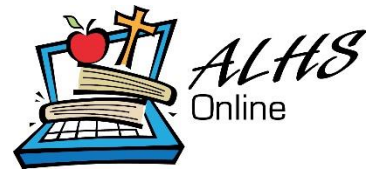


# Game Programming 1

## Course Syllabus 2020-21



<b>School:</b>	ALHS Online
<b>Instructor:</b>	Mrs. Alta Mae Tacke
<b>Level:</b>	High School
<b>Start Date:</b>	January 6, 2020
<b>End Date:</b>	May 19, 2021
<b>Prerequisites:</b>	Introductory programming or consent of instructor.

### Course Description:

This is a course for students with little or no experience in writing computer code. Using the concept of learning by example, students will have the opportunity to manipulate a game manually and then transition into writing the code required to create and execute simple 2D and 3D games. The project(s) developed over the length of this course will encourage students to consider using these skills to pursue a career in technology or to have an impact on the learning of others.

### Course Goals:

- This course will give students the opportunity to thank a gracious God for the skill sets needed to work in the technology structure of computer programming.
- This course will encourage and support students' organizational and computer skills that will develop a foundation for living and working in the environment of the 21<sup>st</sup> Century.
- This course will provide a challenge for students to design God-pleasing computer games that will entertain and instruct without violence or inappropriate language.
- This course will build confidence in using the Unity interface, making it second nature, as they transition into the next semester of more complex game programming.
- This course will give students a foundation for considering the role that game programming can play in providing instruction in education, both academically and spiritually.

### Student Outcomes:

Upon successful completion of the course, students will:

- Understand the concept of a Game Engine platform.
- Demonstrate knowledge of the Unity Game Engine interface and the components that comprise the Integrated Development Environment (IDE) of Unity.
- Demonstrate the ability to use basic concepts and execute them to code simple processes, create simple 2D and 3D games.
- Demonstrate the ability to export games to the web or a browser.
- Use the knowledge of programming structure, keywords, and operators to explore the prewritten code of Unity to appreciate and understand the back side of the programming environment.
- Demonstrate a basic knowledge of scripting syntax for programming in general and game programming specifically.
- Debug simple games and activities that demonstrate programming skills learned.
- Demonstrate the application of 'commenting code' to debug and identify errors in the code, and to describe to other readers of the code what the developer intended.

## Course Outline:

- Unit 1 Game Engine –  
It is What Makes the Game Go  
The Unity Interface – Explore and Create  
Create and Navigate Scenes in 3D Games**
- Unit 2 C# Scripting Language –  
Communicating with the Game Engine  
Adding Interaction and an Interface**
- Unit 3 Polishing Our Game**
- Unit 4 Creating a Simple AI (Artificial Intelligence)**
- Unit 5 C# Review and Two Final Projects**

## Course Materials/Resources:

---

### Software download required

- Unity Hub with Unity 2019.3.13 and WebGL
- Visual Studio Editor (Included with Unity Hub)

### Textbook/Tutorials

**School Bundle consisting of the following pdf books to be downloaded from the instructor's Google Drive:**

- Unity from Zero to Proficiency (Foundations)
- Unity from Zero to Proficiency (Beginner)
- Unity from Zero to Proficiency (Intermediate) (Sem 2)
- Unity from Zero to Proficiency (Advanced) (Sem 2)
- Unity from Proficiency to Mastery (C#)
- Unity from Proficiency to Mastery (Artificial Intelligence)

**Video Courses (accessed with a user name and password – to be assigned on enrollment in the course)**

- [The Video Course: Unity from Zero to Proficiency \(Foundations\)](#)
- [The Video Course: A Beginner's Guide to 2D Platform Games \(2 hours\)](#)
- [The Video Course: A Beginner's Guide to 2D Shooter Games \(2 hours\)](#)
- [The Video Course: A Beginner's Guide to 2D Web and Mobile Games \(2 hours\)](#)
- [The Video Course: Getting Started with 3D Animation in Unity](#) (Sem. 2)
- [The Video Course: Mastering C# and Unity \(Full course\)](#)

All books are available to enrolled students and can be downloaded and saved on a flash drive for use during the course and beyond. All other books and videos published by the author until 30th August 2021 will be added to the School Bundle on the instructor's Google Drive for access and download by enrolled students. The cost of these resources is negotiated by the instructor. Cost to enrolled students will be between \$40-\$50.

