

Let's Dance: A Tool for Teaching Introductory CS Programming Concepts through Dance

Hanan Al Nizami

Department of Computer Science & Information Systems
Youngstown State University
haalnizami@student.yzu.edu

Advisor: Dr. Tiffany Barnes

Department of Computer Science
University of North Carolina at Charlotte, NC
tbarnes2@uncc.edu

ABSTRACT

This paper presents the Dance Choreography Tool as a culturally situated design tool (CSDT) designed to teach Computer Science (CS) concepts while having a good time. While leaving a good impression, we aim to increase general CS knowledge within children and boost creativity. This paper goes about the ways and describes the procedures taken to develop this tool.

KEYWORDS

Cultural Situated Design Tool (CSDT), Dance Tool, Computer Science.

1. INTRODUCTION

The Dance Tool that is designed to teach children of various ages simple concepts of computer science (CS) while having a good time. This web based tool provides an interactive applet in which children can manipulate 3-Dimensional scripting figures to perform dance movements in unique sequences and share them with a community.

The main goals of the project are 1) we want students to learn, and have a better understanding of, some important concepts in computer science while having fun, and 2) this program is one of a different nature that targets the youth/pop groups, therefore, we want to give them a good opportunity of being exposed to CS; a way to increase the possibility of bringing them into the field. Also, 3) we want to disprove the typical stereotypes associated with CS as being boring and uninteresting by creating a platform where students of different ages, colors and backgrounds can play with and enjoy. By attracting the new generations to the field, not only are we benefiting from more enrolments in CS majors in its general aspect but also, 4) we are hoping to bring more women of minorities and different ethnicities into the field and support diversity.

This paper will describe procedures that were followed to achieve our goals and the plan to which we followed to make this idea happen.

2. MOTIVATION AND BACKGROUND

It has been noticed that students lose interest in CS in their middle school and high school years [1], a problem that has reduced the number of enrollments in CS majors. It has also been shown that certain stereotypes are commonly associated with computer scientists, such as “nerds” and “geeks”, a reason for African Americans and some students of minorities in specific to shy away from approaching the field [2].

The idea of the dance tool developed as a part of CSDT. CSDT's main focus was to teach students mathematical principles while creating simulations of cultural art [3]. When targeting user's culture and personality, it is easier for them to accept and absorb the lessons taught to them through an educational application. Through such tools, students of minorities were able to learn about their culture while learning math concepts¹.

Scratch is another inspiration to our tool. It is a Tile-based visual programming environment and toolkit that allow children to make games, animated stories, interactive art, and share with others on the Net². Children are able to show their work to others and they are able to get feedback. Also, they can explain the procedures in how they got their animation to perform a particular task. This way, indirectly, students are learning how to program while having a good time.

Our Dance Tool enforces the strongest points of both ideas and adds more usability for the user. As mentioned earlier, this tool targets youth/pop cultures and presents a fun and effective learning method in which they can create their

¹ See <http://www.rpi.edu/~eglash/csdt.html>

² See <http://scratch.mit.edu/>

own dances and synchronize them to pieces of music of their choice. A student can visually play the animation he/she has created and, accordingly, fix their code to accommodate the certain moves they want to script. Students can be creative with their dances and can present their work to a community of Dance Tool users. By having a place for the dance tool users to upload their work and present it to a community of users, we are creating a support mechanism that enforces good work. By that, students learn to be more and be more creative.

The benefit this tool has over other tools is that it is both event and time oriented and controlled. A timeline is built to keep track of events added by the user that take place on certain times of a song. This will explain in detail later in this paper.

3. DESIGNING THE TOOL

Our Dance Tool was set to be scripted and created in Java programming language. Why? because we needed to create our application to run on any machine. We took into consideration that some schools and/or students might not operate newer machines or applications to run certain programs. We wanted users to be able to access this application with something as simple as having an internet connection. This application will not require any installs for it to run, nor will it require updating any applications. Also, Java is platform independent. It can actually run on any computer. All it needs is an interpreter for Java bytecode [4]. Taking that into consideration, Java was a good candidate for creating our tool. The IDE we use to code and design the application's interface is NetBeans IDE 6.1.

