

How to create a Chess App using Unity

Workshop 4



Prerequisites

- Unity installed (preferably version 2019.3 or newer)
- An IDE (code editor) that is attached to Unity
 - When installing Unity also install Visual Studio with it for it to work automatically
- The completed version: https://github.umn.edu/app-developers-club/Chess_App

Let's Get Started

Adding Game Win conditions:

- Open Unity, and open the Game script
- At the top of the script add: "using UnityEngine.SceneManagement;"
 - This allows us to change scenes
- At the bottom of the script add these four functions
 - Update() is one of the Unity built in functions that gets called every time the game is updated
 - We will check to see if the game is over and someone presses a button to start a new game, if that is true we load the "Game" scene which is the one we are already in, this will reset it to how it looks when we first run the game

```
MovePlate.cs Chessman.cs Game.cs X
Assembly-CSharp - Game

public string GetCurrentPlayer()
{
    return currentPlayer;
}

public bool IsGameOver()
{
    return gameOver;
}

public void NextTurn()
{
    if (currentPlayer == "white")
    {
        currentPlayer = "black";
    }
    else
    {
        currentPlayer = "white";
    }
}

public void Update()
{
    if (gameOver == true && Input.GetMouseButtonDown(0))
    {
        gameOver = false;

        //Using UnityEngine.SceneManagement is needed here
        SceneManager.LoadScene("Game"); //Restarts the game by loading the scene over again
    }
}
```

Add checks for the Chesspiece:

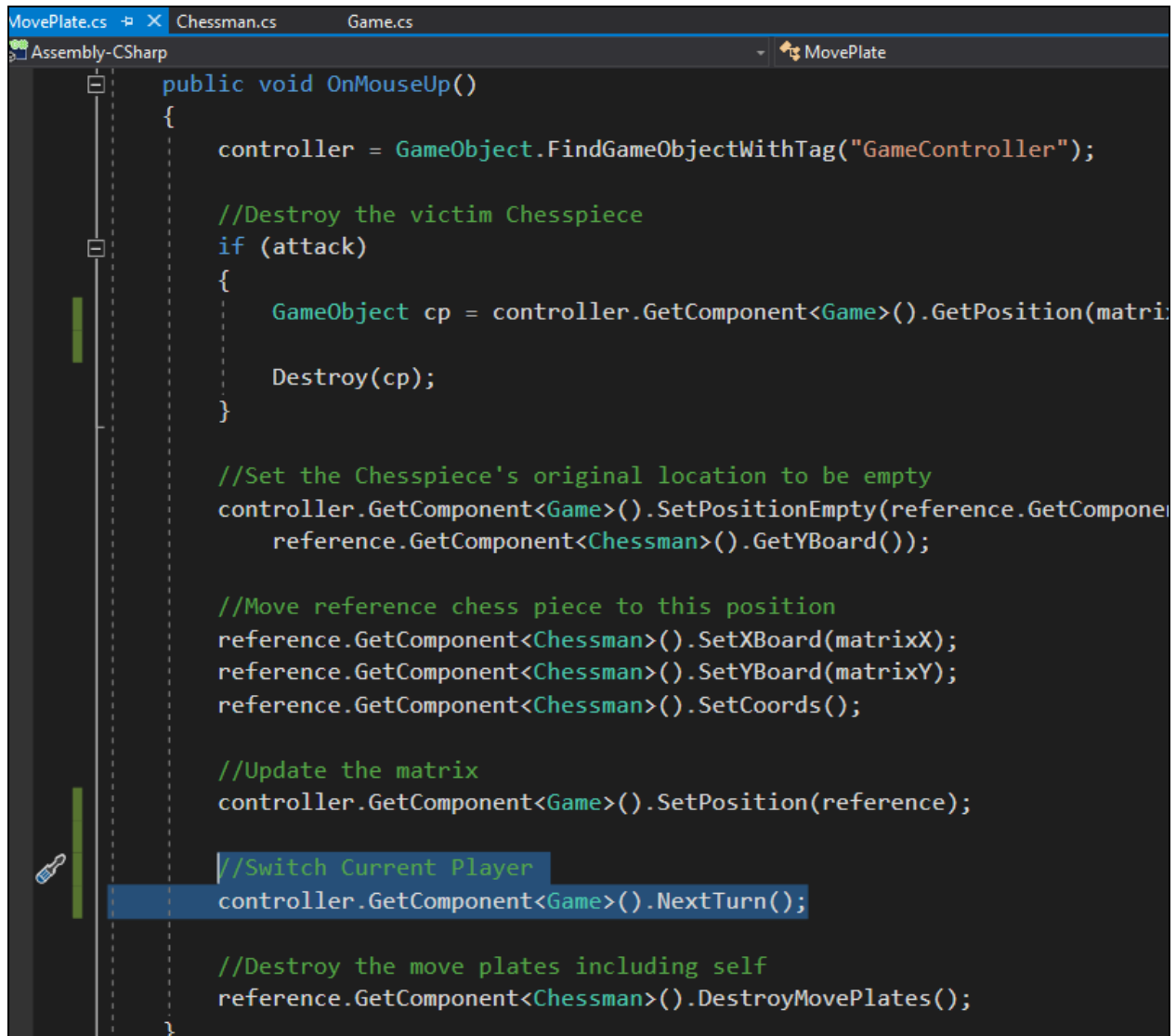
- We want to make it so that it is only one player's turn at a time
- This can be done with an extra check for the Chesspiece's OnMouseUp function
- We simply access boolean returning functions from controller

```
private void OnMouseUp()
{
    if (!controller.GetComponent<Game>().IsGameOver() && controller.GetComponent<Game>().GetCurrentPlayer() == player)
    {
        //Remove all moveplates relating to previously selected piece
        DestroyMovePlates();

        //Create new MovePlates
        InitiateMovePlates();
    }
}
```

