Національний університет «Києво-Могилянська академія»

**ЛАБОРАТОРНА РОБОТА № 2**

Роботу виконали:

Янкін Ігор Сергійович  
студент І року навчання НаУКМА  
Факультет Інформатики  
Спеціальність: інженерія програмного забезпечення

Гінкул Анна Олександрівна  
студент І року навчання НаУКМА  
Факультет Інформатики  
Спеціальність: інженерія програмного забезпечення

**Постановка задачі**

Написати класичну аркадну гру на Java «Breakout».

У программі мають бути:

* Різнокольорові циглинки;
* Ракетка, яка реагує на рух миші користувача;
* М’ячик, який спочатку летить під випадковим кутом та відбивається від цеглинок, ракетки та меж грального світу;
* Повідомлення, яке оголошує перемогу або поразку.

**Розподіл Задачі**

Янкін Ігор:

* Створення цеглин
* Створення м’яча
* Пошук зіткнень м’яча та цеглин
* Створення «особливих» цеглин
* Створення логіки ігри
* Встановлення швидкості м’яча

Гінкул Анна:

* Створення рактеки
* Пошук зіткнень зі стінами
* Визначення стіни, якої доторкнувся м’яч
* Створення меню гри
* Створення фона гри
* Створення допоміжної панелі

**Структура програми**

Програма складається з основного класу (Main), який відповідає за запуск програми та послідовне виконання методів, забезпечуючих працездатність гри. та допоміжних класов об’єктів (GPaddle, Brick, Ball, Menu, ScoreTable), які відповідають за реалізацію нових об’єктів, необхідних під час виконання програми.

**Опис основних методів та класів**

Класс Ball відповідає за створення об'єкту м'яча з полями CentreX та CentreY, у яких зберігається центральні координати м’яча.

Класс Brick відповідає за створення об’екту цеглини за ініціалізацію усіх цеглин (метод createBricks()).

Класс GPaddle відповідає за створення об’єкту рокетки.

Класс ScoreTable відповідає за створення допоміжного вікна, у якому вказується поточний рівень гри, кількість набраних очок та кількість залишившихся життів.

Класс Main відповідає за запуск гри та викликання потрібних методів відповідно до логіки та структури гри.

* setSpeed() - відповідає за випадкове генерування швидкості м’яча.
* setup() - відповідає за початкову ініціалізацію світа для гри.
* mouseMoved() відповідає за рух ракетки відповідно до руху курсора
* jumpSide() відповідає за визначення, якого боку цеглини доторкнувся м’яч
* checkWalls() - відповідає за відштовхування від стін.
* wallNumber() - відповідає за перевірку, якої самої стіни доторкнувся м’яч.
* getCollidingObject - повертає об’єкт, якого торкнувся м’яч, якщо такий існує

**Інструкція Користувача**

**Проблеми під час розробки**

* неправильне визначення моменту зіткнення об’єктів. Рішення – добавлення 28 додаткових точок для перевірки.
* Прилипання м’яча до рактеки під час зіткнення. Рішення – введення нової змінної, яка визначає можливість повторного зіткнення за ракеткою.

**Програмний код**