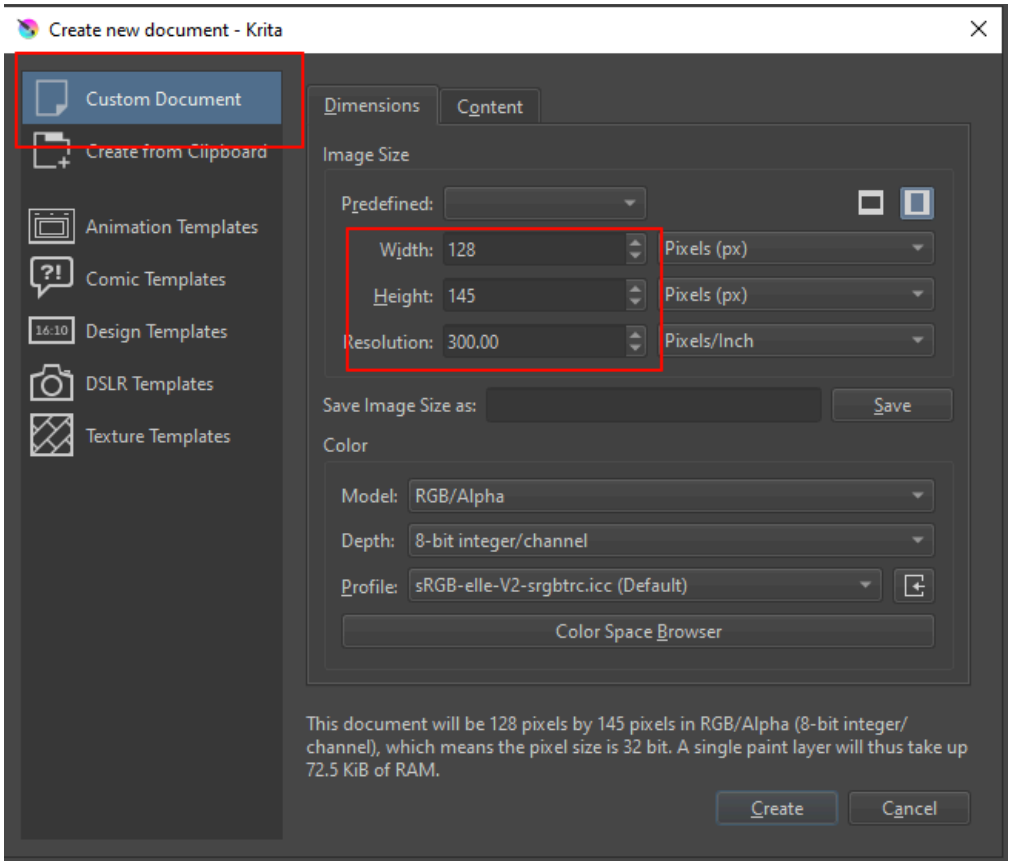
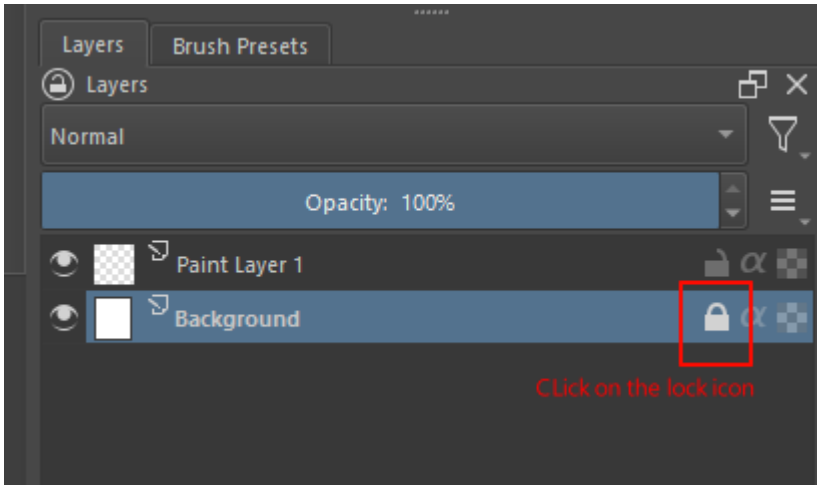
1.Create a 2D character and write a C# script which will allow the player to move and jump

1.1 Steps for Sprite Creation :

1. Open Krita.
2. Press **Ctrl + N** to create a new document **or,** Click on New Document Option in the home screen **or,** Go to file and click new.
3. Select Custom Document in the left document. Set height and width less than 512 pixels.
   a. 