Add a turn switch in MovePlate:

- When someone clicks on a MovePlate, it means we need to switch to the other player
- We just need to add one line of code



```
public void OnMouseUp()
{
    controller = GameObject.FindGameObjectWithTag("GameController");

    //Destroy the victim Chesspiece
    if (attack)
    {
        GameObject cp = controller.GetComponent<Game>().GetPosition(matrixX, matrixY);
        Destroy(cp);
    }

    //Set the Chesspiece's original location to be empty
    controller.GetComponent<Game>().SetPositionEmpty(reference.GetComponent<Game>().GetPosition(),
        reference.GetComponent<Chessman>().GetYBoard());

    //Move reference chess piece to this position
    reference.GetComponent<Chessman>().SetXBoard(matrixX);
    reference.GetComponent<Chessman>().SetYBoard(matrixY);
    reference.GetComponent<Chessman>().SetCoords();

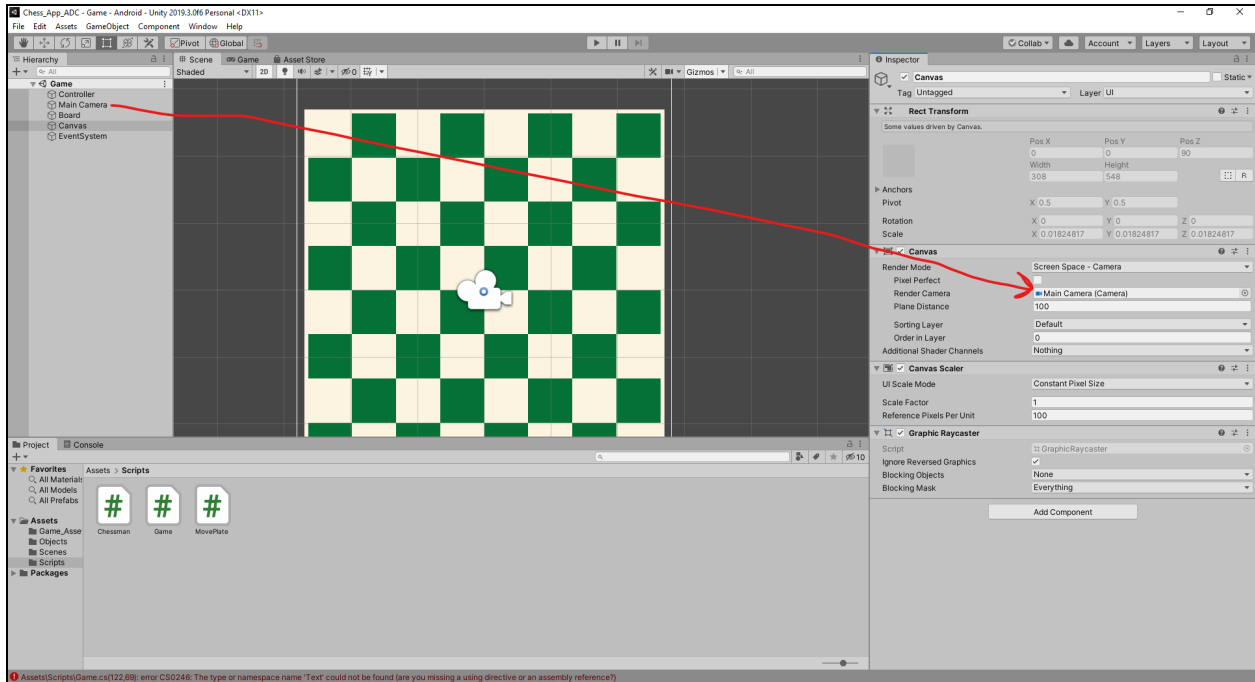
    //Update the matrix
    controller.GetComponent<Game>().SetPosition(reference);

    //Switch Current Player
    controller.GetComponent<Game>().NextTurn();

    //Destroy the move plates including self
    reference.GetComponent<Chessman>().DestroyMovePlates();
}
```

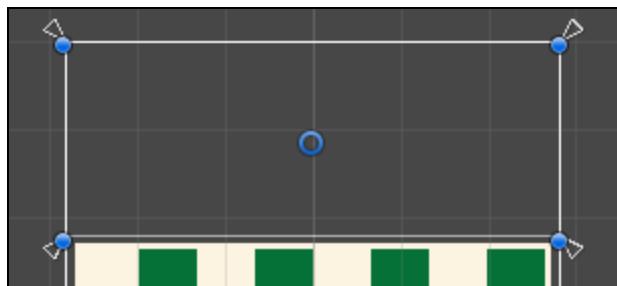
Showing win text:

- We now have the control structures in place, we also want to be able to show who won
- In order to add text we have to add a Canvas object in Unity
- In Unity select GameObject (Very top left) > UI > Canvas
- Click on Canvas and switch Render Mode to "Screen Space - Camera"
 - In the subsection "Render Camera" drag in "Main Camera" from the pane on the left



