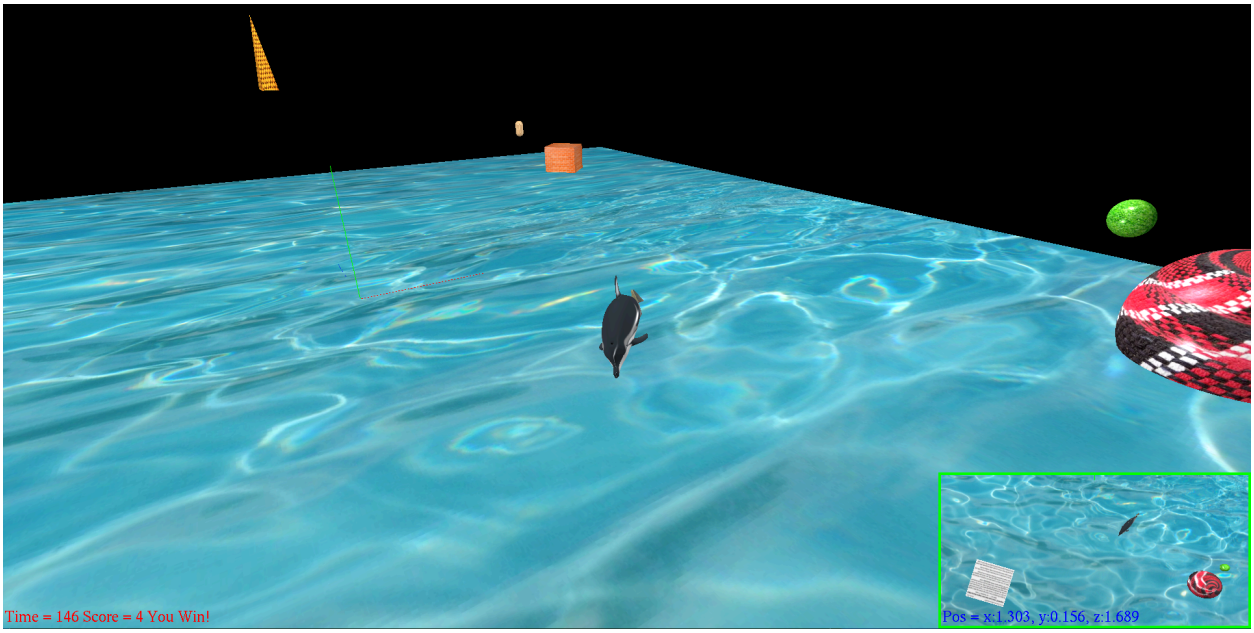
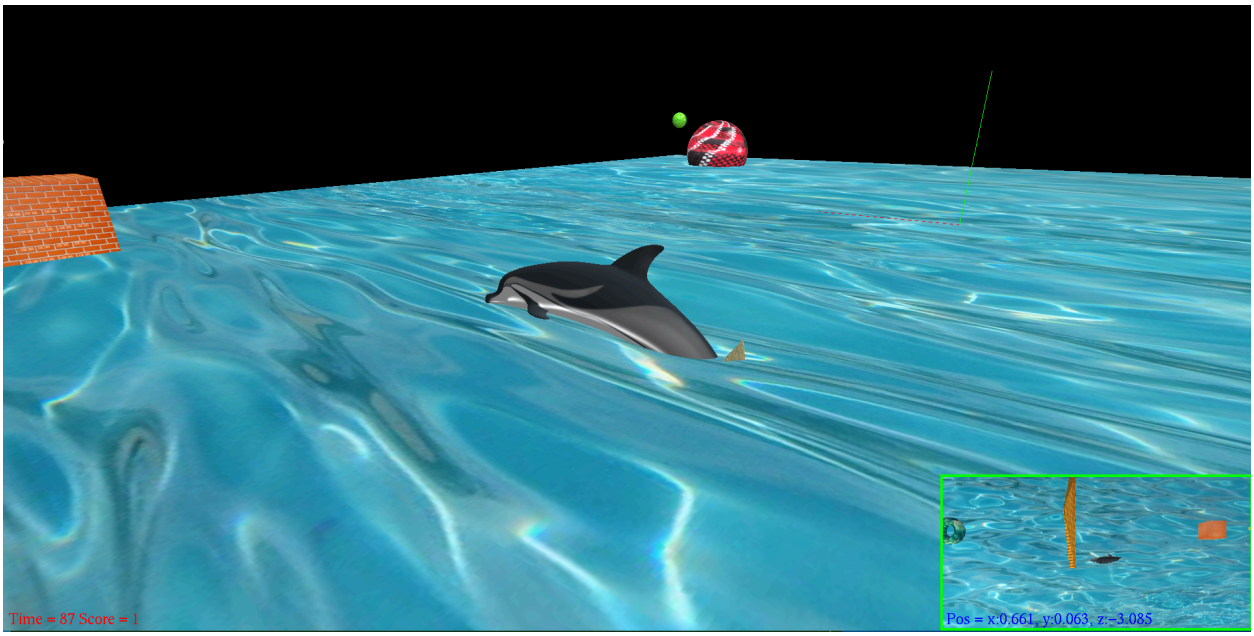
1. Gabriele Nicula, CSC165-01, A2 – Dolphin Tour

