Unity EdLab

ADVANCED PLACEMENT COMPUTER SCIENCE PRINCIPLES

Sample Lesson

Unity EdLab includes a full-year AP Computer Science Principles curriculum with six units and 36 lessons that build skills in computational solution design, algorithms, abstraction, code analysis, and more.

UNITY ED LAB AP COMPUTER SCIENCE PRINCIPLES - CURRICULUM TABLE OF CONTENTS

Unit 1: Computers & Innovation

- 1.1 What is a Computer?
- 1.2 Get Rolling with I/O
- 1.3 Innovation, Function or Impact?

Unit 2: Programming

- 2.1 Planning to Create
- 2.2 Create Objects!
- 2.3 Inspecting the Inspector
- 2.4 Various Variables
- 2.5 Function Junction
- 2.6 Cameras and Conditionals
- 2.7 Design Workshop Check-in
- 2.8 Turning on the Lights
- 2.9 Loopy Loops
- 2.10 Materials Matter

Unit 3: Modeling Algorithms

- 3.1 Exploring Algorithms

Unit 4: Data & Media

Unit 5: Internet & Security

Unit 6: Impacts of Computing

Appendix: Exam Preparation & Task Submission

→

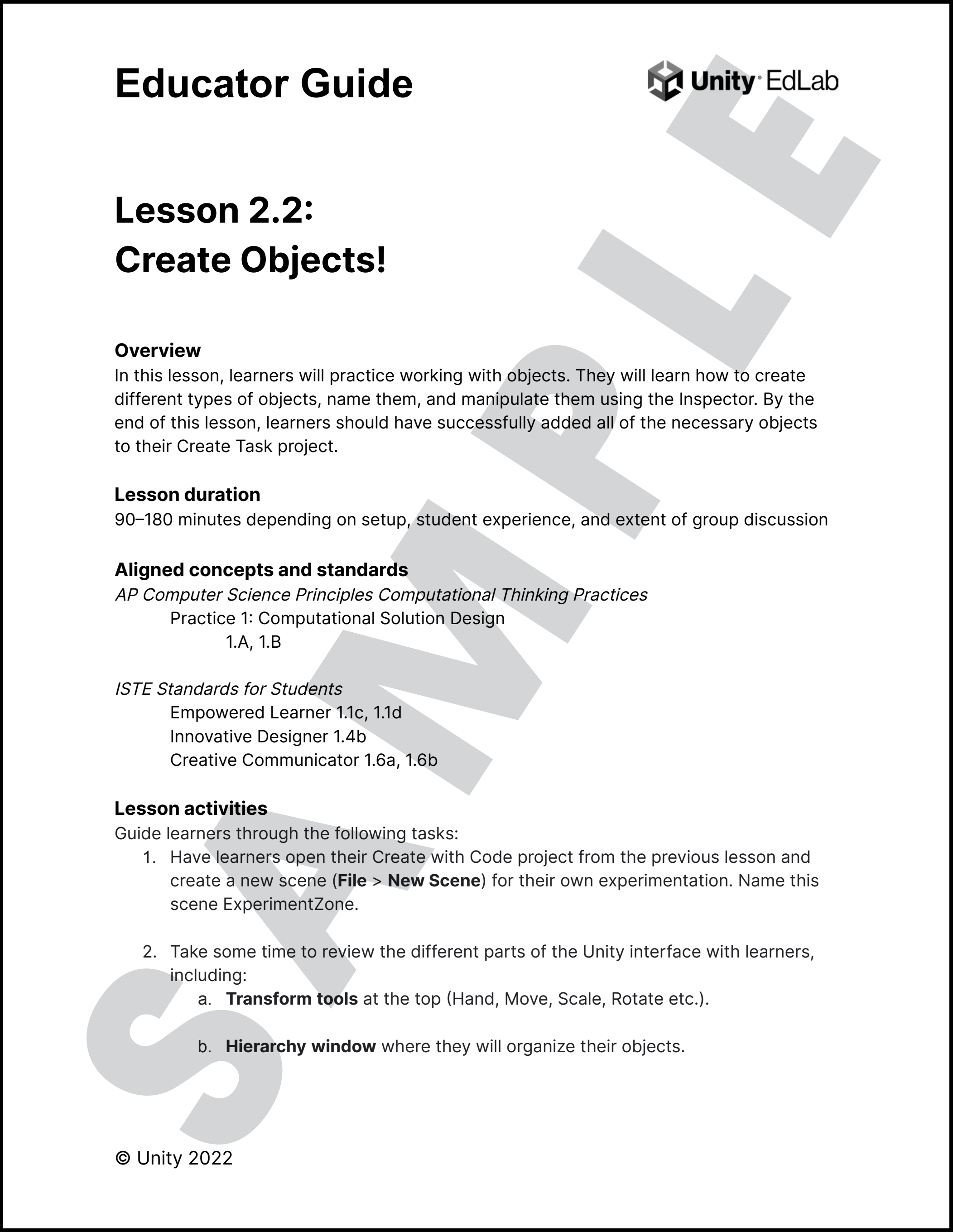
Preparation resources support students in building and submitting a Create Performance Task and taking the AP Computer Science Principles Exam.

