemode. Далі відбувається гра, про це дещо детальніше.

Після натиснення кнопки Play метод Menu() викликає метод Game().

Даний метод заходить у while(true) цикл. Умови з параметром Console.KeyAvailable перевіряє, чи натискає користувач на клавіші руху платформи. Одночасно із цим здійснюється алгоритм руху комп’ютеризованої платформи та м’яча. Якщо м’яч торкається правої або лівої межі ігрового поля, протилежному гравцю зараховується очко, оновлюються змінні firstplayerscore та secondplayerscore. Метод PrintResult() показує очки гравців під нижньою стороною ігрового поля.

Окремо варто розглянути метод Skin(), що встановлює один із трьох кольорів платформи користувача у випадковому порядку. Даний метод викликається з методу Game(), коли м’яч рухається від платформи користувача та перетинається із зоною бонусного елемента.

Гру було вирішено вести до 10. Перший гравець, що набирає таку кількість очків, виграє гру. Цикл переривається через break.

Після завершення роздачі підраховуються очки гравців та виводиться повідомлення про виграш або програш. Потім гра повертається у головне меню.

Рисунок 5.1 – Меню програми

1. ІНСТРУКЦІЯ ПРОГРАМІСТА

Додаткове програмне забезпечення під час проектування програми не використовувалося.

Мова програмування C#. Середовище програмування Visual Studio 2017.

6.1 Умови проектування

Програма проектувалася на ПК з наступними характеристиками: процесор Intel Core i7, кількість оперативної пам’яті 8 ГБ. Операційною системою, яка підтримує Visual Studio є Windows. Було використано Windows 10, що значно поліпшувало роботу в деяких моментах. Для проектування програми знадобилися тільки типові периферійні пристрої такі як комп’ютерна миша, клавіатура та монітор.

* 1. Керівництво розробника програми

Для встановлення Visual Studio потрібне підключення до інтернету. Скачайте безкоштовну Community версію на офіційному сайті Microsoft. Потім слідуйте простій інструкції встановлення середовища програмування.

Спочатку слід створити макет головного меню, ігрове поле з усіма його елементами, та розділ виходу із гри.

Потім необхідно створити необхідні для цієї програми методи Menu, Game, Skin, SetConsole, Greeting, PrintAtPosition, описані вище і реалізувати усі необхідні функції на обробники подій для елементів створеного інтерфейсу Console Application.

В кінці роботи слід скомпілювати програму та зібрати виконавчий файл, який зможуть встановити користувачі.

* 1. Рекомендації щодо подальшої розробки

Програма є досить перспективною для розробки, тому бажано продовжувати її.

На основі розроблених алгоритмів руху м’яча та суперника, а також з використанням дизайну даної програми, можна створити мультиплатформовий додаток, оскільки саме на мобільних платформах ігри стають популярними і прибутковими.

Також, є можливість додати до проекту декілька інших ігор. Найдоречнішими з них могли б стати: змійка, тетріс тощо.

Також, не варто забувати про Unix-подібні системи: саме консольні ігри виглядали б дуже доречно на таких системах.

1. КЕРІВНИЦТВО КОРИСТУВАЧА

Назва ігрової програми — Pong. Ігрова програма створена з розважальною та розвиваючою здібності реакції метою.

7.1 Умови застосування програми

Мінімальні системні вимоги:

* процесор: із тактовою частотою 0.4 ГГц;
* оперативна пам’ять: 128МБ;
* мінімум 10 Мб пам’яті на жорсткому диску;
* графічний пристрій;
* операційна система Windows з встановленим Microsoft NET Framework 4.5;
* клавіатура.

Рекомендовані системні вимоги:

* процесор: із тактовою частотою понад 1 ГГц;
* оперативна пам’ять: 1 ГБ;
* мінімум 15 Мб пам’яті на жорсткому диску;
* операційна система Windows 10 (або новіша версія Windows) з встановленим Microsoft NET Framework 4.5;
* мишка або сенсорний маніпулятор і клавіатура.

