

**PROJECT NAME: FUN ACTIVITIES FOR MIDDLE
SCHOOL STUDENTS**

FINAL REPORT

**RYAN BUSH
OLUWASEUN OGUNGBENLE**

06/21/07

Table of Contents

Executive Summary.....	3
Abstract	4
Introduction	5
Project Analysis.....	5
Requirements.....	6
Scope	6
Project Design	6
Questionnaire for the CMP	8
Student Feedback.....	10
Implementation layout.....	13
Design Details.....	13
Conclusion	20
References.....	21

Executive Summary

Over the last five years, the number of students that have chosen to major in Computer Science has dropped by over 50%, nationwide. Thus, several educators have realized that younger students, specifically those in middle school, need to be encouraged to major in Computer Science.

This project was aimed at familiarizing middle school students with various Computer Science tools and various real-life applications of Computer Science. Our tasks for this project are the following:

- Create a comprehensive list as to why CC4G.net might not work for U.S. students.
- Work with students from the Computer Magnet program in the Denver Public School System to determine their impression of the CC4G modules.
- Create a list of possible modules that might encourage U.S. students to be interested in computer science.
- Decide how to implement the modules (Java, Flash, JavaScript, etc).
- Begin implementing one of the modules.

Results

Our first task was to critique the U.K website—CC4G—that tries to achieve a goal similar to ours. Here is a list of the critique of CC4G:

1. This site was developed specifically for girls
2. It was developed specifically for those that are already interested in Computer Science
3. The character diversity is not the same as what exists in the U.S.
4. The language used (especially the spoken one) might be unfamiliar to students in the U.S.
5. Most of the modules do not make a strong connection to Computer Science
6. The quality of many of the activities is below an acceptable status; some of them are either too complex or too easy for their target group—young girls.

In accordance with our second task, twenty-eight acceptable surveys have been collected from the Computer Magnet Program. From the feedback received, it is suggested that proceeding with this project might not yield results that are any better than what we found.

We have fully designed a module that incorporates a Zoo theme. One activity under this module has been fully implemented. The other seven (our target was eight activities) have been designed and laid out. The section on design details explains how the activities work.

Abstract

Over the last five years, the number of students that have chosen to major in Computer Science has dropped by over 50%, nationwide. Thus, several educators have realized that younger children, specifically those in middle school, need to be encouraged to major in Computer Science.

There is a website known as the Computer Club 4 Girls (<http://www.cc4g.net/>) that has many modules and activities aimed at encouraging young girls to develop an interest in Computer Science. However, this website was developed in the U.K. and may not be practical for the U.S. Perhaps a similar website is needed for the U.S.

Thus, we have been tasked to:

- Create a comprehensive list as to why CC4G.net might not work for U.S. students.
- Work with students from the Computer Magnet program in the Denver Public School System to determine their impression of the CC4G modules.
- Create a list of possible modules that might encourage U.S. students to be interested in computer science.
- Decide how to implement the modules (Java, Flash, JavaScript, etc).
- Begin implementing one of the modules.

Introduction

In an attempt to encourage young girls to consider careers in computer science, a British website, Computer Club For Girls (<http://www.cc4g.net/>), has put together a number of Computer Science applications in form of activities that are supposedly appealing to young girls—especially girls in 7th and 8th grades. Our clients have taken a cue from this idea by deciding to create an American version of this site in an attempt to combat the growing lack of interest in Computer Science among young Americans. This project is aimed at widening the scope of this idea by:

- Extending our target group to young boys as well.
- Making it more diverse.
- Improving the quality of the applications.
- Changing the language from British to American English to make it easier to understand.

The reason for wanting to make this new site more diverse is so that other groups in the United States (Hispanics, Asians, pure Africans, etc.) can be reached either by involving them in the characters used or by making use of their recorded voices.

Improving the quality of the applications entails making it more appropriate and appealing to 7th and 8th grade students. The contents, such as the rules that govern the games, also need to be more clearly defined.

