



Create with Code:

Course Syllabus

In this official high school course from Unity, you will learn to Create with Code as you program your own exciting projects from scratch in C#. As you iterate with prototypes, tackle programming challenges, complete quizzes, and develop your own personal project, you will transform from an absolute beginner to a capable Unity developer. By the end of the course - if you want - you will be ready to put your skills to the test on the Unity Certified User Programming Exam. Most importantly, though, when you complete this course, you will have the confidence that you can Create with Code.

Objectives

C# Skills:	Students will gain a foundational knowledge of programming in C# and will feel confident that they can implement new features on their own with this knowledge
Unity Skills:	Students will have the confidence that, given enough time and resources, they could create anything they want in Unity
Project Management:	As students create their own personal projects, they will learn to manage the process from start to finish: outlining their concept, setting project milestones, and tracking progress
Unity Certified User Exam:	Should they choose, students will have the skills and confidence to pass the Unity Certified User Programming exam, earning an official certificate validating their skills




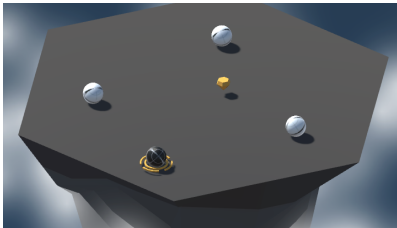
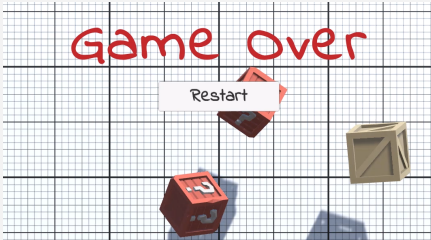
Requirements

Knowledge & Skills	Students are not required to have any prior knowledge or skill with programming, C#, or Unity development
Hardware & Software	Students are required to have access to Unity EdLab or a computer capable of running Unity and a mouse with a scroll wheel. For guidance on setting students up in EdLab see this tutorial

Duration

Independently	If taking the course independently, the full course will take: 40-50+ hours
In a Classroom	If delivering the course to students in a classroom, the full course will take: 50-100+ hours

Course Structure

Unit	Prototypes	Assessments	Personal Project
1	Driving Simulator 	→ Challenge 1 → Quiz 1 → Bonus Features	→ Lab 1
2	Feed the Animals 	→ Challenge 2 → Quiz 2 → Bonus Features	→ Lab 2
3	Run and Jump 	→ Challenge 3 → Quiz 3 → Bonus Features	→ Lab 3
4	Sumo Battle 	→ Challenge 4 → Quiz 4 → Bonus Features	→ Lab 4
5	Quick Click 	→ Challenge 5 → Quiz 5 → Bonus Features	→ Lab 5
Continued Lab work on Personal Projects			

Grading and Rubrics

Overview

- 40%: Prototypes | 5 x 8% each
- 15%: Challenges | 5 x 3% each
- 10%: Quizzes | 5 x 2% each
- 35%: Personal Project | 1 x 35% each

** Note that these weight values are only suggestions*

Prototypes

Weight	40% (5 x 8% each)
Description	Students follow along step-by-step over the course of 4 lessons to create a prototype with the same functionality as the instructor, but with a few of their own creative choices.
Purpose	To teach students all of the concepts and skills they'll need to complete the challenges and quizzes, and to provide examples of core components that they could add to their personal projects

4 - Excellent	3 - Good	2 - Fair	1 - Unsatisfactory
<ul style="list-style-type: none">- Project runs without error- All functionality present and operating as expected- Code and hierarchy are neat & commented, using correct conventions	<ul style="list-style-type: none">- Project runs without error- All functionality present and operating mostly as expected- Code and hierarchy are mostly neat & commented, using correct conventions	<ul style="list-style-type: none">- Project runs with some issues- Some functionality missing, and overall not operating as expected- Code and hierarchy are disorganized, using inconsistent conventions	<ul style="list-style-type: none">- Project barely runs- Most functionality absent- Code and hierarchy are messy, with no sign of consistency in conventions

Challenges

Weight	15% (5 x 3% each)
Description	Students are provided with an incomplete or broken version of a project and tasked with 5 items to implement or resolve, including a couple of Bonus challenges. They are also provided with hints and an example of a completed challenge.
Purpose	To allow students to apply the skills they learned while creating the prototype to a new, but somewhat similar context, solidifying the concepts and extending their skills.

