

Culturally Situated Design Tool: The Dance Choreography Tool

Jasamine Jackson

Department of Business Administration
Winthrop University
jacksonj3@winthrop.edu

Dr. Tiffany Barnes

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

ABSTRACT

This paper presents the design of the Culturally Situated Design Tool (CSDT) known as the Dance Choreography Tool. The Dance Tool was created as a supplementary educational aid for high school students to learn the basics of computer programming by scripting dance moves for a 3D avatar. Even though there are other methods of teaching computer programming, the dance tool will increase motivation in learning programming by providing an environment that encourages creativity and diversity while focusing on the logic behind computer programming. By increasing the interest of computer programming on a high school level, the dance tool will also help solve the problem with the decline of people entering the field of computer science in undergraduate and graduate school.

Keywords

Culturally Situated Design Tools, Dance Choreography Tool, dance tool.

1. INTRODUCTION

Studies have indicated that there is a decline in the number of people choosing computer science as their undergraduate and/or graduate field of study, especially for minorities and underrepresented groups [10]. One of the main reasons for the declining numbers includes the stereotypes behind computer science that are assumed to be true such as “nerdy”, “boring”, and “complex” [2].

In order to solve this problem, we decided to shatter this stereotype on the high school level which is the time that students decide what to do with the rest of their life which includes what subject to study in college. We wanted to create software that is relatable to the youth culture and different from the traditional methods of teaching computer science. We also want this software to be easily applicable to all high school without paying a high cost. Since people stereotype computer science as a complex topic that is only meant for computer science geniuses, we want our software to show that learning computer science is definitely not as impossible as it seems.

2. BACKGROUND AND MOTIVATION

One of the inspirations for the dance choreography tool was Dr. Ron Eglash’s Culturally Situated Design Tools (CSDTs) which uses a combination of anthropology, mathematics and computing to teach mathematical concepts such as using the art of African cornrow braiding to teach transformational geometry [11]. Another inspiration for our software was the user interface of Scratch, software that allows the user to create 2D animations by

dragging and dropping commands that are pre-scripted into puzzle pieces into a panel to create animations, stories, and games to share with the world [9](See Figure 1).

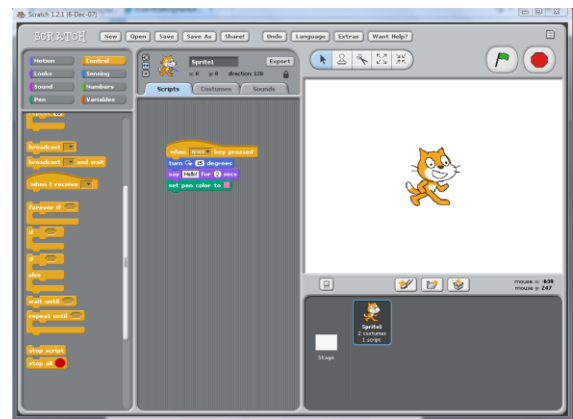


Figure 1: Scratch Interface

In order to make sure that our tool isn’t too complicated to use, we decided to use a similar drag and drop system with pre-scripted code already built in since one of the common problems with beginner programmer’s is syntax errors [1]. Letting the user see the result of the code that was dragged into the panel through animation output allows the user to see how their code.

Our group decided to use the idea behind the CSDTs in order to make our software fun and relatable to the youth culture. Dr. Eglash has already created a dance tool called BreakDancer: Rotation and Sine Function (see Figure 2) which uses the culture of break dancing to relate to geometric concepts.