Project Analysis

This project is aimed at familiarizing middle school students with various Computer Science tools and various real-life applications of Computer Science. To this effect, we have a five-fold task of:

- Creating a comprehensive list of the critique of the existing U.K. site in America.
- Working with students from the Computer Magnet Program in the Denver Public School system to determine their impression of the CC4G modules.
- Creating a list of modules that might spark young Americans' interest in computer science.
- Deciding how to implement these modules (JavaScript, Flash, etc).
- Implementing one of the modules.

Each of the modules that will be created will contain about three sections, with each section containing about three activities. Our target for this project is to create about eight activities that will help these students appreciate and develop interest in Computer Science, with the hope that this will prompt them to consider future careers in Computer Science. This project will explore areas of Computer Science such as web programming, databases, character recognition, image processing, and robotics. Since most of the activities on the existing site are only relevant to girls, this improved site that we are working on creating will incorporate activities for boys as well. For example, the existing site has an activity called fashion show, which is all about female fashion. The onus rests on us to try to create a similar activity for boys. Included in each of these

activities will be step by step instructions that will guide prospective users. Many of these activities will be games.

Requirements

The requirements for this project are divided into functional requirements and non-functional requirements. Non-functional requirements are the constraints imposed on the design or implementation of this project. Functional requirements specify the behavior of this project.

- ***Functional Requirements:***

- Significantly related to computer science
- Diversity in characters used in animated cartoons
- Targeted at 7th and 8th grade American students
- Easily understandable language

- ***Non-functional Requirements:***

- Implemented with Flash
- Web-based

Scope: Our clients are responsible for putting our finished product on the new website that will be created. Thus, we are responsible for implementing the module(s) and providing all the documentation that goes along with the implementation.

Project Design

The design of this project is in accordance with the five tasks that have been assigned to us. The design is such that each successive task builds on the preceding one. Below is a breakdown of our project design.

Task 1: Critiquing CC4G

Our first task was to critique the U.K website—CC4G—that tries to achieve a goal similar to ours. Here is a list of the critique of CC4G:

1. This site was developed specifically for girls
2. It was developed specifically for those that are already interested in Computer Science
3. The character diversity is not the same as what exists in the U.S.
4. The language used (especially the spoken one) might be unfamiliar to students in the U.S.
5. Most of the modules do not make a strong connection to Computer Science
6. The quality of many of the activities is below an acceptable status; some of them are either too complex or too easy for their target group—young girls.

However, in critiquing CC4G we found a number of activities that, we believe, are relevant to our project. While these activities may be relevant to our project, they cannot be copied, and the onus is on us to try as much as possible to be independent of them in implementing our project. It should also be mentioned that if we found any activity we would like to mimic, we could purchase the rights to it from CC4G.

Below is a list of some effective activities from CC4G. For each of the activities, the first word in parenthesis represents the module; the second represents the section of the module under which the activity can be found on the web site.

- Control (Design Time, Dream Room)
- Track It (Investig8, Collect It)
- Match It (Investig8, Collect It)
- Prove it (Investig8, Witness It)
- Whodunnit (Investig8, Question It)
- Long shot (Fit4Sport, Sharp Shooters)
- Fast ‘Nuff (Fit4Sport, FZiKL)
- Star Browser (Fit4sport, Top Teams)
- Slider (Game Girl, Build IT)
- Game Write (Game Girl, Build IT)
- Flipper (Game Girl, Design IT)
- Video Web (Communi-K8, Studio Live)
- EnigmaticKodes (Communi-K8, Cyber Secrets)
- Lets Animate (DanceM8, Dance Write)
- Colour it Right (Fashion, Accessorise)
- Order Online (Fashion, Catalogue)