Primers

Bite-sized concept primer tutorials introduce and assess core concepts of the Unity platform and creating with code.

UNIT 2 PROGRAMMING 2.2 CREATE OBJECTS

The second unit in this series covers the core concepts of computer programming. Learners will create multiple games in Unity and work towards completing their Create Task for the AP Computer Science Principles course.



Educator Guide

Unity EdLab

Lesson 2.2: Create Objects!

Overview
In this lesson, learners will practice working with objects. They will learn how to create different types of objects, name them, and manipulate them using the Inspector. By the end of this lesson, learners should have successfully added all of the necessary objects to their Create Task project.

Lesson duration
90–180 minutes depending on setup, student experience, and extent of group discussion

Aligned concepts and standards
AP Computer Science Principles Computational Thinking Practices
Practice 1: Computational Solution Design
1.A, 1.B

ISTE Standards for Students
Empowered Learner 1.1c, 1.1d
Innovative Designer 1.4b
Creative Communicator 1.6a, 1.6b

Lesson activities
Guide learners through the following tasks:

1. Have learners open their Create with Code project from the previous lesson and create a new scene (**File** > **New Scene**) for their own experimentation. Name this scene ExperimentZone.

2. Take some time to review the different parts of the Unity interface with learners, including:
   a. **Transform tools** at the top (Hand, Move, Scale, Rotate etc.).

   b. **Hierarchy window** where they will organize their objects.

Unity EdLab targets key learning objectives and outcomes highlighted in STEAM, college, and career standards including AP Computer Science Principles and ISTE Standards.

c. **Scene and Game view** where they will build and test their games.

d. **Inspector window** where they can adjust their GameObjects' components.

e. **Project window** where they will store and organize their files.

f. **Console window** where they will find errors, warnings, and logs.
Note: Reference guides to the Unity interface can be found here: [Unity Interface Guide] and [Explore the Unity Editor].