7.2 Характеристики програми

Дана реалізація гри Pong має можливість початку гри із певною складністю та отримання рахунку після завершення гри. Гра має три рівні складності в залежності від швидкості польоту м’яча та досконалості суперника.

Для запуску програми слід запустити виконавчий файл.

Для видалення програми слід скористатися вбудованою «Панеллю керування» в ОС Windows.

7.3 Послідовність дій користування програмою

Після запуску програми слід вибрати бажану кнопку за допомогою стрілок та натиснути клавішу вводу. Можна завершити гру натиснувши кнопку «Exit». Щоб почати гру потрібно натиснути кнопку «Play». Під час гри, користувач може керувати платформою, якою він пересувається по полю, за допомогою кнопок «↑», «↓». Гра автоматично завершиться коли користувач виграє або програє.

Рисунок 7.1 – Ігрове поле з гравцями, м’ячем та бонусом

7.4 Можливі проблеми програми

Під час користування програмою, користувачу необхідно знати про деякі невиправні недоліки. Оскільки програма базується на досить недосконалій платформі Console Application, клік мишкою по ігровому полю може призупинити виконання програми. Щоб виправити це, слід натиснути будь-яку кнопку, що відповідає за управління у грі, або перезапустити додаток.

Також, не варто звертати уваги на жовту вертикальну полосу на ігровому вікні – це елемент прокрутки консолі, що неможливо видалити звідти. Приклад такої помилки можна побачити на рисунку 7.1.

Насамкінець, слід зауважити, що управління ігровою платформою може працювати некоректно, якщо тиснути на кнопки управління безперервно. Ця гра передбачає управління кліками, тобто клавіші слід досить швидко натискати, але не зажимати.

ВИСНОВКИ

Під час виконання курсової роботи було спроектовано та реалізовано версію гри Pong. Усі задачі виконано.

Створена програма має зрозумілий графічний інтерфейс користувача. Процес проектування дозволив використати знання та навички здобуті протягом курсу «[Основи програмування – 2»](http://wiki.kpi.ua/index.php/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20-%202.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%85%20%D0%B4%D0%B0%D0%BD%D0%B8%D1%85_(20002020)) і отримати та удосконалити нові знання з розробки методів, програмування на мові C# та її можливостей на платформi .NET.

В подальшому є перспектива розвитку такої програми, адже її можна удосконалити та розширити, додати нові функції, зробити можливим поширення на інших операційних системах, можливо навіть для Android та iOS, що є дуже важливим у епоху розвитку мобільних додатків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація Visual Studio.

URL: <https://msdn.microsoft.com>

1. Класичні алгоритми руху м’яча

URL: http://www.cyberforum.ru/windows-forms/thread988083.html

1. Ліхоузова Т.А. Методи розробки алгоритмів.

URL: https://drive.google.com/drive/folders/0B-IvU9NLczTMQTFXZ

1. Онлайн версія гри PongGame

URL: http://www.ponggame.org/

1. Онлайн-гра Pong-2

URL: https://pong-2.com

ДОДАТОК А

Текст програми

using System;

using System.Threading;

namespace Ping\_Pong

{

class Program

{

#region vars

static int BallX, BallY;

static int PrevBallX, PrevBallY;

static bool balldirectionup, balldirectionright;

static int FirstPlayerY = 0;

static int SecondPlayerY = 0;

static int firstplayerscore = 0;

static int secondplayerscore = 0;

static bool score = false;

static int SetInitPoints = 0;

static int gamemode = 2;

static int fps = 30;

static int firstplayerx = 0;

static int secondplayerx = 119;

static int ConsoleWidth = 120;

static int ConsoleHeight = 33;

static int LowerBarrierY = 24;

static int SkinX, SkinY;

static int SkinValue = 0;

static bool SkinOnField = false;

static Random randomGenerator = new Random();

#endregion

static void Main(string[] args)

{

SetConsole();

Greeting();

Menu();

}

#region misc

static void PrintAtPosition(int x, int y, char symbol)

{

Console.SetCursorPosition(x, y);

Console.Write(symbol);

}

static void SetConsole()

{

Console.CursorVisible = false;