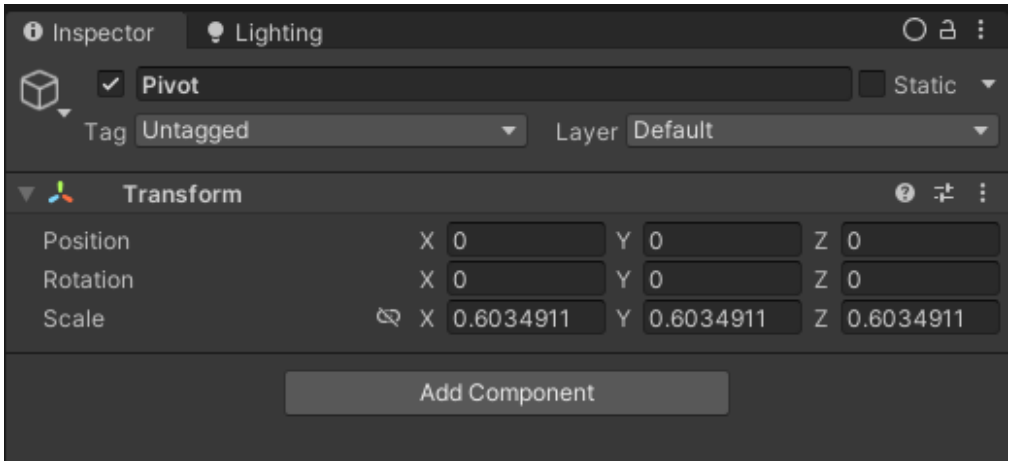
4. Once the document has been created, go to the layers section on the bottom right part of the screen and uncheck the lock icon of the Background layer.

a. 

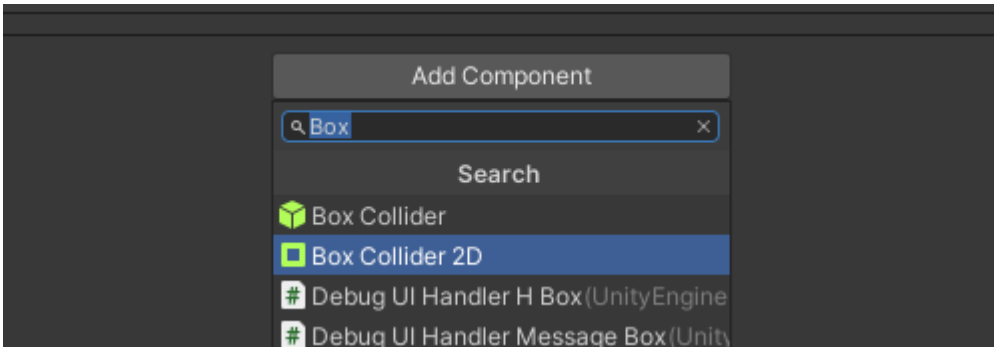
b. After clicking the lock icon, it changes to an unlock icon. Then, simply **right-click** on the **Background layer** and select **Remove Layer** to make the image transparent.
c. **Press B** to shift to draw mode or click on the brush icon in the leftmost panel.
d. Draw your character.
e. Once the drawing is complete, Go to **File > Save As**. Save your image as a **PNG** File.

1.2 Unity

1. Create a Unity Project with the **2D Core** Template.
2. Read about the different windows in the unity editor using this[1] link.
3. Drag and drop the PNG into the **Project** window. It will appear black in the project window.
4. Click on the image, go to the **Inspector** window and then set the texture type to **sprite** from the dropdown menu.
5. Click on the **Apply** button in the bottom of the inspector window. Your character should now be visible.
6. Drag and drop the sprite into the **scene** window.
7. Set the transform values of the sprite to (0,0,0) in the inspector window.

a. 

