Slice It Workshop Walkthrough

1. Workshop Introduction

- Start the workshop by opening the **Slice It introduction** slide from the provided materials.
- **Step:** Open the workshop slides on your computer and project them on the screen.
    - **Click**: Open the presentation software (PowerPoint, Google Slides, etc.) and click "Present" or "Slideshow" to begin.
- **Explain** what **TAP** (Technology Assistance Program) is, focusing on its goals and how participants can get involved.
- **Skip or introduce** the **team slide** depending on who is present.

2. Overview of Technology

- **Step:** Go to the slide that introduces the technologies used in the workshop.
    - **Click**: On the slide deck, click the "Next" arrow or "Page Down" key to navigate.
- **Explain** briefly the tools being used:
    - **Unity**: “This is the game engine we’ll use to build levels. It might seem complex, but don't worry, we’ll walk you through it step by step.”
    - **GitHub**: "This is where we store the project files, but you don’t need to know the details for this workshop."

3. Teaching Boolean Expressions

- Boolean expressions are essential for understanding game logic. Explain how these expressions work in programming.
- **Step:** Go to the **Boolean Expressions** slide and introduce the concept.
    - **Click**: Navigate to the Boolean slide using the "Next" button.
- **Example:** “A Boolean expression evaluates to either `true` or `false`. We’ll use this logic to control game conditions.”
- **Step:** Start with **Question 1**:
    - "Which of the following makes `color is green == true`?"
    - **Click**: Show the options on the slide, and ask participants to pick an answer.
    - **Incentivize**: Offer candy or points for correct answers to encourage interaction.

- **Step:** Continue with **Question 2**:
    - "Which shape will make `shape is cube == true`?"
    - **Click**: Highlight the answer options and involve different participants to answer.
- Repeat this process for **Question 3**, again offering rewards to increase engagement.

4. Starting the Game

- Direct participants to open the game from the workshop website.
- **Step:** Provide the link to the game and instruct participants to open it in their browsers.
    - **Click**: Open a browser and type in the provided URL (e.g., **https://sliceitgame.com**).
    - **Click**: Press the **Enter** key to load the page.
    - **Click**: On the website, click the **"Play"** button to start the game.

5. Creating Levels in Unity

- Now, move to the part of the workshop where participants will create their own levels in **Unity**.
- **Step:** Open **Unity Hub**.
    - **Click**: Double-click the **Unity Hub** icon on your desktop (or find it in the Start menu).
    - **Click**: On the **Projects** tab, click **Open** and browse to the downloaded project files.
- **Step:** Open the specific project.
    - **Click**: Double-click the **Slice It** project to load it in Unity.
- **Step:** Navigate to the **Scene** view.
    - **Click**: In the Unity window, click the **Scene** tab at the top to open the workspace.
    - **Click**: Find the **Hierarchy** window on the left, then double-click **Level 1** to load the scene.
- **Step:** Start adding **prefabs** (game objects) to the scene.
    - **Click**: Open the **Prefabs** folder in the **Project** window.
    - **Drag and Drop**: Select a **Green Box** prefab and drag it into the scene.
- **Step:** Add a **tag** to the object.
    - **Click**: In the **Inspector** window on the right, click the **Tag dropdown**.

    - **Click**: Choose **Add Tag**, then type `green box` and assign it to the prefab.
- **Repeat the process** for other objects except the **bomb** prefab, as adding tags to bombs may cause issues.

6. Editing Game Scripts

- Participants will now edit the game logic in **Visual Studio**.
- **Step:** Open **Visual Studio** (or the default code editor).
    - **Click**: In Unity, go to **Assets > Scripts** and double-click the **Blade.cs** script to open it in Visual Studio.
- **Step:** Navigate to **line 19** in the code.
    - **Click**: Scroll down to **line 19**, where a comment in the code indicates where to start adding logic.
    - **Type**: Add a **public integer object** to count the number of objects the player interacts with.
    - Example Code:

```
csharp
Copy code
public int objectCount = 0;
```

- **Step:** Modify the script to detect collisions.
    - **Click**: Scroll down to **line 98**.
    - **Type**: Add a method for detecting collisions, like:

```
csharp
Copy code
private void OnTriggerEnter(Collider other) {
  if (other.gameObject.tag == "green box") {
    objectCount++;
  }
}
```

- **Explain** that this code checks if an object with the tag `green box` has collided and increments the count when it does.

7. Level Progression and Scene Management

- **Step:** Teach participants how to manage scenes and transition between levels.
    - **Click**: In Unity, go to **File > Build Settings** to view the scene list.

    - **Drag and Drop**: Drag scenes like **Level 1** and **Level 2** into the build settings.
- **Step:** Implement a condition to move to the next level.
    - **Click**: In the **Blade.cs** script, under the object count logic, add:

```
csharp
Copy code
if (objectCount >= 10) {
  SceneManager.LoadScene("Level 2");
}
```

    - **Explain**: “This code moves the player to Level 2 when they have interacted with 10 objects.”

8. Handling Bombs and Resetting Progress

- **Step:** Teach participants how to handle bombs.
    - **Click**: Select the **bomb prefab** in Unity.
    - **Click**: In the **Inspector**, add a **Bomb** tag.
- **Step:** Modify the script to reset the game when a bomb is hit.
    - **Click**: In the **Blade.cs** script, add:

```
csharp
Copy code
if (other.gameObject.tag == "bomb") {
  objectCount = 0;
}
```

9. Testing and Playing the Created Level

- **Step:** Test the newly created level in Unity.
    - **Click**: In Unity, click the **Play** button at the top to run the game.
    - **Test**: Let participants play through the level and see the scene transition when they reach the required object count.

10. Conclusion

- **Step:** Ask if participants have any **questions**.
    - Open the floor for questions and assist with any issues.
- **Step:** Provide a link to the **post-survey**.

- **Click**: On the final slide of the presentation, display the **post-survey** link and encourage participants to complete it.

This in-depth guide includes where to click, what to open, and how to guide participants through each part of the workshop.