Console.SetWindowSize(ConsoleWidth, ConsoleHeight);

Console.BufferHeight = Console.WindowHeight;

Console.BufferWidth = Console.WindowWidth;

}

static void SetBorder()

{

for (int x = 0; x < ConsoleWidth; x++)

{

PrintAtPosition(x, LowerBarrierY, '█');

}

}

static void SetInitialPoints()

{

switch (SetInitPoints)

{

case 0:

FirstPlayerY = 15;

SecondPlayerY = 15;

SetBall(59, randomGenerator.Next(10) + 10);

SetBorder();

break;

case 1:

FirstPlayerY = randomGenerator.Next(15) + 5;

SecondPlayerY = 15;

SetBall(2, FirstPlayerY + randomGenerator.Next(2) - 1);

SetBorder();

break;

case 2:

FirstPlayerY = 15;

SecondPlayerY = randomGenerator.Next(15) + 5;

SetBall(117, SecondPlayerY + randomGenerator.Next(2) - 1);

SetBorder();

break;

}

}

static void PrintResult()

{

int FirstPlayerScoreX = 54;

int FirstPlayerScoreY = 27;

int SecondPlayerScoreX = 62;

int SecondPlayerScoreY = 27;

PrintNum(FirstPlayerScoreX, FirstPlayerScoreY, firstplayerscore);

PrintAtPosition(FirstPlayerScoreX + 4, FirstPlayerScoreY + 1, '▀');

PrintAtPosition(FirstPlayerScoreX + 5, FirstPlayerScoreY + 1, '▀');

PrintAtPosition(FirstPlayerScoreX + 6, FirstPlayerScoreY + 1, '▀');

PrintNum(SecondPlayerScoreX, SecondPlayerScoreY, secondplayerscore);

}

static void ClearFieldNearPlayers()

{

for (int y = 0; y <= LowerBarrierY - 1; y++)

{

PrintAtPosition(firstplayerx + 2, y, ' ');

PrintAtPosition(secondplayerx - 2, y, ' ');

}

for (int i = firstplayerx + 2; i < ConsoleWidth - 3; i++)

{

PrintAtPosition(i, LowerBarrierY - 1, ' ');

}

}

static void PrintNum (int x, int y, int num)

{

switch (num)

{

case 0:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, zero[a, b]);

}

}

break;

case 1:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, one[a, b]);

}

}

break;

case 2:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, two[a, b]);

}

}

break;

case 3:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, three[a, b]);

}

}

break;

case 4:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, four[a, b]);

}

}

break;

case 5:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, five[a, b]);

}

}

break;

case 6:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, six[a, b]);

}

}

break;

case 7:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, seven[a, b]);

}

}

break;

case 8:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, eight[a, b]);

}

}

break;

case 9:

for (int a = 0; a <= 2; a++)

{

for (int b = 0; b <= 2; b++)

{

PrintAtPosition(x + b, y + a, nine[a, b]);

}

}

break;

}

}

static void Greeting()

{

for (int a = 0; a < 9; a++)

{

switch (a)

{

case 0:

Console.ForegroundColor = ConsoleColor.Blue;

break;

case 1:

Console.ForegroundColor = ConsoleColor.Cyan;

break;

case 2:

Console.ForegroundColor = ConsoleColor.DarkMagenta;

break;

case 3:

Console.ForegroundColor = ConsoleColor.Gray;

break;

case 4:

Console.ForegroundColor = ConsoleColor.Green;

break;

case 5:

Console.ForegroundColor = ConsoleColor.Magenta;

break;

case 6:

Console.ForegroundColor = ConsoleColor.Red;

break;

case 7:

Console.ForegroundColor = ConsoleColor.Yellow;

break;

case 8:

Console.ForegroundColor = ConsoleColor.White;

break;

}

for (int x = 0; x < 35; x++)

{

for (int y = 0; y < 3; y++)

{

PrintAtPosition(40 + x, 13 + y, welcome[y, x]);

}

}

Thread.Sleep(100);

if (a == 8)

{

Console.Clear();

}

}

for (int a = 0; a < 9; a++)

{

switch (a)