2. Screenshot(s):





3. List of keyboard and gamepad controls for moving the dolphin:

Move Forward:	W, Gamepad Left Joystick Y-axis up
Move Backward:	S, Gamepad Left Joystick Y-axis down

Yaw Left:	A, Gamepad Left Joystick X-axis left
Yaw Right:	D, Gamepad Left Joystick X-axis right
Roll Left:	Q
Roll Right:	E
Pitch Up:	Up Arrow, Gamepad Button 0
Pitch Down:	Down Arrow, Gamepad Button 1
Toggle World Axes On/Off:	1
Dolphin Wireframe On:	2
Dolphin Wireframe Off:	3
Exit Game:	ESC

4. List of keyboard and gamepad controls for the orbit controller, and for zooming and panning the overhead viewport:

Camera Rotate Left:	Gamepad Right Joystick RX-axis left
Camera Rotate Right:	Gamepad Right Joystick RX-axis right
Camera Elevation Up:	Gamepad Right Joystick RY-axis up
Camera Elevation Down:	Gamepad Right Joystick RY-axis down
Camera Zoom In:	Gamepad Button 2
Camera Zoom Out:	Gamepad Button 3
Pan Mini-Map Up:	0 (zero)
Pan Mini-Map Right:	P
Pan Mini-Map Down:	L
Pan Mini-Map Right:	O
Mini-Map Zoom In:	]
Mini-Map Zoom Out:	[

5. Description of node controllers, what they do, and how they are used in the game:

   RotationController - a node controller that rotates the GameObject around a given axis with a constant amount at each step. This class is available in tage\NodeControllers and I used as is. All 4 reachable game objects (cube, torus, plane, sphere) use this node controller and the controller is enabled only after the object is “visited” by the dolphin.
   Additionally, each magnet that is attached to the dolphin is controlled by a RotationController which rotates the magnets once they are attached to the dolphin.

   StretchController - a node controller I added to tage\NodeControllers. This

controller stretches the controlled object alternatively on two axis, X and Z. I implemented this class following the code given in the course material. The stretching follows first axis X and, after a full cycle, the axis is flipped to Z and the object is stretched back and forth on Z axis. The process then repeats.
The manual golden Pyramid object is controlled by this StretchController.

6. Description of scenegraph parent/child relationships and their effect in the game:

   Conceptually there are 3 parent<->child relationship between the object in the game:
   a) Dolphin<->magnets: the magnets are attached to the dolphin and move/rotate with the parent object. There are 4 magnets that are attached as children objects.
   b) Brick Cube<->Capsule: the capsule is attached to the brick cube and rotates with it after it is visited
   c) Sphere<->Satellite: the small sphere satellite is attached to the red texture reachable sphere object and rotates with it after the reachable sphere object is visited.

7. List of all changes you made to the TAGE engine:

   a) tage\NodeControllers:
   Added tage\NodeControllers\StretchController.java, a NodeController that stretches controlled object alternatively on axis X, Z
   b) tage\CameraOrbit3D.java
   Java class that implements a 3-D moveable camera around an avatar. CameraOrbit3D implements methods for controlling the azimuth, elevation and zoom.
   b) tage\GameObject.java: Added methods for computing movements of the game object, specifically:
   public void moveForwardBack(float speed, Vector3f cameraLoc)
   public void pitch(float speed)
   public void yaw(float speed)
   public void roll(float speed)
   c) tage\input\action: Added classes that implement movements of the dolphin and controlling the secondary viewport, specifically:
   tage\input\action\SecondaryViewportPanXActionK.java
   tage\input\action\SecondaryViewportPanYActionK.java
   tage\input\action\SecondaryViewportZoomActionK.java
   tage\input\action\ForwardBackActionJ.java
   tage\input\action\ForwardBackActionK.java

tage\input\action\PitchActionJ.java
tage\input\action\PitchActionK.java
tage\input\action\RollActionK.java
tage\input\action\YawActionJ.java
tage\input\action\YawActionK.java

8. List of any requirements that you weren't able to get working: None

9. List of anything special that you added beyond what was specified in the requirements:
    a) Added 3 examples of a hierarchical relationship between two or more GameObjects
    b) Added an ImportedModel shape for a child object using the TAGE provided assets\defaultAssets\capsule.obj
    c) Roll left/right for the dolphin

10. List of every asset used in the game:
    assets\textures\assign1.png - made by me
    assets\textures\brick1.jpg - made by me
    assets\textures\corvette1.jpg - Closeup of Red and Black Striped Woven Fabric · Free Stock Photo (pexels.com)
    assets\textures\gold1.jpg - Photo of Cone Pattern · Free Stock Photo (pexels.com)
    assets\textures\grass1.jpg - Green Grass · Free Stock Photo (pexels.com)
    assets\textures\magnet1.jpg - Metal Sheet · Free Stock Photo (pexels.com)
    assets\textures\water.jpg - Body of Water · Free Stock Photo (pexels.com)
    assets\textures\waterTorus.jpg - https://www.pexels.com/photo/aerial-shot-of-blue-water-3894157/
    assets\textures\fur1.jpg - Close-Up Photo of Brown Furry Texture · Free Stock Photo (pexels.com)