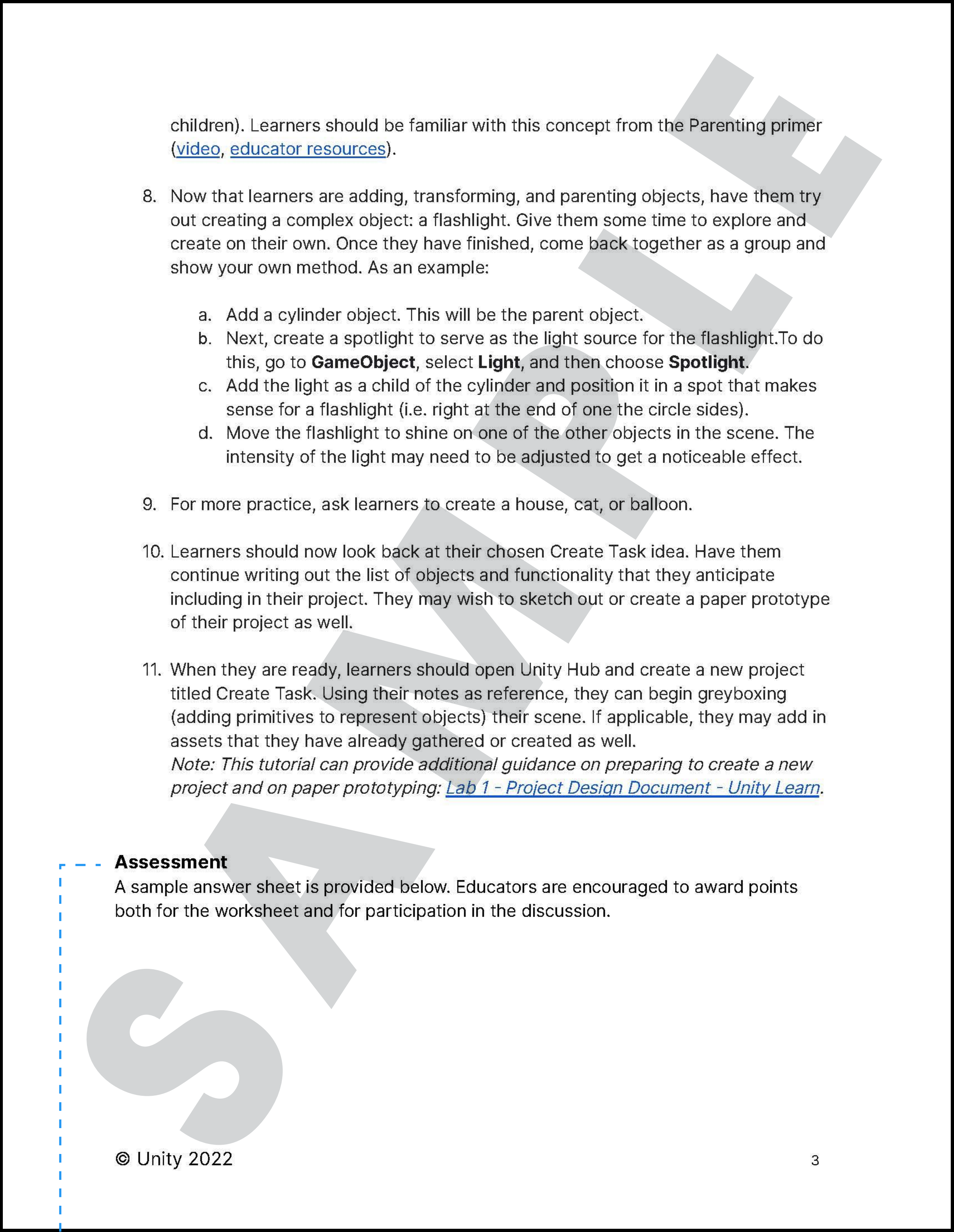
3. Ask learners to identify any GameObjects in the scene. They should notice that there are two in the Hierarchy already: Main Camera and Directional Light. GameObjects can be many different things, including 3D shapes, lights, cameras, and more. They are the fundamental objects in a scene. Each one contains components which make them behave or appear in certain ways. All of the GameObjects in a scene can be found in the Hierarchy.

4. Have learners create their own simple primitive, a 3D cube. You can use the following tutorial as guidance: Work with GameObjects in a 3D Scene - Unity Learn. Have learners create cubes in multiple ways:
   a. GameObjects can be added by going to the top toolbar, selecting **GameObject** and then selecting the desired object in the dropdown.
   b. GameObjects can be added by right clicking in the Hierarchy window and then selecting the desired object in the pop up.
   c. GameObjects can be added by clicking on the plus button in the Hierarchy window and selecting the desired object in the dropdown.
   d. For all options, learners will be going to **3D Object** and selecting **Cube**.