8. Ensure that the character is inside the camera box and visible in the game window.

    a. If not, then adjust the position[2] of the camera till the character becomes visible.
9. In the **Hierarchy** Window right click > 2D Objects > Sprites > Square
10. Tweak the square sprite to make the ground below your character using the transform tools.
11. Add **Box Collider 2D** component in the inspector window. Click on the sprite and click Add Component button in the inspector window. Search for Box Collider 2D in the search bar.
    a. 

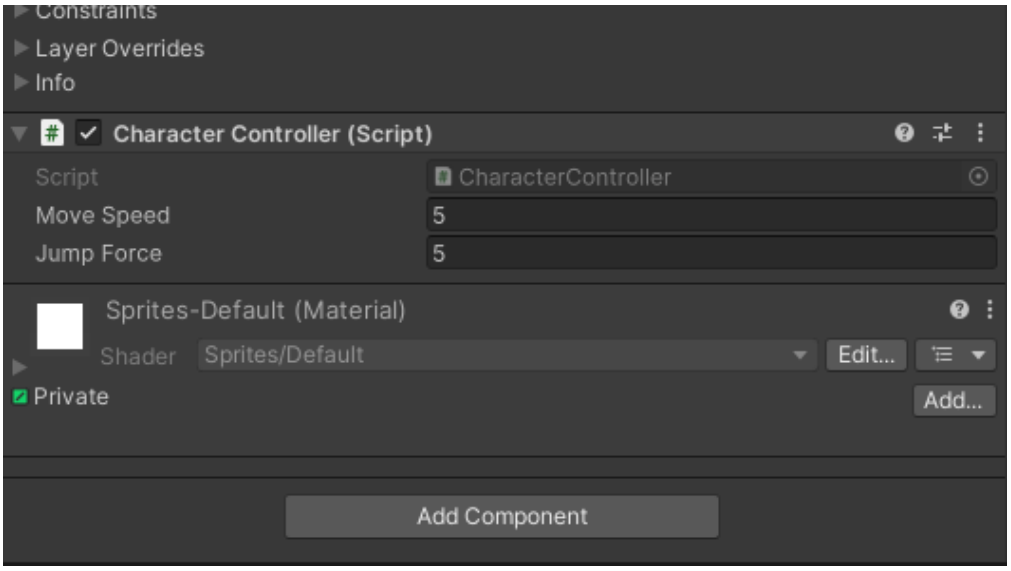
12. Add Box Collider 2D component to the character sprite as well in the inspector window.
13. Add Rigidbody 2D component to the character sprite.
14. C# scripts are created in the project window. Right click in the project window > Create > C# script
15. Create a new c# script in the Project window, call it CharacterController. An empty script will look like the following codeblock.

```C/C++
using UnityEngine;
public class CharacterController : MonoBehaviour
{
  void Start()
  {

  }

  void Update()
  {

  }
}
```

16. Add the following code to the above script.

```C/C++
using UnityEngine;
```

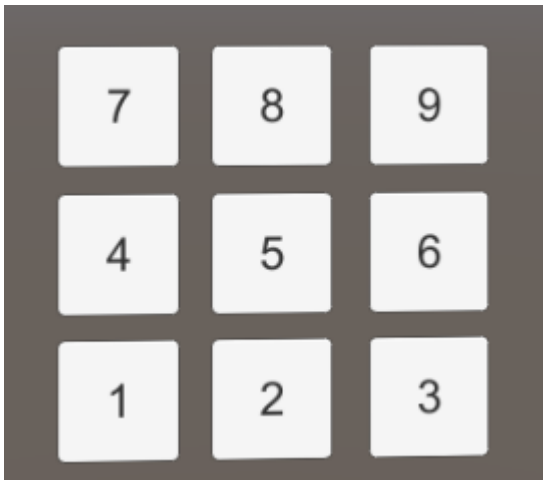
```
public class CharacterController : MonoBehaviour
{
    public float moveSpeed = 5f;
    public float jumpForce = 5f;

    private Rigidbody2D rb;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        Vector3 movement = new Vector3(Input.GetAxis("Horizontal"), 0f, 0f);
        transform.position += movement * Time.deltaTime * moveSpeed;

        if (Input.GetKeyDown(KeyCode.Space))
        {
            rb.AddForce(new Vector3(0f, jumpForce), ForceMode2D.Impulse);

        }
    }
}
```

17. Attach this script to the character gameobject by using the **Add Component** button in the Inspector or by dragging and dropping the script directly onto the gameobject in the **Hierarchy** or the Scene window.
    a. 

