



# Welcome to Number Wizard UI

@UnityCourse  
[facebook.com/UnityCourse](https://facebook.com/UnityCourse)





**GDD**





**Section Notes**





# Creating New Scenes And Levels

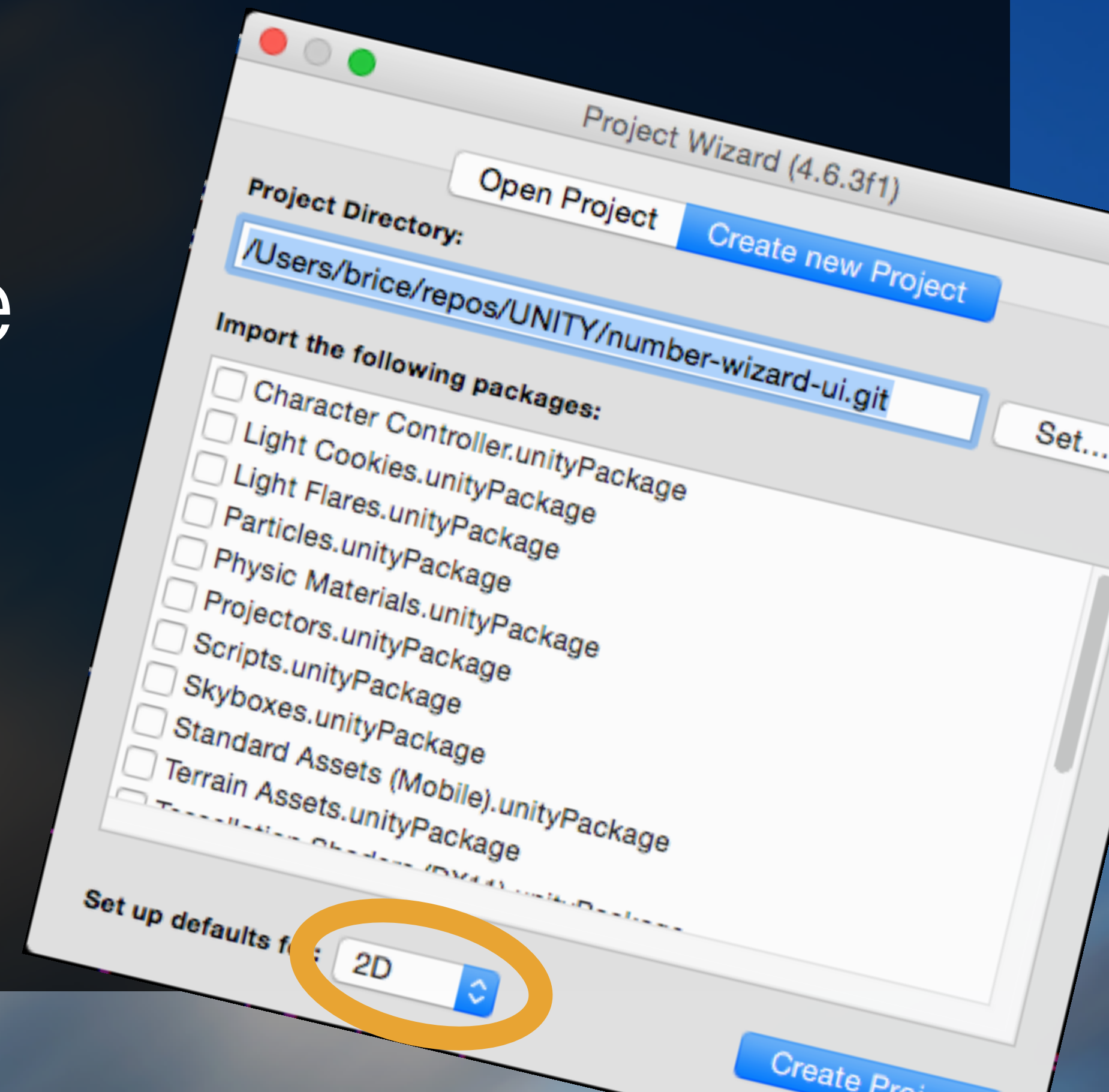
@UnityCourse

[facebook.com/UnityCourse](https://facebook.com/UnityCourse)