3.1 Getting Started

First, we got around getting familiar with Java. We started out by testing Java's ability to run some of the tasks carried out by the Dance Tool to achieve the primary goals. Our tool contained functionalities such as adding 3-Dimensional objects to a canvas (the environment in which the 3-D dancers and objects will be added into), rendering the objects, adding drag and drop functionalities to buttons that will contain pre-scripted functionality that will affect the 3-D figures. We broke down the tasks to the following:

- Making a simple program that can be loaded to the web since our tool is going to be web based. We had to make sure it is possible to upload Java applications to the internet.

- Building an interface that contains a canvas then coding it to have the ability to contain 3-D objects within it.
- Rendering 3-D objects in a canvas. As a continuation to the success of the previous bullet, we needed to make sure a 3-D object was able to be rendered and manipulated in a canvas.
- Applying drop functionality to JPanels. The framework for drop can be applied to JList, JTree, JTable by default. JPanels, on the other hand, do not contain that ability to accept drag enabled Swing components.

In order to achieve that, we had to do the following: 1) Plan little assignments that contain some functionality; as we needed to test the ability to script the functionalities we require for our dance tool in Java. 2) Research and implement code to those assignments and make them work.

In figure 1, this assignment was one of the first ones we worked on towards testing Java's ability to upload to the web.

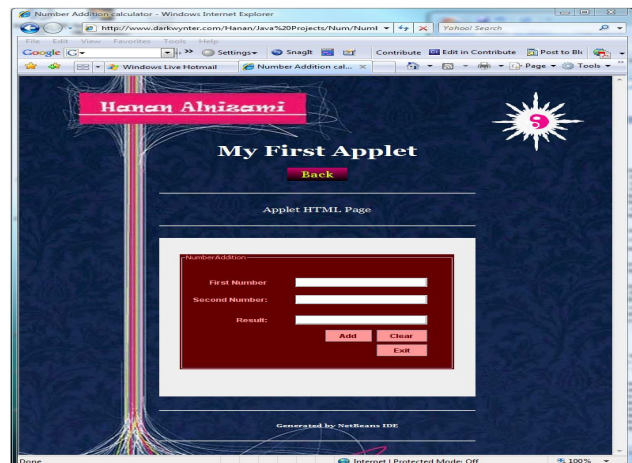


Figure 1: Calculator Java application Uploaded to the web.

Here, we created a simple calculator in NetBeans then we tried uploading the application to the web. Fortunately, that was an easy task. JApplets in NetBeans, generate an HTML page containing the application. Accordingly, our first task was achieved and we were able to upload applications to the internet.

Then, we had to work on building an interface that contained a canvas. That canvas should be able to paint anything the program tells it to draw within it. Figure 2 below illustrated our work for assignment 2.