Task 2: Student Interview

Our second task is to work with students from the Computer Magnet Program in the Denver Public School system in order to determine their impression of the activities that are on CC4G. Since the whole idea behind this project is to create a set of activities to demonstrate Computer Science applications that might interest middle school students, we thought it might be a good idea to have the students themselves critique CC4G and tell us what they like and do not like about it. To this effect, we have created a questionnaire that will be made available to a randomly selected number of students (about ten of them) during the Computer Magnet Program during the week of June 4th at the University of Colorado at Boulder. The existing site has twelve modules, each of which has about three sections; each of the sections in turn has about three activities. The idea is to select certain activities that we think are relevant to Computer Science, and have the students critique them. Their feedback should help us achieve our aims for this project by giving us a better idea of the kinds of activities we should incorporate into our project, that would interest them (obtaining student feedback is important because our implementation will initially be based on our own perception of fun activities, which may be quite different from middle school students' ideas). A draft of the questionnaire is included in this report. It can also be accessed at <http://alamode.mines.edu/~rybush/survey.html>

Questionnaire for the Computer Magnet Program

The following is a prototype of the survey that was administered to the students that participated in the Computer Magnet Program.

In one to three sentences, write your thoughts about this activity using the comment box below

Comments:

1. How fun was this activity to you?

Not at all Fun
Slightly Fun
Somewhat Fun
Very Fun
Extremely Fun

2. How fun would this activity be for middle school boys?

Not at all Fun
Slightly Fun
Somewhat Fun
Very Fun
Extremely Fun

3. How fun would this activity be for middle school girls?

Not at all Fun
Slightly Fun
Somewhat Fun
Very Fun
Extremely Fun

4. How interesting was this activity to you?

Not at all Interesting
Slightly Interesting
Somewhat Interesting
Very Interesting
Extremely Interesting

5. How interesting would this activity be for middle school boys?

Not at all Interesting
Slightly Interesting
Somewhat Interesting
Very Interesting
Extremely Interesting

6. How interesting would this activity be for middle school girls?

Not at all Interesting
Slightly Interesting
Somewhat Interesting
Very Interesting
Extremely Interesting

7. How difficult was this activity for you?

Not at all Difficult
Slightly Difficult
Somewhat Difficult
Very Difficult
Extremely Difficult

8. How difficult would this activity be for middle school students?

Not at all Difficult
Slightly Difficult
Somewhat Difficult
Very Difficult
Extremely Difficult

9. What topic did this activity concern?

Web Programming
Databases
Animation
Security
Application Programming

10. How easy was it to understand the British accent spoken in this activity?

Not at all Easy
Slightly Easy
Somewhat Easy
Very Easy
Extremely Easy

11. How often did the British speaker use words/terms that American middle school children would not understand?

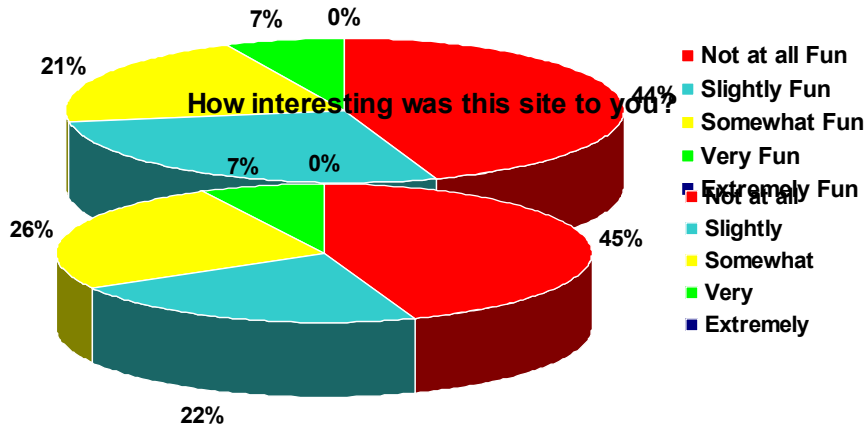
Not at all Often
Slightly Often
Somewhat Often
Very Often
Extremely Often

