unity

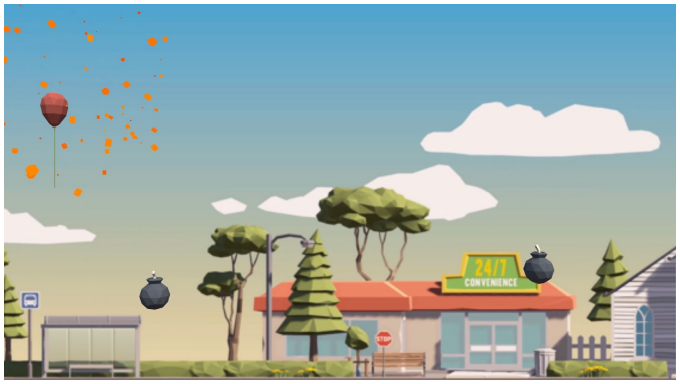
Create with Code:

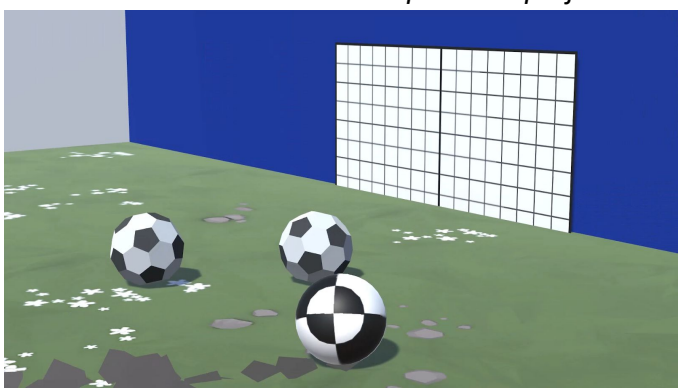
Scope & Sequence

In this introductory course, students will use Unity to learn the fundamentals of programming in the context of creating their own projects. During the course, they will create several **prototypes** along with the instructor, manage a larger **personal project** more independently, and complete **challenges and quizzes** along the way to solidify and expand their new knowledge. The skills that they learn will align with the learning objectives from the Unity Certified User exam, providing them with a foundational understanding of Unity and C# programming, preparing them for certification. In addition to these core technical competencies, students will learn how to manage a project from start to finish: coming up with a concept, creating a project plan, prioritizing tasks, and hitting milestones. By the end of the course, students will have the confidence that, given enough time and resources, they can create anything they want with Unity and C#.

Table of Contents

Examples demonstrating the variety of projects students will be able to make for their personal projects





Unit 1 - Player Movement (Driving Simulator Prototype)

UNIT SCOPE	
In this Unit, students will be introduced to Unity for the first time as they create a very basic mini-project in which they control the side-to-side movement of an object to avoid colliding with obstacles. In addition to becoming familiar with the Unity editor and workflow, they will learn how to create new C# scripts and do some simple programming. By the end of the Unit, students will be able to call basic functions, then declare and tweak new variables to modify the results of those functions.	