4 - Excellent	3 - Good	2 - Fair	1 - Unsatisfactory
<ul style="list-style-type: none">- All 5 tasks have been completed fully	<ul style="list-style-type: none">- 4 out of 5 of the tasks have been completed	<ul style="list-style-type: none">- 3 out of 5 of the tasks have been completed	<ul style="list-style-type: none">- 2 or less of the tasks have been completed

Quizzes

Weight	10% (5 x 2% each)		
Description	Students complete 10 multiple choice questions.		
Purpose	To give students the opportunity to apply and check their knowledge in a decontextualized environment, which will also help prepare them for the Unity Certified User exam.		

4 - Excellent	3 - Good	2 - Fair	1 - Unsatisfactory
- 9-10 out of 10 correct	- 7-8 out of 10 correct	- 5-6 out of 10 correct	- Less than 5 out of 10 correct

Personal Project

Weight	35% (1 x 35% each)
Description	Students conceptualize, plan, and complete their own personal project throughout the course, integrating features they learned during the prototypes and extending them beyond. Students will be evaluated on completeness and uniqueness of their project.
Purpose	To give students an opportunity to <i>extend</i> their skills to a project that is uniquely their own, further solidifying the skills they learned and giving them the confidence that they can create whatever they want with the power of Unity and C#.

4 - Excellent	3 - Good	2 - Fair	1 - Unsatisfactory
<ul style="list-style-type: none">- Project contains all of the features outlined in their project plan- Stayed on track with their planned milestones- Used their Unity and/or C# skills in a novel and creative ways- Code and hierarchy are neat & commented, using correct conventions	<ul style="list-style-type: none">- Project contains most of the features outlined in their project plan- Stayed mostly on track with their planned milestones- Used their Unity and/or C# skills in new, but not necessarily creative ways- Code and hierarchy are mostly neat & commented, using correct conventions	<ul style="list-style-type: none">- Project contains a few of the features outlined in their project plan- Did not really stay on track with their planned milestones- Did not use their Unity or C# skills in any new ways- Code and hierarchy are disorganized, using inconsistent conventions	<ul style="list-style-type: none">- Project does not contain any of the features outlined in their project plan- Did not stick to their planned milestones at all- Did not demonstrate an ability to apply skills they learned in the course- Code and hierarchy are messy, with no sign of consistency in conventions

Growth Mindset

Derived from the work of Stanford University Professor Carol Dweck, “growth mindset” is the understanding that we can incrementally develop our abilities and intelligence indefinitely through practice and effort. The

opposite of a growth mindset is a “fixed mindset”, the belief that your intelligence and abilities are fixed or limited. A fixed mindset can be dangerous for learners because it can cause them to give up. Growth mindset underpins the desirable learning behaviors of resilience and grit, i.e. perseverance when learning becomes difficult. It is the ability to self-regulate and work through problems in learning, rather than giving up. Even when we fail, we learn.

When learners push themselves out of their comfort zones to learn something challenging and stick to it, neurons in the brain form new, stronger connections. Unity is a great tool for developing resilience as it provides challenges, interesting problems to solve, and opportunities to fail and grow.

Growth	Mixed	Fixed
<ul style="list-style-type: none"> - You look forward to the next challenge. - Saw mistakes as temporary setbacks, something to be overcome. - Reflected on what you learned and apply that learning. - Invited feedback and criticism and applied it to improve your project. - Used different strategies and asked others about their strategies. - Showed stamina while working on your project until it was complete. - Asked specific questions, including questions about your own thinking, and challenged others' ideas. - Showed confidence in taking risks and happily shared mistakes you made and what you learned. 	<ul style="list-style-type: none"> - Took on challenges after having success in related challenges. - Motivated by feedback when it wasn't too critical and if you are comfortable with the person providing feedback. - Open to strategies that help to meet a challenge but tend to work on things you are already “good at.” - Persevered with prompting and support. - Asked questions about things in Unity that you feel more confident with but less likely to do so if it is outside your comfort zone. - Took risks if the task was already fairly familiar to you. 	<ul style="list-style-type: none"> - Did not want to take on challenges alone. - Saw mistakes and failures as proof the task was too difficult and not worth pursuing. - Avoided feedback or saw it as a reason to give up. - Did not demonstrate any effective strategies for accomplishing project tasks or goals. - Showed little or no persistence through challenges. - Did not ask questions or seek guidance and support. - Did not take risks