# Creating new Scenes and Levels

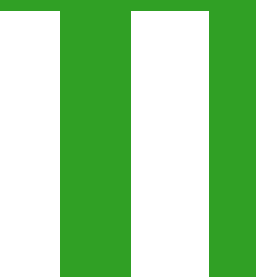
- We'll create a new **Start, Win, Game and Lose** scene in a new 2D project
- Scenes and Levels are the same
- That's it :-)





# Create three new scenes

- In the same way as we created the **Start** scene create the **Game**, **Lose**, and **Win** scenes







# How To Add Text To The Scene

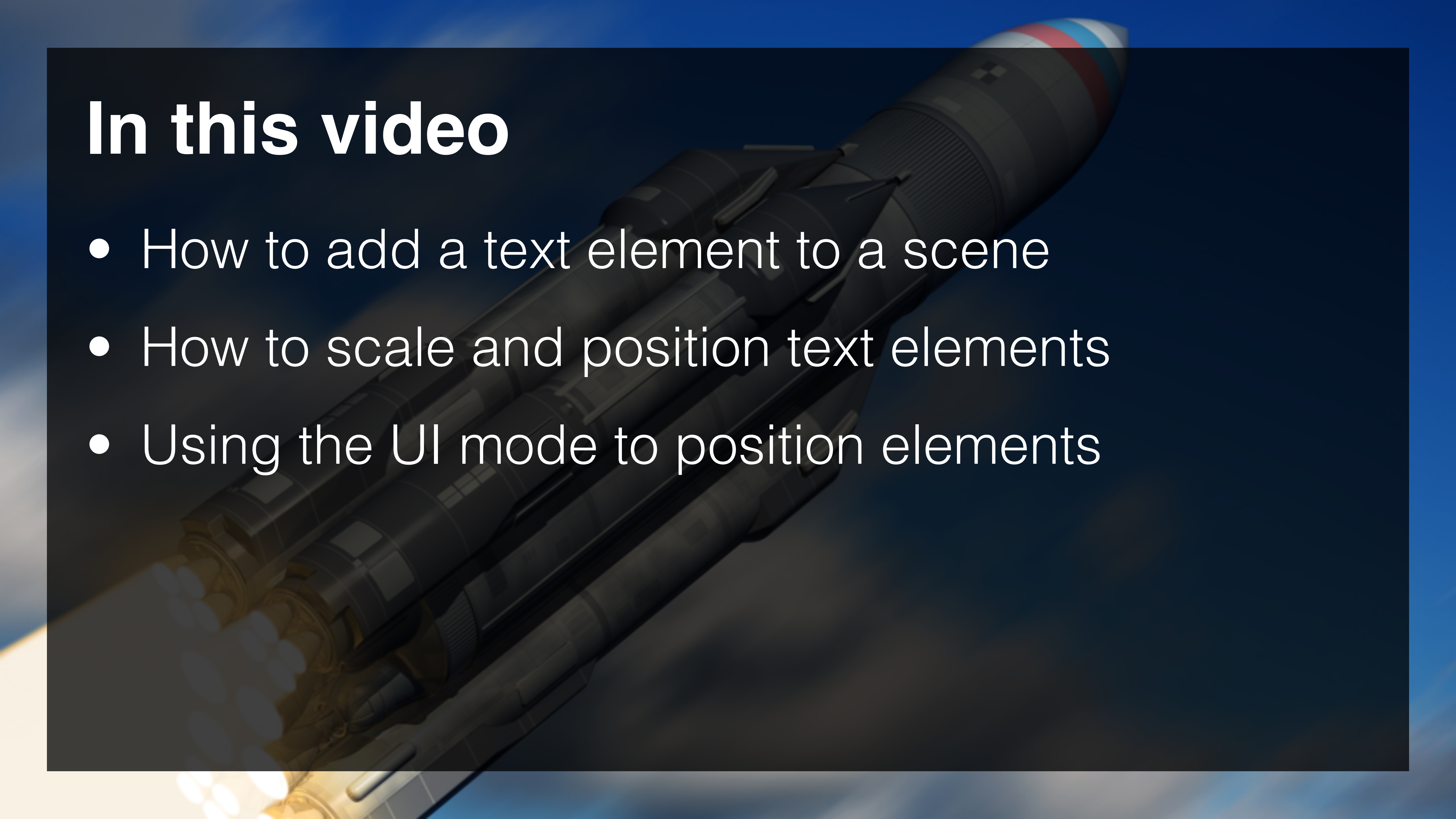
@UnityCourse

[facebook.com/UnityCourse](https://facebook.com/UnityCourse)