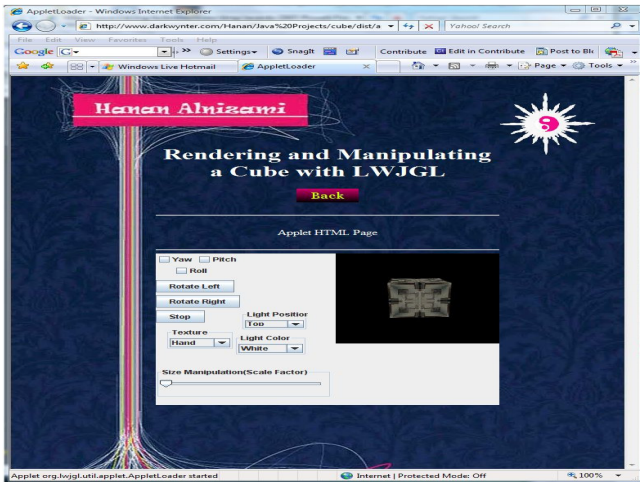


Figure 2: Rendering Cube Application

In this particular assignment, we created a 3-D cubical shape in the canvas. That required adding the LWJGL library. Through such libraries, different functionalities in Java are possible through pre-scripted methods and functions that can be called into the application. As you can see in this illustration, we added swing elements that are responsible for rendering different properties of that particular cube. First, we worked on scripting code to render the yaw, pitch and roll rotations of the cube. That was made possible by checking or un-checking the box relating to the specific rotation direction then clicking the “Rotate left” and “Rotate Right” buttons for navigation purposes. Also, the “Stop” button is scripted to stop the rotation. We added swing elements that would handle texture, light position and Light color. Selecting a texture from a drop down menu, a user is able to change different textures on the cube. Through another drop down menu, a user can change light positions in the canvas space at the top, bottom, right, left, and front of the cube. Also, a user is able to manipulate the light color pointed at the cube. Colors such as white, red and green were available. It was very important that we add a slider bar to manipulate size of the cube. What that slider bar actually does is changing the camera’s position in space by moving it closer to the shape making the shape look bigger, or moving it away from the shape making the shape look smaller. Summing up the results, we were able to achieve our task. We were able to know that Java will allow 3-D object-rendering and camera-manipulating.

In our Dance Tool application, some buttons are dragged and dropped into a panel. Therefore, the assignment that followed consisted of making the drop functionality possible for a JPanel where it is able to accept a dragged button. Those buttons are responsible for telling the 3-D object what to do and how to move in the canvas. This assignment required lots of research and reading.

It was learnt afterwards that the drop functionality is made possible by pre-scripted handlers that tell allows the program to perform a drop act. A new handler was created and added to the project in means to allow panels to accept dropping a button within it. Below is the final applet.

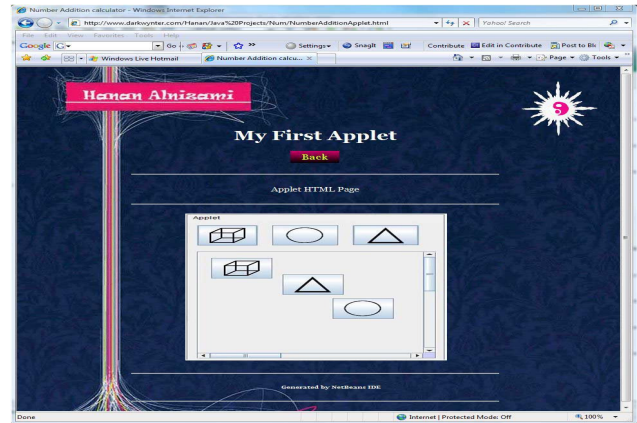


Figure3: Drag and dropping button instances to a panel

By that, we have achieved the main functionalities of our tool, and we were ready to lay out the basic design for the Dance Tool.

3.2 Taking It to the Next Step

3.2.1 Overview

It was time we started creating a visual interpretation of the Dance Tool we have been mentioning for the first half of this paper. A mock up sketch designed in 2007 by Eve Powell was a starting point for our design (check figure 4). First, we read some very important publications that helped us decide how to sketch our tool. We took time to critique the mock up design by adding and subtracting elements to make the design as effective, straight to the point and clear as possible.



Figure 4: DanceTool Mock Up

We came to an agreement on which elements of the Dance Tool were more important and highly recommended to

keep, and which ones were unclear and needed changed. In the final design, we have decided the Dance Tool is going to include 3 panels. Each panel is going to contain a different functionality. There will also be a timeline in which all events are added and synchronized to some music. All elements of the application interact together to manipulate a 3-D figure that is placed in the canvas. Check figure 5.

By that, we are building an environment that would resemble a dancer's surroundings. The closer the user can relate to our tool, the more responsive he/she would be to the goals we want the tool to achieve.

After scripting the dances, a user should be able to save and publish their work. They should also be able to open a previous project or create a new one without having to exit the current project. They should also be able to pick a dancer. We have 2 dancers available for scripting; Maya and Max. They are both created in an application called Autodesk Maya 8. A few dance moves are already scripted for Maya but not yet applied to the Dance Tool.

3.2.2 Panels

The First Panel was agreed upon to have 6 tabs. The first tab is titled the "Music Marker". In this tab, the user can create a time variable in which an event is added on the timeline. Variables are placed in a panel after being created and can be dragged to the panel below (panel 2).



Another tab is the "Dance" moves tab. In this tab, a user is able to drag a pre-scripted dance move presented by buttons and dropping it in a panel 2.

The third tab is the "Beat" tab. Depending on the musical background of a user will be able to add musical notes to which the dance moves are synchronized. For example, a user will be able to tell the dancer to perform a specific dance move every whole note, half a note, or quarter of a note and so forth. We are still working on the functionality of this tab.

