Hyperiondev

TASK

Capstone Project II

Visit our website

Introduction

WELCOME TO THE SECOND CAPSTONE PROJECT!

This Capstone is a milestone in your learning so far. In this project, you will be consolidating the knowledge that you have gained and applying it to create a game! This Capstone project will allow you to demonstrate your competence in using variables, various data types, if statements, loops and nested structures. Remember, it is worth putting some extra time and effort into this project — it can become part of your developer portfolio.

Get in touch

Connect for support

Remember that with our courses, you're not alone! You can contact your mentor to get support on any aspect of your course.

The best way to get help is to login to **www.hyperiondev.com/portal** to start a chat with your mentor. You can also schedule a call or get support via email.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

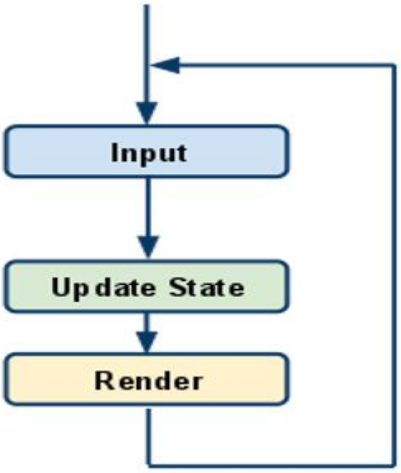
A note from the

HyperionDev Team

Programming games is a great way to learn how to code. Look at any game and think to yourself: How can I do that? What statements or loops will I need? What data types and structures would I use?

For example, using control structures, you could perhaps make a turn-based strategy game. What about games like Sudoku or Hangman even?

The workings of a game can be represented as follows:



The above diagram represents a 'game loop'. This loop is executed many times a second. Inside the loop, the game logic is executed and the screen is updated and rendered accordingly.

Even if you do not have all the programming tools at your disposal yet, it is often useful to prime the mind to think in this way.

Using the pygame module is one way to make games in Python. Visit the **pygame website** to find out more information about pygame.

DEVELOPER PORTFOLIO

Developers who have the edge are those who find ways to apply their newfound skills from the get-go. A **developer portfolio** (a collection of software that you have made) allows you to demonstrate your skills rather than just telling people about them. It's a way of bringing your CV to life and introducing yourself to the world. As you learn more skills and put these into practice, each project that you complete will become more efficient and eye-catching.

This application series offers you the means to create the first project of your very own developer portfolio, allowing you to walk away from this course not only with a certificate but, more importantly, with a headstart into your career!

THE TASK AT HAND

In this Capstone Project, you will be creating a simple game using Pygame. Pygame is a library of code that allows one to create simple games using Python.

A key skill for any software developer is being able to apply what you know to new technologies and environments. In this project, you will be required to demonstrate that you can teach yourself to adapt to working with code libraries that are new to you. This will require a little bit of research and extra effort. Don't worry! You will be provided with a code example of how to create a game. This example contains detailed comments that explain the example. Also, remember that your mentor is always there to help you if you get stuck.

This project will also test your ability to understand and modify code that someone else has written. This is another vital skill for any software developer!

Before you begin:

A key focus of this project will be ensuring that your code is correct and adheres to style guides. In this regard, make sure that you do the following before submitting your work:

1. Make sure that you have identified and removed all syntax, runtime and logical errors from your code.

2. Make sure that your code is readable. To ensure this, add comments to your code, use descriptive variable names and make good use of whitespace and indentation.

3. Look through **this style guide** for best-practice guidelines regarding how your Python code should be written.

4. Use a tool such as **PyLint** to check your code.

Compulsory Task

In this project, you will be creating a simple game. Follow these steps:

- You will need to download pygame. See instructions for this here: **https://www.pygame.org/wiki/GettingStarted**

- Open the **example.py** file in the **game** folder. **example.py** contains a very simple game coded with pygame. The player can move up and down with the arrow keys and must avoid colliding with the 'enemy' to win. Read through the code and comments and run the game to see how to create a game with pygame.

- Create your own game based on the code in **example.py** (you do not have to write your game from scratch). Your game should include:
    - One 'player' object. You should be able to move the player around the screen by using the up, down, left and right arrows.
    - At least 3 'enemy' objects. The enemies should all move across the screen from different positions. If the player object collides with any enemy object, they lose, and the game ends.
    - At least 1 'prize' object. If the player object collides with the prize object, they win, and the game ends.

- Feel free to use the images in this folder or your own images. Remember to respect copyright laws if you choose to use images from the web.

- For more information about creating a game with pygame, see here: **https://www.pygame.org/docs/tut/MakeGames.html**

Thing(s) to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python or Notepad++** may not be installed correctly.
2. Before submitting your task, make sure that your code runs without any errors and that you have followed the guidelines in the 'Before you begin' section of this document.

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we've done a good job?

Click here to share your thoughts anonymously.