12. How easy was it to understand the instructions?

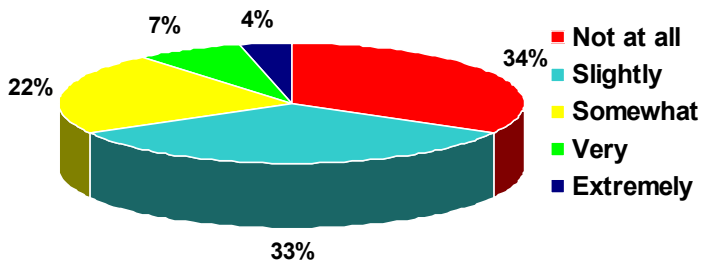
Not at all Easy
Slightly Easy
Somewhat Easy
Very Easy
Extremely Easy

Student Feedback

How fun was this site to you?



How often did the British speaker use words/terms that American middle school children would not understand?



The figures above show the response of the middle school students that participated in the Computer Magnet Program to the survey. These three questions encapsulate the main ideas of our project. From these data, it may be implausible to continue with the project. The results tend to be in disfavor of the CC4G website. Making a new website, that incorporates the same ideas, would probably not yield results that are satisfactory.

Task 3: List of modules

The third task is to create a list of modules—based on the feedback from the students—that might spark young Americans’ interest in computer science. Given the fact that we cannot get any feedback from the students before June 8th, we have decided to create a tentative list of modules that might accomplish the same purpose. Each module has a theme. Each of the three sections of the module represents a Computer Science application, and each of the activities under a given section has some sort of connection to the Computer Science application indicated by the section name. In addition to this, the activities within a given section must follow a logical sequence. An example of this setup would be having a section on programming and logic, and having three activities that make use of programming, such as instructions, condition statement, and functions. A more comprehensive example of the whole arrangement will be shown in Figure 1.

Task 4: Mode of implementation

Another task, sequel to the one above, is deciding how to implement these modules. We had options of JavaScript and Flash. We have decided to use Flash for the implementation of the modules. The specific version of Flash that will be used is Adobe Flash CS3 Professional. The reason for our choice of Flash is because it supports dynamic sites, making it very appropriate for our project, since our project requires texts, images, and animations to move around on the screen.

Task 5: Design layout

Figure 1 is a visual description of our final task –implementing one of our modules. As seen in Figure 1 below, the theme (or name) of the module is Zoo, it is a module centered on animals. Each of the three sections of the module is a Computer Science application that can be used in this context, hence meeting our goal of familiarizing these students with real world applications of Computer Science. For the security section of this module, the two activities so far created, encryption and protection, form a sequence of events in securing the animals in a given zoo. For the web programming section, the activities, html and database, are aimed at some record keeping. The idea behind the programming/logic section is to show how a robot can be made to:

- Follow a set of instructions based on some written code –first activity
- React in a given situation or respond to some disturbance –second activity
- Perform some functions –third activity

