TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

🞠◊🞠◊🞠

****

**Report**

**Subject: IT PROJECT**

DESIGN A SIMPLE GAME

**Class: S2\_PROJ215879E\_07CLC**

**Lecturer name: Hoang Van Dung**

**Semester: 3– School year: 2022-2023**

**TP.HỒ CHÍ MINH – Month 12/Year 2022 List of members**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student name** | **Contribution (%)** |
| 20110428 | Nguyễn Sỹ Khải | **100%** |
| 20110497 | Phạm Hoàng Huy | **100%** |

**Score:**

**Comment:**

**Sign**

# Acknowledgment

We would like to appreciate Mr. Hoang Van Dung who guided us through the project, made sure that we finished our work on time. He gave us such value suggestions, helped us improve our product, solve some difficulties, understand the issues and make it more complete which was otherwise result in a lot of failures. Our project could not turn out great without his guidance.

This project is made with in five – six weeks. Due to the lack of knowledge as well as the work we divided for each week is not optimal, result in many trails and errors, which is inevitable. Despite all of those, our project still made major progress. We are looking forward to receiving all the comments of teacher to help us improve our limited knowledge.

Sincerely thanks.

# Abbreviation list

UI: User Interface

GUI: Graphical User Interface

**Table of content**

[**Acknowledgment 2**](#_Toc89673767)

[**Abbreviation list 2**](#_Toc89673768)

[**Figure list 3**](#_Toc89673769)

[**Table list 3**](#_Toc89673770)

[**I. Project description 5**](#_Toc89673771)

[**1. Objectives 5**](#_Toc89673772)

[**2. User benefits 5**](#_Toc89673773)

[**II. Task Assignment 6**](#_Toc89673774)

[**III. Design 7**](#_Toc89673775)

[**1. Designing the source code 7**](#_Toc89673776)

[**2. Designing the UI 9**](#_Toc89673777)

[**IV. Test cases 11**](#_Toc89673778)

[**V. Conclusion 12**](#_Toc89673779)

[**1. Evaluation 12**](#_Toc89673780)

[**2. Difficulties 12**](#_Toc89673781)

[**3. Advantages 12**](#_Toc89673782)

[**4. Disadvantages 12**](#_Toc89673783)

[**5. Development ideas 12**](#_Toc89673784)

1. **Project description**
2. ***Objectives***

We have created a simple game. To help people relax after stressful working hours. Along side with that is being able to apply the knowledge learned in school and thereby know where your strength is. And this is also a stepping stone for us to complete bigger projects.

1. ***User benefits***

As a user, gaming is really a workout for your mind disguised as fun. Studies have shown that playing video games regularly may increase gray matter in the brain and boost brain connectivity. (Gray matter is associated with muscle control, memories, perception, and spatial navigation.). SpaceShooter have some of the perks such as:

* Colorful and eye-catching UI
* Easy to play
* For all ages
* Rogue like game for multiple playability

1. ***Technology***

Python is a high-level programming language that serves many different programming purposes. This is a solid stepping stone when entering the world of programming, although it has been developed for more than thirty years, but Python at the present time is still very popular at the present time thanks to its simplicity, usefulness, and strong support. Python is easy to read and understand, doesn't require huge programming skills and is suitable for a wide range of audiences, in addition to having a lot of support and libraries for the users themselves. That's why Python is so popular and most of the organizations and companies that require recruitment at the moment need to be able to use Python in their work.

As an open source language, the software we can use to write Python such as: IDEs (Anaconda/spyder; Pycharm...), Editors (VIM; Visual Code...), Browsers (Jypyter, Notebook...). The use of Visual Studio community is encouraged. Download and install the Python version, there are many versions available but it is recommended to install the Python 3.6 version because of its existing, up-to-date libraries.



*Basic principles:*

• Variables: somewhat similar to C / C++ such as containing no spaces, special characters, not starting with numbers and being case sensitive. Also variables in Python do not require declaration. When naming variables, it needs to be meaningful so that when looking back or the number of lines of code is a lot, we do not get confused.

• Without opening curly braces, Python relies on spaces to align user commands.

• Regarding the If, While or for statements, the form is like the C/C++ side.

• Array in Python can contain many values such as numbers, letters ...

