CSc-165 Spring 2023 Week 2 (b)

TAGE
GamePad & Lines

```
. . .
import tage.input.*;
import tage.input.action.*;
import net.java.games.input.*;
import net.java.games.input.Component.Identifier.*;

public class MyGame extends VariableFrameRateGame
{  . . .
   private InputManager im;
   private GameObject tor, avatar, x, y, z;
   private ObjShape torS, dolS, linxS, linyS, linzS;
   . . .
-----------------------------------------------------------

   @Override
   public void loadShapes()
   {  dolS = new ImportedModel("dolphinHighPoly.obj");
      torS = new Torus(0.5f, 0.2f, 48);
      linxS = new Line(new Vector3f(0f,0f,0f), new Vector3f(3f,0f,0f));
      linyS = new Line(new Vector3f(0f,0f,0f), new Vector3f(0f,3f,0f));
      linzS = new Line(new Vector3f(0f,0f,0f), new Vector3f(0f,0f,-3f));
   }
   . . .
   @Override
   public void buildObjects()
   {  . . .
      // build dolphin avatar
      avatar = new GameObject(GameObject.root(), dolS, doltx);
      initialTranslation = (new Matrix4f()).translation(-1f,0f,1f);
      avatar.setLocalTranslation(initialTranslation);
      initialRotation = (new Matrix4f()).rotationY(
                     (float)java.lang.Math.toRadians(135.0f));
      avatar.setLocalRotation(initialRotation);

      // build torus along X axis
      tor = new GameObject(GameObject.root(), torS);
      initialTranslation = (new Matrix4f()).translation(1,0,0);
      tor.setLocalTranslation(initialTranslation);
      initialScale = (new Matrix4f()).scaling(0.25f);
      tor.setLocalScale(initialScale);

      // add X,Y,-Z axes
      x = new GameObject(GameObject.root(), linxS);
      y = new GameObject(GameObject.root(), linyS);
      z = new GameObject(GameObject.root(), linzS);
      (x.getRenderStates()).setColor(new Vector3f(1f,0f,0f));
      (y.getRenderStates()).setColor(new Vector3f(0f,1f,0f));
      (z.getRenderStates()).setColor(new Vector3f(0f,0f,1f));
   }

   @Override
   public void update()
   {  . . .
      // update inputs and camera
      im.update((float)elapsedTime);
      positionCameraBehindAvatar();
   }

   private void positionCameraBehindAvatar()
            *** (same as in DolphinRide) ***

   public GameObject getAvatar() { return avatar; }
```

```
@Override
public void initializeGame()
{  . . .
   positionCameraBehindAvatar();

   // ---------------- INPUTS SECTION -----------------------------
   im = engine.getInputManager();

   FwdAction fwdAction = new FwdAction(this);
   TurnAction turnAction = new TurnAction(this);

   im.associateActionWithAllGamepads(
      net.java.games.input.Component.Identifier.Button._1, fwdAction,
      InputManager.INPUT_ACTION_TYPE.REPEAT_WHILE_DOWN);
   im.associateActionWithAllGamepads(
      net.java.games.input.Component.Identifier.Axis.X, turnAction,
      InputManager.INPUT_ACTION_TYPE.REPEAT_WHILE_DOWN);
   im.associateActionWithAllKeyboards(
      net.java.games.input.Component.Identifier.Key.W, fwdAction,
      InputManager.INPUT_ACTION_TYPE.REPEAT_WHILE_DOWN);
}
```

FwdAction.java

```
. . .
import tage.input.action.AbstractInputAction;
import net.java.games.input.Event;
import org.joml.*;

public class FwdAction extends AbstractInputAction
{  private MyGame game;
   private GameObject av;
   private Vector3f oldPosition, newPosition;
   private Vector4f fwdDirection;

   public FwdAction(MyGame g)
   {  game = g;
   }

   @Override
   public void performAction(float time, Event e)
   {  av = game.getAvatar();
      oldPosition = av.getWorldLocation();
      fwdDirection = new Vector4f(0f,0f,1f,1f);
      fwdDirection.mul(av.getWorldRotation());
      fwdDirection.mul(0.01f);
      newPosition = oldPosition.add(fwdDirection.x(),
                           fwdDirection.y(), fwdDirection.z());
      av.setLocalLocation(newPosition);
}  }
```

TurnAction.java

```
. . .
public class TurnAction extends AbstractInputAction
{  . . .
   @Override
   public void performAction(float time, Event e)
   {  float keyValue = e.getValue();
      if (keyValue > -.2 && keyValue < .2) return;  // deadzone

      av = game.getAvatar();
      oldRotation = new Matrix4f(av.getWorldRotation());
      oldUp = new Vector4f(0f,1f,0f,1f).mul(oldRotation);
      rotAroundAvatarUp = new Matrix4f().rotation(-.005f,
                  new Vector3f(oldUp.x(), oldUp.y(), oldUp.z()));
      newRotation = oldRotation;
      newRotation.mul(rotAroundAvatarUp);
      av.setLocalRotation(newRotation);
}  }
```