{

case 0:

Console.ForegroundColor = ConsoleColor.Blue;

break;

case 1:

Console.ForegroundColor = ConsoleColor.Cyan;

break;

case 2:

Console.ForegroundColor = ConsoleColor.DarkMagenta;

break;

case 3:

Console.ForegroundColor = ConsoleColor.Gray;

break;

case 4:

Console.ForegroundColor = ConsoleColor.Green;

break;

case 5:

Console.ForegroundColor = ConsoleColor.Magenta;

break;

case 6:

Console.ForegroundColor = ConsoleColor.Red;

break;

case 7:

Console.ForegroundColor = ConsoleColor.Yellow;

break;

case 8:

Console.ForegroundColor = ConsoleColor.White;

break;

}

for (int x = 0; x <= 51; x++)

{

for (int y = 0; y <= 6; y++)

{

PrintAtPosition(35 + x, 13 + y, Ping\_Pong[y, x]);

}

}

Thread.Sleep(100);

}

Console.ReadKey();

Console.Clear();

}

static void Menu()

{

int selected = 1;

Console.BackgroundColor = ConsoleColor.DarkYellow;

for (int x = 0; x <= ConsoleWidth - 1; x++)

{

for (int y = 0; y <= ConsoleHeight - 1; y++)

{

PrintAtPosition(x, y, ' ');

}

}

while (true)

{

if (selected == 1)

{

Console.ForegroundColor = ConsoleColor.DarkRed;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

}

for (int x = 20; x <= 100; x++)

{

for (int y = 2; y <= 10; y++)

{

PrintAtPosition(x, y, button\_play[y - 2, x - 20]);

}

}

if (selected == 2)

{

Console.ForegroundColor = ConsoleColor.DarkRed;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

}

for (int x = 20; x <= 100; x++)

{

for (int y = 12; y <= 20; y++)

{

PrintAtPosition(x, y, button\_mode[y - 12, x - 20]);

}

}

if (selected == 3)

{

Console.ForegroundColor = ConsoleColor.DarkRed;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

}

for (int x = 20; x <= 100; x++)

{

for (int y = 22; y <= 30; y++)

{

PrintAtPosition(x, y, button\_exit[y - 22, x - 20]);

}

}

if (Console.KeyAvailable)

{

ConsoleKeyInfo keyInfo = Console.ReadKey();

if ((keyInfo.Key == ConsoleKey.UpArrow || keyInfo.Key == ConsoleKey.W || keyInfo.Key == ConsoleKey.NumPad8) && selected > 1)

{

selected--;

}

if ((keyInfo.Key == ConsoleKey.DownArrow || keyInfo.Key == ConsoleKey.S || keyInfo.Key == ConsoleKey.NumPad2)&& selected < 3)

{

selected++;

}

if (keyInfo.Key == ConsoleKey.Enter && selected == 1)

{

Console.ResetColor();

Console.Clear();

Game();

Console.BackgroundColor = ConsoleColor.DarkYellow;

for (int x = 0; x <= ConsoleWidth - 1; x++)

{

for (int y = 0; y <= ConsoleHeight - 1; y++)

{

PrintAtPosition(x, y, ' ');

}

}

}

if (keyInfo.Key == ConsoleKey.Enter && selected == 2)

{

selected = 1;

Console.BackgroundColor = ConsoleColor.DarkYellow;

for (int x = 0; x <= ConsoleWidth - 1; x++)

{

for (int y = 0; y <= ConsoleHeight - 1; y++)

{

PrintAtPosition(x, y, ' ');

}

}

while (true)

{

if (selected == 1)

{

Console.ForegroundColor = ConsoleColor.DarkRed;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

}

for (int x = 20; x <= 100; x++)

{

for (int y = 2; y <= 10; y++)

{

PrintAtPosition(x, y, button\_easy[y - 2, x - 20]);

}

}

if (selected == 2)

{

Console.ForegroundColor = ConsoleColor.DarkRed;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

}

for (int x = 20; x <= 100; x++)

{

for (int y = 12; y <= 20; y++)