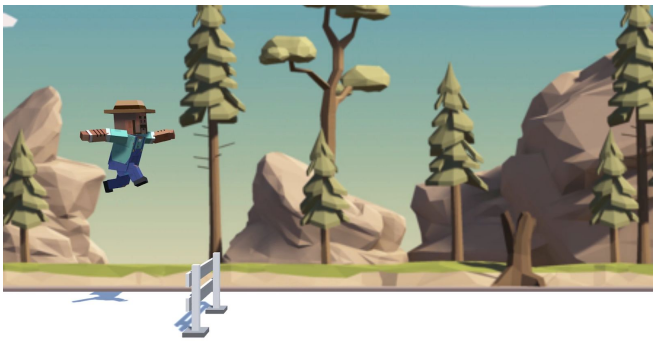
UNIT SEQUENCE	
Lesson 1.1 Start your 3D Engines	In this lesson, you will create your very first game project in Unity Hub. You will choose and position a vehicle for the player to drive and an obstacle for them to hit or avoid. You will also set up a camera for the player to see through, giving them a perfect view of the scene. Throughout this process, you will learn to navigate the Unity Editor and grow comfortable moving around in 3D Space. Lastly, you will customize your own window layout for the Unity Editor.
Lesson 1.2 Pedal to the Metal	In this lesson you will make your driving simulator come alive. First you will write your very first lines of code in C#, changing the vehicle's position and allowing it to move forward. Next you will add physics components to your objects, allowing them to collide with one another. Lastly, you will learn how to duplicate objects in the hierarchy and position them along the road.
Lesson 1.3 High Speed Chase	Keep your eyes on the road! In this lesson you will code a new C# script for your camera, which will allow it to follow the vehicle down the road and give the player a proper view of the scene. In order to do this, you'll have to use a very important concept in programming: variables.
Lesson 1.4 Step into the Driver's Seat	In this lesson, we need to hit the road and gain control of the vehicle. In order to do so, we need to detect when the player is pressing the arrow keys, then accelerate and turn the vehicle based on that input. Using new methods, Vectors, and variables, you will allow the vehicle to move forwards or backwards and turn left to right.
Challenge 1 & Quiz Plane Programming	Use the skills you learned in the driving simulation to fly a plane around obstacles in the sky. You will have to get the user's input from the up and down arrows in order to control the plane's pitch up and down. You will also have to make the camera follow alongside the plane so you can keep it in view.
Lab 1 Project Design Document	In this first ever Lab session, you will begin the preliminary work required to successfully create a personal project in this course. First, you'll learn what a personal project is, what the goals for it are, and what the potential limitations are. Then you will take the time to come up with an idea and outline it in detail in your Design Document, including a timeline for when you hope to complete certain features. Finally, you will take some time to draw a sketch of your project to help you visualize it and share your idea with others.

Unit 2 - Basic Gameplay (Feed the Animals Prototype)

UNIT SCOPE	
In this Unit, you will program a top-down game with the objective of throwing food to hungry animals - who are stampeding towards you - before they can run by. In order to do this, you will become much more familiar with some of the most important programming and Unity concepts, including if-statements, random value generation, arrays, collision detection, prefabs, and instantiation. In completing this Unit, you will learn how to program a basic game with the ability to launch projectiles and control the player to keep the game alive.	

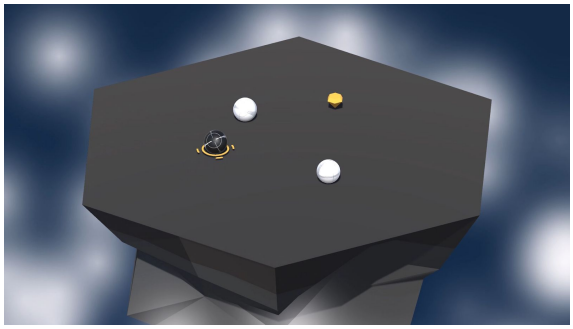
UNIT SEQUENCE	
Lesson 2.1 Player Positioning	You will begin this unit by creating a new project for your second Prototype and getting basic player movement working. You will first choose which character you would like, which types of animals you would like to interact with, and which food you would like to feed those animals. You will give the player basic side-to-side movement just like you did in Prototype 1, but then you will use if-then statements to keep the Player in bounds.
Lesson 2.2 Food Flight	In this lesson, you will allow the player to launch the projectile through the scene. First you will write a new script to send the projectile forwards. Next you will store the projectile along with all of its scripts and properties using an important new concept in Unity called Prefabs. The player will be able to launch the projectile prefab with a tap of the spacebar. Finally, you will add boundaries to the scene, removing any objects that leave the screen.
Lesson 2.3 Random Animal Stampede	Our animal prefabs walk across the screen and get destroyed out of bounds, but they don't actually appear in the game unless we drag them in! In this lesson we will allow the animals to spawn on their own, in a random location at the top of the screen. In order to do so, we will create a new object and a new script to manage the entire spawning process.
Lesson 2.4 Collision Decisions	Our game is coming along nicely, but there are are some critical things we must add before it's finished. First off, instead of pressing S to spawn the animals, we will spawn them on a timer so that they appear every few seconds. Next we will add colliders to all of our prefabs and make it so launching a projectile into an animal will destroy it. Finally, we will display a "Game Over" message if any animals make it past the player.
Challenge 2 & Quiz Play Fetch	Use your array and random number generation skills to program this challenge where balls are randomly falling from the sky and you have to send your dog out to catch them before they hit the ground. To complete this challenge, you will have to make sure your variables are assigned properly, your if-statements are programmed correctly, your collisions are being detected perfectly, and that objects are being generated randomly.
Lab 2 New Project with Primitives	You will create and set up the project that will soon transform into your very own Personal Project. For now, you will use "primitive" shapes (such as spheres, cubes, and planes) as placeholders for your objects so that you can add functionality as efficiently as possible without getting bogged down by graphics. To make it clear which object is which, you will also give each object a unique colored material.

