Sands of Egypt

ABSTRACT

This project was designed relative to the scope of object-oriented principles with primary focus on inheritance, abstraction and two-dimensional transformations, which included a rotational moment of a subclass object. Variations of dynamic and fixed arrays were integrated for a pyramid staircase and sandstorm, respectively. The goal was to design a functional graphical user interface (GUI), which integrated three level grandparent-parent-child inheritance relationships including interfaces, abstract classes and 2D stack element transformations within those designed subclasses, therefore proper layering of the interface stack elements represented the most vital aspect of this project. Egypt is constructed as an interface similar to a Shape interface, which contains 3 main elements: human traveler with a hand wave gesture; a pyramid with staircase added and the background of Egypt with multiple transformations including an array generated sandstorm, eclipse of the sun and a day to night background effect. The result was an application that meets the six main goals as outlined on the project criteria guideline and executes seamlessly.

INTRODUCTION

The object-oriented approach taken to construct the project, which began with a simple idea of day and night transition based on a reference image presented below:

Figure 1. Reference Image Guide

OVERALL DESIGN PROCESS

Using this as a point of reference and template the night-based transition was modified to a dark evening sky with an eclipse of the sun occurring in the background stack both of which occur through a mouse and keyboard interaction. The transformation is displayed in the figure below:

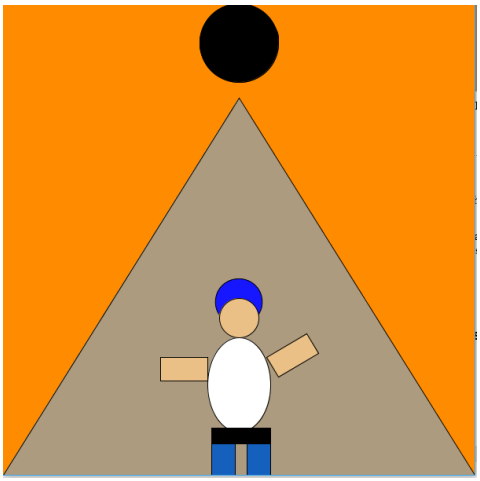

Figure 2. Egypt Eclipse

Secondly, the pyramid object element was added to the stack of the interface. This introduced the first of two multi-level hierarchies in which the Egypt interface represents the grandparent, abstract class being Pyramid and PyBase as the child class of Pyramid. The code for this approach is shown as the following:

```
abstract class Pyramid implements Egypt{
  abstract void display();
   color c;
   int x1;
   int y1;
   int x2;
   int y2;
   int x3;
   int y3;



   Pyramid(color pyC, int x1pos, int y1pos, int x2pos, int y2pos, int x3pos, int y3pos){
     c = pyC;
     x1 = x1pos;
     y1 = y1pos;
     x2 = x2pos;
     y2 = y2pos;
     x3 = x3pos;
     y3 = y3pos;
   }
}
class PyBase extends Pyramid implements Egypt{

  PyBase(color c, int x1, int y1, int x2, int y2, int x3, int y3){
    super(c, x1, y1, x2, y2, x3, y3);
  }
   void display(){
     fill(c);
     triangle(x1, y1, x2, y2, x3, y3);
   }
}
```

Figure 3. Abstract Class & Constructor

Following the pyramid, a staircase was added to the stack based on a dynamic ArrayList class, which implemented a forward for loop to add an incrementing vertical set of stairs along the y-axis upon mouse press allowing the user to create

stairs to ascend the pyramid which fade away as stairs are removed from the stack through a backwards for loop variation.