In the fourth tab, or the "Random" tab, a random variable is picked from the list of available variables of the same type to perform a specific task. This tab is still being modified.

The fifth tab is the "Math" tab. Users are able to add mathematical operations to statements. Buttons in this tab are drag enabled.

"Control" is the last tab in that panel. It includes pre-defined loops that a user can drag and drop them into panel 2. The user is able to add conditional loops to the dance script being built by dragging them from their location on the tab and dropping them into the notepad in the panel 2. The tab contains if statements, while loops, for loops, for each loops and forever if loops.

The Second panel is where buttons from panel 1 are dropped. An empty notepad sticker first is created by clicking on the "create new" icon  that is located under the panel. Variables, methods, loops and operations from panel 1 can be dropped within the notepad to script moves. The use of such a panel is allowing the user to see the code they are building and learn its functionality as they do so. The user can also delete a notepad by clicking the "delete" icon .

The Third panel is the one that contains the canvas. This is where all 3-D objects are painted. Some elements to be added to the canvas are still being worked on. In the near future, a user will be able to pick a character from a list of characters and modify them the way they please. By default, we will include Maya, our female character, for the user to operate. A user will be able to pick an environment where they would like their 3-D figure to be. For example, we have a stage available by default at the moment. The user will be able to fill the surroundings with 3-D object of his/her choice. This will trigger a user's creativity. Available 3-D objects a user can pick as of now are a disco ball, a camera, a spot light, subwoofers and speakers, chairs. All those elements were created and rendered in Autodesk Maya 8.5 and then exported over to the dance tool with the help of the COLLADA exporter³. In this panel, a user will be able to move the camera, add lights and preview the dance moves that are already scripted for Maya and/or Max.

3.2.3 Timeline

The timeline added to the Dance Tool is a slider bar that is scripted to fit the length of the music. It shows where events are taking place according to time. A user is able to play, stop, rewind, forward, go to the beginning and/or go to the end of an animation. Figure 5 illustrates the final design.



Figure 5: The Dance Tool

³ Maya COLLADA plug-in supports importing and exporting Maya scenes according to the COLLADA format specifications. It preserves information about the objects exported.

4. ACCOMPLISHED

As it has been noted, our Dance Tool is still under development. Now, a user is able to create and delete a notepad in panel 2. Functions and parameters from panel 1 can also be dragged and dropped into panel 2 within the notepad. They are able to appear and take position in the notepad but carry no functionality yet. We currently have a default figure (Maya) for a user to see in the canvas. The timeline is able to read the length of the music. The music can be played, paused and stopped. Maya can only move moving left and right, up and down. A loader is scripted to read the 3-D objects created, rendered and textured in Maya after importing the COLLADA file to the project.

5. Future Work

In the future, the user will have more freedom selecting a dancer of their choice. They will be able to chose a stage or background and insert 3-D objects to it. Buttons in all the tabs of panel 1 will also carry a functionality that will affect the 3-d figure in the canvas directly. A user will be able to use the buttons new, open, save and publish successfully. The timeline will be an interactive one to which all events will appear in different colors. Clicking on one timeline events will highlight the corresponding note in panel 2. A clue featured in a little pop up note will appear by hovering over a button to access a description of the functionality and how it can contribute to the existing script.

6. ACKNOWLEDGMENTS

Our thanks to the CRA Distributed, Dr. Tiffany Barnes for mentoring the project, and Eve Powell for supervising.

7. REFERENCES

- [1] Kelleher, C., Pausch, R., and Kiesler, S. 2007. Storytelling Alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1455-1464. DOI= <http://doi.acm.org/10.1145/1240624.1240844>
- [2] Burge, J. D. and Suarez, T. L. 2005. Preliminary analysis of factors affecting women and African Americans in the computing sciences. In *Proceedings of the 2005 Conference on Diversity in Computing*
- [3] Eglash, R., Bennett, A., O'Donnell, C., Jennings, S. and Cintonino, M. Culturally Situated Design Tools: Ethnocomputing from Field Site to Classroom. *Am. Anthropol.*, 108 (2). 347--362.
- [4] Eck, David J., 2007. Introduction to Programming Using Java. (Geneva, NY, USA, November, 2007).