5. Now that learners have a variety of cubes in their scene, challenge them to transform them. Ask them to translate, rotate, and scale them using both the transform tools in the scene and by using the transform component.

6. Guide learners through creating another primitive, this time a 3D sphere. Learners should find the sphere in the Hierarchy window and drag it over to the cube. Doing this will highlight the cube and letting go of the mouse button will drop the sphere "into" the cube. The cube should now have a dropdown arrow that can hide and show the sphere underneath.

7. Learners should experiment with moving the cube, moving the sphere, and swapping the order in which they appear in the Hierarchy. This experiment will introduce the concept of **parents and children**, an essential concept in Unity that involves one GameObject (the parent) directly influencing the objects below it (the



children). Learners should be familiar with this concept from the Parenting primer (video, educator resources).

8. Now that learners are adding, transforming, and parenting objects, have them try out creating a complex object: a flashlight. Give them some time to explore and create on their own. Once they have finished, come back together as a group and show your own method. As an example:

    a. Add a cylinder object. This will be the parent object.
    b. Next, create a spotlight to serve as the light source for the flashlight.To do this, go to **GameObject**, select **Light**, and then choose **Spotlight**.
    c. Add the light as a child of the cylinder and position it in a spot that makes sense for a flashlight (i.e. right at the end of one the circle sides).
    d. Move the flashlight to shine on one of the other objects in the scene. The intensity of the light may need to be adjusted to get a noticeable effect.

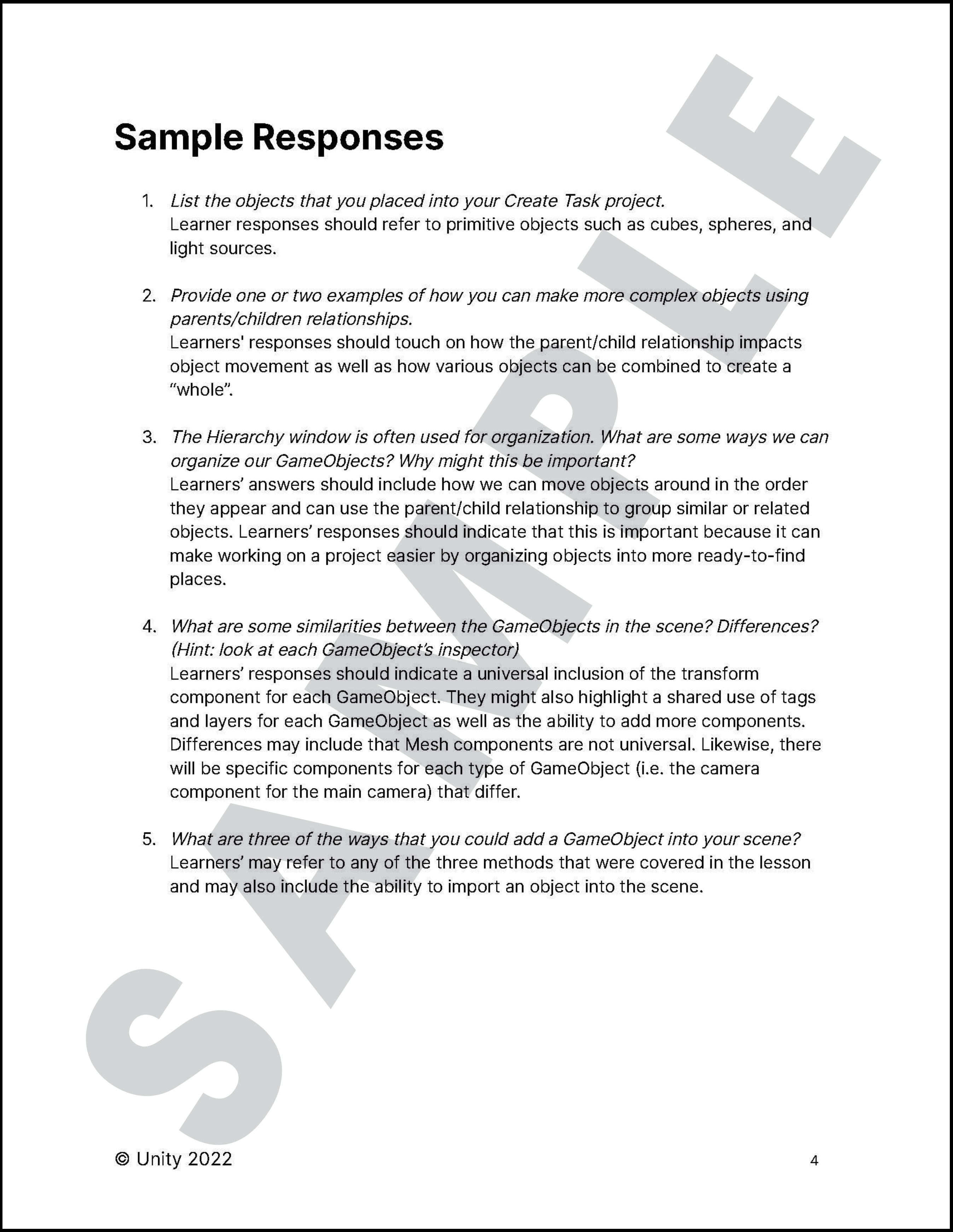
9. For more practice, ask learners to create a house, cat, or balloon.

