

# Unity Workshops

## Certified Programmer Exam Prep: 2017.4



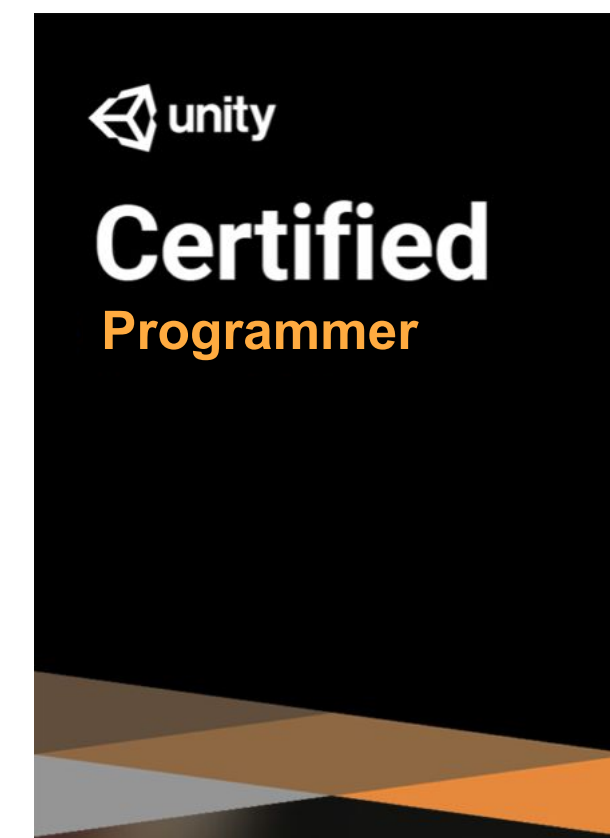
# Workshop Overview



This workshop covers the following four topics over three days of study:

- Core Interaction Programming
- Application Systems Programming
- 3D Interactions, Cameras, and Navigation
- 3D Art & Audio Pipeline

By the end of this workshop, participants will be prepared to take the Unity Certified Programmer Certification Exam.




# Core Interaction Programming

---

By the end of this series of activities, you will be able to:

- Analyze and structure scripts required to implement behaviors and interactions of GameObjects and environments
- Recognize techniques for structuring scripts for modularity, readability, and reusability.
- Identify methods to implement inputs and controls, and GameObject instantiation, destruction, and management.
- Evaluate errors and performance issues using tools such as the Unity Profiler.
- Recognize concepts associated with uses and impacts of version control.
- Interpret scripts for application interface flow, such as menu systems, UI navigation, and application settings.
- Interpret scripts for user-controlled customization, such as character-creators, inventories, storefronts, and in-app purchases.
- Analyze scripts required to implement user progression features such as scoring, leveling, and in-game economies.
- Analyze scripts required to implement 2D overlays such as heads-up displays (HUDs), mini-maps, and advertisements.
- Demonstrate knowledge of developer testing and its impact on the software development process.

The logo for 'Asteria X' is displayed against a dark, starry space background. The word 'Asteria' is in a stylized, italicized font, and the 'X' is a large, bold, geometric shape. Both are outlined in a bright blue, glowing neon-like light. A small blue sphere is positioned between the 'Asteria' and the 'X'.A rectangular button with a blue gradient and a glowing blue border. The word 'Start!' is written in a bold, white, italicized font in the center of the button.

**Start!**

# Activity 1

## Core Interaction Programming

Implement movement and firing while practicing collaboration and version control



Form work groups of 3-4 people



Set the order and times for mobbing



Implement movement and firing



Discuss and compare approaches

# Activity 2

## Core Interaction Programming

Implement Screen Wrapping  
and Turret Aiming



Form work groups  
of 3-4 people



Set the order and  
times for mobbing



Implement aiming  
and screen  
wrapping.