# In this video

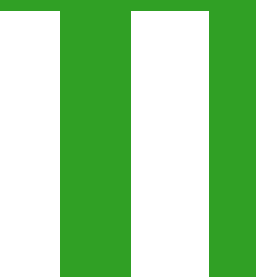
- How to add a text element to a scene
- How to scale and position text elements
- Using the UI mode to position elements





# Add a subtitle

- In the same manner as we added a title, add a subtitle prompting the user to make a choice of number





# Be Aware of old style GUI code

- + Physics
- + Scripting
- + Audio
- + Animation
- UI
  - UI Overview
    - Canvas
    - Basic Layout
    - Visual Components
    - Interaction Components
    - Animation Integration
    - Auto Layout
    - Rich Text
  - Legacy GUI Scripting Guide
    - GUI Basics (Legacy)
    - Controls (Legacy)
    - Customization (Legacy)
    - Layout Modes (Legacy)
    - Extending GUI (Legacy)
- + UI Reference
- + UI How Tos
- + Navigation and Pathfinding
- + Architecture
- + How to contribute to Unity
- + Platform Specific

## Making Controls with UnityGUI

UnityGUI controls make use of a special function called `OnGUI()`. The `OnGUI()` function gets called every frame as long as the containing script is enabled - just like the `Update()` function.

GUI controls themselves are very simple in structure. This structure is evident in the following example.

```
/* Example level loader */

// JavaScript
function OnGUI () {
    // Make a background box
    GUI.Box (Rect (10,10,100,90), "Loader Menu");

    // Make the first button. If it is pressed, Application.Loadlevel (1) will be executed
    if (GUI.Button (Rect (20,40,80,20), "Level 1")) {
        Application.LoadLevel (1);
    }

    // Make the second button.
    if (GUI.Button (Rect (20,70,80,20), "Level 2")) {
        Application.LoadLevel (2);
    }
}

//C#
using UnityEngine;
using System.Collections;
```





# Adding A Start Button

@UnityCourse

[facebook.com/UnityCourse](https://facebook.com/UnityCourse)



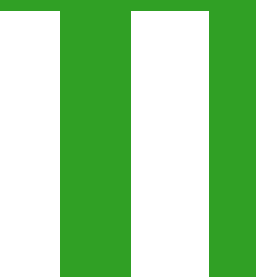
# In this video

- How to change a text element to be a button
- How to make the button respond to mouse hover and click
- Adding a quick button



# Add a Quit button

- Add a quit button for yourself below the start button





A detailed 3D rendering of a rocket launching. The rocket is grey with blue and red accents near the nose. It is angled upwards from the bottom left towards the top right. A large, bright orange and yellow flame is visible at the base, with white smoke trailing behind it. The background is a clear blue sky with some light clouds.

# Calling Scripts From Buttons

**@UnityCourse**

**[facebook.com/UnityCourse](https://facebook.com/UnityCourse)**



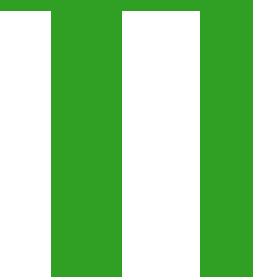
# Calling scripts from buttons

- Create a new script to handle level managment
- Call the script from the buttons
- Check everything works by logging out to the console



# Do the same for the Quit button

- Wire up the quit button so that every click logs out to the console







# How To Load Scenes And Quit

@UnityCourse

[facebook.com/UnityCourse](https://facebook.com/UnityCourse)



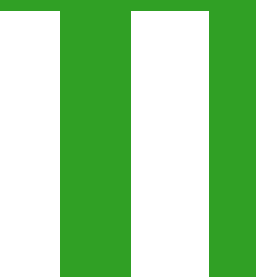
# How To Load Scenes And Quit

- We'll use **Application.LoadLevel()** to load a new level
- We'll use **Application.Quit()** to quit the game
- We'll look at the limitations of **Application.Quit()** and when it won't work as expected



# Create a Button to return to the start

- Using what you learned so far, create a button to return to the start scene from the game scene.
- You don't need to re-write a **LevelManager** script or re-create an object. It's ok to copy the existing one.