10. Learners should now look back at their chosen Create Task idea. Have them continue writing out the list of objects and functionality that they anticipate including in their project. They may wish to sketch out or create a paper prototype of their project as well.

11. When they are ready, learners should open Unity Hub and create a new project titled Create Task. Using their notes as reference, they can begin greyboxing (adding primitives to represent objects) their scene. If applicable, they may add in assets that they have already gathered or created as well.
Note: This tutorial can provide additional guidance on preparing to create a new project and on paper prototyping: Lab 1 - Project Design Document - Unity Learn.

Assessment
A sample answer sheet is provided below. Educators are encouraged to award points both for the worksheet and for participation in the discussion.

Student assessments for each lesson gauge progress and comprehension of computer science concepts.



Sample Responses

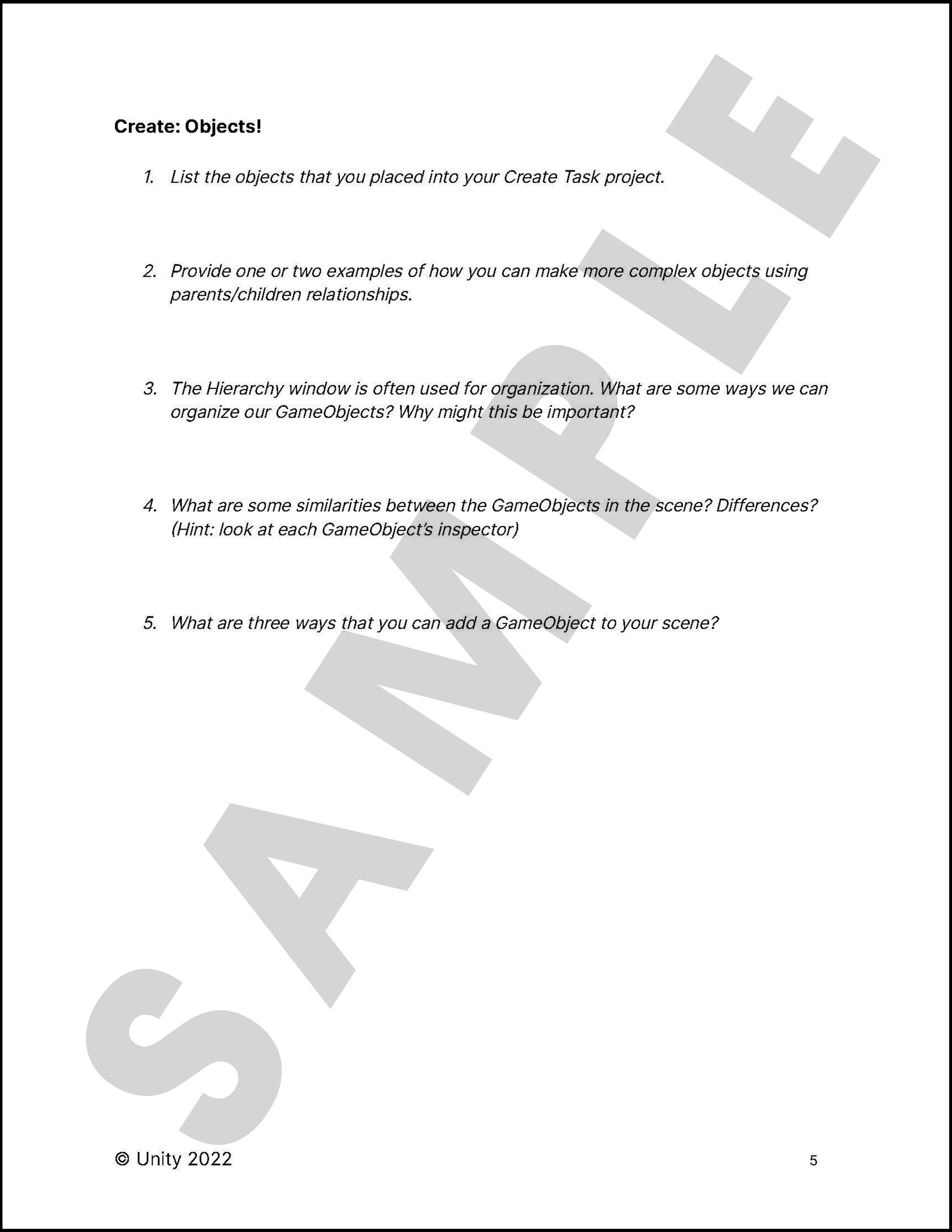
1. *List the objects that you placed into your Create Task project.*
   Learner responses should refer to primitive objects such as cubes, spheres, and light sources.

2. *Provide one or two examples of how you can make more complex objects using parents/children relationships.*
   Learners' responses should touch on how the parent/child relationship impacts object movement as well as how various objects can be combined to create a "whole".

3. *The Hierarchy window is often used for organization. What are some ways we can organize our GameObjects? Why might this be important?*
   Learners' answers should include how we can move objects around in the order they appear and can use the parent/child relationship to group similar or related objects. Learners' responses should indicate that this is important because it can make working on a project easier by organizing objects into more ready-to-find places.

4. *What are some similarities between the GameObjects in the scene? Differences? (Hint: look at each GameObject's inspector)*
   Learners' responses should indicate a universal inclusion of the transform component for each GameObject. They might also highlight a shared use of tags and layers for each GameObject as well as the ability to add more components. Differences may include that Mesh components are not universal. Likewise, there will be specific components for each type of GameObject (i.e. the camera component for the main camera) that differ.

5. *What are three of the ways that you could add a GameObject into your scene?*
   Learners' may refer to any of the three methods that were covered in the lesson and may also include the ability to import an object into the scene.



Create: Objects!

1. *List the objects that you placed into your Create Task project.*

2. *Provide one or two examples of how you can make more complex objects using parents/children relationships.*

3. *The Hierarchy window is often used for organization. What are some ways we can organize our GameObjects? Why might this be important?*

4. *What are some similarities between the GameObjects in the scene? Differences? (Hint: look at each GameObject's inspector)*

5. *What are three ways that you can add a GameObject to your scene?*

Unity®