Fairfax Collegiate

703 481-3080 · www.FairfaxCollegiate.com

C# Game Programming 7-9 Syllabus

Course Goals

1 Development Interface
Students learn to navigate the Unity 3D development environment.

2 C# Coding
Students interpret and write C# scripts to manipulate game objects.

3 Assets
Students learn to design their own assets and use open-source assets in game development.

4 Creative Game Design
Students apply their knowledge from the course to add original ideas to games they developed.

Course Topics

1 Unity 3D Interface
Students learn the components of the Unity 3D development environment, including the hierarchy, scene view, project browser, and object inspector.

2 Asset Manipulation
Students manipulate assets by changing their physical properties including shape, size, rotation, color and texture.

3 Using Physics
Students implement physics into their games by adding gravity and friction to objects.

4 C# Scripting
Students learn to code with basic C# and add functionality to their game objects.

5 Maze Project
Students combine techniques learned to create their first multi-scene game involving a ball and a maze.

6 Space Shooter Project
Students use Unity and C# knowledge and open-source assets to develop an outer space game.

7 Advanced Creative Design
Students expand upon their favorite game in a creative way to add a personal touch.

Course Schedule

Day 1

Introduction to Programming
Students receive an overview of what programming is and what programs are used for.

The Unity Interface
Students receive an overview of the panels in Unity and what they are used for.

Building Scenes
Students build their first scene in Unity while learning about the tools available.

Creating an Object
Students add an object to their first scene and learn to maneuver around the scene.

Transforming Objects
Students learn how to move, rotate, and scale an object.

Snapping
Students learn how to snap object positions and sizes and determine when snapping is useful.

Using the Hierarchy
Students learn to use Unity's hierarchy to organize their game objects and manipulate multiple at once.

Scene Gizmo
Students navigate through a scene with the Scene Gizmo.

Lighting
Students learn how to add lighting and textures to their scenes.

Additional Resources
Students review additional resources that can be used.

Day 2

Game Physics
Students learn to control how physics affects interactions between objects in their games.

Gravity and Collisions
Stuents learn how to have objects interact with the Nvidia physics engine by using the RigidBody component.

Physic Materials
Students use Physic Materials to change the way objects in their game can bounce and slide.

Physics Challenges
Students complete a series of challenges involving the Unity Physics game engine and materials.

Intro to Scripting
Students learn how to control their game's behavior with scripts.

Day 3

Variables
Students learn how to control game behavior through the use of variables.

Adding Interaction
Students learn how to add an interaction component to their game, particularly with mouse clicks and console messages.

Supplement on Scripting
Students explore the documentation on scripting.

Scripting Challenges
Students work through a series of programming challenges to practice what they've learned.

Day 4

Roll-a-Ball Project
Students create a project in which they design a maze to roll a ball around in and pick up floating cubes. This involves writing some C# code as well as using GUI tools, physics elements, assets, and the Inspector.

Setting up the Game
Students set up the initial game area for the Roll-A-Ball game.

Moving the Player
Students set up keyboard control to move their player around the playing field.

Setting up the Camera
Students make the camera follow the player in order to get a full view of the playing field.

Setting up the Maze
Students learn to create walls to create a defined maze area.

Creating Collectibles
Students create collectible objects using Prefabs, allowing them to manipulate multiple similar game objects at once.

Day 5

Collecting the Collectibles
Students learn how to handle collisions in their programs in order to have their player collect the pickups.

Displaying Count and Text
Students learn how to display information in a User Interface on screen in their game. Students use this to display the game's score and a win message.

Building and Beyond
Students extend, build, and deploy their games for use on PC, Mac, or Linux machines.

Day 6

Space Shooter Game
Students create a project where they design a space shooter scene involving a moving space ship that shoots lasers, moving asteroids that act as enemies, and a score counter. This involves the students writing C# scripts, using game physics, prefab objects, and the Inspector.

Setting Up the Environment
Students create a new Space Shooter game and import the necessary pre-made assets. They set up the proper dimensions and resolution for the scene.

The Player GameObject
Students add the Player ship to their scene and learn how to use prefab objects

Camera and Lighting
Students position their camera correctly and add lighting to the game.

Game Background
Students add a background to their scene to give it a more space-like feel.

Moving the Player
Students begin with scripting in C# to move their player ship around the game area.

Day 7

Shooting Shots
Students use prefab objects and C# scrips to make their Player ship shoot lasers.

Boundary
Students create a boundary around their game playing area using C# scripts and game assets.

Creating Hazards
Students use prefab objects and scripts to add enemy asteroids to their game.

Explosions
Students create explosions in their game on contact with an asteroid or a laser.

Game Controller
Students begin writing a script that will control the movement of asteroids in their game.

Day 8

Spawning Waves of Asteroids
Students design scripts to unleash spawn waves of asteroids at their Player.

Points and Scoring
Students add points and scoring to their game using interactions with the GameController script.

Ending the Game
Students finish their Space Shooter Game by adding text and adding a game over script.

Final Project
Students plan, design, and program an interactive game.

Day 9

Final Project
Students plan, design, and program an interactive game.

Day 10

Final Project
Students plan, design, and program an interactive game.

Wrap-Up
Students wrap up the class and take home flash drives with their game files on them.

Updated on 3/22/2019