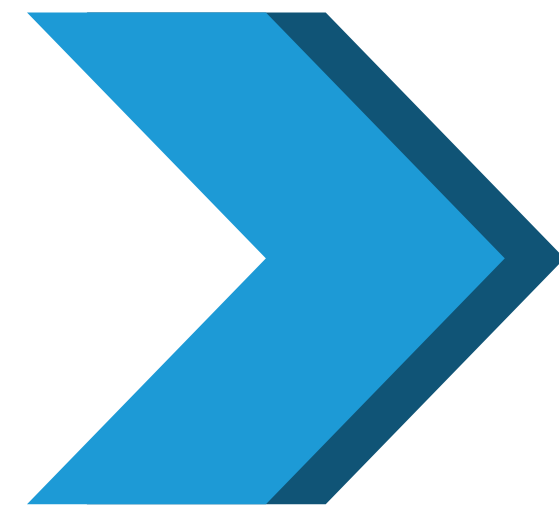


Discuss and  
compare  
approaches

# Activity 3

## Core Interaction Programming

Implement asteroids and their behaviors



Review the Requirements Document to understand asteroid behavior



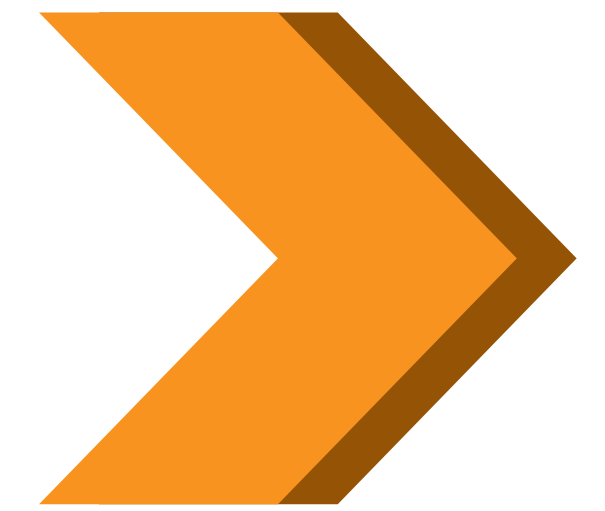
Set the Turn Order for



Implement asteroid creation, behaviors, and destruction



Try the Bonus Challenge as time allows



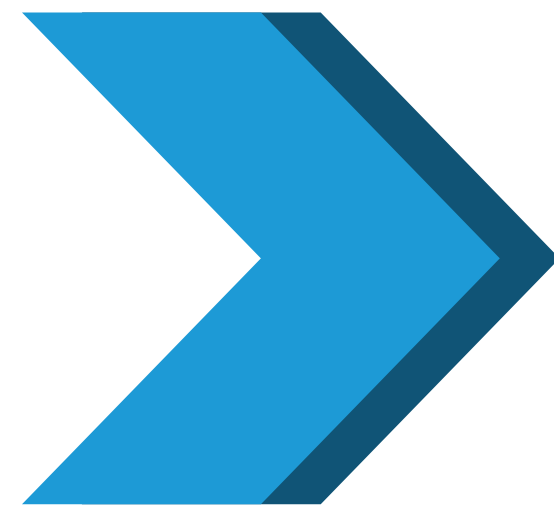
Discuss and compare approaches



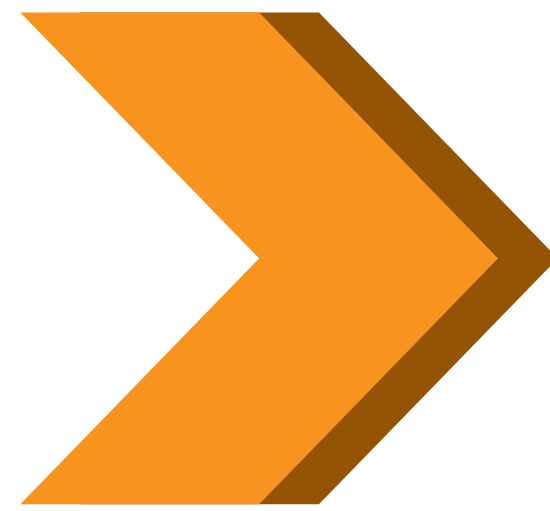
# Activity 4

## Core Interaction Programming

Implement points, jumps, and UI



Implement a Heads-Up Display (HUD) to show the player points and lives.



