Міністерство освіти і науки України

Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Програмування інтелектуальних інформаційних систем

**ЗВІТ**

до лабораторних робіт

| **Виконав**  **студент** |  | ІТ-92 Кобець Ірина Олександрівна |  |  |
| --- | --- | --- | --- | --- |
|  |  | (№ групи, прізвище, ім’я, по батькові ) |  |  |
|  |  |  |  |  |
| **Прийняв** |  | ас. Очеретяний О. К. |  |  |
|  |  | (посада, прізвище, ім’я, по батькові ) |  |  |

Київ 2021

# **Завдання лабораторної роботи**

Завдання лабораторної роботи булу створити базову частину гри Pacman для подальшого її удосконалення алгоритмами.

# **Опис використаних технологій**

Для написання гри використовувала мову програмування Python та бібліотеку pygame, що є потрібною для написання комп’ютерних ігор та мультимедійних додатків.

# **Опис програмного коду**

Весь програмний код складається з 3 класів: App (додаток, гра в цілому), Player (гравець), Enemy (ворог).

Функції класів та їх основні функції:

**App** (у даному класі знаходиться основна зв’язуюча частина):

* ініціалізуються всі локальні змінні та задаються необхідні параметри у функції **\_\_init\_\_()**
* іде процес гри та оприділяється її стан у функції **run()**
* зчитується поле (лабіринт гри, що береться із текстового файлу) у функції **load()**
* задаються вороги: **set\_enemies()** та сам гравець
* а також описуються команди, що відбуваються при початку гри і те, що саме відображається: **start(), start\_draw()**
* команди, що визначають поведінку під час гри, а саме: **play()** визначає поведінку при натисканні на кнопки управління героєм, **play\_update()** оновлює дані гравця та ворогів та визначає чи потрапив гравець на ворога, чи ні, **play\_draw()** малює поле гри на екрані та всі необхідні деталі
* також функції, що спрацьовують при виграші або програші гравця: **game\_win(), game\_over(),** ...

Також є допоміжний файл, що містить усі дефолтні налаштування для гри, а тобто кольори, різні стилістичні параметри та розміри, кількості рядків/стовпчиків.

**Player** відповідає за ініціалізацію та поведінку гравця

* У функції **init** задаються базові параметри, швидкість, позиція.
* **update()** - оновлює гравця, оновлює позицію, вказує на поведінку при попаданні на монету (**on\_coin()**) або при переході у loop (тунель) (**on\_loop()**)
* у таких функціях як **can\_move(), time\_to\_move()** оприділяється можливість руху, який блокується стінами лабіринту
* у функції **eat\_coin()** описується процес зникнення монетки та нарахування балів
* а функція **draw()** відображає значок гравця на екрані, підгружає картинку пакмена та задає їй положення, відображення горизонтально відповідно до напрямку героя
* **get\_xy()** відповідає за піксельні параметри ширини та довжини місцезнаходження героя на полі

**Enemy** відповідає за дію ворога, що є схожою на таку як і у гравця, проте менш розширеною із автоматично заданою поведінкою та напрямком (що задається у функції **update()**)

# **Скріншоти роботи програмного застосунку**



Рис.1. Початковий екран гри.

При натисканні на ‘SPACE’ користувач запускає гру:

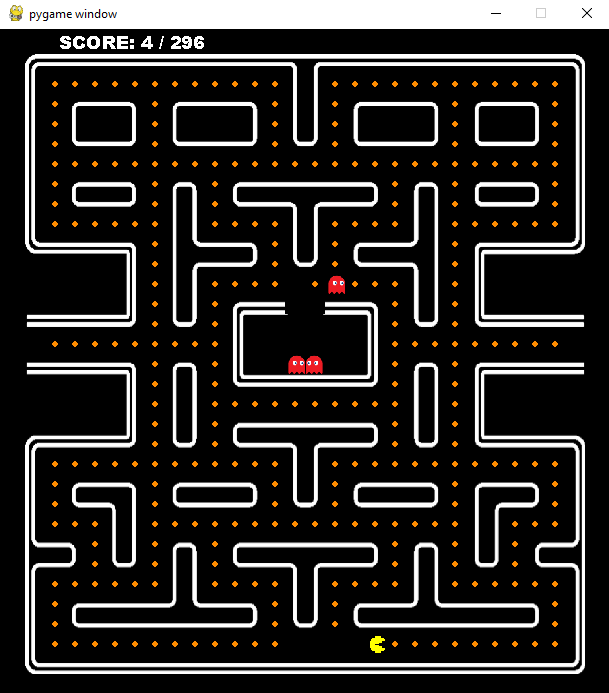


Рис.2. Процес гри.

Гра одразу починається і герой починає рухатись. При потраплянні на монетки, герой отримує бали. При набиранні максимальної кількості (що зазначена у лівому верхньому кутку) гра закінчується, гравець виграє, проте при потраплянні на ворогів, що є червоними привидами на малюнку, гравець програє.

При виграші чи програші користувач має змогу зіграти ще раз натиснувши необхідну клавішу або вийти.

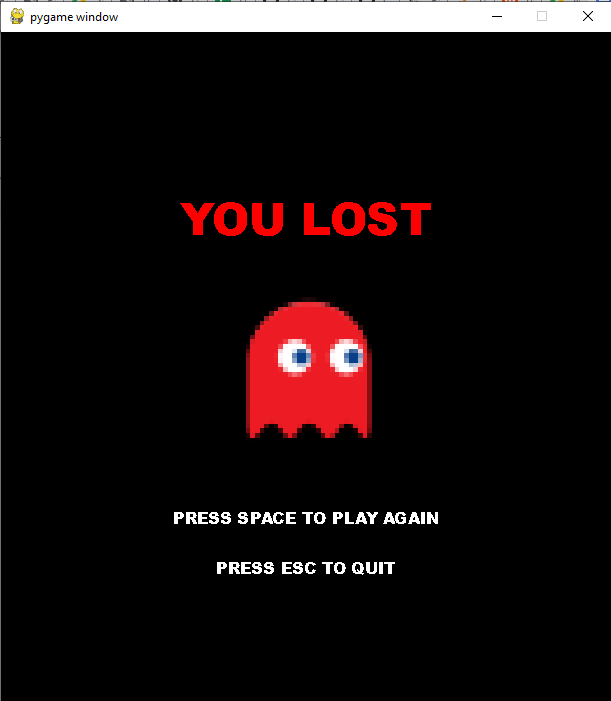


Рис 3-4. Кінець гри.