# ВСТУП

Гра Asteroids зроблена в найкращих традиціях класичних ігор 90-х, проте зі значно більш свіжим виглядом! Сбивайте астероїди, ухиляйтесь від уламків та ставте нові рекорди, літаючи на Тисячолітньому Соколі!

# 1 Опис роботи програми

Гра побудована на одному таймері. 30 разів у секунду екран відновлюється та перемальовується. Оскільки таймер один, то ігрова механіка прив’язана до цієї постійної кадрової частоти гри.

Після того, як гравець нажимає кнопку “Start Game”, в центрі екрана спавниться космічний корабель, за основу якого взято прототип «Тисячолітнього сокола» із всесвіту Зоряних Війн. Одночасно з цим при кожному перемальовуванні екрану може заспавнитися об’єкт астероїду (з вірогідністю в 5% на кожний кадр). Напрямок руху, початкова точка, швидкість руху, швидкість обертання навколо власної осі, напрямок обертання та текстура (одна з трьох доступних) – випадкові.

У разі попадання снаряду в астероїд він розпадається на випадкову кількість уламків (від 2 до 5) того ж типу, що й сам астероїд. Уламки розлітаються від місця розпаду астероїду в різні сторони та мають таку ж загрозу для космічного корабля як і великі цілі астероди. За збиття кораблем великого астероїду надається 100 очок, уламку – 50.

Між собою астероїди не зіштовхуються, тому що простір уявляється як тривимірний.

Після закінчення гри (розбиття корабля або натискання гравцем кнопки виходу до головного меню) текстовий файл рекордів переглядається, та новий рекорд (якщо такий є) записується відповідному гравцю.

# 2 РОЗРОБКА ПРОГРАМИ

2.1. Загальні відомості

Найменування програми “Asteroids”. Для функціонування програми необхідна операційна система Windows не старіша ніж Windows Vista. Для роботи з програмою необхідна комп’ютерна миша, клавіатура та монітор. Програма написана на мові програмування С++.

2.2. Функціональне призначення

Програма являє собою гру, що дозволить користувачеві приємно провести час та позмагатися з іншими гравцями у максимальному рекорді.

2.3 Опис логічної структури

Перелік елементів форми програми:

* Чотири кнопки головного меню дозволяють повноцінно керувати грою.
* Кнопка виходу та лічільник очок у вікні самої гри.
* Може використовуватися екземпляр класу QMessageBox для виводу різноманітних сповіщень та інформації на користувацький екран.

2.4 Використовувані технічні засоби

Програма проектувалася на ПК з наступними характеристиками: процесор Intel Core i3, кількість оперативної пам’яті 4 ГБ. Операційна система - Windows 10. Для проектування програми знадобилися тільки типові периферійні пристрої такі як комп’ютерна миш, клавіатура та монітор.

2.5 Виклик і завантаження

Для заватаженя програми необхідно завантажити інсталяційний файл - файл AsteroidsSetup.exe, запустити його та слідувати простій інструкції інсталяції.

2.6 Вхідні дані

Файл з гравцями та їх рекордами.

2.7 Вихідні дані

Файл з гравцями та їх рекордами.

# 3 КЕРІВНИЦТВО РОЗРОБНИКА

3.1 Вибір середовища для розробки

Для розробки слід використовувати мову програмування С++, середовище розробки Qt Creator, Для встановлення Qt Creator потрібне підключення до інтернету.

Скачайте безкоштовну Community версію на офіційному сайті Qt. Потім слідуйте простій інструкції встановлення середовища програмування. Після встановлення та налаштування середовища розробки необхідно створити проект типу Asteroids. Потім додати в проект необхідні класи. Далі треба створити необхідні функції та змінні для коректної роботи програми. Після закінчення написання структури і логіки потрібно скомпілювати і зібрати виконавчий файл за допомогою стандартного інтерфейсу Qt.

3.2 Структура функціональних можливостей програми

У ході роботи були розроблені наступні класи: Asteroid, Bullet, Button, doubleEntity, funcCol, Game, reloadBar, SpaceShip, staticEntity.

3.2.1 Клас Asteroid

Даний клас описує астероїди. Він може зберігати як цілий астероїд, так і уламок. Наслідується від doubleEntity.