Implement the teleportation of the PlayerShip each time a life is lost



Make a copy of the completed Unity folder on each team member's machine



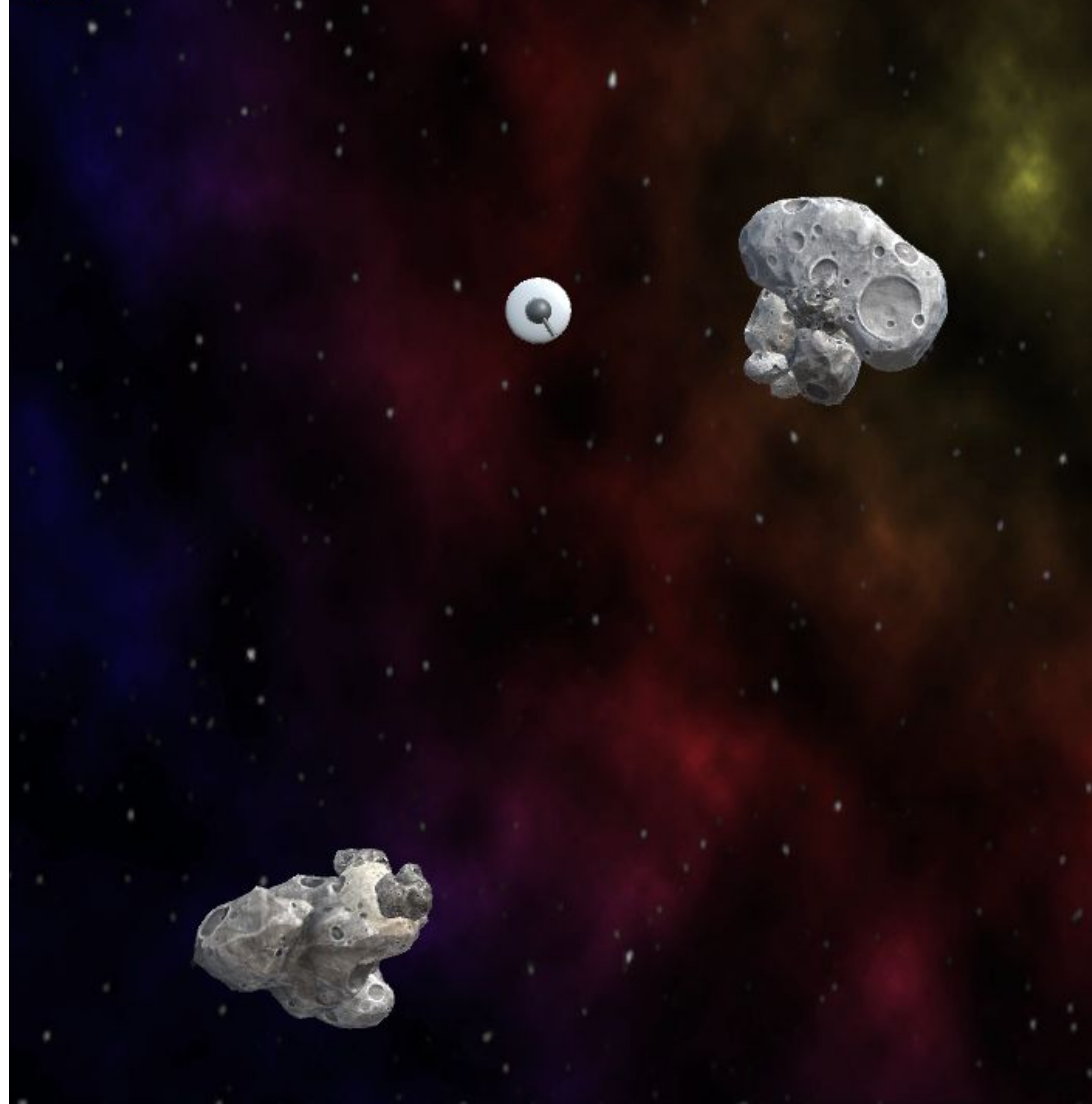
Wrap up the first Activity

# Application Systems Programming

---

By the end of this series of activities, you will be able to:

- Analyze scripts required to implement behaviors and interactions of GameObjects and environments.
- Demonstrate foundational knowledge of Particle Systems and how they can be controlled with Unity's particles API.
- Identify methods for implementing GameObject instantiation, destruction, and management, and inputs, controls, camera views, and movement.
- Recognize techniques for structuring scripts for modularity, readability, and reusability.
- Interpret scripts for application interface flow such as menu systems, UI navigation, and application settings.
- Analyze scripts required to implement user progression features, such as scoring, leveling, and in-game economies.
- Analyze scripts required to implement 2D overlays, such as heads-up displays (HUDs), mini-maps, and advertisements.
- Identify scripts for saving and retrieving application and user data.
- Interpret scripts for user-controlled customization, such as character-creators, inventories, storefronts, and in-app purchases.
- Analyze scripts required to implement 2D overlays, such as heads-up displays (HUDs), mini-maps, and advertisements.
- Demonstrate foundational knowledge of Materials, Textures, and Shaders and how they can be controlled with Unity's rendering API.
- Identify optimizations to address requirements for specific build platforms and/or hardware configurations.