# Review of Scene Loading

Start Button

Level  
Manager

## Legend

Game  
Objects



# Review of Scene Loading

Start Button

Button

Level  
Manager

Level  
Manager

## Legend

Game  
Objects

Component



# Review of Scene Loading

Start Button

Button

OnClick()

Level  
Manager

Level  
Manager

## Legend

Game  
Objects

Component

Event Handler



# Review of Scene Loading



## Legend

Game Objects

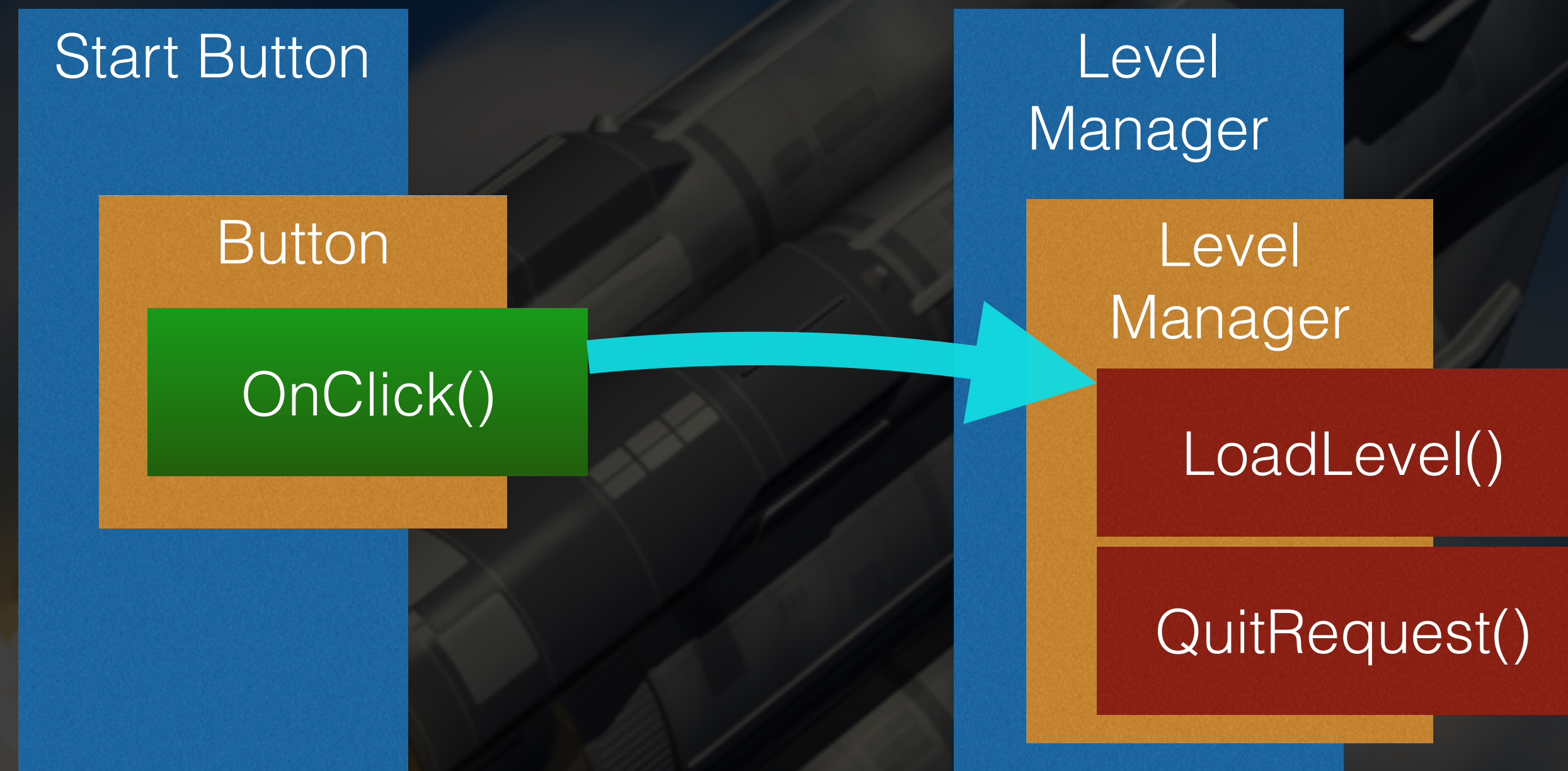
Component

Event Handler

Methods



# Review of Scene Loading



## Legend

Game Objects

Component

Event Handler

Methods





# Using a Game Design Document

@UnityCourse

[facebook.com/UnityCourse](https://facebook.com/UnityCourse)