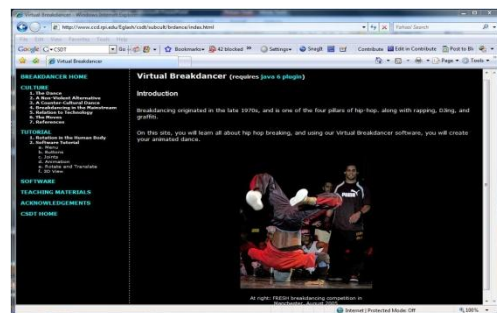


Figure 2: BreakDancer Tool

Using the break dancing idea, we've decided to create a modified version of the BreakDancer tool to teach the basics of programming. Besides the different concepts that our tool and the BreakDancer Tool teaches, our dance tool and the BreakDancer Tool have three major differences:

1. With our dance tool, the user will be able to use drag and drop controls to script dance movements.
2. With our tool, the user will be able to use a timeline to schedule dance movements for an entire song.
3. The BreakDancer tool allows the user to choreograph dance movements frame by frame. Our tool will already have the dancing animations programmed into the drag and drop components. These animations include popular break dancing moves such as the flare and the worm.

In order to make the dance tool easily accessible to high schools, our group has decided to take advantage of java programming language (since it is an independent -platform) and to create the dance tool as an applet. This way, any high school that has access to the internet will be able to use our software.

3. DESIGN AND IMPLEMENTATION OF DANCE CHOREOGRAPHY TOOL

3.1 The Design Layout

Since the idea for the project started in 2007, the layout for the project was already completed before starting this project. Our task was to implement the design layout of the dance tool and finish a prototype to test at the local high schools. Since we are using the same design layout for the implementation of the dance tool, detailed description of the layout is included in this paper.

3.1.1 The Overall Design

Figure 3 show the mock-up design for the Choreography Dance Tool.

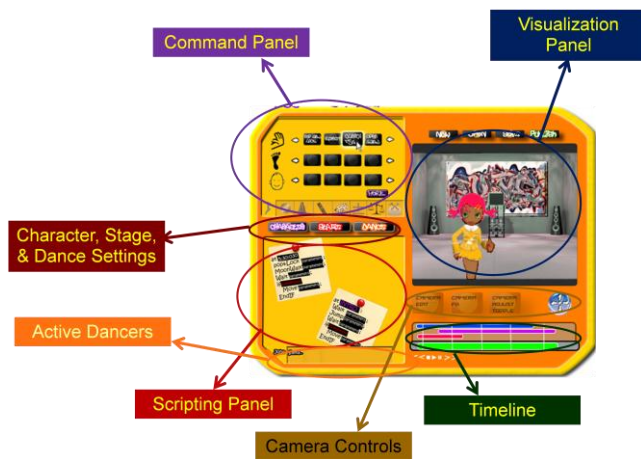


Figure 3: Mock-up Design of the Dance Choreography Tool

Instead of using the predictable shades of grays and blues for the applet and buttons, we've decided to use bright and exciting colors and font that will attract the user's attention. The tool has three buttons located above the control panel: stage mode (control the stage settings for the dancer), character mode (controls the character settings or appearance of the 3D avatar dancer), and

dance mode (allows the user to script dance movements). The layout also includes a timeline that allows the user to schedule events at different times in the song. Now we are going to go more into detail about the capabilities of each panel in the dance tool.

3.1.2 The Command Panel

The command panel (also seen in Figure 4) contains several tabs which contain all of the drag and drop components that the user needs to choreograph dance movements.

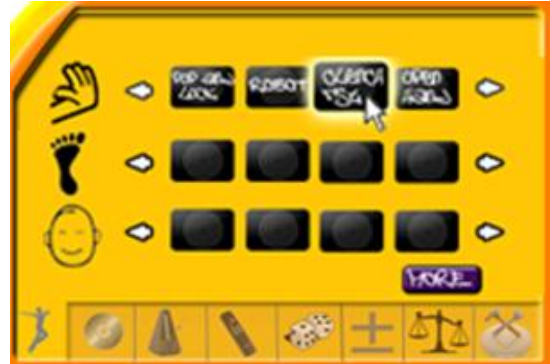


Figure 4: The Command Panel

All of the features of the command panel will be discussed later in this paper.

3.1.3 The Scripting Panel

The scripting panel (also seen in Figure 5) is where the user will be able to script the dance movements for their 3D avatar.