Unit 3 - Animation, Sound, & Effects (Run and Jump Prototype)

UNIT SCOPE	
In this Unit, you will program a fast-paced side-scrolling game where the player jumps over oncoming obstacles to avoid crashing. In creating this prototype, you will learn how to add music and sound effects, completely transforming the experience of your projects. You will also learn how to create dynamic endless repeating backgrounds, which are critical for any side-scrolling games. Finally, you will learn to incorporate particle effects like splatters and explosions, which make your games so much more satisfying to play.	

UNIT SEQUENCE	
Lesson 3.1 Jump Force	The goal of this lesson is to set up the basic gameplay for this prototype. We will start by creating a new project and importing the starter files. Next we will choose a beautiful background and a character for the player to control, and allow that character to jump with a tap of the spacebar. We will also choose an obstacle for the player, and create a spawn manager that throws them in the player's path at timed intervals.
Lesson 3.2 Make the World Whiz By	We've got the core mechanics of this game figured out: The player can tap the spacebar to jump over incoming obstacles. However, the player appears to be running for the first few seconds, but then the background just disappears! In order to fix this, we need to repeat the background seamlessly to make it look like the world is rushing by! We also need the game to halt when the player collides with an obstacle, stopping the background from repeating and stopping the obstacles from spawning. Lastly, we must destroy any obstacles that get past the player.
Lesson 3.3 Don't Just Stand There	The game is looking great so far, but the player character is a bit… lifeless. Instead of the character simply sliding across the ground, we're going to give it animations for running, jumping, and even death! We will also tweak the speed of these animations, timing them so they look perfect in the game environment.
Lesson 3.4 Particles and Sound Effects	This game is looking extremely good, but it's missing something critical: Sound effects and Particle effects! Sounds and music will breathe life into an otherwise silent game world, and particles will make the player's actions more dynamic and eye-popping. In this lesson, we will add cool sounds and particles when the character is running, jumping, and crashing.
Challenge 3 & Quiz Balloons & Booleans	Apply your knowledge of physics, scrolling backgrounds, and special effects to a balloon floating through town, picking up tokens while avoiding explosives. You will have to do a lot of troubleshooting in this project because it is riddled with errors.
Lab 3 Player Control	In this lesson, you program the player's basic movement, including the code that limits that movement. Since there are a lot of different ways a player can move, depending on the type of project you're working on, you will not be given step-by-step instructions on how to do it. In order to do this, you will need to do research, reference other code, and problem-solve when things go wrong.

Unit 4 - Gameplay Mechanics (Sumo Battle Prototype)

UNIT SCOPE	
In this Unit, you will program an arcade-style Sumo battle with the objective of knocking increasingly difficult waves of enemies off of a floating island, using power ups to help defeat them. In creating this prototype, you will learn how to implement new gameplay mechanics into your projects, which are new rules or systems that make the game more interesting to play.	