• Range function: create a larger and longer array

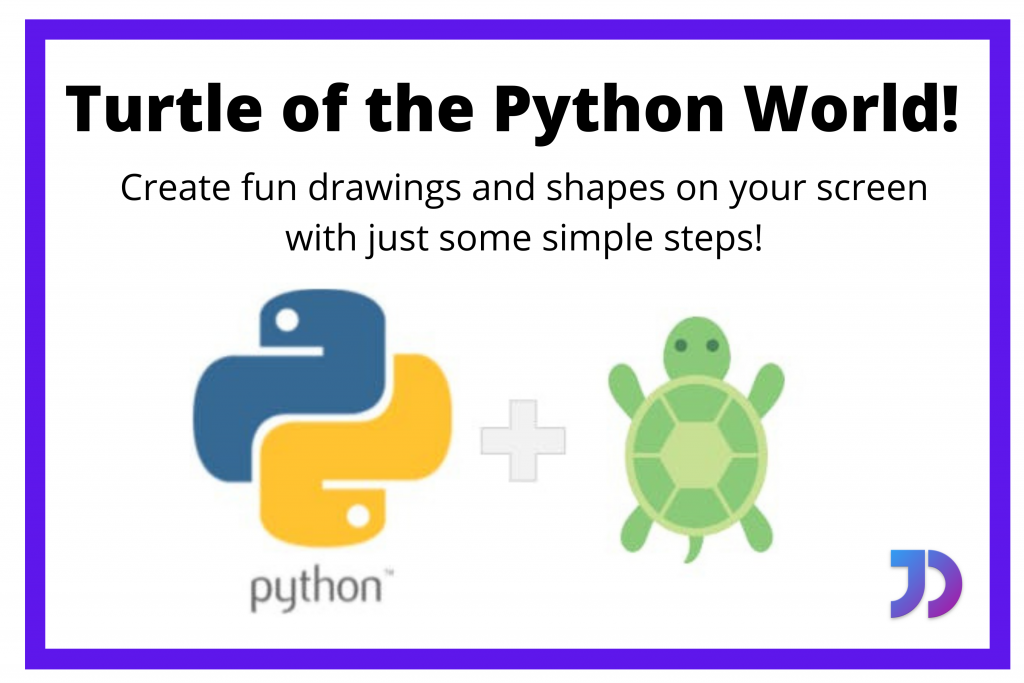
• To support Python commands need to use Python Debug Interactive.

Packages: equivalent to the concept of Library, is a library of pre-written functions for user needs, there are more than 130,000 packages that exist on the net, some Packages learn such as: Numpy (will better support List than List). ), matplotlib (supports plotting), Pandas (reads and manipulates data from psv fils). To use the library just simply import the library and use it. To install the library, go to the CMD of the computer and use the command pip install <library name>, the machine will automatically install the library.

The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, width, and on/off state (also called down and up).

The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it, setting its color, or setting its width. A student could understand (and predict and reason about) the turtle's motion by imagining what they would do if they were the turtle. Seymour Papert called this "body syntonic" reasoning.

A full turtle graphics system requires control flow, procedures, and recursion: many turtle drawing programs fall short. From these building blocks one can build more complex shapes like squares, triangles, circles and other composite figures. The idea of turtle graphics, for example is useful in a Lindenmayer system for generating fractals.

Turtle geometry is also sometimes used in graphics environments as an alternative to a strictly coordinate-addressed graphics system.

1. **Task Assignment**

*Table 1 : Task Assignment*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Student’s Name | Task work | Evaluate contribution |
| 20110428 | Nguyễn Sỹ Khải | Leader + Designer | 100% |
| 20110497 | Phạm Hoàng Huy | Report | 100% |

1. **Design**
2. ***Designing the source code***

With the game in mind, this game is programmed in Python language using the design tool Turtle. The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, width, and on/off state (also called down and up).

The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it, setting its color, or setting its width. With these tool, we can make some core feature of the game.

Here are the code :