{

PrintAtPosition(x, y, button\_normal[y - 12, x - 20]);

}

}

if (selected == 3)

{

Console.ForegroundColor = ConsoleColor.DarkRed;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

}

for (int x = 20; x <= 100; x++)

{

for (int y = 22; y <= 30; y++)

{

PrintAtPosition(x, y, button\_hard[y - 22, x - 20]);

}

}

if (Console.KeyAvailable)

{

ConsoleKeyInfo keyInfo2 = Console.ReadKey();

if ((keyInfo2.Key == ConsoleKey.UpArrow || keyInfo2.Key == ConsoleKey.W || keyInfo2.Key == ConsoleKey.NumPad8) && selected > 1)

{

selected--;

}

if ((keyInfo2.Key == ConsoleKey.DownArrow || keyInfo2.Key == ConsoleKey.S || keyInfo2.Key == ConsoleKey.NumPad2) && selected < 3)

{

selected++;

}

if (keyInfo2.Key == ConsoleKey.Enter && selected == 1)

{

gamemode = 1;

for (int x = 0; x <= ConsoleWidth - 1; x++)

{

for (int y = 0; y <= ConsoleHeight - 1; y++)

{

PrintAtPosition(x, y, ' ');

}

}

selected = 1;

break;

}

if (keyInfo2.Key == ConsoleKey.Enter && selected == 2)

{

gamemode = 2;

for (int x = 0; x <= ConsoleWidth - 1; x++)

{

for (int y = 0; y <= ConsoleHeight - 1; y++)

{

PrintAtPosition(x, y, ' ');

}

}

selected = 1;

break;

}

if (keyInfo2.Key == ConsoleKey.Enter && selected == 3)

{

gamemode = 3;

for (int x = 0; x <= ConsoleWidth - 1; x++)

{

for (int y = 0; y <= ConsoleHeight - 1; y++)

{

PrintAtPosition(x, y, ' ');

}

}

selected = 1;

break;

}

}

}

}

if (keyInfo.Key == ConsoleKey.Enter && selected == 3)

{

EndGame();

}

}

}

}

static void Game()

{

for (int a = 0; a < 9;)

{

SetInitialPoints();

if (gamemode == 1)

{

fps = 45;

}

if (gamemode == 3)

{

fps = 25;

}

while (true)

{

DrawFirstPlayer(SkinValue);

DrawSecondPlayer();

DrawBall();

PrintResult();

Skin();

MoveBall();

int k = randomGenerator.Next(3);

if (gamemode == 1 && (k == 0))

{

SecondPlayerAIMove();

}

if (gamemode == 2 && (k == 1 || k == 2))

{

SecondPlayerAIMove();

}

if (gamemode == 3 && (k == 0 || k == 1 || k == 2))

{

SecondPlayerAIMove();

}

if (Console.KeyAvailable)

{

ConsoleKeyInfo keyInfo = Console.ReadKey();

if (keyInfo.Key == ConsoleKey.UpArrow || keyInfo.Key == ConsoleKey.W || keyInfo.Key == ConsoleKey.NumPad8)

{

MoveFirstPlayerUp();

}

if (keyInfo.Key == ConsoleKey.DownArrow || keyInfo.Key == ConsoleKey.S || keyInfo.Key == ConsoleKey.NumPad2)

{

MoveFirstPlayerDown();

}

}

if (score)

{

break;

}

ClearFieldNearPlayers();

Thread.Sleep(fps);

}

SkinOnField = false;

SkinValue = 0;

if (firstplayerscore > secondplayerscore)

{

a = firstplayerscore;

}

else

{

a = secondplayerscore;

}

score = false;

Console.Clear();

}

SetInitPoints = 0;

firstplayerscore = 0;

secondplayerscore = 0;

}

static void Skin()

{

int k = randomGenerator.Next(50);

if (k > 45 && !SkinOnField)

{

int a = randomGenerator.Next(4);

switch (a)

{

case 0:

SkinX = 40;

SkinY = 10;

break;

case 1:

SkinX = 80;

SkinY = 10;

break;

case 2:

SkinX = 40;

SkinY = 20;

break;

case 3:

SkinX = 80;

SkinY = 20;

break;

}

for (int i = 0; i <= 2; i++)

{

for (int j = 0; j <= 2; j++)

{

PrintAtPosition(SkinX + i - 1, SkinY + j - 1, skin\_block[j, i]);

}

}

SkinOnField = true;

}

if (BallX > SkinX - 2 && BallX < SkinX + 2 && BallY > SkinY - 2 && BallY < SkinY + 2)

{

SkinOnField = false;

for (int i = 0; i <= 2; i++)

{

for (int j = 0; j <= 2; j++)

{

PrintAtPosition(SkinX + i - 1, SkinY + j - 1, ' ');

}

}

SkinValue = randomGenerator.Next(2) + 1;

}

}

static void EndGame()

{

//TODO thx for playing

Environment.Exit(0);

}

#endregion

//all about first player (person)

#region FirstPlayer

static void DrawFirstPlayer(int SkinValue)

{

for (int y = FirstPlayerY - 2; y <= FirstPlayerY + 2; y++)

{

switch (SkinValue)

{

case 0:

Console.ForegroundColor = ConsoleColor.Green;

PrintAtPosition(firstplayerx, y, '█');

PrintAtPosition(firstplayerx + 1, y, '█');

Console.ResetColor();

break;

case 1:

Console.ForegroundColor = ConsoleColor.DarkBlue;

PrintAtPosition(firstplayerx, y, '█');

Console.ForegroundColor = ConsoleColor.DarkYellow;

PrintAtPosition(firstplayerx + 1, y, '█');

Console.ResetColor();

break;

case 2:

Console.ForegroundColor = ConsoleColor.Black;

PrintAtPosition(firstplayerx, y, '█');

PrintAtPosition(firstplayerx + 1, y, '█');

Console.ResetColor();

break;

}

}

}

static void MoveFirstPlayerDown()

{

if (FirstPlayerY <= 20)

{

FirstPlayerY++;

PrintAtPosition(0, FirstPlayerY - 3, ' ');

PrintAtPosition(1, FirstPlayerY - 3, ' ');

}

}

static void MoveFirstPlayerUp()

{

if (FirstPlayerY >= 3)

{

FirstPlayerY--;

PrintAtPosition(0, FirstPlayerY + 3, ' ');

PrintAtPosition(1, FirstPlayerY + 3, ' ');

}

}

#endregion

//all about second player (AI)

#region SecondPlayer

static void DrawSecondPlayer()

{

for (int y = SecondPlayerY - 2; y <= SecondPlayerY + 2; y++)

{

Console.ForegroundColor = ConsoleColor.Red;

PrintAtPosition(ConsoleWidth - 1, y, '█');

PrintAtPosition(118, y, '█');

Console.ResetColor();

}

}

static void SecondPlayerAIMove()

{

int MinXForMove = 0;

switch (gamemode)

{

case 1:

MinXForMove = 70;

break;

case 2:

MinXForMove = 40;

break;

case 3:

MinXForMove = 20;

break;

}

if (balldirectionup && SecondPlayerY >= 3 && BallX > MinXForMove && balldirectionright)

{

MoveSecondPlayerUp();

}

if (!balldirectionup && SecondPlayerY <= 20 && BallX > MinXForMove && balldirectionright)

{

MoveSecondPlayerDown();

}

}

static void MoveSecondPlayerUp()

{

SecondPlayerY--;

PrintAtPosition(ConsoleWidth - 1, SecondPlayerY + 3, ' ');

PrintAtPosition(118, SecondPlayerY + 3, ' ');

}

static void MoveSecondPlayerDown()

{

SecondPlayerY++;

PrintAtPosition(ConsoleWidth - 1, SecondPlayerY - 3, ' ');

PrintAtPosition(118, SecondPlayerY - 3, ' ');

}

#endregion

//all about ball

#region Ball

static void SetBall(int bx, int by)

{

BallX = bx;

BallY = by;

}

static void BallRandomDirection()

{

int b = randomGenerator.Next(2);

switch (b)