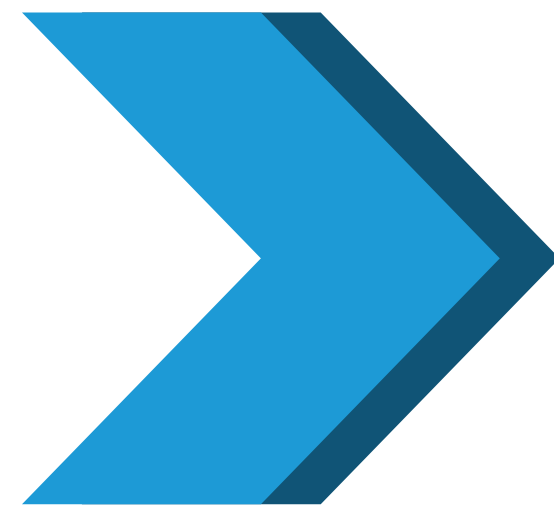




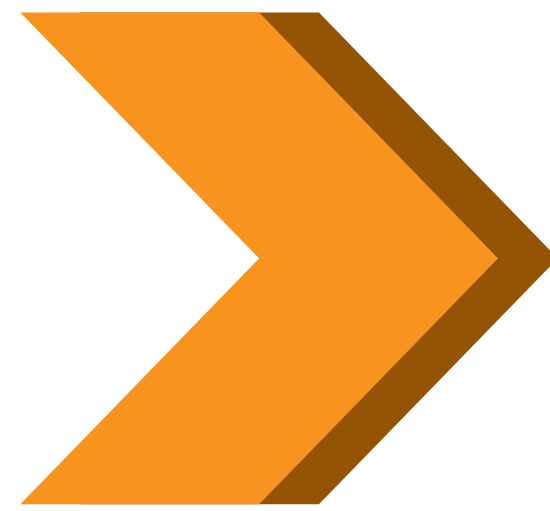
# Activity 1

## Application Systems Programming

Implement particle effects  
and explosions



Import starting  
assets and observe  
starting point



Create the particle  
effects



Try the Bonus  
Challenge as time  
allows

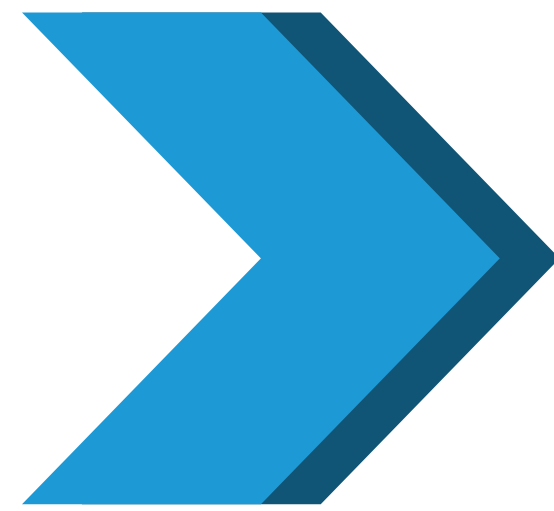


Discuss and  
compare  
approaches

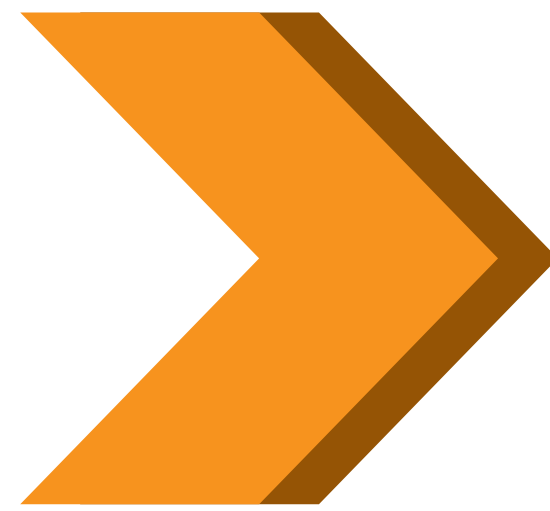
# Activity 2

## Application Systems Programming

Create multiple levels and pause



Create 10 levels of difficulty



Implement a pause option



Discuss and compare approaches



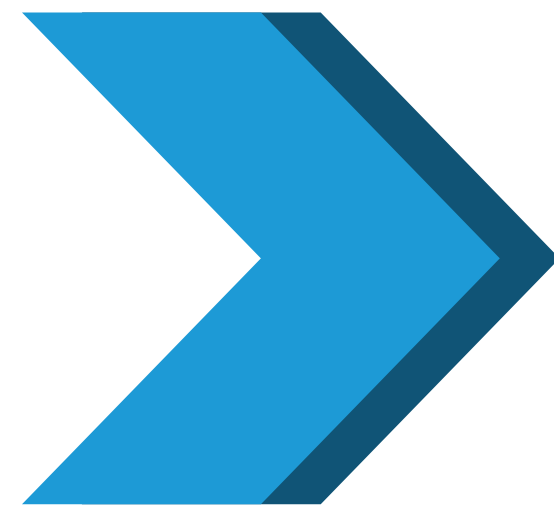
Prepare to Implement Achievements



# Activity 3

## Application Systems Programming

### Achievements



Reference  
Requirements Doc  
for achievement list



Build an  
Achievement  
Manager and Popup  
notification

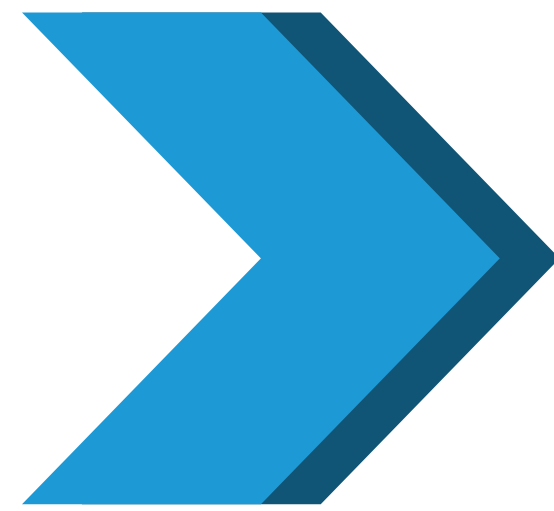


Discuss and  
compare  
approaches

# Activity 4

## Application Systems Programming

Save information locally



Discuss what elements are necessary to create a usable local save function



Implement saving, loading, and deleting game data



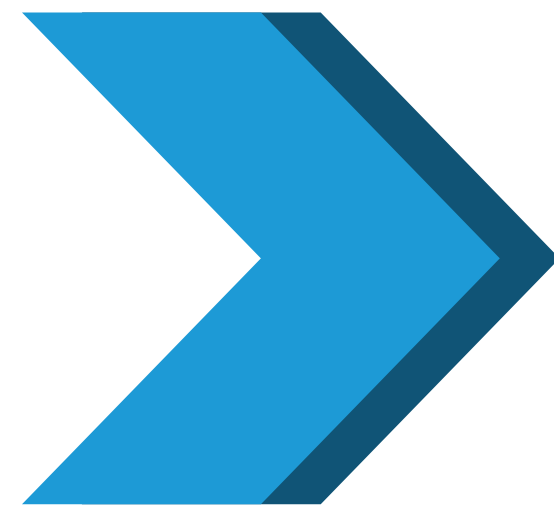
Discuss and compare approaches



# Activity 5

## Application Systems Programming

Create UI for players to customize their ship



Create UI to allow player to select from four Body and four Turret ShipParts



Write a script to replace parts on PlayerShip with those selected by the player



Discuss and compare approaches

# Activity 6

## Application Systems Programming

Implement Unity Analytics and Remote Settings



Download Analytics  
Setup starter  
project and set up  
Unity Analytics