Figure 5: Scripting Panel

First the user will generate a notepad in the control panel (notepad generator will be added to the final design). This is where the user will be able to script the dance movements for their avatar. To delete a notepad or a command that was scripted onto the notepad, the user will use the delete notepad button (created in the final design implementation). The active dancer area in the control panel (also in Figure 6) allows the user to keep track of which avatar the user is choreographing dance moves to just in case the user is scripting dance movements to more than one avatar.



Figure 6: Active Dancers Section

In Figure 6, for example, the dancers include two avatars, John and Sama. John is the current avatar that the user is scripting dance movements to.

3.1.4 The Visualization Panel

The visualization panel (also shown in Figure 8) is where the user will be able to see their 3D avatar in action.



Figure 8: Visualization Panel

The visualization panel will display the current avatar(s) the user has choreographed dance movements for, the 3D stage and stage props the user has chosen (based on the stage setting), and the dance animations of the avatar which depends on the scripted movements of the user.

3.2 Implementation of the Design Layout

We decided that easiest way to start implementation is to examine the functionality of each panel and figure out how to execute the functionalities through java programming language, NetBeans IDE 6.1, and the LWJGL (Light Weight Java Gaming Library). After analyzing the panels, our next step would be to figure out how to implement the timeline for our dance tool.

Since this project was started by a group of researchers in 2007, there are earlier version of the dance tool that was created using NetBeans and java. We used these earlier versions to assist us in figuring out how to implement a better and more modern version of the dance tool. Since the earlier versions used NetBeans and java, we decided to follow the trend.

3.2.1 Drag and Drop Controls for the Control Panel

To create the drag and drop components for the control panel, we've decided to run a simple test of dragging one button from one panel and dropping it into another panel (Figure 9).

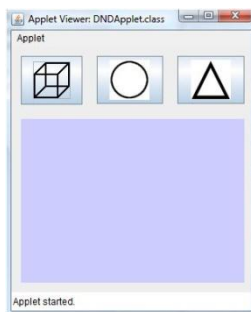


Figure 9: Drag and Drop Applet

After dropping the button into the panel, we want to be able to drag the button around the panel according to the position of the mouse. We soon realized that the task was a little trickier than expected.

However, after a few adjustments in our drag and drop program, we were able to take out all of the bugs.

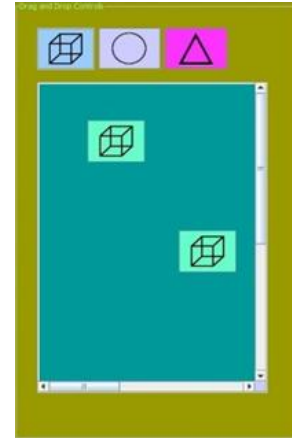


Figure 10: Drag and Drop Applet with color added

3.2.2 Using Drag and Drop Operations to Control 3D graphics

Whatever commands the user drags into the notepad, the avatar will execute. Thanks to Su Hyung Cho (grad student that have worked on the project in the past); we already have a 3D avatar that was created using 3D Studio Max and animations of break dancing moves. Our next task was to see figure out how to use drag and drop controls to manipulate a 3D figure in a canvas. We used drag and drop controls for rendering a cube using the LWJGL (Light Weight Java Game Library) in Figure 14. When we drag the cube icon into the panel, a 3D cube is rendered to the canvas.

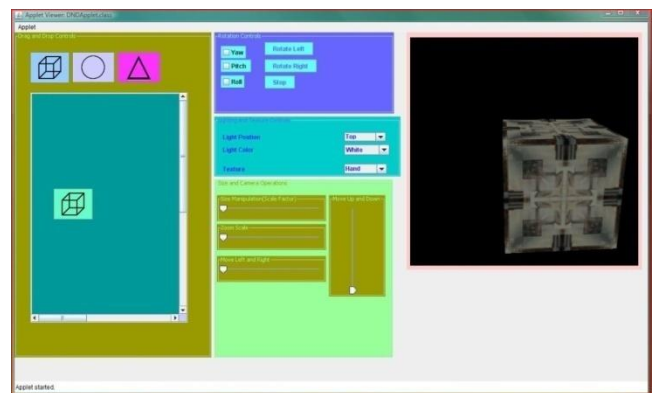


Figure 14: Rendering 3D cube to a canvas through drag and drop controls

3.2.3 Loading 3D figures into NetBeans IDE 6.1

One of the most difficult tasks that we are still in the process of finishing is loading the 3D stage and stage props that we created from Autodesk Maya 8.5 and the 3D avatars that were created

