Lesson 9: Obstacles

In this lesson, students will learn about the concept of a game over state in a 2D game and how to set it up. They will understand the importance of a game over state in creating a challenging and engaging game experience. The lesson will focus on creating an enemy scene and implementing spikes as obstacles that trigger the game over function when the player collides with them. Students will have the opportunity to practice setting up collision detection and the game over function in their own game projects, as well as experiment with different types of enemies and obstacles.

## **Objectives:**

- Students will understand the concept of a game over state in a 2D game.

- Students will be able to set up a game over function in their game.

- Students will be able to create an enemy scene in a 2D game.

- Students will be able to implement spikes as nodes that trigger the game over function when the player collides with them.

## **Materials:**

- Computers with game development software installed

- Projector or smart board

- Internet access

## **Bell-Ringer Activity (5 minutes):**

1. Display an image or video clip of a game over screen from a popular 2D game.

2. Ask the students to briefly discuss what they think a game over state is and why it is important in a game.

## **Introduction (10 minutes):**

1. Explain to the students that in game development, a game over state is a condition that occurs when the player loses the game.

2. Discuss the importance of a game over state in creating a challenging and engaging game experience.

3. Explain that in this lesson, we will be focusing on setting up a game over state in a 2D game by implementing spikes as obstacles that trigger the game over function when the player collides with them.

## **Direct Instruction (20 minutes):**

1. Demonstrate how to create an enemy scene in a 2D game using the game development software.

2. Show the students how to add spikes as nodes in the enemy scene.

3. Explain how to set up collision detection between the player and the spikes.

4. Demonstrate how to call the game over function when the player collides with the spikes.

5. Provide step-by-step instructions and explanations as you go through the process.

## **Guided Practice (30 minutes):**

1. Divide the students into pairs or small groups.

2. Instruct each group to open their game development software and create a new project.

3. Guide the students through the process of setting up an enemy scene and adding spikes as nodes.

4. Assist the students in implementing collision detection and the game over function.

5. Circulate around the classroom to provide support and answer any questions.

## **Independent Practice (20 minutes):**

1. Instruct the students to continue working on their game projects individually.

2. Encourage them to experiment with different types of enemies and obstacles in their game.

3. Remind them to test their game frequently to ensure that the game over state is triggered correctly when the player collides with the spikes.

## **Exit Ticket (5 minutes):**

1. Ask the students to write a brief summary of the steps they took to set up a game over state in their game.

2. Collect the exit tickets to assess the students' understanding of the lesson objectives.

## **Closure (5 minutes):**

1. Recap the main points of the lesson, emphasizing the importance of a game over state in creating a challenging game experience.

2. Encourage the students to continue exploring and experimenting with game development to enhance their skills.

3. Thank the students for their participation and effort in the lesson.

**Common Core Standards:**

- CCSS.ELA-LITERACY.RST.9-10.2: Determine the central ideas or conclusions of a text; trace the text's explanation or depiction of a complex process, phenomenon, or concept; provide an accurate summary of the text.

- CCSS.ELA-LITERACY.RST.9-10.3: Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.