*Table 2 : Source code*

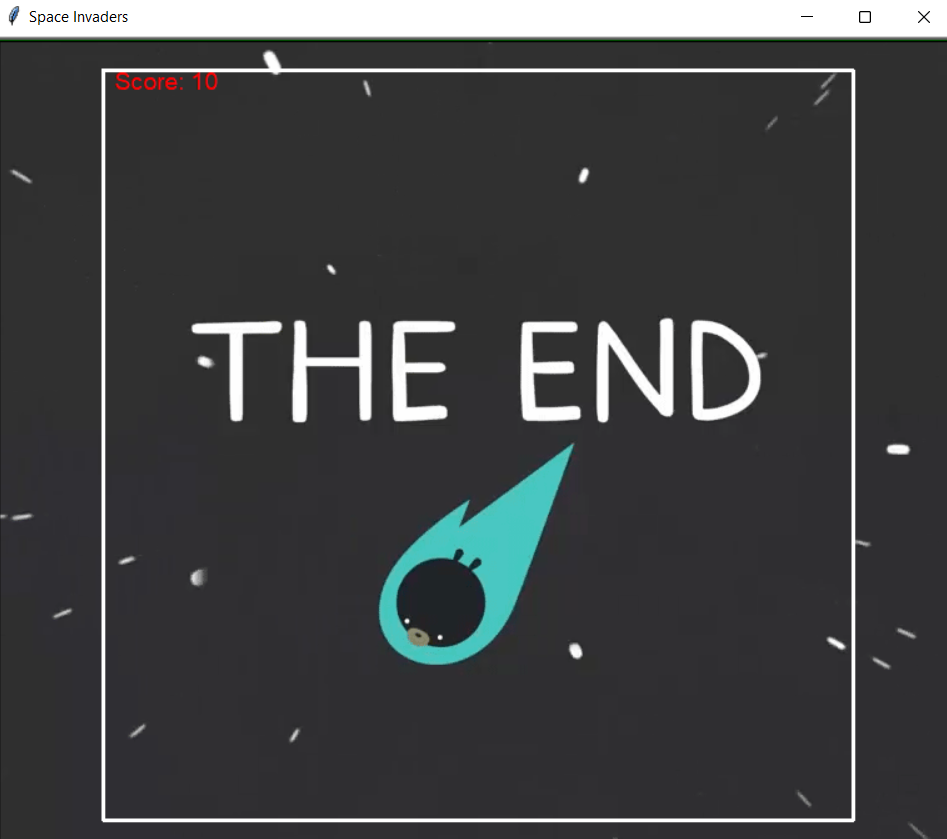
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Method | Purpose | File name, Line | Responsible |
| 1 | **window = turtle.Screen()** | Create a new screen or return to an existing one | SpaceShooter.py(7) | **Nguyễn Sỹ Khải** |
| 2 | **turtle.register\_shape()** | Register the shape for the turtle to draw | SpaceShooter.py(13,14) | **Nguyễn Sỹ Khải** |
| 3 | **border\_pen=turtle.Turtle()** | Draw the border | SpaceShooter.py(17) | **Nguyễn Sỹ Khải** |
| 4 | **border\_pen.hideturtle()** | Hide the turtle | SpaceShooter.py(27) | **Nguyễn Sỹ Khải** |
| 5 | **score\_pen = turtle.Turtle()** | Draw the score board | SpaceShooter.py(33) | **Nguyễn Sỹ Khải** |
| 6 | **player = turtle.Turtle()** | Create a player turtle | SpaceShooter.py(43) | **Nguyễn Sỹ Khải** |
| 7 | **enemies = []** | Create a list of enemy | SpaceShooter.py(55) | **Nguyễn Sỹ Khải** |
| 8 | **for i in range(number\_of\_enemies):**  **enemies.append(turtle.Turtle())** | Add enemy to the list | SpaceShooter.py(58-60) | **Nguyễn Sỹ Khải** |
| 9 | **for enemy in enemies:** | A loop to draw the enemy in the invader.gif | SpaceShooter.py(62) | **Nguyễn Sỹ Khải** |
| 10 | **bullet = turtle.Turtle()** | Create the player’s bullet | SpaceShooter.py(73) | **Nguyễn Sỹ Khải** |
| 11 | **bulletstate = "ready"** | Define bullet state | SpaceShooter.py(87) | **Nguyễn Sỹ Khải** |
| 12 | **def move\_left():** | Move the player to the left | SpaceShooter.py(91) | **Nguyễn Sỹ Khải** |
| 13 | **def move\_right():** | Move the player to the right | SpaceShooter.py(99) | **Nguyễn Sỹ Khải** |
| 14 | **def fire\_bullet():** | Fire the bullet from player | SpaceShooter.py(107) | **Nguyễn Sỹ Khải** |
| 15 | **def isCollision\_enemy\_bullet(t1, t2):** | Check for collision between enemy and bullet | SpaceShooter.py(120) | **Nguyễn Sỹ Khải** |
| 16 | **def isCollision\_enemy\_player(t1, t2):** | Check for collision between enemy and player | SpaceShooter.py() | **Nguyễn Sỹ Khải** |
| 17 | **turtle.listen()** | Create keyboard bindings | SpaceShooter.py(142) | **Nguyễn Sỹ Khải** |
| 18 | **turtle.onkey()** | Set the keyboard bindings up | SpaceShooter.py(143) | **Nguyễn Sỹ Khải** |

List of methods of the support class

1. ***Designing the UI***

We have used a colorful and yet simple UI to make it easy to reach the users. The cosmic background plus the spaceship makes the game more exciting, and the scoreboard creates a sense of achievement.

Picture 1:UI main game



Picture 2:Game over UI

1. **Test cases**

*Table 2 : Test cases*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test cases | Purpose | Brief Explanation |
| 1 | **Test case 1:**  Input:  Left arrow and right arrow | The player move the right position | The code for moving the player is correct |
| 2 | **Test case 2:**  Input:  Spacebar | The player doesn’t do anything | The code for shooting is missing a guide for the turtle to draw the bullet |
| 3 | **Test case 3:**  Input:  Spacebar | The bullet doesn’t eliminate the enemy | The code checking the collision of the bullet and the enemy is incorrect. |
| 4 | **Test case 4:**  **Input:**  Contact to enemy | The game suddenly end without the ending screen. | Missing an ending component to end the game. |

1. **Conclusion**
2. ***Evaluation***

* Almost requirements are met
* The code is clean and reusable
* Doing what it supposed to do

1. ***Difficulties***

* Learning new technology is a problem because it slows down the project progress

1. ***Advantages***

* Quite clean code
* Meets the requirements of the project
* Simple GUI, users easy to use this application
* Reuse, recycling, and maintainability

1. ***Disadvantages***

* May have some bugs in the process that we did not know about
* May take a large portion of memory
* Missing some exciting feature like power up

1. ***Development ideas***

* We can have more practical information and more option to add
* Upgrade the GUI for better display
* Make it more difficult by adding new features

**Github link:** [**https://github.com/20110428/Space-Shooter**](https://github.com/20110428/Space-Shooter)