**Game Design**

Performance Task

# **Create — Player Control**

## **Overview**

In this performance task, students will program a car moving side-to-side on a floating road, trying to avoid (or hit) obstacles in the way. In addition to becoming familiar with the Unity editor and workflow, you will learn how to create new C# scripts and do some simple programming. By the end of the Unit, you will be able to call basic functions, then declare and tweak new variables to modify the results of those functions. You will also explore the game design industry and find out what it means to work in the game industry.

## **Assessment**

You will be provided with 16 hours of class time to complete and submit the following:

* A video of your game being planned or a trailer for the game
* Written responses about your game and design process
* Digital Game Prototype

Your teacher will share submission guidelines that include suggestions for creating video and PDF files.

## **General Requirements**

You are required to:

* Define control structures and boolean expressions
* Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
* Identify data types and memory models
* Define events, event handler and validation in terms of algorithms.
* Identify the cause of an incorrect behaviour (i.e., bug) in a program.
* Identify the rationale or purpose behind programs and digital tools commonly used. Explain how programs are built for a purpose.
* Identify the basic principles of design thinking (i.e., empathy, brainstorming, prototyping, testing, iterating).
* Write responses to questions about your game.
* Include your entire prototype.

## **Submission Requirements**

### 1. **Video**

Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of your game. Your video must not exceed 1 minute in length and must not exceed 30MB in size.

### 2**. Written Responses**

Submit one PDF document in which you respond directly to each prompt. Clearly label your responses. Your response to all prompts combined must not exceed 750 words.

You may use images to show off or to explain things in your written responses.

## **Purpose and Development**

1. Provide a written response or audio narration in your video that:

Identifies the changes you made to your prototype. Identifies the changes.

* Explains what the video illustrates.

(Approximately 150 words)

1. Identify the cause of an incorrect behaviour (i.e., bug) in a program. Identify the rationale or purpose behind programs and digital tools commonly used. Explain how programs are built for a purpose. Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

(Approximately 200 words)

1. Identify the basic principles of design thinking (i.e., empathy, brainstorming, prototyping, testing, iterating).

(Approximately 200 words)

1. Define control structures and boolean expressions. Identify data types and memory models. Define events, event handler and validation in terms of algorithms. (Approximately 200 words)

## Tasks

### **Activity 1 - Explore**

**Description**

Students learn about careers in the video game industry. They’ll watch videos highlighting various careers and opportunities in the industry and complete self-reflection assignments to consider various careers. students learn about video game studios and research a studio of their choice. Students then develop a one-page web presentation to share their findings and reflect on types of roles within game studios.

students learn about video game studios and research a studio of their choice. Students then develop a one-page web presentation to share their findings and reflect on types of roles within game studios.

* Careers in Game Design
* Scavenger Hunt
* How to get into Game Design

Time To Complete: 1-3 Hours

### **Activity 2 - Research**

**Description**

Define keywords related to programming. What is a function? What is a variable? How do we define a variable and use it in code. How do we use unity to create games and manipulate objects?

* Introduction to Unity
* Unity Basics
* Scripting Basics

Time To Complete: 1-2 Hours

### **Activity 3 - Ideate**

**Description**

Students dive deeper into common elements that all fun games possess. Students 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.

* What makes a good Game
* Project - Game Design Document

Time To Complete: 1-2 Hours

### **Activity 4 - Evaluate**

**Description**

Students will self evaluate their game idea, and then peer review the idea looking for input and helpful feedback to finalize the idea before building the prototype.

Time To Complete: 1-2 Hours

### **Activity 5 - Construct a Prototype**

**Description**

Students 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.

Students 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.

Keep your eyes on the road! 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.

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.

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.

* Start your 3D engine
* Pedal to the Metal
* High Speed Chase
* Step into the Driver’s Seat
* Challenge 1

Time To Complete: 6-8 Hours

### **Activity 6 - Improve the Design**

**Description**

Once the prototype is built the students will now test the games. Students should play 2-3 games and provide useful feedback to the creator. Students will then take their feedback and plan how to improve their game, play test it once more and get some final feedback.

* Extras

Time To Complete: 2-3 Hours

### **Activity 7 - Share**

**Description**

Students will share their feedback, prototype and whole design process with the class in a formal presentation, or gallery walk.

Time To Complete: 1-2 Hours

### **Activity 8 - Reflect**

**Description**

Students will reflect on their design process and how they went about creating their game, what struggles they encounter and how what worked and didn’t work for them. They will submit a video of game play, along with their written reflection.

Time To Complete: 1 Hour