Fired up!

Python Foundations

Time: 60 mins

Introduction

In this class, students will be introduced to the concept of debugging. Students will apply the concepts of adding a turtle and detecting collisions. By the end of the class, they can add a bullet object to the game, destroy asteroids, identify bugs and errors, and debug their code.

Python Commands Introduced

global - keyword
Makes a variable accessible throughout the code

Vocabulary

- **Bug:** When a code runs but gives an unexpected result, it is said to have a bug
- **Debugging:** Finding and removing bugs from the code is called debugging
- **Error:** When the code is incorrect and does not run, it is called an error

Learning Objectives

Student(s) should be able to:

- ***State*** the commands used to fetch the position of a turtle
- ***Identify*** collision between turtles using the **distance()** function
- ***Implement*** collision detection with key press
- ***Investigate*** the code to debug and fix errors

Activities

1. **Class Narrative:***(2 mins)*
    - Summarize the progress the student has made in the game and what the spaceship can do
    - Introduce the outcome of the class - Make the spaceship fire bullets on the press of the “Space” key

2. **Concept Introduction Activity:** *(3 mins)*
    - Encourage the students to play and explore the game and destroy the asteroids
    - Guide the students to fire bullets using the “**Space**” key and destroy the asteroids

3. **Activity 1: Fire the Bullet:** *(10 mins)*

   Student Activity 1.1
   - Guide the students to define a function using the **def** keyword.
   - Guide the students to write the code within the **fire()** function. Students need to use variables to store the current **x** and **y** positions of the **spaceship** using **xcor()** and **ycor()** functions, and position the bullet using the **goto()** function.
   - Remind the students to write the code to call the **fire()** function when the **spacebar** is pressed.
   - Draw the students' attention to the bullets' lack of movement.

   Student Activity 1.2
   - Guide the student to write a **while()** loop to move the bullet upwards by increasing its **y-position** till the bullet reaches the top of the game screen.
   - Note: Help the students to understand the difference between adding and subtracting a number to change the direction of movement of the bullet.

4. **Activity 2: Destroy the Asteroid:** *(10 mins)*
   Student Activity
   - Guide the students to create a function **destroy_asteroid().**
   - Remind the students to use the **distance()** function to detect collisions between the bullet and the asteroid, and **hide** and **reset** the bullet and the asteroid on collision.
   - Probing Question: “What would happen if the distance mentioned in the condition is increased”? Answer: “The collision will be detected before the asteroid and the bullet appear to collide”

5. **Activity 3: Debug the Code:** *(20 mins)*
   Teacher Activity
   - Ask the students to play the game and observe the outcome when they press the ‘spacebar’ continuously and when it is held down.
   - Introduce the concept of **bugs** in code and **debugging** with the outcome of the game as an example.
   - Demonstrate debugging using the **variable bullet_state** in the **fire()** function and run the code
   - Draw the students' attention to the error message and Introduce the concept of **errors**
   - Fix the code using the keyword **global** and declare **bullet_state** as a global variable within the fire function.
   - Explain that the error was caused since the variable bullet_state was not modifiable in the **fire()** function.

   Student Activity
   - Guide the students to go through the **fire()** function and set the **bullet_state** to **“fired”** if the **bullet_state** is “**loaded”**.
   - Guide the students to declare the **bullet_state** as a global variable within the **destroy_asteroid()** function and reload the bullet by updating the value of the variable **bullet_state** to **“loaded”.**
   - Congratulate students for powering up the spaceship and also learning to **debug** their code

6. **Introduce the Post class project:***(2 min)*
   - Write the code to start the soccer game when the ‘s’ key is pressed.

7. **Test and Summarize the class learnings:***(5 mins)*

- Check for understanding through quizzes and summarize learning after respective missions.
- Summarize the overall class learning towards the end of the class.

8. **Additional activities:**
    - Encourage the students to move the enemy spaceship to the right by updating the value of it's x-position within a **while()** loop.
    - Encourage the students to reset the enemy to its starting position after it moves out of the screen.

9. **State the Next Class Objective:***(1 min)*
    - In the next class, we will add game states to the space shooter game to run the game smoothly and display scores and lives to make our game more fun and engaging.

U.S. Standards:

CSTA: 2-AP-11, 2-AP-12, 2-AP-13

Links Table		
Activity	**Activity Name**	**Link**
Class Presentation	Fire up!	https://s3-whjr-curriculum-uploads.whjr.online/23b78ff1-5366-47dc-92f3-ebff1b9179dd.html#/48
Explore Activity	Space Shooter Playable Link	https://tynker.me/code/view/62bd35e89eacb751786eabe2/
Student Activity 1.1	Fire Up	https://tynker.me/code/project/6322f27d1e448a4499694732
Teacher Reference: Student Activity 1.1 Solution	Solution of SA1.1	https://tynker.me/code/project/6322f484ce59eb62803feb92
Student Activity 1.2	Move the bullet	https://bfs-dev.tynker.com/code/project/6322f5a5cfa5487dea1a4ad2
Teacher Reference: Student Activity 1.2 Solution	Solution of SA1.2	https://tynker.me/code/project/6322f88a56a70b5a080b26e2
Student Activity 2	Destroy the Asteroid	https://tynker.me/code/project/6322f8f77abb867053240042
Teacher Reference: Student	Solution of SA2	https://tynker.me/code/project/632

Activity 2 Solution		2f8da261dc579c24926d2
Student Explore Activity	Debug Playable Link	https://tynker.me/code/view/62b42d11e56d7c57cd3d8812/
Teacher Activity 3	Debug	https://www.tynker.com/code/project/6322fda6d68d8e60af028c52
Teacher Reference: Teacher Activity 3 Solution	Solution of TA3	https://tynker.me/code/project/6322fdce8bf22e7b4233ca82
Student Activity 3	Debug	https://tynker.me/code/project/632307c905ada9718c2bc962
Teacher Reference: Student Activity 3 Solution	Solution of SA3	https://tynker.me/code/project/632307fb69b866155917a3c2
Student Additional Activity 1	Move the Enemy	https://tynker.me/code/project/6329ae6e635a5a50ed7de9c2
Teacher Reference: Student Additional Activity 1 Solution	Solution of SAA1	https://tynker.me/code/project/6329ae3fadd50f1bf9152aa2
Student Additional Activity 2	Reset the Enemy	https://tynker.me/code/project/6329ad7e8aed443e69753ae2
Teacher Reference: Student Additional Activity 2 Solution	Solution of SAA2	https://tynker.me/code/project/632994d873b8e051955c6772
Post Class Project	Start the Game	https://tynker.com/code/project/6332961be6ff02270224c6f2
Teacher Reference: Post Class Project Solution	Solution of Post Class Project	https://tynker.me/code/project/633287578bf612664a75b1b2