Space Wreck

Python Foundation

Time: 60 mins

Introduction

In this class, students will be introduced to the concept of collision detection by tracking the distance between two objects. Students will revisit the concepts of randomness and add a blast image on the spaceship turtle on collision.

Python Commands Introduced

● random.randint(lower_limit, upper_limit)	Generates a random number between lower_limit value and upper_limit value including both limits
● turtle1.distance(turtle2)	Moves the specified object towards left when the specified arrow key is pressed
● print()	Print values in the console
● break statement	A while loop can be stopped using "break" statement

Vocabulary

- **Collision** is detected when two objects intersect each other.

Learning Objectives

Student(s) should be able to:

- ***Recall*** the concept of randomness.
- ***Describe*** how to break the while loop using a break statement.
- ***Explain*** how to detect collision between two objects by tracking distance between them.
- ***Create*** the blasting effect on the spaceship when an asteroid collides with it.

Activities

1. **Class Narrative:**
    - Create a sense of excitement about how the game can be more entertaining.
    - Lead them to think about how the game can end and how collision detection and randomness can be implemented to make the game challenging.

2. **Concept Introduction Activity:**
    - Recall the concept of randomness to make the asteroids falling from random position.
    - Emphasize that the start and end numbers are included in **random.randint()**.

3. **Activity 1: Spawn the Asteroids randomly:** *(5 min)*

   Student Activity:
   - Guide students to generate and assign random **x** positions from **-170, 170** for asteroids in **reset_asteroid()** function.
   - Note: Help students understand why the range is chosen between -170, 170.

4. **Activity 2: Detect Collision:** *(15 min)*

   Teacher Activity:
   - Introduce students about collision and its detection by tracking distance between two objects.
   - Introduce the commands used in turtle for tracking the distance between two objects from its center.

   Student Activity:
   - Guide the students to check distance between asteroid and spaceship using **distance()** function and print its value in the console window using **print()** function.
     Note: Help them to open the console window by clicking on the small icon at the right bottom corner to check the result of print.
   - Probe them to identify the required distance value when the asteroid is touching the spaceship.
   - Recall the use and syntax of if statement and guide them to detect collision between asteroid and spaceship.
     Note: In this activity, **print()** is used for printing variables as well as messages. Emphasis use of **" "** while printing the message directly.

5. **Activity 3: End the Game:** *(10 mins)*

   Teacher Activity:
   - Introduce the problem that the game keeps running even after collision. Lead them to think why and how this can be resolved.
   - Recall use of while loop, which is still executing even after collision is detected.
   - Explain how to use the **break** statement to stop the while loop.

   Student Activity:
   - Guide students to add a blast image and replace the image of a spaceship with a blast image on collision with asteroids.

6. **Introduce the Post class project:** *(2 min)*
   - Defend the goalpost by detecting the collision between goalkeeper and football.
   - On collision detection, move the football backwards.

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 them to create a function to select an image for an asteroid randomly.
   - Encourage them to hide the asteroid and spaceship after the blast.

9. **State the Next Class Objective:** *(1 min)*
    - You will learn to add shooting capability to the spaceship.

U.S. Standards:

CSTA: 2-AP-11, 2-AP-12, 2-AP-13, 2-AP-14, 2-AP-16

Links Table		
Activity	**Activity Name**	**Link**
Class Presentation	Space Wreck	https://s3-whjr-curriculum-uploads.whjr.online/8549f691-c8bd-457c-9a28-4170e5f5a973.html
Student Activity 1	Spawn the Asteroids Randomly	https://tynker.com/code/project/62b0312c13227d6ebc5772c2
Teacher Reference: Student Activity 1 Solution	Solution of SA1	https://tynker.com/code/project/62b0303aee7e9a6fa645ed22
Student Activity 2.1	Detect Collisions	https://tynker.com/code/project/62b0315f0991535b5e7e2a82
Teacher Reference: Student Activity 2.1 Solution	Solution of SA2.1	https://tynker.com/code/project/62b032a5001b03155c08a962
Student Activity 2.2	Detect Collisions	https://tynker.com/code/project/62b1a9560b9bbf700970f9a2
Teacher Reference: Student Activity 2.2 Solution	Solution of SA2.2	https://tynker.com/code/project/62b1a8fe609e7b447b2c1e42
Teacher Activity 3	End the Game	https://tynker.com/code/project/62b0332be7df28781e404d02
Teacher Reference: Teacher Activity 3 Solution	Solution of TA3	https://tynker.com/code/project/62b0334e81cd39789e49eb12
Student Activity 3	End the Game	https://tynker.com/code/project/62b033dd2eb40d76ec6f8b33
Teacher Reference: Student Activity 3 Solution	Solution of SA3	https://tynker.com/code/project/62b0340b1fe6990bbd3c06e2
Student Additional Activity 1	Spawn Different Asteroids	https://tynker.com/code/project/62b05a2b1cc8f64e574a2d92
Teacher Reference: Student Additional Activity 1 Solution	Solution of SAA1	https://tynker.com/code/project/62b05850a52974155a66aa82
Student Additional Activity 2	Blast On Collision	https://tynker.com/code/project/62b03493b13aeb07a3182b92
Teacher Reference: Student Additional Activity 2 Solution	Solution of SAA2	https://tynker.com/code/project/62b034b4d7229a28e1096b32
Post Class Project	Kick the Football	https://tynker.com/code/project/62b04c6c74162f38391097e2
Teacher Reference: Post Class Project Solution	Solution of Post Class Project	https://tynker.com/code/project/62b046df3dff874196180193