Name(s)_____ Period _____ Date _____

Practice PT - Design a Digital Scene

Background

In this Practice Performance Task you will use many of the concepts and skills that you have learned through this unit. Here's a quick summary.

Abstractions simplify the representation of something more complex. They allow you to hide details to help you manage complexity, focus on relevant concepts, and reason about problems at a higher level.

Functions (also called "procedures") are an example of an abstraction in programming. They are named blocks of code. Grouping and naming many simple commands allows programmers to reason about their program in chunks, rather than having to think about individual commands.

Top-Down Design is a programming technique that breaks down a large programming task into chunks. Each chunk becomes a function in your program. If a chunk is still too large to easily program then it is divided into even smaller chunks which are given additional functions of their own. Programs written in this way will have layers of functions with high level functions calling lower level ones.

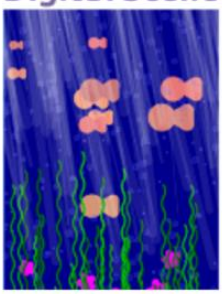
Collaboration in programming is much easier when using Top Down Design. Once a task is divided into many chunks, they can then be assigned to individual programmers who are responsible for writing the functions (or layers of functions) that solve that piece. Their code can later be combined to solve the original problem.

Target Image

Broken Into Functions

```
hide();
penUp();
drawBackground();
drawAllSeaGrass();
drawAllStarfish();
drawAllFish();
drawAllSeaGrass();
drawAllBubbles();
drawAllSunBeams();
```

Digital Scene



Project Description

Description: Work with a group to design and program a digital scene. You will use Top Down Design to divide your scene into logical chunks. Then each member of your group will write the code to complete a portion of the scene. Finally your group will combine your code into a single program that draws the scene.

AP Create Performance Task: This project is designed to closely match the Create Performance Task. The submission guidelines provide more details on what elements are taken from the Create PT.

Programming Concepts: Your program code will need to use many of the programming concepts you learned in this unit. More details can be found in the rubric but you should expect to use layers of functions, loops, parameters, and commenting. The "Under the Sea" project is a good example.

You will submit individually

- **PDF of program code** with a rectangle around an abstraction that you developed.
- **PDF of written responses** describing the purpose of you program, your development process, and the abstraction that you developed

You will submit as a group

- **Group Planning Guide**
- **Link to your Digital Scene project.** Only one group member needs to submit this project

Practice PT Submission Guide - Design a Digital Scene

Reading these Guidelines

The language in this document is taken in large part from the Create Performance Task Description (pg 113) and 2018 Create Scoring Guidelines. It is highly recommended that you read those documents.

Written Responses Submission Guidelines

Selected portions for this Practice Performance Task

AP Performance Task - Create	Tips and Comments
2. Written Responses Submit one PDF file in which you respond directly to each prompt. Clearly label your responses 2a, 2b, 2d in order. Your response to all prompts combined must not exceed 550 words, exclusive of the Program Code. **Program Purpose and Development** **2a**. Provide a written response that: ● identifies the programming language ● identifies the purpose of your program ● *and explains what the user should expect to see when they run the program.* *(Must not exceed 150 words)* **2b**. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)* **2c.** *(omitted for this project)* **2d**. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a rectangle). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*	*These tips and comments are not part of the official Create PT. They are intended to help you better understand the prompts. Many tips are taken directly from the Scoring Guidelines.* ***2a*** ● *JavaScript is the programming language used in App Lab.* ● *The purpose can simply be to make a randomly generated drawing.* ● *"Explains what user should expect.."* is added to this version of the activity since you will not submit a video of your program. ***2b*** ● *Keep a few notes as you program. It will make it easier to identify opportunities / difficulties later.* ● *Make sure you describe the entire process, not just the two distinct points* ● *"Incremental and iterative" means that you continuously improved your program based on feedback, testing, or reflection. For credit you need to refer to this feedback, testing, or reflection in your response.* ● *For the two distinct opportunities / difficulties make sure you include how they were resolved* ***2d*** ● *Make sure you copy and paste the code of your abstraction into your written responses, even though you also are drawing a rectangle around it in your code. Switching to text mode may help.* ● *The "how" is important. Don't just explain what your abstraction is. Explain how it makes your program easier to read or reason about.* ● *Sentence starters: "An abstraction in my program that I developed is the [blank]. It manages complexity in my program by [blank]"*

PDF of Program Code Submission Guidelines

Selected portions for this Practice PT

AP Performance Task - Create	Tips and Comments
3. Program Code Capture and paste your entire program code into a document. ● Mark with a **rectangle** the segment of program code that represents an abstraction you developed. ● Include comments or acknowledgments for program code that has been written by someone else.	*These tips and comments are not part of the official Create PT. They are intended to help you better understand the prompts.* ● *"Include comments….for program code" means inserting comments (by writing // before a line of code) into the code itself before printing.* ● *On the Create Performance Task you are allowed to collaborate with a partner but must carefully indicate whose work is whose with comments.* ● *Make sure the abstraction you identify is program code you wrote.* ● *In almost all cases the abstraction you choose will be a function you wrote. You'll want to draw a box around the place where you wrote that function code. Suggest: A good one to choose would be a function that you wrote that calls other functions you wrote, because it demonstrates abstraction and managing complexity.* ● *To make a PDF and draw a rectangle, we suggest using the "Code Print" program here: https://bakerfranke.github.io/codePrint/*

Group Planning Guide Submission Guidelines

Your collaborative programming group will submit a **single copy** of your group planning guide. Make sure that all group members' names are listed on the document.

App Lab Project Submission Guidelines

When you have completed your digital scene, submit it by clicking the ***Submit Project*** button on the proper App Lab project in Code Studio (the project associated with this lesson). Submitting indicates that your project is ready to be reviewed.

Group Planning Guide

Choosing a topic: Before you begin make sure you've reviewed the submission guidelines and rubric. Then pick a scene that has the following features:

- Have several components that allow it to be broken into logical chunks (functions)
- Have repeated elements that will allow you to use loops and random values
- Use a function with a parameter (same figure but different size / color / dimensions, etc.)

Scene Description What is the topic? What are the different pieces of the scene?	**Scene Sketch** Make a quick sketch of the scene. Write notes to clarify points. Use the back of this sheet if you need.

Identify Top-Level Functions and Assign to Group Members

Use Top Down Design to identify the major components of your scene. Give each component a top-level function. In the Under the Sea project these were `drawAllFish()`, `drawAllSeagrass()`, `drawAllBubbles()`, etc. Then assign each function to a member of the team to program individually. They may need to further divide their component into smaller functions later.

Scene Component	Function Name	Group Member

(use back side of page if you need more)