from 3D Studio Max into NetBeans IDE 6.1. 3D models are shown in Figures 15-16.

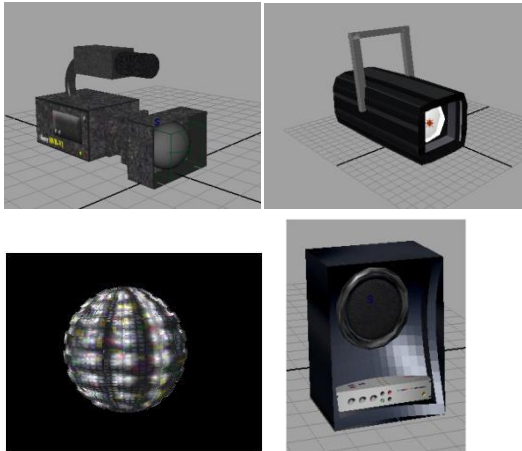


Figure 15: Stage props created using Maya
Top Left: Camera Top Right: Stage Light
Bottom Left: Disco Ball Bottom Right: Speaker



Figure 16: Avatar dancer (Maya) created using 3D Studio Max

Since we are using LWJGL, we want a loader that is compatible with this library. However, there isn't a loader that preserves the skeleton of the 3D figures which is what we need for to load these 3D figures in java. Based on research that was done on loaders, we decided to create a COLLADA loader from scratch that can have all the capabilities that we need for loading 3D figures (See Figure 17 and Figure 18).

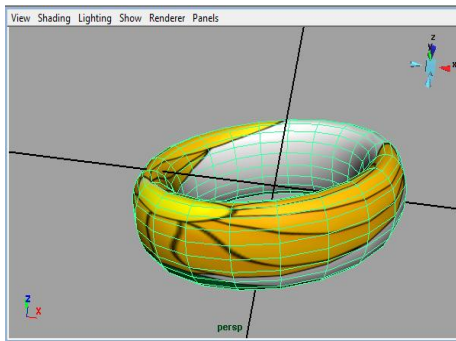


Figure 17: 3D Model in Maya

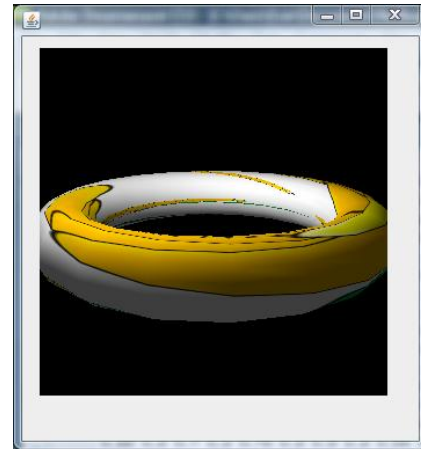


Figure 18: 3D Model in Java Applet using COLLADA loader

3.2.4 Timeline Design: Adjusting the position of a JSlider according to the time past playing a song

We are still working on the design for the timeline. We have decided that the first task in the time line is figuring out how the applet is going to keep track of the time that pasts in a song. In order to accomplish this task, we decided to use a JSlider and JButtons to create an image similar to that seen in Figure 18.



Figure 19: Screenshot from Windows Media Player

4. CONCLUSION

The dance tool is still a work in progress. However, we were able to make great strides towards making a great educational tool. Figure 18 is a screenshot of what we have so far.