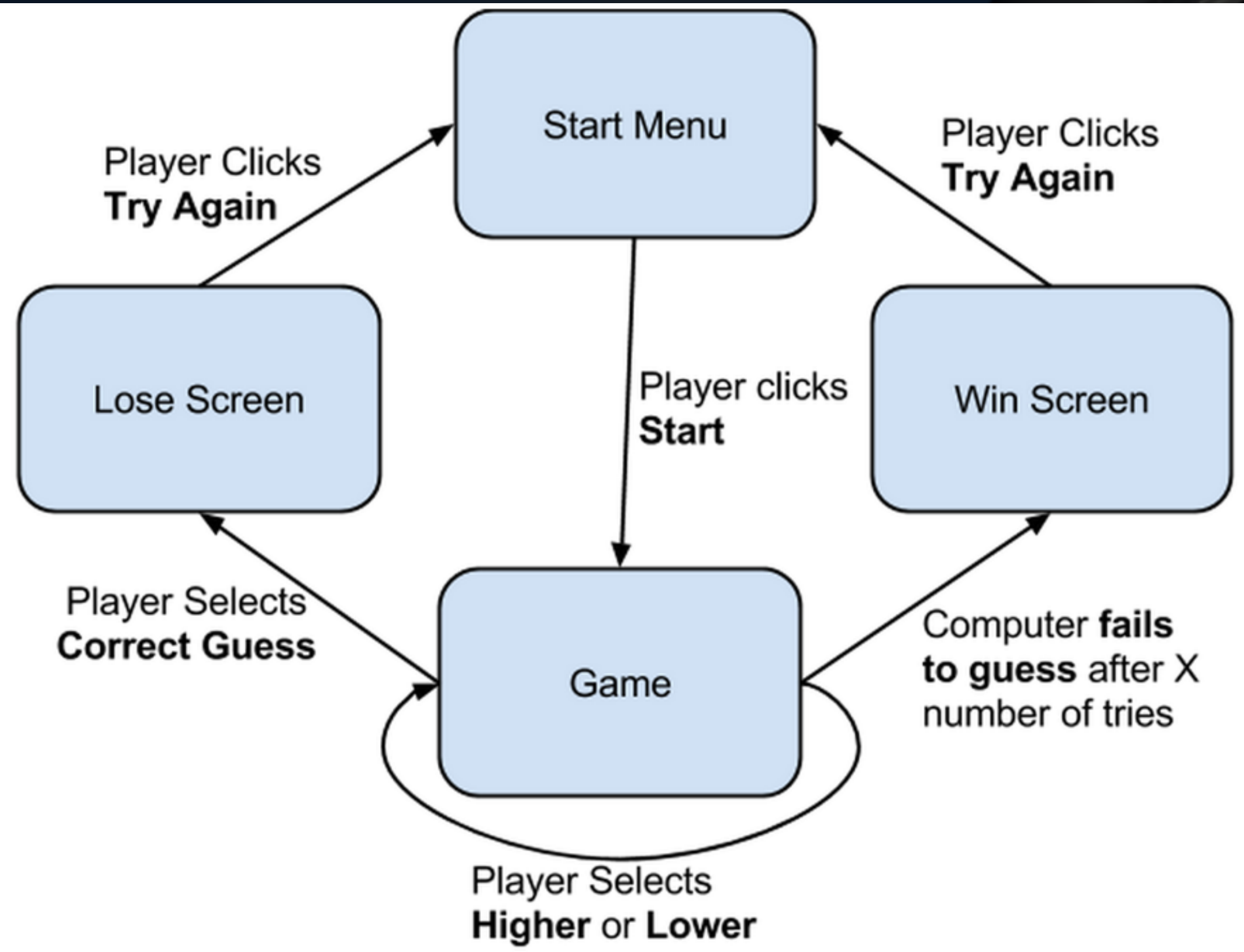


# What is a Game Design Document?

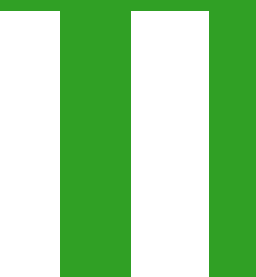
- A tool to **communicate** between members of your team (Artists, Developers, Designers)
- A place to **record decisions** made before and during development. What's **in**. What's **out**.
- **Not a plan :)**



# Create the scene transitions according to the GDD



- Create buttons that transition between all scenes according to the GDD.







# Importing Number Wizard Code

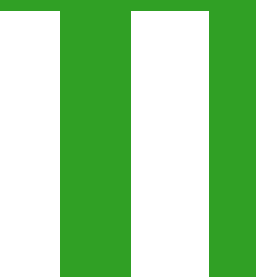
@UnityCourse

[facebook.com/UnityCourse](https://facebook.com/UnityCourse)



# Populate the Game Scene

- With text appropriate for your game
- Link up the guess higher and guess lower buttons to the Number Wizard game object
- Link up the correct button to the Level Manager's **LoadLevel()**







# Finishing Number Wizard

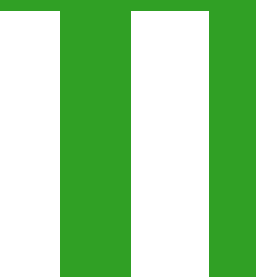
@UnityCourse

[facebook.com/UnityCourse](https://facebook.com/UnityCourse)



# Guess the numbers randomly

- Use **Random.Range()** to guess a random number instead of always choosing the halfway point in the NumberWizard script
- Optionally, ensure that the game starts on a anew guess instead of the initial 500