Змінні та константи класу Asteroid:

|  |  |  |
| --- | --- | --- |
| Ідентифікатор (-и) | Тип змінної | Призначення |
| anglefly, angle | short | Кут, під яким летить астероїд від горизонту та кут повороту навколо власної осі. |
| angspeed | char | Швидкість обертання навколо власної осі. |
| type | unsigned char | Тип астероїду (максимум 3). |
| speed | unsigned char | Швидкість польоту астероїду. |

Функції:

* Конструктор – слугує для присвоєння полів за замовчуванням;
* isBig() – повертає true, якщо астероїд цілий (не є уламком);
* draw() – малює астероїд на ігровій мапі;
* move() – рух астероїду на необхідний шаг;
* turn() – поворот астероїду на необхідний кут;
* checkBUM() – перевірка на зіткнення із снарядом;
* getType() – повертає тип астероїду (1 – 3).
* Деструктор – звільняє пам’ять при видаленні об’єкту;

3.2.2 Клас Bullet

Даний клас описує снаряд, який випускає «Тисячолітній Сокіл». Наслідується від doubleEntity.

Змінні та константи класу Bullet:

|  |  |  |
| --- | --- | --- |
| Ідентифікатор (-и) | Тип змінної | Призначення |
| angle | short | Кут нахилу снаряду відносно горизонталі. |

Функції:

* draw() – функція, що малює снаряд;
* move() – рух снаряду на необхідну кількість пікселів;

3.2.3 Клас Button

Даний клас містить опис кнопки. Може описувати кнопку будь-якого розміру та з будь-якою текстурою. Наслідується від staticEntity.

|  |  |  |
| --- | --- | --- |
| Ідентифікатор (-и) | Тип змінної | Призначення |
| text | QString | Текст даної кнопки. |
| list | QSound\* | Вказівник на .wav файл звуку перегортання сторінок. |
| over | bool | Булева змінна, що позначає знаходження/не знаходження курсору над кнопкою. |
| Sizehor, sizevert, factsizevert | short | Розміри даної кнопки. |

Функції:

* isOver() – повертає true, якщо курсор знаходиться над кнопкою;
* draw() – малює кнопку на формі;

3.2.4 Клас doubleEntity

Даний клас містить базові змінні класу, в якому потрібні дробні координати.

|  |  |  |
| --- | --- | --- |
| Ідентифікатор (-и) | Тип змінної | Призначення |
| x, y | double | Дробні координати об’єкта. |
| img | QPixmap\* | Вказівник на текстуру об’єкта. |

Функції:

* getCoord() – повертає потрібну координату, приведену до цілого числа;
* draw() – малює елемент на формі;

3.2.5 Клас funcCol

Даний клас є статичним та містить методи для виконання часто потрібних дій.

Функції:

* turn() – повертає текстуру на задану кількість градусів;
* movestraight() – рух вказаного об’єкта під заданим кутом до горизонту;

3.2.6 Клас Game

Даний клас використовує шаблон програмування “Singleton” та описує основний клас ігри, в якому зосереджена основна логіка роботи програми.

|  |  |  |
| --- | --- | --- |
| Ідентифікатор (-и) | Тип змінної | Призначення |
| ship | SpaceShip | Об’єкт космічного корабля. |
| aster, smallaster, backgrounds, buts, backbut, spaceship, bul, rel | QPixmap\* | Вказівники на текстури гри. |
| step | unsigned short\* | Вказівник на змінну «мікростепу». |
| texts | QString[] | Масив текстів кнопок. |
| inGame | bool | Булева змінна уві грі / поза грою. |
| buttons | Button\*[] | Масив кнопок головного меню. |
| back | Button\* | Кнопка виходу у головне меню. |
| score | unsigned int | Змінна поточного рахунку. |
| directions | bool\* | Вказівник на 1 елемент масиву натиснутих клавіш. |
| curback | unsigned char\* | Поточний фон. |
| bullets | QList<Bullet\*> | Список снарядів. |
| asteroids | QList<Asteroid\*> | Список астероїдів. |
| names | QList<std::string> | Список імен гравців. |
| records | QList<unsigned int> | Список рекордів відповідних гравців. |
| crack, list | QSound\* | Вказівники на звуки. |
| line | QLineEdit\* | Лайнедіт для вводу імені гравця. |
| mes | QMessageBox | Екземпляр класу QMessageBox для виводу потрібної інформації на користувацький екран. |
| relBar | reloadBar\* | Вказівник на екземпляр класу reloadBar. |