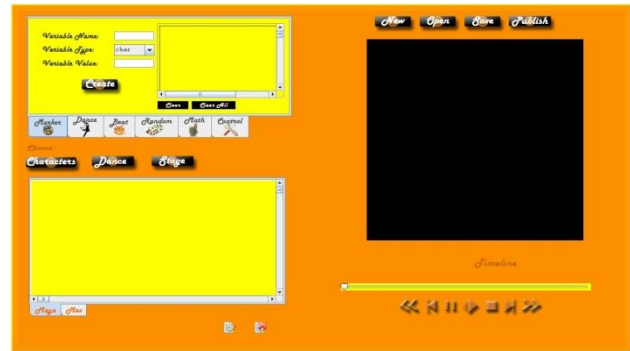


Figure 20: Implementation of the Mock-Up Design in NetBeans

Our implementation has the same features as the mock-up version with a few differences. One of the differences is the timeline (seen in Figure 18).



Figure 21: Our version of the timeline

One of our main concerns for the timeline was figuring out how the applet will be able to keep track of how much time has passed as a song plays. In order to accomplish this task, we decided to have a JSlider change its position to reflect how much time has past while a song is playing.

The song controls below the JSlider were created from JButtons and each button executes the appropriate function. Our applet is able to play songs from a default list that we provide. However, we are still figuring out the details of how to schedule events on the timeline.

Figures 22 – 27 shows the features of the command panel.



Figure 22: Variable Generator



Figure 25: Random Variable



Figure 23: Dance Move Controls



Figure 26: Math Operations



Figure 24: Dance Beat Controls



Figure 27: Conditional Controls

Figure 22 is the variable generator which allows the user to create any variable that is needed to script their dance moves. For example, when the user generates a notepad, the first variable that a user will need will specify at what point in the song the avatar will start dancing (example shown in Figure 28).

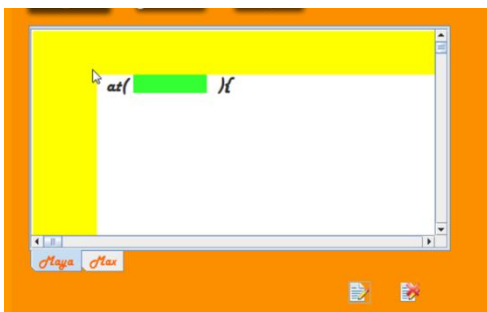


Figure 29: Empty Notepad

If the user wants to specify at what point in the song that they want choreograph the dance moves for their avatar, they would create the variable or marker in the variable generator. The user will enter name of their variable, the variable type and value. If the user creates the variable correctly, a button will appear in the panel with the name of their variable (shown in Figure 28).



Figure 28: Generating Variable in Marker Panel

The button contains the data that the user entered for the variable (name, data type, and value) and can be dragged and dropped into the notepad. If the user enter invalid data for the variable (ex: an integer value for a boolean type variable) a button won't show in the panel. The name of each panel describes the functions of each panel:

- The dance panel (Figure 23) contains drag and drop buttons of dance animations that the user will use to script their dance routine.
- The beat panel (Figure 24) contains drag and drop buttons that gives the user the choice of what beat they want the avatar to “dance”. For example, the user can choose the quarter note button if they want the avatar to move at every quarter beat.
- The random panel (Figure 25) allows that user to drag and drop a random variable button to obtain a random variable value when needed.
- The math panel (Figure 26) allows the user to drag and drop buttons that contain mathematical operations into the panel when needed (addition, subtraction, etc.).
- The control panel (Figure 27) allows the user to drag and drop conditional statements into the notepad when needed (while loop, for loop, etc.).

The last difference between our tool and the mock-up version of the dance tool is the addition of the notepad generator button and notepad delete button (shown in Figure 30).



Figure 30: Left Button: Notepad Generator

Right Button: Notepad Delete Button

To generate a notepad into the scripting panel, the user must drag and drop the notepad generator button into the panel. To delete a button from the panel, the user must select what notepad they wish to delete and press the button to delete the notepad.