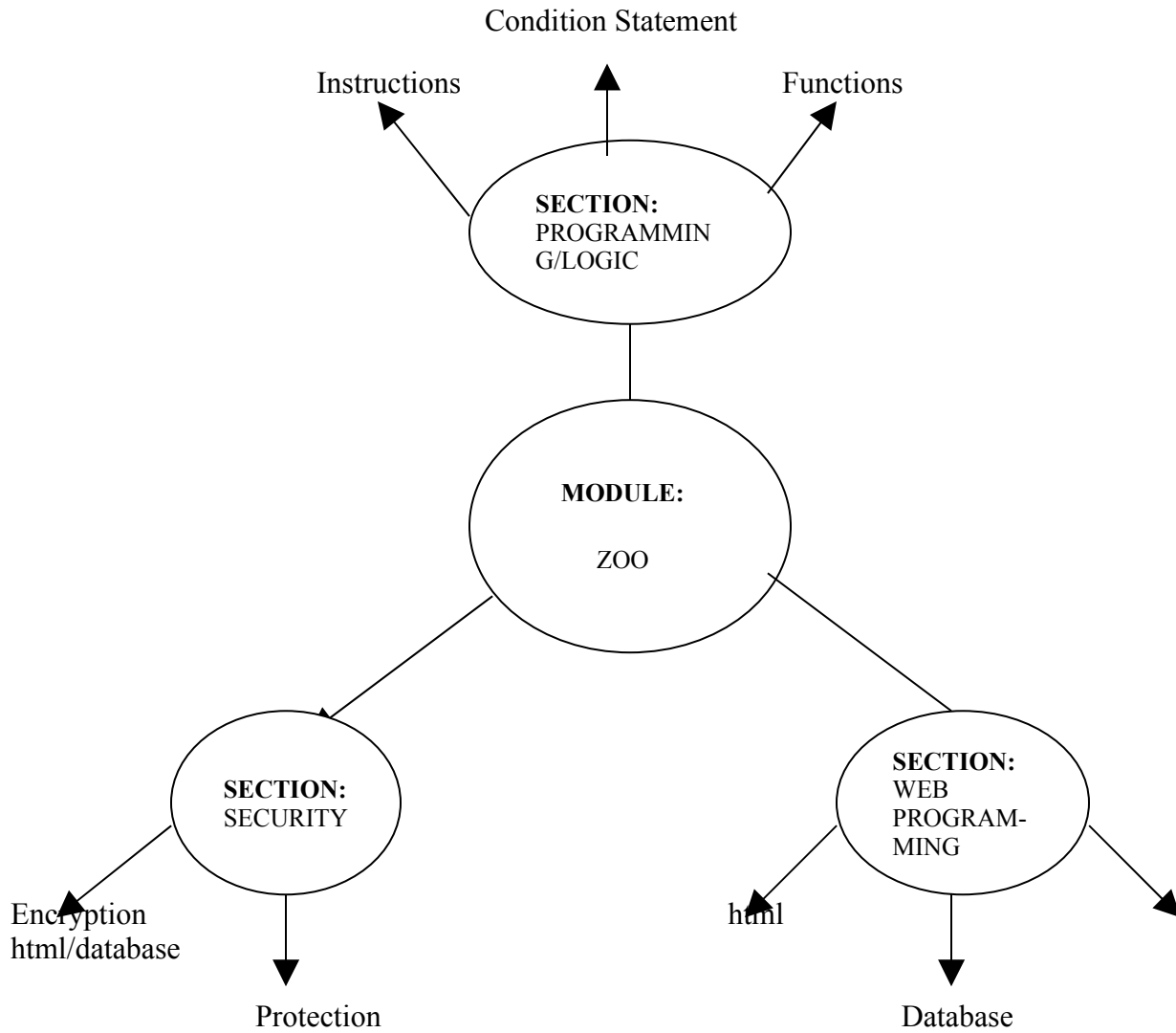


Figure 1. A flowchart showing the layout of the implementation of one of our modules, with its activities

Design Details

HTML/Database: HTML: Website for an animal —Figure 2

- The right side of the screen contains objects that can be clicked to change content.
- Instructions on left side explain how to input a title, text, and choose an image. A screen that reads “Animate image of some animal for decoration” will be displayed on the left.
 - Instructions are static text.
 - Animation (not here, but in the implementation) is a movie clip
 - Input Title and Input Text are both input text fields.
- Choose Image button brings up a new screen from a shape tween that displays 4 pictures. Each picture is a button that will set variables to remember what picture the user chose.
- The View Code button and a View Site button are first created. The View code button changes color when pressed and brings up a new screen from a shape tween that displays the HTML code that would be used to create a similar site. Lines such as body and font tags will be hard coded with static text, and dynamic text will take the input from the previous screen and insert it into the correct space. The image tag will have first part hard coded with remaining space being dynamic that decides which image was chosen and puts the correct file description into the space.
- The View Site button brings up a screen from a shape tween that displays what the site would look like if the code was implemented. Dynamic text will take the input from the previous screen and the image the user chose will be displayed at bottom of page.
- Items such as title, pictures, and text can be changed.

