



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 4
Creating & Assigning Materials, UI, Programming
Date of Performance:
Date of Submission:
Marks:
Sign:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

EXPERIMENT NO. 4

Aim: Creating & Assigning Materials, UI, Programming

Theory:

UI (User Interface) programming involves creating the visual and interactive elements of a software application that users interact with.

Languages and Frameworks

1. HTML/CSS:

- **HTML (HyperText Markup Language):** The standard language for creating web pages.
- **CSS (Cascading Style Sheets):** Used for describing the presentation of HTML, including layouts, colors, and fonts.

2. JavaScript:

- A scripting language used to create dynamic content and interactive elements on web pages. Popular libraries and frameworks include:
 - **React:** A JavaScript library for building user interfaces.
 - **Angular:** A platform for building mobile and desktop web applications.
 - **Vue.js:** A progressive framework for building user interfaces.

3. Swift/Objective-C:

- Used for developing iOS applications. Swift is the modern programming language preferred for iOS development, while Objective-C is older but still used.

4. Java/Kotlin:

- Used for developing Android applications. Kotlin is now the preferred language for Android development, while Java remains widely used.

5. C#/.NET:

- Used for developing Windows applications. The .NET framework and its successor, .NET Core, provide a robust platform for building desktop and web applications.

6. Flutter/Dart:

- **Flutter:** A UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase.
- **Dart:** The programming language used by Flutter.

7. Python:

- Often used for prototyping and developing applications with frameworks like Django (web) and Kivy (cross-platform).

Tools and Libraries

1. UI Design Tools:

- **Sketch:** A vector graphics editor for macOS.
- **Figma:** A web-based design tool with real-time collaboration.
- **Adobe XD:** A vector-based user experience design tool.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

2. **Component Libraries:**
 - **Material-UI:** React components for faster and easier web development.
 - **Bootstrap:** A front-end framework for developing responsive websites and web applications.
 - **Ant Design:** A design system for enterprise-level products.

Problem Statement:

Develop a scene in Unity that includes a cube, plane & sphere. Create a new material & texture separately for three GameObjects. Change the color, material & texture of each GameObject separately in the scene. Write a C# program in visual studio to change the color & material/texture of the gameobjects dynamically on button click.

Steps:

Scene creation:

- Create a scene named Exp 4 in the similar manner to experiment 3 with the same folder structure.
- Create a cube, plane & sphere in the scene.
- Position them & the camera in such a way that all of them are visible in the Game view in the similar manner.

Material creation:

1. Material is an asset that is used to define how a gameobject with a renderer component will look in the game.
2. Create a "Materials" folder inside the experiment 4 folder.
3. Create a material inside the folder by right click -> Create->Material.
4. Click on the material & check if it has the standard shader assigned to it in the inspector.
5. Name the material "Plane". Create 2 other materials for cube & sphere.
6. Give a different color to each material from the inspector by changing the albedo.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Assigning Materials:

Go to the Mesh Renderer component of each gameobject - cube, sphere, plane & assign the respective material in the materials list of it.

2. Each GameObject will only have one material in its materials list. Creating Custom Components from code:

- We will be using C# to create custom components in Unity.
- Create a new folder, "Scripts" inside the experiment folder.
- Create -> C# script
- Name the script "ColorChanger"
- Double click the script to open it up in Visual Studio.

Understanding MonoBehaviour:

MonoBehaviour is the base class from which every Unity script derives.

If you need to add a script as a Component on a GameObject, you need to inherit the class from MonoBehaviour.

When a script is created the Start & Update functions are already present in it by default.

Write the script as following:

Editing the script

1. Remove the Update function from the script as we won't be needing that.
2. We need to implement the color changing functionality inside this script.
3. To do that we need a list of colors & textures that we will be changing on runtime at the button click.

4. Write the script as following:

5. Save the script by Ctrl + S & switch to Unity.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Creating User Interface (UI)

UI is the graphical interface where we can display information such as numbers, messages & add interactable buttons for the user to interact with.

2. Go back to Unity & in the hierarchy window, right click -> UI -> Canvas.
3. Now right click on the Canvas GameObject created & select UI -> Button - TextMeshPro.
4. Name the gameobject "Color Changing Button".
5. Click on the child text object of it & in the inspector in the text field type- "Switch Color".

Rect Transform Component:

1. The Rect Transform component is the 2D layout counterpart of the Transform component. Where Transform represents a single point, Rect Transform represent a rectangle that a UI element can be placed inside.
2. All UI elements don't have a Transform but a RectTransform component.
3. It's used to store and manipulate the position, size, and anchoring of a rectangle and supports various forms of scaling based on a parent RectTransform.
4. We have to position the button from the left side of the screen so we will do that by clicking on the box shaped icon on the top left corner of the RectTransform component in the inspector. Then press shift+alt on the keyboard & click on the 1st icon from the middle row in the grid.

Prefabs:

1. Unity's Prefab system allows you to create, configure, and store a GameObject complete with all its components, property values, and child GameObjects as a reusable Asset.
2. The Prefab Asset acts as a template from which you can create newPrefab instances in the Scene.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

3. When you want to reuse a GameObject configured in a particular way – like a non-player character (NPC), prop or piece of scenery – in multiple places in your Scene, or across multiple Scenes in your Project, you should convert it to a Prefab.
4. This is better than simply copying and pasting the GameObject, because the Prefab system allows you to automatically keep all the copies in sync.

Creating the Button Prefab:

1. To create a Prefab, simply drag & drop a gameobject from the hierarchy to a folder in the project window.
2. Drag & drop the Color changing button from the hierarchy into a “Prefabs” folder inside the Experiment 4 folder.

Arranging the Buttons

We will be needing the buttons for changing the colors of 3 GameObjects so create 2 more duplicates of the button GameObject in hierarchy view by clicking on it & pressing Ctrl + D.

We will need to reposition the buttons so they don't overlap.

Edit the “Pos Y” of the RectTransform of 2 buttons to make them position in a similar way.

Click on the arrow besides the Button gameobject in the hierarchy to expand it. Then Change the text of the buttons by selecting the Text gameobject of the button child & Changing the text in the inspector.

Setting Up ColorChangers:

- Select the Plane, Cube, Sphere in the Hierarchy & in the inspector click on Add component.
- Search for Color Changer Script & press Enter.
- Setup each color changer of each gameobject individually



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

- For instance: Setup the Plane by clicking on it. In the Inspector window to add colors in the Colors List field, click on the plus icon on the right of the field. Click on the new color added & select the color from the palette & close the palette window
- Similarly add Textures to the Textures list.

Calling the Functions:

1. To call a script's function from a button click click on a button.
2. In the Inspector, look at the OnClick Event of the Button component.
3. Click on the + sign on the bottom right of the Event.
4. Drag & drop one of the 3 GameObjects on the field written as "None (Object)". Then from the dropdown menu select the ColorChanger script & then the ChangeColor() function.
5. Setup the other 2 buttons too for the other Color Changing GameObjects in a similar way.

Click on Play to view the output.

Output:

Conclusion:

What do you mean by material and how it is created?