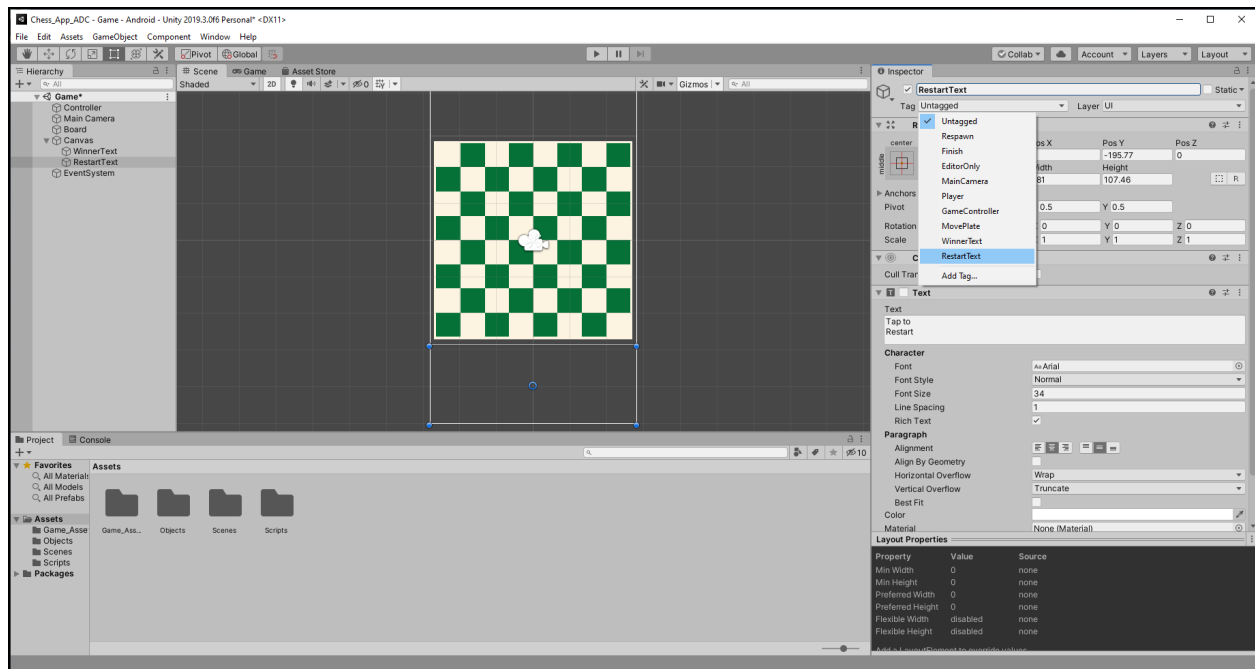
Formatting the Text:

- Right click on the Canvas and select UI > Text
- This will create a Text object as a child node
- You can take the Text box on the screen and drag it to the empty top portion
 - Drag the box to how you like it
 - There are little white arrows when selecting the text, these are the anchors
 - Make sure the anchors for the same box that the text forms



- Set the Color to white on the Inspector Pane (Right side)
- Set the Paragraph Alignment to center and center
- Change the font, font style, and font size to your liking
 - You can try downloading a more interesting font on your own
 - Simply download a font and put it in the assets folder, then add it to the font category of Text
- Delete the text
- Do this all again for the bottom part of the screen
- On the bottom Text inside the Text field type "Tap to [newline] Restart"
- Uncheck the Text checkbox so the text disappears from the scene
 - We will be activating this in the code when the game is won
- Up top create two new tags
 - For the top text: WinnerText
 - For the bottom text: LoserText

- Assign the correct tag to each and also rename them the same as the tags



Showing the winner:

- Open the Game script and add the following code
- Make sure to add "using UnityEngine.UI;" to the top of the code

```
public void Update()
{
    if (gameOver == true && Input.GetMouseButtonDown(0))
    {
        gameOver = false;

        //Using UnityEngine.SceneManagement is needed here
        SceneManager.LoadScene("Game"); //Restarts the game by loading the scene over again
    }
}

public void Winner(string playerWinner)
{
    gameOver = true;

    //Using UnityEngine.UI is needed here
    GameObject.FindGameObjectWithTag("WinnerText").GetComponent<Text>().enabled = true;
    GameObject.FindGameObjectWithTag("WinnerText").GetComponent<Text>().text = playerWinner + " is the winner";

    GameObject.FindGameObjectWithTag("RestartText").GetComponent<Text>().enabled = true;
}
```

Running the Winner() function:

- Right now nothing is calling the Winner() function to display who won
- A good place to do that is inside the MovePlate when it is destroying an enemy piece
 - If the enemy piece is the king then the game has been won by the other player
- Add the following code to the MovePlate script

```

public void OnMouseUp()
{
    controller = GameObject.FindGameObjectWithTag("GameController");

    //Destroy the victim Chesspiece
    if (attack)
    {
        GameObject cp = controller.GetComponent<Game>().GetPosition(matrixX, matrixY);

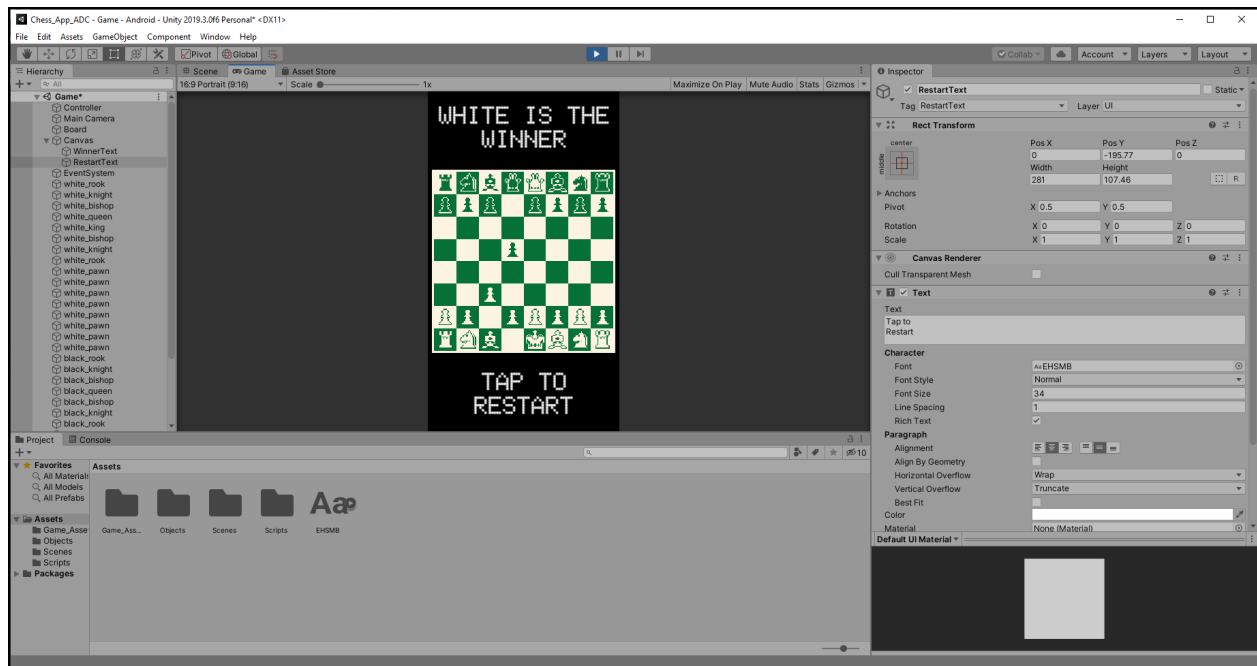
        if (cp.name == "white_king") controller.GetComponent<Game>().Winner("black");
        if (cp.name == "black_king") controller.GetComponent<Game>().Winner("white");

        Destroy(cp);
    }
}

```

The entire game now works:

- You can play a full round of chess and also restart



Running the app on an Android Phone:

- Plug your phone into the computer and make sure developer settings are enabled
 - Lookup online how to do that for your phone
- Go to File > Build Settings and make sure Android is selected
 - Once you get it running you can play around with the settings
- Click canvas and set UI Scale Mode to "Scale With Screen Size"
- Then click Build and Run
- Name the apk file
- *Note: There are many things that can go wrong at this step, but if it works that's awesome. Once you get it working Unity is a really quick way to make an app for Android!*

Thank you for working through the whole tutorial series:

- There's still more that could be done
 - Special pawn movement not implemented
 - Check and Checkmate not implemented
 - Upgrading pawns not implemented
 - Multiplayer
 - Saving
- Created by Evan Tredal and everyone in the App Developers Club at the University of Minnesota, College of Science and Engineering, 2020