### Tutorials at

[www.learntocreategames.com](http://www.learntocreategames.com)

Getting Started tutorials and videos : <https://unity3d.com/learn>

### Hardware requirements:

- Windows PC with Windows 10 (64-bit version)
  - Video card supporting DirectX10
  - (Ensure your graphics card is using the latest driver)
  - Shader Model 4.0
  - Alt: macOS 10.12.6
  - Mac: Xcode 9.4 or higher
-

### Course Procedures:

- Course lessons and assigned materials provide a step-by-step approach with examples and plenty of graphics that show students exactly what to do.
- Tips to maximize workflow are included in the materials.
- **Self-paced learning is a goal for all students based on the ability and motivation to work independently and meet assessment guidelines and deadlines.**
- Challenges and projects are used to perfect and assess skills learned in each chapter or unit.
- Quizzes will test knowledge at the end of each chapter or unit.
- Critical thinking will be encouraged with frequently asked questions organized by themes.

Students will complete course work that includes quizzes, demonstration activities, discussions, writing activities and programming activities. The final weeks' assessment will use a project from Unity's Learn Tutorial that will demonstrate proficiency of skills learned to date and to become familiar with the concept of Artificial Intelligence. All content covered throughout Programming 1 will be included in the final week's activities.

### Daily Routine and Grading:

- Students are expected to use their scheduled period working on the assignments for this course. Time outside the classroom is usually required to complete the assigned projects. Students will be able to work on a computer other than the school computer if the hardware and software requirements are met, and the solution files are saved to a flash drive or external portable hard drive. A student laptop meeting the hardware requirements and having the software installed is the most efficient way to take the course. It should have at least 8GB of memory (RAM).
- First quarter grade will be calculated as close to the first quarter grading period as possible depending on the individual school calendar (app. January 6 – March 12, 2021). Second quarter grade and final grade will be calculated after completion of the course May 19.
- Enrolled students represent several schools with schedules that may not mesh with the timeline of this course. Adjustments in due dates to reflect school breaks and days off will be made to the calendar as needed.
- Each assignment and activity will have a point value. A student's grade will be calculated on percentage of points possible and converted to a letter grade based on the following scale:

Letter Grade	Percentage	Letter Grade	Percentage
A+	98-100%	C	73-76%
A	93-97%	C-	70-72%
A-	90-92%	D+	67-69%
B+	87-89%	D	63-66%
B	83-86%	D-	60-62%
B-	80-82%	F	0-59%
C+	77-79%		

- Students are expected to abide by the attendance policy of their respective school. Communication with the instructor is important if there are absences due to illness or other circumstances.
- This is a sequential project-based course. All assignments must be completed to move on to the next step in the process. Timely submission of assignments will serve to help the student demonstrate accountability and develop organizational skills.
- To meet the goal of self-paced learning, due dates will be set. Assignments not turned in by the due date will receive a grade of zero and commented as missing. Full credit is restored if submission is made within 24 hours of due date time. Additional late days are graded at 5% reduction of points earned for each day late.

- Bonus points for early submissions can be used to offset any points lost on an assignment or project. Those points are recorded by the instructor as a Bonus Points Extra Credit line in the Moodle Grade book. Bonus points are calculated as 5% of points earned with a maximum of 5 points per assignment.
- Cheating is a sin. Cheating can be defined either as copying or using content created by someone else. It is a violation of God's will and prevents the student from truly learning. The nature of this course is very hands on and collaborative. Collaboration to develop solutions is permissible. Submission of work that is at least in part, not your own, is not acceptable. Students are expected to use the forums to discuss syntax, programming structure and possible solutions to solve a programming problem. Any obvious cheating will be dealt with through the ALHS class supervisor.

### **ALHS Online Policies:**

Current ALHS Online policies are listed in the *Handbook for ALHS Online Students and Parents*, available on the [ALHSO.org](http://ALHSO.org) website. This includes policies on non-discrimination, anti-harassment, student expectations, attendance, academic honesty, student discipline, student grades, course add/drop, etc.

Please note the policy on **class attendance** which states in part:

***“Even if a student’s local school does not have school on a particular day (snow day, teacher’s conference, quarter break, choral fest, class trip, etc.) ALHS Online courses will continue to meet and students are expected to complete required work on time.”***

Students also fall under the policies of the school where they attend as a full-time student. When applicable, these same local school policies will be applied to enrollment in this ALHSO course.