**End of section QUIZ**



# Download your files

The screenshot shows a Udemy lecture interface. At the top, navigation links include 'Back to Course' and 'Previous Lecture'. The current section is 'SECTION 5' and the lecture is 'LECTURE 50: DOWNLOAD YOUR GHOUL GARDEN ASSETS'. A 'View resources' button is circled in red and labeled with a red circle containing the number '1'. Below the video player, a 'NEXT LECTURE' button is visible. On the right side, a sidebar contains a 'Download' icon (a yellow button with a download symbol) circled in red. Below this, the 'Downloadable resources' section lists 'Ghoul Garden Draft Assets' with a download icon circled in red and labeled with a red circle containing the number '2'. The 'Download this lecture' section lists 'HowtoDownload.pdf' with a download icon. A red dashed box encloses the entire sidebar area, with the text 'Downloads Here' in red at the bottom. The video player shows two men in a room, with a play button in the center. The video progress bar indicates 00:18 / 02:15.

Back to Course Previous Lecture

SECTION 5 LECTURE 50 DOWNLOAD YOUR GHOUL GARDEN ASSETS

View resources

Lecture Description

You will find a copy of the draft assets we used in this section attached to this lecture.

Downloadable resources

Ghoul Garden Draft Assets

Download this lecture

HowtoDownload.pdf

Downloads Here







# Danger

**Construction work  
in progress**

THS 123 - TIGER SUPPLIES LTD 01376 345554





# Review and Congratulation

@UnityCourse  
[facebook.com/UnityCourse](https://facebook.com/UnityCourse)