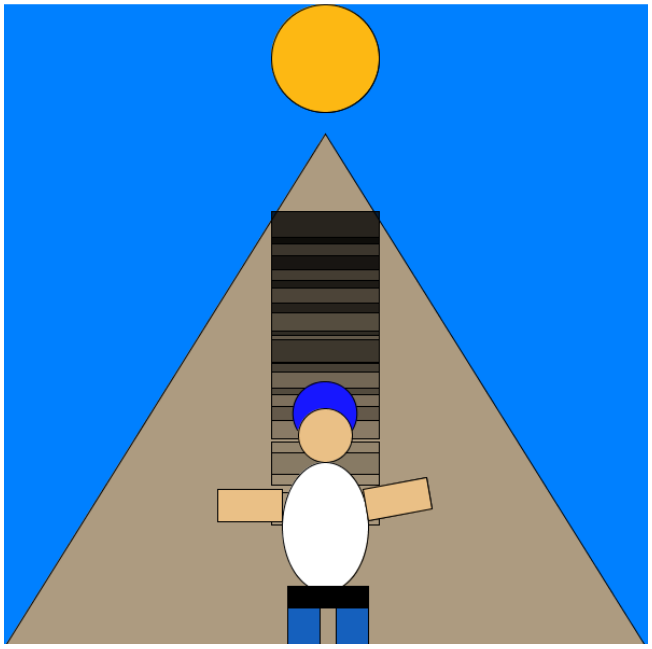

Figure 4. ArrayList Staircase 1

A sandstorm effect was created using basic array to store a fixed amount of elliptical sand elements that populate in the background and form a trailing pattern as the user moves the mouse as shown in the following:

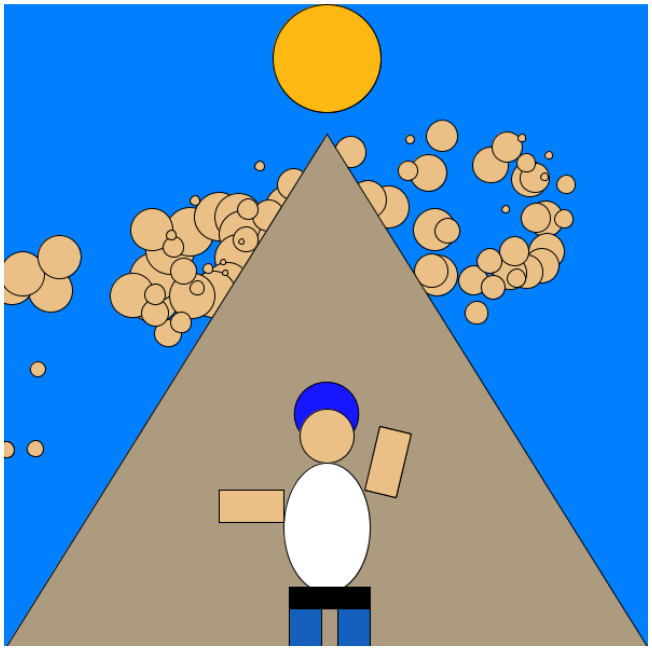

Figure 5. Sand Trail Array

The important point here was the way the stack was structured in this array transformation relative to its location as the third component in front of the background and sun with the effect occurring behind the pyramid and traveler, which were the fourth and fifth stack-based object elements, respectively.

Following the pyramid and staircase a human traveler was added to the stack, which was designed as a multi-level hierarchy with a grandparent, parent and subclasses relationships each associated with a component that make up the entire human figure with an added 2D hand wave transformation. The multi-level hierarchy is shown on the following UML:

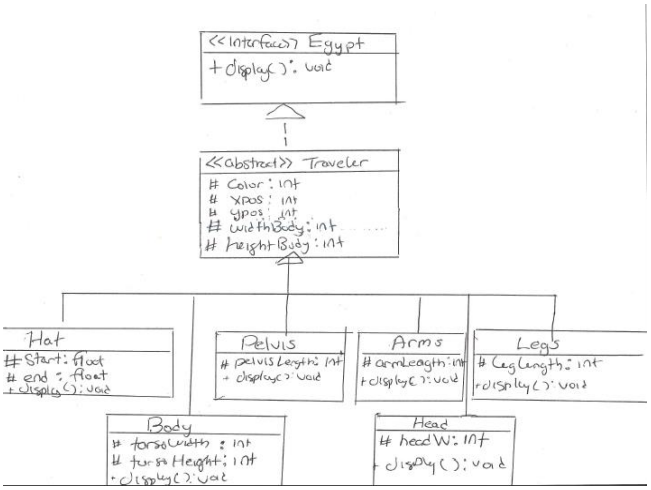

Figure 6. Human Traveler Figure UML

We begin with an interface Egypt which is then implemented by abstract class Traveler which has subclasses consisting of the head, pelvis, arms, legs and hat of the traveler each of which forms an IS-A Traveler relationship, head for instance IS-A Traveler component. The hat object is a key component that distinguishes the traveler from just a human that could or could not be wearing a hat. For the traveler's hat component an arc-based object was created with reference to its radian conversion. The image below represents a reference used in creation of the hat object:

Figure 7. Arc Based Hat Design Reference 1

Lastly, a 2D arm wave transformation was added which rotated the right arm object pivot point relative to the origin counterclockwise, hence a negative value of 45 degrees was used which converted into radians. The human traveler being the outermost and last element added to the stack, which appears to be waving with the other stack elements of the Egypt interface appear in the background. The transformation reference, which served as a template is as follows:

```
void drawLeftArm()
{
  pushMatrix();
  translate(12, 32);
  rotate(radians(135));
  rect(-12, 0, 12, 37); // left arm
  popMatrix();
}

void drawRightArm()
{
  pushMatrix();
  translate(66, 32);
  rotate(radians(-45));
  rect(0, 0, 12, 37); // right arm
  popMatrix();
}
```

Figure 8. 2D Transformation Reference

ALGORITHM DESIGN

The algorithms used to design the project began with an initial step of creating UML representation of the two key multi-level hierarchical objects: human traveler and pyramid, both of which formed the basis of the Egypt interface. Once the UML was designed it made it simple to visualize how each object component of traveler fit together. The draft version of the code for traveler was then written on paper, allowing mistakes to be seen clearly. Following this the code was transferred to processing, where a top-down approach was taken starting from the most abstract aspects to the most concrete components. Once classes were designed constructors were created and invoked by child classes such as the constructor for the Traveler class, which is then invoked through extended subclasses through super(), the hat object for instance used a modified or mutated version of the Traveler constructor, which integrated elements of an arc, such as start and end.

```
abstract class Traveler implements Egypt {
  abstract void display();
  color c;
  int x;
  int y;
  int w;
  int h;

  Traveler(color tempC, int xpos, int ypos, int widthBody, int heightBody){
    c = tempC;
    x = xpos;
    y = ypos;
    w = widthBody;
    h = heightBody;


  }
}
  class Hat extends Traveler {
    float s;
    float e;

    Hat(color c, int x, int y, int w, int h, float start, float end){
      super(c, x, y, w, h);
      s = start;
      e = end;
    }
    void display(){
      pushMatrix();
      fill(c);
      arc(x, y, w, h, s, e);
      popMatrix();
    }
  }
```

Figure 9. Multi-Level Hierarchy Traveler

In the last step functionality was added to each subclass with various custom methods and object calls such as Boolean stairDisappear(), which helped created the stairs fading away effect. The algorithm approach used for traveler then served as a template to construct the second three level hierarchy pyramid object.

CONCLUSION

The project application delivered a GUI that represents a human traveler in Egypt making a hand wave gesture while behind him represents the transforming stack object elements that are part of Egypt including pyramids, sandstorms and a day to evening, morning sun to eclipse transition. The result is a fully functional end application that meets all project requirements and guidelines.

REFERENCES

2D Transformation

https://www.processing.org/tutorials/transform2d/

Arc Design for Traveler Hat

https://processing.org/reference/arc_.html

Day and Night Pyramid Reference Image

https://www.google.ca/search?q=pyramid+night+and+day&rlz=1C1CHBF_enCA814CA814&source=lnms&tbm=isch&sa=X&ved=0ahUKEwj4q4XP85jeAhVrh1QKHSWnDhcQ_AUIDigB&biw=1604&bih=744#imgrc=q_CoUv8SaWeU1M:

Egyptian Hat Reference

https://www.google.ca/search?q=egyptian+hats&rlz=1C1CHBF_enCA814CA814&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjwrayZrJjeAhUiLX0KHeYzAigQ_AUIDigB&biw=1604&bih=792#imgrc=28qbAN0mPMOgLM:

Staircase ArrayList

https://processing.org/reference/ArrayList.html

Sandstorm based Array

https://processing.org/reference/Array.html