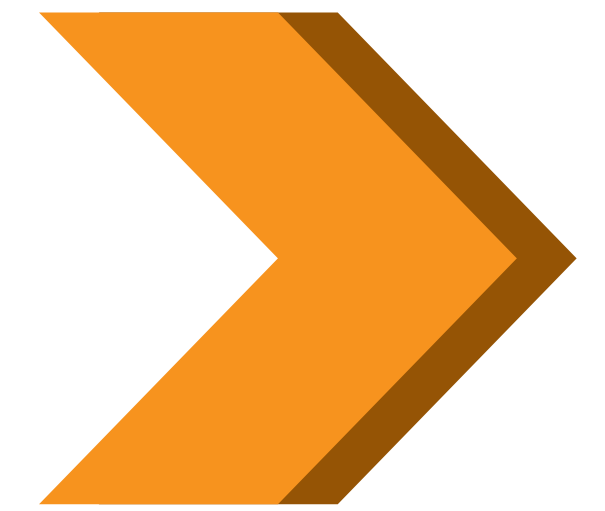
Work in pairs to to  
create Standard  
Event and Custom  
Event calls to Unity  
Analytics



Include a dictionary  
entry for each part  
type



Add Remote  
Settings by creating  
a levelProgression  
key



Discuss results in  
pairs



# 3D Interactions, Cameras, and Navigation

---

By the end of this series of activities, you will be able to:

- Analyze scripts required to implement behaviors and interactions of GameObjects and environments.
- Determine scripts for pathfinding with the Unity navigation system.
- Demonstrate foundational knowledge of 2D and 3D animation and how they can be controlled with Unity's animation API.
- Identify methods to implement inputs and controls.
- Identify methods to implement camera views and movement.
- Identify methods for implementing GameObject instantiation, destruction, and management.
- Demonstrate foundational knowledge of Materials, Textures, and Shaders and how they can be controlled with Unity's rendering API.
- Demonstrate foundational knowledge of lighting and how it can be controlled with Unity's lighting API.
- Recognize techniques for structuring scripts for modularity, readability, and reusability.



# Activity 1

3D Interactions,  
Cameras, and  
Navigation

Implement navigation of an  
enemy sentry



Set up the patrol  
system: bake a  
NavMesh and  
implement the robot  
as a NavMeshAgent



Write a Waypoint  
script to indicate  
the position and  
orientation of the  
waypoint



Implement the  
movement  
sequence of the  
EnemyBot



Implement the  
enemy animation  
using the Animator.  
CrossFade()  
method



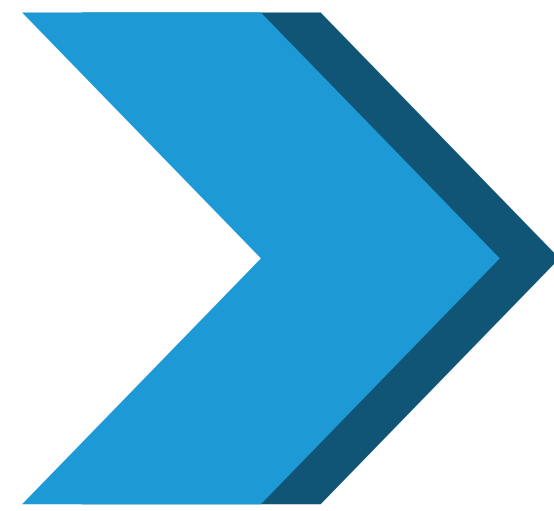
Try the Bonus  
Challenge as time  
allows