Функції:

* startGame() – функція, що розпочинає гру;
* draw() – малює елементи класу на формі;
* backToMain() – функція виходу в головне меню гри;
* setDef() – задання дефолтних параметрів;
* checkCoord() – функція, що перевіряє координати руху курсора по формі;
* action() – функція, що ритмічно викликається таймером;
* shoot() – функція, що реалізує постріл із «Тисячолітнього Сокола»;
* setSounds() – завантаження до класу необхідних звуків із ресурсів програми;
* isInGame() – повертає булеву змінну inGame;

3.2.7 Клас reloadBar

Даний клас містить опис процесу перезаряджання. Наслідується від staticEntity.

|  |  |  |
| --- | --- | --- |
| Ідентифікатор (-и) | Тип змінної | Призначення |
| curstate | unsigned char | Поточна потужнність заряду. |

Функції:

* action() – функція, що ритмічно викликається таймером;
* draw() – малює стан перезаряджання на формі;
* shoot() – повертає true, якщо корабель готовий зробити постріл;

3.2.8 Клас SpaceShip

Даний клас містить опис космічного корабля.

|  |  |  |
| --- | --- | --- |
| Ідентифікатор (-и) | Тип змінної | Призначення |
| x, y | double | Координати. |
| img, bul | QPixmap\* | Текстури корабля та снаряду. |
| shootsound | QSound\* | Звук пострілу. |
| curimg | bool | Поточна текстура космічного корабля. |
| curangle | short | Поточний кут повороту відносно горизонту. |
| step | unsigned short\* | «Мікростеп». |

Функції:

* draw() – малює корабель на формі;
* setDef() – задання дефолтних параметрів;
* moveside() – поворот космічного судна;
* movestraight() – рух вперед/назад;
* getAngle() – повертає поточний кут повороту кораблю;
* getCoord() – повертає потрібну координату;

3.2.9 Клас staticEntity

Даний клас містить базові змінні для класів з цілочисленими координатами.

|  |  |  |
| --- | --- | --- |
| Ідентифікатор (-и) | Тип змінної | Призначення |
| x, y | int | Координати. |
| img | QPixmap\* | Текстура. |

Функції:

* draw() – малює елемент на формі;
* setDef() – задання дефолтних параметрів;

# 4 КЕРІВНИЦТВО КОРИСТУВАЧА

Для інсталяції програми слід запустити інсталяційний файл і слідкуючи за простою інструкцією інсталяції встановити програму

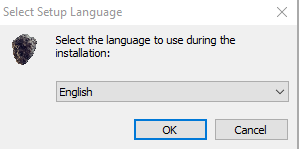


Рис. 4.1 – Виглід інсталяційного файлу

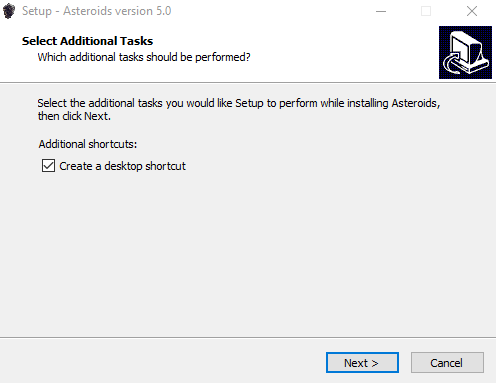


Рис. 4.2 – Процес інсталяції.

