

## CS 4730: Computer Game Design

### Display Tree and Animation

#### **Overview**

In this assignment, you will implement a basic display tree in your engine and a simple animated sprite.

#### **PART 1: Create the *DisplayObjectContainer* class**

Begin by creating a new class called *DisplayObjectContainer*. This class should extend *DisplayObject* (Sprite, Game, etc. should now extend *DisplayObjectContainer* instead).

Your class should contain a list of *DisplayObject* objects that are the children of this container. Thus, containers are essentially nodes in a tree and this list is the list of child objects (which might be *DisplayObject* type which represents a leaf node OR *DisplayObjectContainer* types representing another internal node).

The header file is included in the course repository (see the homework page for details). You may alter the header file as you see fit, but the methods provided will all be useful as we move forward so we highly recommend you implement all of them.

#### **Test Demo: *Solar System Simulator***

Create a simple solar system that shows off your engine's new features:

- Make a sun sprite that is the root node of the whole system.
- Planets should be children of the sun.
- Moons of planets should be children of the planets
- The whole system should move continuously.
- The planets and moons should rotate around their parent at different rates. Think carefully about how to do this.
- Planets and moons should have elliptical orbits. Again, try to think of an easy way to accomplish this in code.

Your demo should also support the following commands:

- Q&W:           Zooms in and out towards center of sun OR center of current screen position.
- Arrows:       Pans up/down/left/right around the system.
- A&S:           Rotates the system around the sun's center point.

#### **PART 2: *AnimatedSprite* and *Sprite***

Next, you will implement *Sprite* class (extends *DisplayObjectContainer*) and a new class called *AnimatedSprite* that extends *Sprite*. The *Sprite* class will essentially be empty for now. Your next task is to implement the *AnimatedSprite*, which should have the following additional functionality:

- Class should contain a list of frames (images, not *DisplayObjects*) that contain all of the images in the animation(s). You'll need some mechanism for instantiating these (sprite sheet support, or perhaps just indexing the individual images with some naming convention like *Mario\_run\_0.png*, *Mario\_run\_1.png*, etc.)

- Class should contain a current frame integer as well as a startIndex and endIndex. These last two integers will be used to support multiple animations (see next bullet below).
- Class should support multiple animations (e.g., jump, walk, run)
- Animations can be stopped on a single frame, or playing.
- The speed of the animation can be adjusted through a setter.
- I can set which animation to play through a method, like animate("run");

A header file is provided for you in the course repository.

***Test Demo: Updating our Solar System***

Update the Solar System demo so that:

- The sun is animated with multiple sprites. For now, it is ok if each frame is a separate image file.
- Holding the 'P' key changes the animation of the sun to a different animation (maybe it shines brighter or looks different some other way).
- Releasing 'P' returns the sun to its original animation.

***Turn In***

Zip up your entire project and submit it on Collab.