# Activity 2

3D Interactions,  
Cameras, and  
Navigation

Control the camera



Modify the  
StealthPlayer-  
Camera script to  
move to look down  
hallways



Try the bonus  
challenge as time  
allows

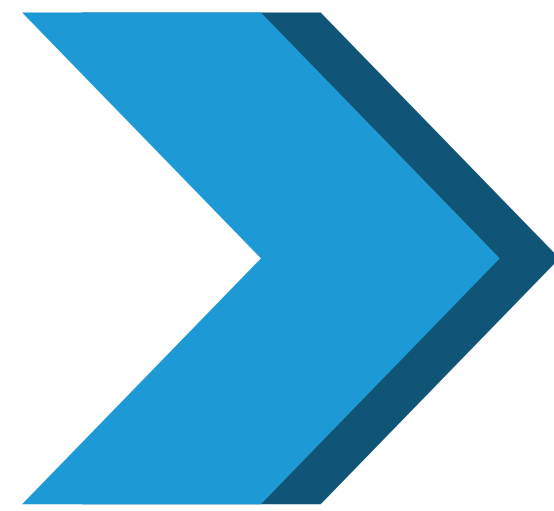


Discuss and  
compare  
approaches

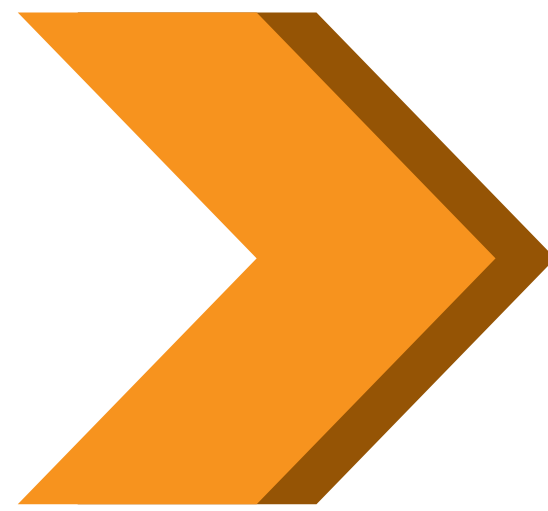
# Activity 3

3D Interactions,  
Cameras, and  
Navigation

Add environmental  
interactions



Implement a  
SecurityGate



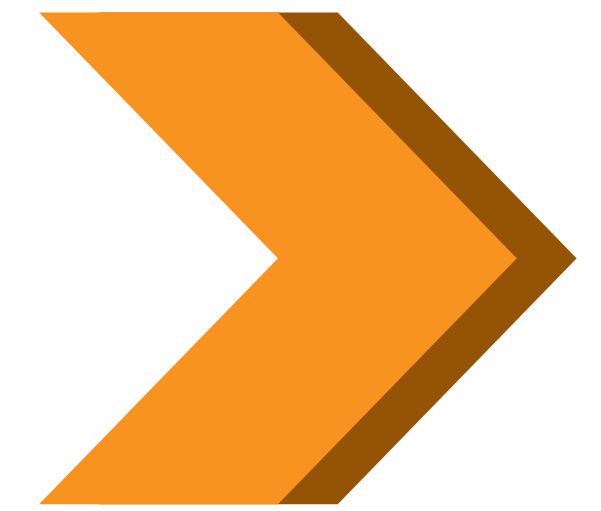
Implement a Desk  
panel to disable the  
SecurityGate and  
automated  
SecurityCameras



Make volumetric  
lights on the  
SecurityCameras  
and enemy robot  
sense the Player



