A Visual Designer for Generating Multimedia Simulations

Robert Smyly

(Mentor: Dr. Jeff Gray)

Mountain Brook High School

Software Composition and Modeling Laboratory
SoftCom
Department of Computer and Information Sciences
University of Alabama at Birmingham

Project Information

Abstract

The primary objective of this research project is to enable end-users with the ability to animate a multimedia simulation using a visual design tool. The specific application context is the recreation of an automobile accident that was witnessed by bystanders. When an automobile accident is disputed in court, a helpful aid toward understanding the accident is a computer animated model. Typically, the creation of such a model involves expensive software and trained professionals who know how to use the software. Such animation software often has many advanced features such as using physics to find unknown variables. However, such expensive features are usually not necessary when applied to smaller accidents and make it difficult for average end-users. Furthermore, most commercial animation tools use generic vehicle models and scenery, which limits the specific recall of the accident scene. The solution that was investigated in this research led to the creation of a visual designer that allows an average end-user with no previous experience to recreate the accident using real-world customized images. This is done by creating an easy to use graphical interface in which the user can simply click on where the animated objects should be at a certain point in time and allow custom images to be used in the animation. The approach is general enough to find application in other domains, such as air traffic control recreation, or other domains that may require an end-user to recreate a visual image of their recollection of an event.

Project Objectives

The objective of this project is to:

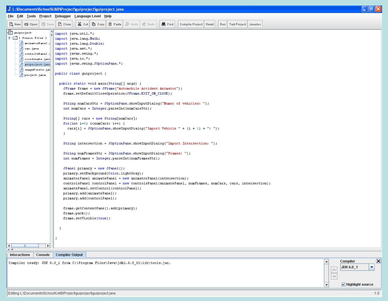
- Create a visual designer that will allow inexperienced end-users to easily create a multimedia simulation
- Allow for the animation to be created simply by clicking where the objects should move
- Make the program able to be applied to the domain of the user's expertise
- Keep interaction with the program as simple as possible while still meeting the primary goal

Key Challenges

- The interface needs to be easy to use without sacrificing the user's ability to control the program to the level that they prefer.
- Support input and processing of any external image.
- Make program versatile enough to be used for many different scenarios

Background: Java and Media Computation

The visual designer is written in Java, which is a widely-used programming language created by Sun Microsystems. The code for this program was written in the Dr. Java IDE, which is a light-weight integrated development environment (IDE) written especially for students. The visual designer used a multimedia Java library of classes from Georgia Tech that expands on Java's standard media computation libraries.



The Dr. Java IDE

```
for(int i=0; i<testCars.length; i++) {

      imagePixels[] tempPixels = testCars[i].getCarPixels();

      for (int j=0; j<tempPixels.length; j++) {
        if (!(tempPixels[j].getX() >= output.getWidth() || tempPixels[j].getX() <= 0 ||
              tempPixels[j].getY() >= output.getHeight() || tempPixels[j].getY() <= 0)) {

          Pixel outputPixel = new Pixel(output, tempPixels[j].getX(), tempPixels[j].getY());
          outputPixel.setRed(tempPixels[j].getRed());
          outputPixel.setGreen(tempPixels[j].getGreen());
          outputPixel.setBlue(tempPixels[j].getBlue());

        }
      }
    }
```

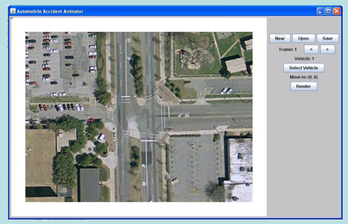
A segment of the code that stacks two image layers.

Example: An Automobile Accident

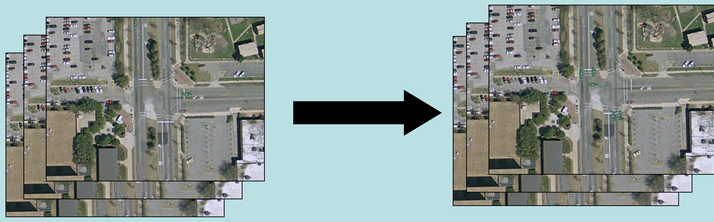
In this example, the visual designer is used to recreate an automobile accident scene.

Step 1: When the visual designer starts, the user is asked to input basic information about the simulation (e.g., how many vehicle are involved, which images to use for the vehicle, which image to use for the intersection, and how long the simulation will last).

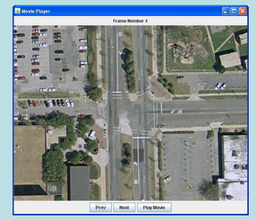
Step 2: From the main screen of the visual designer, the user clicks on the location of the first vehicle for the first frame. The user then progresses through all of the frames and vehicles, selecting their locations and positions. Once completed, the user then press "render".



Step 3: After the user selects "render", the program takes all of the location information and generates an image for each frame, placing the vehicles in the correct location.



Step 4: The images are loaded back into the visual designer and sequenced into a video that is displayed in a new window.



Images taken from Google Earth: http://earth.google.com/

Limitations Representing Future Work

- Add support for key frames: Using key frames, the program can automatically create frames when the objects move in a predictable pattern, such as a straight line.
- Export Video to MPEG: Currently, the visual designer can only export the individual frames and must reconstruct them into a video each time that it is viewed.

Acknowledgements

- This project was created in the Dr. Java development environment. More information can be found at *http://www.drjava.org/*
- For image processing, the visual designer uses a set of classes developed by Georgia Tech. More information can be found at *http://home.cc.gatech.edu/TeaParty*