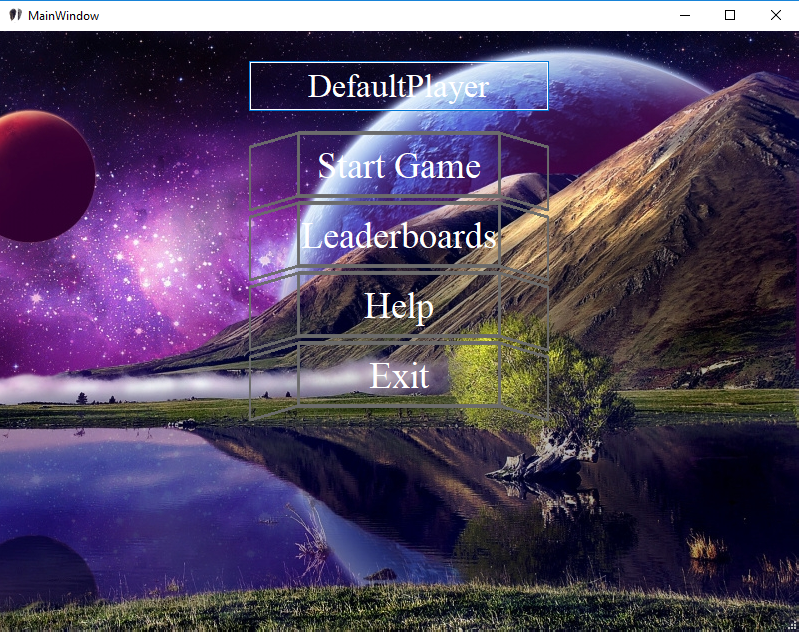


Рис. 4.3 – Меню програми

При запуску програми користувач переходить до головного меню гри, де розташовано чотири кнопки: “Start Game”, “Leaderboards”, “Help” та “Exit”.

При натисканні на кнопку “Start Game” ми починаємо гру, “Leaderboards” виводить список рекордів, “Help” виводить інформацію про гру та “Exit” виходить із гри.

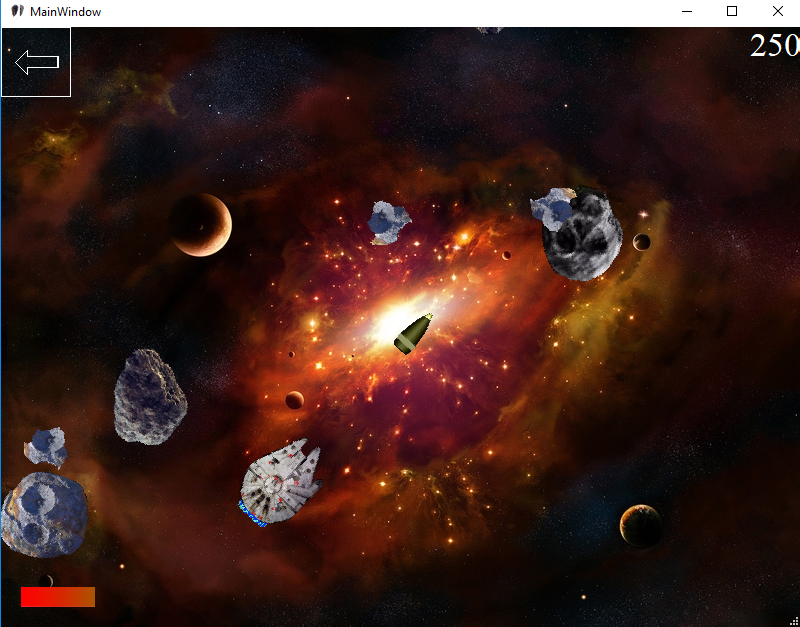


Рис. 4.4 – Ігровий процес.

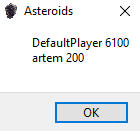


Рис. 4.5 – Таблиця рекордів

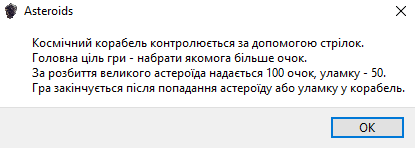


Рис. 4.6 – Допомога

# 5 ОТРИМАНІ РЕЗУЛЬТАТИ

У ході перевірок виявилося, що гра коректно працює в будь-яких умовах, не займає зайвої оперативної пам’яті та в розпакованому вигляді на диску важить лише 56 МБ (разом з усіма бібліотеками Qt).

# ВИСНОВОК

Написання даної ігрової програми змусило мене глибше вникнути в тему ООП, вивчити деякі фундаментальні принципи програмування та відкрило мені дорогу далі на шляху становлення програмістом. Також програма використовує звукові ефекти, що було для мене досить в новинку. І звісно ж, фундаментальне розуміння сенсу синуса/косинуса теж ще нікому не заважало.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація Qt.

http://doc.qt.io/

2. Форум cyberforum.

<http://www.cyberforum.ru>

# ДОДАТОК

UML діаграма класів