Try the bonus  
challenge as time  
allows



Discuss and  
compare  
approaches

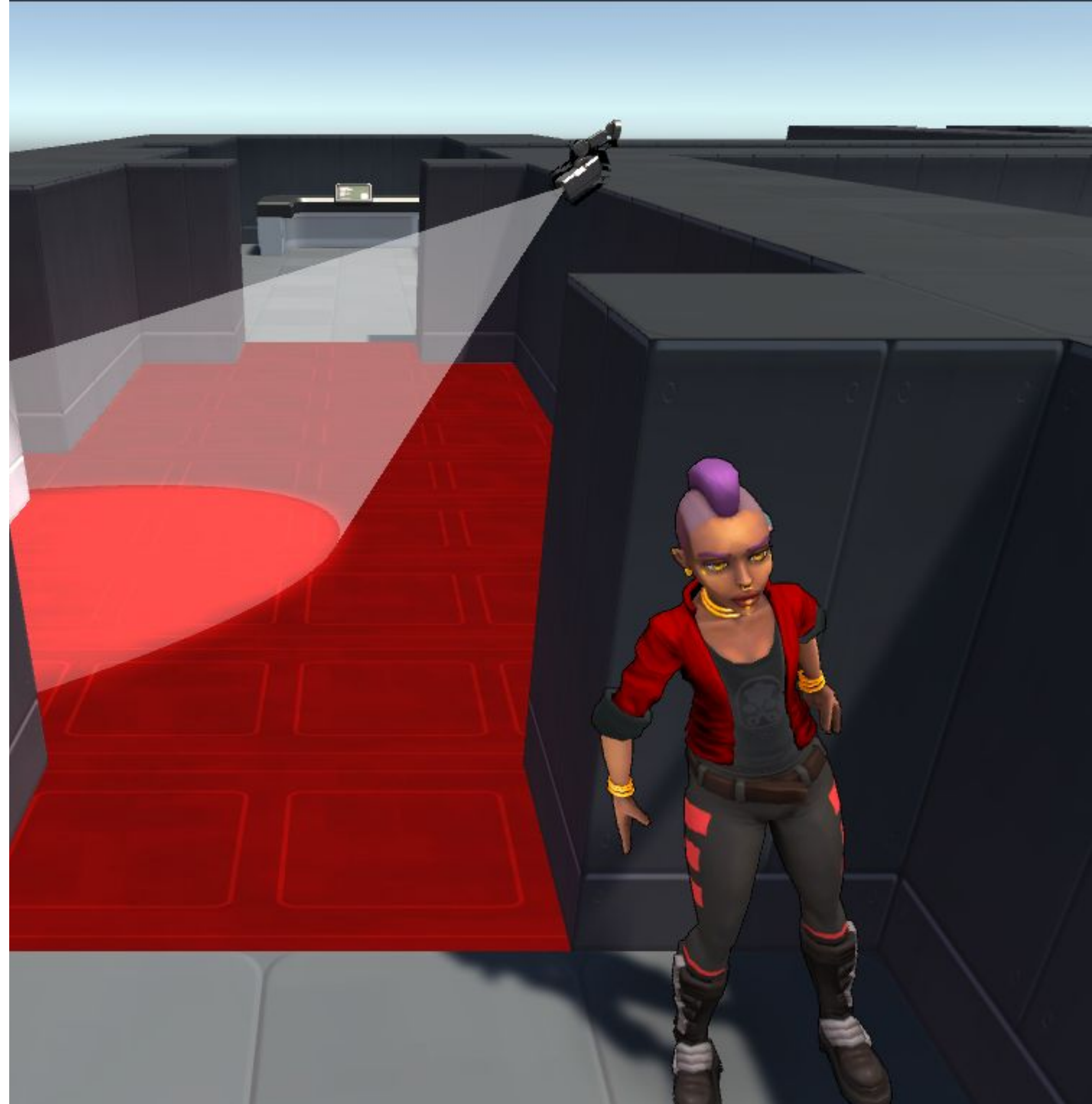


# 3D Art & Audio Pipeline

---

By the end of this series of activities, you will be able to:

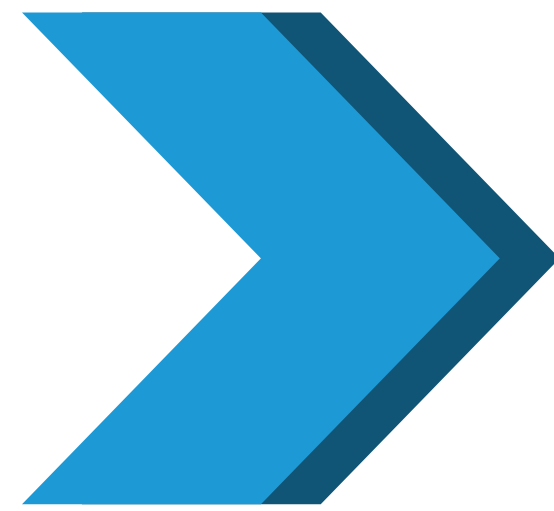
- Analyze scripts required to implement behaviors and interactions of GameObjects and environments.
- Demonstrate foundational knowledge of Materials, Textures, and Shaders and how they can be controlled with Unity's rendering API.
- Demonstrate foundational knowledge of lighting and how it can be controlled with Unity's lighting API.
- Demonstrate foundational knowledge of 2D and 3D animation and how they can be controlled with Unity's animation API.
- Identify methods for implementing GameObject instantiation, destruction, and management.
- Determine scripts for pathfinding with the Unity navigation system.
- Recognize techniques for structuring scripts for modularity, readability, and reusability.
- Determine scripts for implementing audio assets.
- Interpret scripts for application interface flow, such as menu systems, UI navigation, and application settings.



# Activity 1

## 3D Art & Audio Pipeline

Create a Red Alert



Import assets and  
project files



Implement the  
changes that occur  
when Alert Mode is  
activated and  
deactivated