UNIT SEQUENCE	
Lesson 4.1 Watch Where You're Going	First thing's first, we will create a new prototype and download the starter files. You'll notice a beautiful island, sky, and particle effect... all of which can be customized. Next you will allow the player to rotate the camera around the island in a perfect radius, providing a glorious view of the scene. The player will be represented by a sphere, wrapped in a detailed texture of your choice. Finally you will add force to the player, allowing them to move forwards or backwards in the direction of the camera.
Lesson 4.2 Follow the Player	The player can roll around to its heart's content… but it has no purpose. In this lesson, we fill that purpose by creating an enemy to challenge the player! First we will give the enemy a texture of your choice, then give it the ability to bounce the player away... potentially knocking them off the cliff. Lastly, we will let the enemy chase the player around the island and spawn in random positions.
Lesson 4.3 PowerUp and CountDown	The enemy chases the player around the island, but the player needs a better way to defend themselves... especially if we add more enemies. In this lesson, we're going to create a powerup that gives the player a temporary strength boost, shoving away enemies that come into contact! The powerup will spawn in a random position on the island, and highlight the player with an indicator when it is picked up. The powerup indicator and the powerup itself will be represented by stylish game assets of your choice.
Lesson 4.4 For-Loops For Waves	In this lesson we will wrap things up by putting a bunch of the pieces together. First we will enhance the enemy spawn manager, allowing it to spawn multiple enemies and increase their number every time a wave is defeated. Lastly we will spawn the powerup with every wave, giving the player a chance to fight back against the ever-increasing horde of enemies.
Challenge 4 & Quiz Soccer Scripting	Use the skills you learned in the Sumo Battle prototype in a completely different context: the soccer field. Just like in the prototype, you will control a ball by rotating the camera around it and applying a forward force, but instead of knocking them off the edge, your goal is to knock them into the opposing net while they try to get into your net. Just like in the Sumo Battle, after every round a new wave will spawn with more enemy balls. However, almost nothing in this project is functioning! It's your job to get it working correctly.
Lab 4 Basic Gameplay	In this lab, you will work with all of your non-player objects in order to bring your project to life with its basic gameplay. You will give your projectiles, pickups, or enemies their basic movement and collision detection, make them into prefabs, and have them spawned randomly by a spawn manager. By the end of this lab, you should have a glimpse into the core functionality of your game.

Unit 5 - User Interface (Quick Click Prototype)

UNIT SCOPE

In this Unit, you will program a game where the goal is to click and destroy objects randomly tossed in the air before they can fall off the screen. In creating this prototype, you will learn how to implement a User Interface - or UI - into your projects. You will add a title screen with a difficulty select menu, you will add a score display that will track the player's points, and you will add a Game Over screen, which will allow the player to restart and try again. In learning these skills, you will be able to create a fully "playable" experience.

UNIT SEQUENCE

Lesson 5.1 Clicky Mouse	It's time for the final unit! We will start off by creating a new project and importing the starter files, then switching the game's view to 2D. Next we will make a list of target objects for the player to click on: Three "good" objects and one "bad". The targets will launch spinning into the air after spawning at a random position at the bottom of the map. Lastly, we will allow the player to destroy them with a click!
Lesson 5.2 Keeping Score	Objects fly into the scene and the player can click to destroy them, but nothing happens. In this lesson, we will display a score in the user interface that tracks and displays the player's points. We will give each target object a different point value, adding or subtracting points on click. Lastly, we will add cool explosions when each target is destroyed.
Lesson 5.3 Game Over	We added a great score counter to the game, but there are plenty of other game-changing UI elements that we could add. In this lesson, we will create some "Game Over" text that displays when a "good" target object drops below the sensor. During game over, targets will cease to spawn and the score will be reset. Lastly, we will add a "Restart Game" button that allows the player to restart the game after they have lost.
Lesson 5.4 What's the Difficulty?	It's time for the final lesson! To finish our game, we will add a Menu and Title Screen of sorts. You will create your own title, and style the text to make it look nice. You will create three new buttons that set the difficulty of the game. The higher the difficulty, the faster the targets spawn!
Challenge 5 & Quiz Whack-a-Food	Put your User Interface skills to the test with this whack-a-mole-like challenge in which you have to get all the food that pops up on a grid while avoiding the skulls. You will have to debug buttons, mouse clicks, score tracking, restart sequences, and difficulty setting to get to the bottom of this one.
Lab 5 Swap out your Assets	In this lab, you will finally replace those boring primitive objects with beautiful dynamic ones. You will either use assets from the provided course library or browse the asset store for completely new ones to give your game exactly the look and feel that you want. Then, you will go through the process of actually swapping in those new assets in the place of your placeholder primitives. By the end of this lab, your project will be looking a lot better.