{

case 0:

balldirectionup = true;

break;

case 1:

balldirectionup = false;

break;

}

}

static void DrawBall()

{

PrintAtPosition(BallX, BallY, '█');

}

static void MoveBall()

{

//ball near higher border

if (BallY == 0)

{

balldirectionup = false;

}

//ball near lower border

if (BallY == LowerBarrierY - 1 && balldirectionright)

{

balldirectionup = true;

if (BallX < 100)

{

BallX += randomGenerator.Next(5);

}

}

if (BallY == LowerBarrierY - 1 && !balldirectionright)

{

balldirectionup = true;

if (BallX > 10)

{

BallX -= randomGenerator.Next(5);

}

}

//ball collides first player

if (BallX == firstplayerx + 2 && !balldirectionright && BallY <= FirstPlayerY + 2 && BallY >= FirstPlayerY - 2)

{

balldirectionright = true;

}

//ball collides second player

if (BallX == secondplayerx - 2 && balldirectionright && BallY <= SecondPlayerY + 2 && BallY >= SecondPlayerY - 2)

{

balldirectionright = false;

}

//ball collides left border - goal for second player

if (BallX == 0 && !balldirectionright)

{

secondplayerscore++;

balldirectionright = true;

SetInitPoints = 1;

score = true;

}

//ball collides right border - goal for first player

if (BallX == ConsoleWidth - 1 && balldirectionright)

{

firstplayerscore++;

balldirectionright = false;

SetInitPoints = 2;

score = true;

}

//remove previous ball

PrintAtPosition(PrevBallX, PrevBallY, ' ');

//move ball

if (balldirectionright)

{

PrevBallX = BallX;

BallX++;

}

else

{

PrevBallX = BallX;

BallX--;

}

if (balldirectionup)

{

PrevBallY = BallY;

BallY--;

}

else

{

PrevBallY = BallY;

BallY++;

}

}

#endregion

//needed char arrays

#region Chars

static char[,] welcome = new char[,] {{'█', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', '▀', '▀', ' ', '█', ' ', ' ', ' ', '█', '▀', '▀', ' ', '█', '▀', '█', ' ', '█', '█', ' ', ' ', ' ', '█', '█', ' ', '█', '▀', '▀'},

{' ', '█', ' ', '█', ' ', '█', ' ', ' ', '█', '▀', '▀', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', ' ', '█', ' ', '█', ' ', '█', ' ', '█', ' ', '█', '▀', '▀'},

{' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', '▄', '▄', ' ', '█', '▄', '▄', ' ', '█', '▄', '▄', ' ', '█', '▄', '█', ' ', '█', ' ', ' ', '█', ' ', ' ', '█', ' ', '█', '▄', '▄'} };

static char[,] Ping\_Pong = new char[,] { {'█', '▀', '▀', '▀', '▀', '▀', '▄', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▀', '▀', '▀', '▀', '▀', '▄', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},

{'█', ' ', ' ', ' ', ' ', ' ', '█', ' ', '▀', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},

{'█', '▄', '▄', '▄', '▄', '▄', '▀', ' ', '█', ' ', '█', '▄', '▀', '▀', '▄', ' ', '▄', '▀', '▀', '▀', '▄', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', '▄', '▄', '▀', ' ', '█', '▀', '█', ' ', '█', '▄', '▀', '▀', '▄', ' ', '▄', '▀', '▀', '▀', '▄'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', '▀', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', '▀', '▀', '▀', '▀', '▀', '▀', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', '█', '▀', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', '▀', '▄', '▄', '▄', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', '▀', '▄', '▄', '▄', '█'},

{' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '▄', '▄', '▀', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '▄', '▄', '▀'}};

static char[,] zero = new char[,] { {'█', '▀', '█'},

{'█', ' ', '█'},

{'█', '▄', '█'} };

static char[,] one = new char[,] { {' ', ' ', '█'},

{' ', ' ', '█'},

{' ', ' ', '█'} };

static char[,] two = new char[,] { {'█', '▀', '█'},

{'▄', '▄', '█'},

{'█', '▄', '▄'} };

static char[,] three = new char[,] { {'█', '▀', '█'},

{' ', '▀', '█'},

{'█', '▄', '█'} };

static char[,] four = new char[,] { {'█', ' ', '█'},

{'█', '▄', '█'},

{' ', ' ', '█'} };

static char[,] five = new char[,] { {'█', '▀', '█'},

{'█', '▄', '▄'},

{'▄', '▄', '█'} };

static char[,] six = new char[,] { {'█', '▀', '▀'},

{'█', '▄', '▄'},

{'█', '▄', '█'} };

static char[,] seven = new char[,] { {'█', '▀', '█'},

{' ', ' ', '█'},

{' ', ' ', '█'} };

static char[,] eight = new char[,] { {'█', '▀', '█'},

{'█', '▄', '█'},

{'█', '▄', '█'} };

static char[,] nine = new char[,] { {'█', '▀', '█'},

{'█', '▄', '█'},

{'▄', '▄', '█'} };

static char[,] button\_play = new char[,] { {'█', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▀', '▀', '▀', '▀', '▀', '▄', ' ', '█', ' ', ' ', ' ', ' ', '▄', '▀', '▀', '▀', '█', ' ', '█', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', '▄', '▄', '▀', ' ', '█', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', '█', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '█'}};

static char[,] button\_exit = new char[,] { {'█', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▀', '▀', '▀', '▀', '▀', ' ', '█', ' ', ' ', ' ', '█', ' ', '▀', '█', '▀', ' ', ' ', '█', '▀', '▀', '█', '▀', '▀', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', '▄', '▄', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', '▄', '▄', ' ', '█', ' ', ' ', ' ', '█', ' ', '▄', '█', '▄', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '█'}};

static char[,] button\_mode = new char[,] { {'█', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '█', ' ', ' ', ' ', ' ', ' ', '█', '█', ' ', '█', '▀', '▀', '█', ' ', '█', '▀', '▄', ' ', ' ', '█', '▀', '▀', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', '▀', '▀', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', '▄', '▄', '█', ' ', '█', '▄', '▀', ' ', ' ', '█', '▄', '▄', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '█'}};

static char[,] button\_easy = new char[,] { {'█', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▀', '▀', '▀', '▀', '▀', ' ', '▄', '▀', '▀', '▀', '█', ' ', '▄', '▀', '▀', '▀', '█', ' ', '█', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', '▄', '▄', ' ', '█', '▄', '▄', '▄', '█', ' ', '▀', '▄', '▄', '▄', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', '▄', '▄', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', '▄', '▄', '▄', '▀', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '█'}};

static char[,] button\_normal = new char[,] { {'█', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '█', ' ', ' ', ' ', '█', ' ', '█', '▀', '▀', '█', ' ', '█', '▀', '▀', '▄', ' ', '█', '█', ' ', ' ', ' ', ' ', ' ', '█', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', '▄', '▄', '▀', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', '█', ' ', '█', ' ', ' ', '█', ' ', '█', '▄', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', '█', '▄', '▄', '█', ' ', '█', ' ', '▀', '▄', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '█'} };

static char[,] button\_hard = new char[,] { {'█', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '▀', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', '▄', '▀', '▀', '▀', '█', ' ', '█', '▀', '▀', '▄', ' ', '█', '▀', '▄', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', '█', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', '▄', '▄', '▄', '▄', '█', ' ', '█', '▄', '▄', '▄', '█', ' ', '█', '▄', '▄', '▀', ' ', '█', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', '▄', ' ', ' ', ' ', '█', ' ', ' ', '█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█', ' ', ' ', ' ', ' ', '█', ' ', '█', ' ', ' ', ' ', '█', ' ', '█', ' ', '▀', '▄', ' ', '█', '▄', '▀', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '█'},

{'█', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '▄', '█'}};

static char[,] skin\_block = new char[,] { {'█', '▀', '█'},

{'█', ' ', '█'},

{'█', '▄', '█'} };

#endregion

}7

}

ДОДАТОК Б

Скріншоти роботи програми



Рисунок Б.1 – Не пощастило з бонусом – користувач не бачить своєї платформи