Report 8

CS 152 - B

Yazan Bawaqna

Abstract:

The goal of this project is to further use Zelle graphics and more complex python structures to simulate and visualize an object moving between different blocks and getting knocked back. The first part of the project outlines the basics of dictionaries and inheritance, which are fundamental concepts in python and in programming in general. To explain, a dictionary in python is simply a set carrying keys with each key carrying a value. For instance, in the project, I have created a dictionary containing some countries as keys and the user has to input the capitals as values for those keys. Dictionaries are of great significance as they allow for an easy building of the relation between the key and the value. The other essential concept to mention is “inheritance”. In Python, we can create objects using classes. A class of an object just carries the characteristics of that object. Imagine a cow class, for instance, its characteristics would be its height, weight, color, etc. From here, we can create a child class and name in “calf” for example. The child class can inherit the characteristics of the parent class (color, species) but can also have its own characteristics (gender and height).

The project starts with creating classes. In this project, I created a ball, triangle, block, and star. All of those classes are child classes of the “thing” class, which carries attributes that can be given to any shape (Color, height, velocity, etc.) Following that, I created a collision main function that visualizes the shapes and their interactions. I will give the user control over one of the items to make the video more interactive and entertaining.

The project offers a wide usage of computational thinking commands, including “if statements” and “nested if statements”. The way they work is by offering a condition for the program to run, for instance, if it is a collision between the ball and the block, then run this, else run this. Moreover, the use of the random library offers real-life simulations, as it gives different simulations for the same collision to account for unaccounted random errors.

Solution to tasks:

The first real struggle I encounter was with defining class set position. Though I could have copied it from the last project, I wanted to understand how it actually functions. First, I knew I need to get the change in the displacement (dx, dy), so I was able to figure out this part. Then I needed to implement the move, the issue is that I forgot about the built-in “move” method. However, after I recheck my previous project, I was able to recoil.

Another issue that took considerable effort was drawing the star object. I have used the ball object as a parent class and only overridden the reshape function. Simply, I used an if statement that would undraw the previous object if already drawn, so I can redraw the ball into a circle. Lastly, following the previous examples, I have used the vis list and tried to visualize how to draw a star. It did not turn out perfect, yet it is still considerably effective.

The most challenging part was the last task requiring recording a video of an interactive game or film. The way I approached this challenge is by using a main function. This will allow my function to be run all the time using if name == main and running the terminal. Next, since I am using Zelle graphics, I knew that I need a window to visualize my graphics. Simply after I defined my objects “the ball” and “the block” by connecting them to the ball in the physical objects file.

My next step which took me time to figure was to set the attributes. I have made the mistake of confusing the get and set functions due to their complementary functions. However, eventually, Ellen helped me correct the mistake and I changed the function into a setter function.

The difficult part came next, I used checkMouse() and getMouse() to control the object using the letters “a” and “d”. If the user clicked on “a” for instance, the set position function will be called which will reset the position of the ball by decreasing it vertically by 2 points. The issue I faced with this method is that I can lose the ball outside the window if the user kept clicking upward or downward. For this case, I used an if statement to reset the position of the ball to the center of the window if the ball got outside.

Reflection:

What was really fascinating about the project is how easy is it to level up the complexity of the project by importing outside-sourced libraries. Although ignorant as I may seem. I never comprehended the volume of outside libraries we can import into our code. While I was doing the project, I went to check for some online libraries out of curiosity. To my surprise, I came to know that there are endless options of codes to be reused at all levels of complexity. The power of this is that it allows beginners like me to have the ability to invent advanced projects, like

sensors or 3D graphing. What's most beneficial is that code reuse here allowed me to decrease my code lines by at least 75%, which is what is needed for a code to be efficient. While many think that reusing the code takes on the advantage of learning the actual code, however, this is misleading. From my experience, when I was reusing the Zelle library, I needed to teach myself far more concepts than I would otherwise have learned. That made me research getters setters, windows in python, and even more libraries. Reusing the code made me watch videos to use those libraries, perhaps that's how I came to learn some trinker graphics as well.

I also came to realize the usefulness of inheritance in python. Since currently I am spending my time thinking of the last project for my CS class, The implications of inheritance can be used for a card game simulator, where I can define the class deck (full cards) and the class hand (the cards being held in the hand) as a whole class.

Follow-up Questions:

1. ***What is inheritance?***

   The concept of inheritance in python allows the user to create to classes with one class inheriting the properties and methods from the other class.

   The class being inherited is named the parent class while the class inheriting the methods is classed a child class.

2. ***What does it mean for a child class to override a method?***

   Overriding a method simply means the ability to have two methods with the same name but performing different tasks. However, this is also doable via different classes since you cannot have two functions with the same name performing different functions within the same class. If the child class has the same name as a method existing within a parent class, we can say that the child class overrides this method.

3. ***What is a class variable or class global variable?***

   A global variable is a variable that can be accessed throughout the whole code unless over shadowed. To create one, just define the variable at the start of the program.

4. ***What is a field of an object***

A reference class for all the classes who are child classes of the field parent class

Reference:

David Roberts, contributed to the last task and help me achieve a functioning video output.

Kalyan, Fixed my star object.

Magzhan, Fixed some major issues in my lab