18. Hit the play button to move around the character on the ground.

2. Create a simple “Guess the Number” game using Unity’s UI features

2.1 Setup UI Buttons

1. Create new unity project using the 2D Core template
2. In the hierarchy right click > UI > Canvas
3. Right click on the canvas > Legacy > Button
4. Arrange the buttons in a 3 x 3 grid. Make sure the buttons are visible in the game window, if not move them inside the camera bounds using the transform tools.
5. Each button has a Text gameobject as its child. Choose and rename it to be a respective number on the number keys.
    a. 

6. Create another Text gameobject inside canvas and rename it AnswerText.
7. Place it above the grid seen in point 5 above.

2.2 Scripting

1. Create a new c# script. Call it GameController.

```
C/C++
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class GameController : MonoBehaviour
{
  public Button button1;
  public Button button2;
  public Button button3;
  public Button button4;
  public Button button5;
  public Button button6;
  public Button button7;
```

```
  public Button button8;
  public Button button9;
  public int answer;
  public Text answerTxt;
  int targetNumber = 0;

  void Start()
  {
    targetNumber = Random.Range(1, 9);

    button1.onClick.AddListener(()=>ButtonInputBehaviour(1));
    button2.onClick.AddListener(()=>ButtonInputBehaviour(2));
    button3.onClick.AddListener(()=>ButtonInputBehaviour(3));
    button4.onClick.AddListener(()=>ButtonInputBehaviour(4));
    button5.onClick.AddListener(()=>ButtonInputBehaviour(5));
    button6.onClick.AddListener(()=>ButtonInputBehaviour(6));
    button7.onClick.AddListener(()=>ButtonInputBehaviour(7));
    button8.onClick.AddListener(()=>ButtonInputBehaviour(8));
    button9.onClick.AddListener(()=>ButtonInputBehaviour(9));
  }

  public void ButtonInputBehaviour(int answer)
  {
    targetNumber = Random.Range(1, 9);
    if (answer == targetNumber)
    {
      answerTxt.text = "Congrats!!";
    }
    else
    {
      answerTxt.text = "Try again, Value was : "+targetNumber;
    }
  }
}
```

2. Attach this script to an empty gameobject
3. Attach each button to the GameController component by either dragging and dropping from the hierarchy or by selecting from the field itself using the Bullseye icon.
4. Attach the answerText text gameobject to the component
5. Hit play to start the game.

3.Create a simple 2D clock with moving seconds arm

3.1 Setup

1. Create a circle sprite in the hierarchy window.Make sure it's position is at the origin by setting the transform position to (0,0,0) in the inspector.
2. Change its colour in the sprite renderer component in the inspector.
3. Create a duplicate of the circle sprite and scale it down a bit.
4. Set sorting order of the duplicated circle as 2.
5. Change colour of the duplicated circle
6. Create another duplicate of the circle sprite and scale it down much further to make the centre of the clock.
7. Set the sorting order of this sprite gameobject to 3.
8. Create a square sprite and tweak it to make the second hand of the clock.
9. Create an empty gameobject and place it in the centre.Rename this gameobject to “Pivot”.
10. Make the seconds hand of the sprite a child of the pivot gameobject.

3.2 Scripting

1. Create a C# script, name it ClockController.
2. Add the following code to that script.

```
C/C++
using System;
using UnityEngine;

public class ClockController : MonoBehaviour
{
  const float secondsToDegrees = -6f;
  public Transform secondsPivot;

  void Update()
  {
    var time = DateTime.Now;
    if (secondsPivot != null)
      secondsPivot.localRotation = Quaternion.Euler(0f, 0f, secondsToDegrees *
time.Second);
  }
}
```

3. Add the script to an empty gameobject. You’ll see one of the fields is empty in the inspector.
4. Attach the pivot gameobject made in step 9 in 3.1 Setup section to the empty field in the ClockController component.

References

1. https://docs.unity3d.com/2022.3/Documentation/Manual/UsingTheEditor.html
2. https://docs.unity3d.com/2022.3/Documentation/Manual/PositioningGameObjects.html
3. https://docs.unity3d.com/2022.3/Documentation/Manual/SceneViewNavigation.html