5. FUTURE WORK

Figure 30 shows a mock-up of what our tool should look like (excluding the final design of the timeline which we are still deciding).



Figure 28: Mock-Up Implementation

5.1 More Functionality to the Buttons

The buttons in the control panel needs to have the appropriate functionality programmed into each one. Each button has drag and drop capabilities. However the only buttons that have full functionality at this time is the variable buttons generated by the user. Right now, we are still in the process of adding functionality to the buttons.

5.2 Character and Stage Settings

We want the user to have the option of having more than one dancer performing on stage at the same time. Right now, we are focusing on getting one dancer (Maya) on stage and performing. Once we have a working and successful prototype of our dance tool, we can look into providing options for the user to create their own avatar. We are also looking into providing the user with choices of different stages and stage props to choose from.

5.3 Camera Controls

We are in the process of creating the camera controls for the dance tool.

5.4 Timeline Design

Right now, our applet can play songs and keep track of time that past while the song is playing. We are still trying to figure out what would be the simplest and most efficient way to design the timeline. We are doing research to see if there is some related work that could help us solve this problem.

5.5 Developing and Prototype and Testing

We have a partially working version of the dance tool, but we are aiming towards having a fully functional prototype to test at the local high schools to see what modifications needs to be made and if this is a very effective educational tool.

5.6 COLLADA Loader

We are still in the process of finishing the COLLADA loader. We are able to load figures into the loader and display our figure through an applet, however, we are still having issues with the

textures of our 3D models. After the loader is complete, we will be able to load 3D models into our dance tool.

6. ACKNOWLEDGMENTS

Our thanks go out to CRA-W DMP which provided us with the opportunity and funding to participate in the research program at the University of North Carolina in Charlotte. We would also like to thank Dr. Tiffany Barnes, our mentor for this project. We would also like to thank Eve Powell, our project leader and creator of our COLLADA loader. We would also like to thank Su-Hyung Cho, the creator of Maya, one of our first avatar dancers.

7. REFERENCES

- [1] Alice 2.0. <http://www.alice.org>
- [2] Eglash, R., Bennett, A., O'Donnell, C., Jennings, S. and Cintorino, M. Culturally Situated Design Tools: Ethnocomputing from Field Site to Classroom. *Am. Anthropol.*, 108 (2). 347--362.
- [3] The Java Tutorials: Getting Started with Swing. Retrieved July 29, 2008. <http://java.sun.com/docs/books/tutorial/uiswing/start/index.html>.
- [4] J.D. Burge, T.L. Suarez, "Preliminary analysis of factors affecting women and African Americans in the computing sciences," *rtcdc*, pp. 53-56, 2005 Richard Tapia Celebration of Diversity in Computing Conference, 2005.
- [5] Lightweight Java Game Library. Retrieved July 29, 2008. WWW: <http://lwjgl.org/>.
- [6] NetBeans IDE 4.0 Applet Tutorial. Retrieved May 28, 2008. WWW: <http://www.netbeans.org/kb/articles/tutorial-applets-40.html#embed>.
- [7] NetBeans: Introduction to GUI Building. Retrieved May 28, 2008. <http://www.netbeans.org/kb/60/java/gui-functionality.html>.
- [8] OpenGL Apron Tutorials: The Rotating Cube. Retrieved May 29, 2008. WWW: http://www.morrowland.com/apron/tutorials/gl/gl_rotating_cube.php
- [9] Scratch. Retrieved July 28, 2008 from <http://scratch.mit.edu/>
- [10] Taulbee Survey. Internet. Available WWW: <http://www.cra.org/CRN/articles/may05/taulbee.9to16>.
- [11] Transformational Geometry and Iteration in Cornrow Hairstyles. Retrieved June 2008. WWW: http://www.ccd.rpi.edu/Eglash/csdt/african/CORNR OW_CURVES/.
- [12] Virtual BreakDancer. Retrieved July 2008. <http://www.ccd.rpi.edu/Eglash/csdt/subcult/brdance/index.html>.