Discuss and  
compare  
approaches

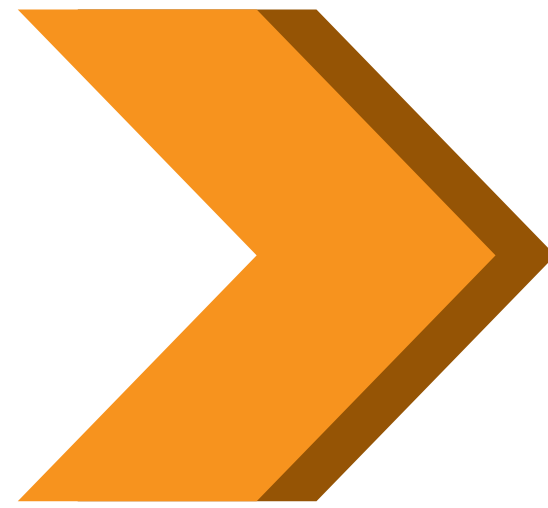
# Activity 2

## 3D Art & Audio Pipeline

Add audio effects



Set up the Audio  
Listeners and  
Reverb Zones



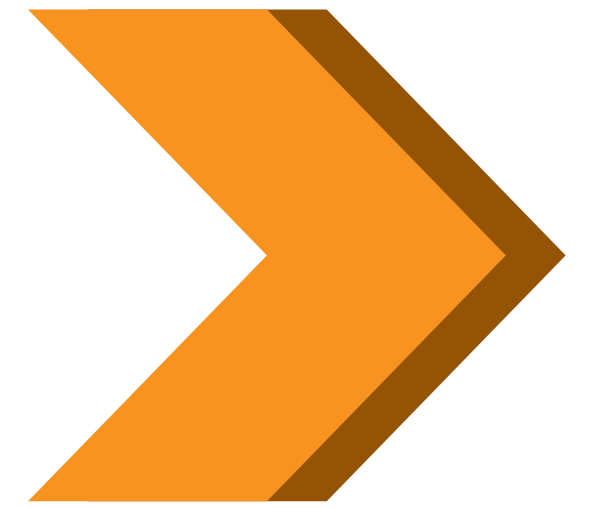
Create an Audio  
Mixer



Add background  
music, ambient  
sound, and the alert  
sound



Set up  
Location-Based 3D  
Sound



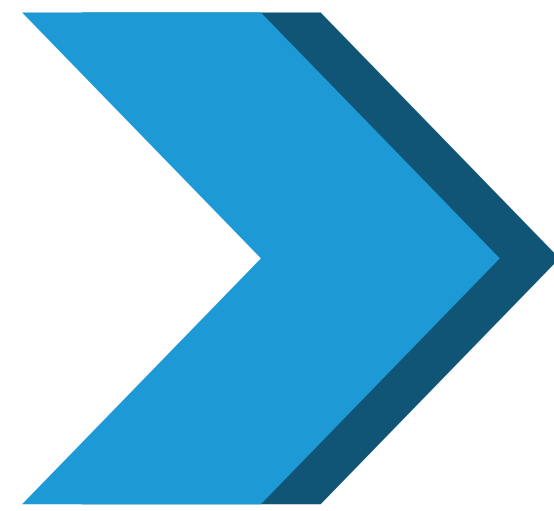
Add footsteps



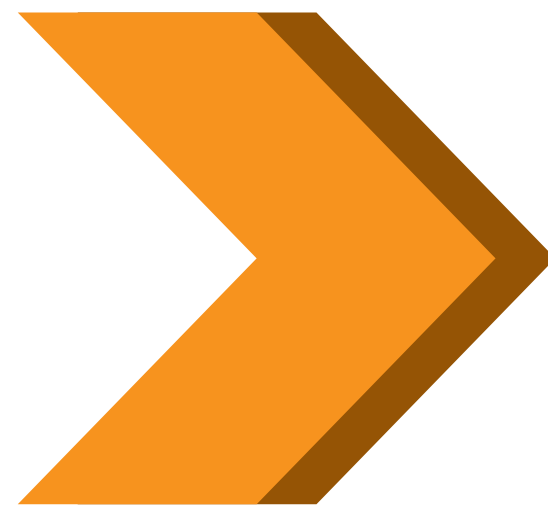
# Activity 3

## 3D Art & Audio Pipeline

Add a second level to *Stealth*



Create a second level



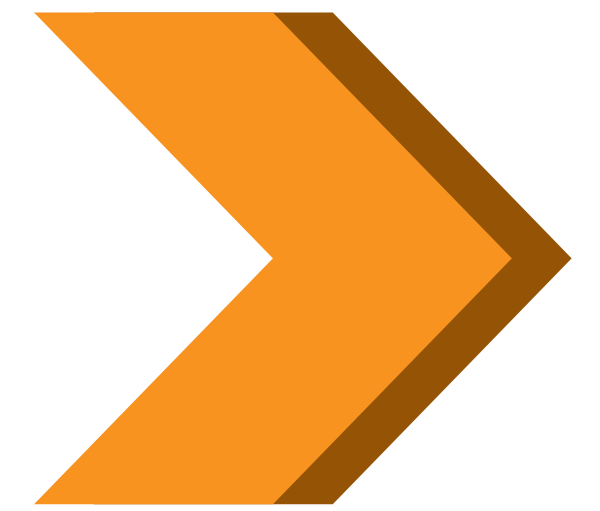
Manage  
GameObjects



Move elements of  
the game to a scene  
named  
\_PersistentScene



Make a copy of the  
completed Unity  
folder on each team  
member's machine

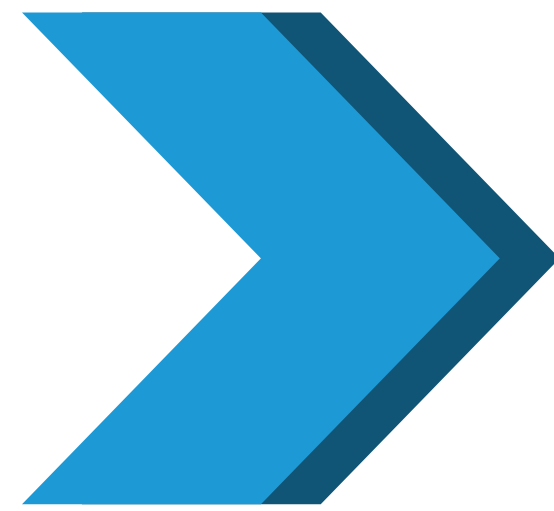


Prepare for peer  
review

# Activity 4

## 3D Art & Audio Pipeline

Conduct peer review



Trade machines  
with a partner



Follow the rubric to  
review and score  
your classmate's  
project



Share and compare  
feedback

# Grading Rubric

## ➤ GameManager Script

The programmer has created a GameManager script that manages the additive loading and subtractive unloading of levels from \_PersistentScene: **1 point**

## ➤ Level Restart

When the player's character is "caught" by the enemy robot, the level restarts: **1 point**

## ➤ Next Level Loading

When the player's character reaches the level goal area, the LevelAdvancePanel appears and the next level loads (or the first level, if the player has just completed the last level in the game): **1 point**

## ➤ Starting Position

When a new level loads, the player's character is placed in the correct starting position and orientation for that level: **1 point**

## ➤ MiniMap Reload

When a new level loads, the MiniMap reloads and represents the new level and its contents accurately: **1 point**





Final  
Review



Any remaining topics

Common challenges

Creative approaches

“Aha” moments

What to expect from the exam



# Congratulations!



# Unity Workshops

## Certified Programmer Exam Prep: 2017.4