Assignment #2: Camera Control, Node Control, Viewports, Hierarchical Objects

"Dolphin Adventure 2" ------- Due: Friday March 10th (3 weeks)

The objective of this assignment is to learn a variety of game and game engine details, by upgrading your *"Dolphin Aventure 1"* game from A1, to become *"Dolphin Adventure 2"*. Your game will now treat the dolphin as an avatar, with an associated *orbit camera controller* added to TAGE. It will also provide a second (smaller) *viewport* with an overhead view of the world, the dolphin, and the prizes. At least one new *node controller* will be added to TAGE and utilized in the game. And, the game will make greater use of TAGE's scenegraph support by including a hierarchical object or system.

"Dolphin Adventure 2" (A2) is similar to A1, except the "game world" now takes place on a flat ground plane (e.g., the Y=0 plane) on which the objects reside nearby, and on which the player manipulates the avatar. The game now places the two HUD elements so that there is one for each viewport.

The prizes now must all reside on or near the ground plane. Initially they are unmoving. When a prize is "collected" by a player, it starts moving in place (such as rotating, or bobbing up and down).

You may add other features, or even change the game, as long as your program satisfies the implementation details described below. Make sure that any changes you make to the game are described in the "Player's Guide" that you also submit.

Game Program Requirements

- Implement a 3rd-person Orbit Camera Controller as a new class in TAGE called **CameraOrbit3D**. It must be possible to: (1) orbit the camera without altering the avatar's heading, (2) adjust the camera elevation angle, (3) zoom the camera in and out on the avatar, and (4) move and turn the dolphin while maintaining the camera's relative position and orientation relative to the dolphin.
- Support a second, smaller viewport with its own camera that shows the same game world but from an overhead perspective. This second viewport will not use an orbit camera controller, but should have its own controls to *zoom* and *pan* this camera.
- Provide HUDs for each viewport. Both HUDs must maintain their relative position within their respective viewports if the player resizes the window. The HUD for the main viewport includes the player's score. The HUD for the second viewport includes the avatar's world position.
- Include a *ground plane* – a flat surface on which the game takes place. You may if you wish allow players to leave the ground for a time ("jumping"), but they should not have the ability to move continuously in arbitrary 3D directions like they did in A1. You will need to build the plane using the TAGE Plane class. You may also want to apply **`scale()`** and **`rotate()`** to adjust its size and position. Retain the XYZ axes from A1, but add a control for disabling it (make them invisible).
- Two different types of *Node controllers*, at least one of which is written from scratch by you. One of the controllers is applied to a prize when it is first visited by the player, while the second controller can be used for whatever reasonable purpose you like. Since TAGE currently only has one built-in node controller (a rotation controller), you will need to add a new node controller to TAGE. The new controller can do whatever you like - for example, it could cause an object to bounce or change color. It must be implemented by you, and be added to the **`nodeController`** sub-package in TAGE, and consist of a class that extends the TAGE **`NodeController`** class.

- Your game must include at least one example of a hierarchical relationship between two or more **GameObject**s. That is, there must be at least one node that has a "child" node, and a clear *reason* for the relationship. For example, you could add miniature versions of captured prizes as child nodes of the dolphin to make it look like the dolphin is carrying them. Use your imagination!
- In this version of the game, there is nothing stopping the player from moving the camera (or the dolphin) completely off the ground plane – although your code should prevent them from going below the ground plane. There is also nothing stopping the player from positioning the dolphin avatar or the camera in such a way that some other object is blocking the camera's view of the avatar. That is ok for now – we will fix this (at least partially) in Project 3.
- It is no longer a requirement to be able to hop on and off the dolphin. You should remove the code for hopping off the dolphin to visit prizes – simply have the dolphin visit the prizes.

Deliverables and Coding Style Requirements

- This is an INDIVIDUAL assignment.
- The submission format and file structure is the same as in A1 (except that it is called **A2**). Refer to the A1 assignment specification for details on how to format and submit your work to Canvas. Again, you will submit a ZIP folder and a separate TEXT file. The ZIP folder will contain folders for "tage", "assets", "**a2**", batch files for compiling/running, and a "readme.pdf" as outlined below. The TEXT file indicates which RVR-5029 machine you used to test your A2 program.
- As before, the "**tage**" folder contains the complete TAGE game engine, including your changes. Your new camera orbit controller class should be included here, and your new node controller should be included in the **nodeControllers** subpackage. As before, you are encouraged to make other changes and additions to your copy of TAGE – just document them in your **readme** PDF.
- All of your additions to TAGE should include corresponding additions to the TAGE *javadocs*.
- Your "**readme.pdf**" document (approx. 2 pages) must include the following 12 numbered items:
  1. your full name, CSc-165 section number, and "A2 – Dolphin Adventure 2"
  2. a screenshot (JPG file) showing a typical scene from your game
  3. how your game is played, and the scoring
  4. a list of keyboard and gamepad controls for moving the dolphin
  5. a list of keyboard and gamepad controls for the orbit controller
  6. a list of keyboard and gamepad controls for zooming and panning the overhead viewport
  7. a description of your node controllers and what they do, and how they are used in your game
  8. a description of your use of scenegraph parent/child relationships and their effect in the game
  9. a clear list of all changes you made to the TAGE engine (don't forget to also update the javadocs!)
  10. a list of any requirements that you *weren't* able to get working
  11. a list of anything special that you added beyond what was specified in the requirements
  12. a list of every asset used in your game, and whether you made it. For each asset that you didn't create yourself, indicate where you got it, and provide clear evidence that it is legal for you to use in this game (such as written permission, or a posted license). If an asset was copied from the distributed TAGE examples, just state that.

Note that the submitted files must be organized in the proper hierarchy *in the ZIP file*, as indicated in the coding style requirements listed above, *except* the TEXT file which must be *outside* the ZIP folder.