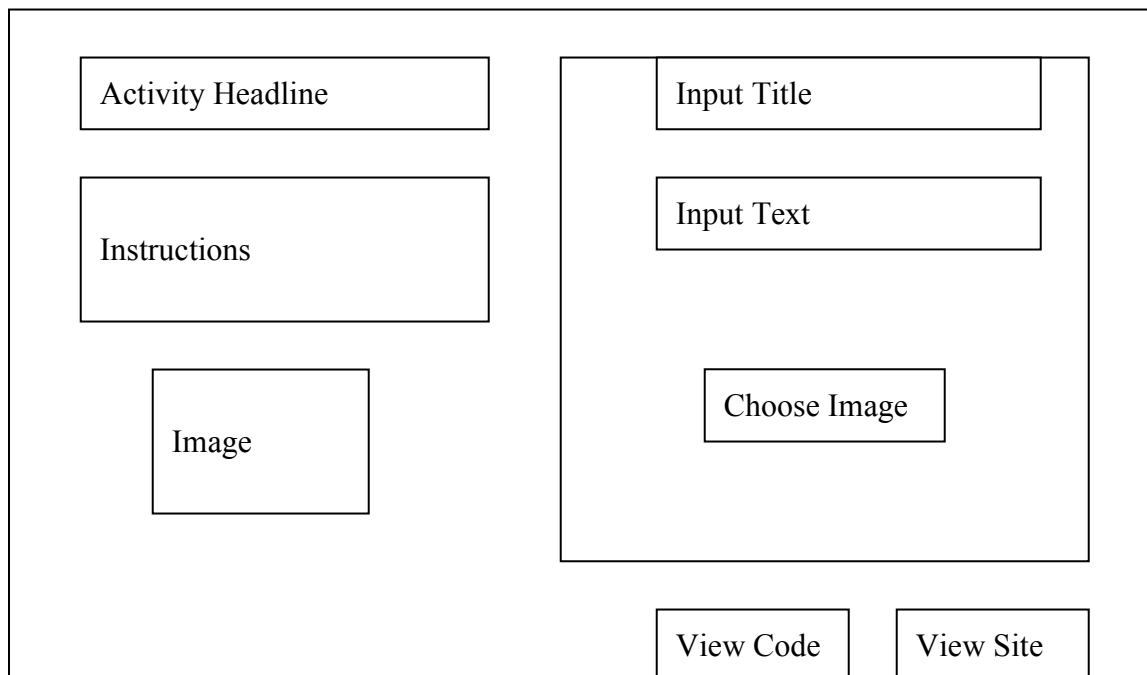


Figure 2: A visual display of Activity 1 under the HTML/Database section.

HTML/Database: Database: Buy Animals for Zoo —Figure 3

- Instructions explain that the user is supposed to select a habitat, and recognize how the query statement changes. Then, that they may buy animals and have only so much money.
- Drop box for region you want to buy from (Safari, Rain Forest, Arctic).
 - Drop box will be a movie clip that will roll out displaying several habitat selections that will be buttons. The buttons will then alter the text in the Query Statement and the list of possible animals to buy.
- A dynamic un-selectable text box that shows the query generated when the option is selected. EX: `SELECT * FROM * WHERE Habitat='Safari'`
- List of animals to buy is generated when a region is selected.
 - Each entry will be a button that will bring up a screen of information about the animal. Such information as picture, habitat, food preferences, and price. Box will also contain a Buy button.
- Budget is a dynamic display of amount of money available for user to use to purchase animals.
- Display of animals bought may possibly need to be scrollable if list gets to large.

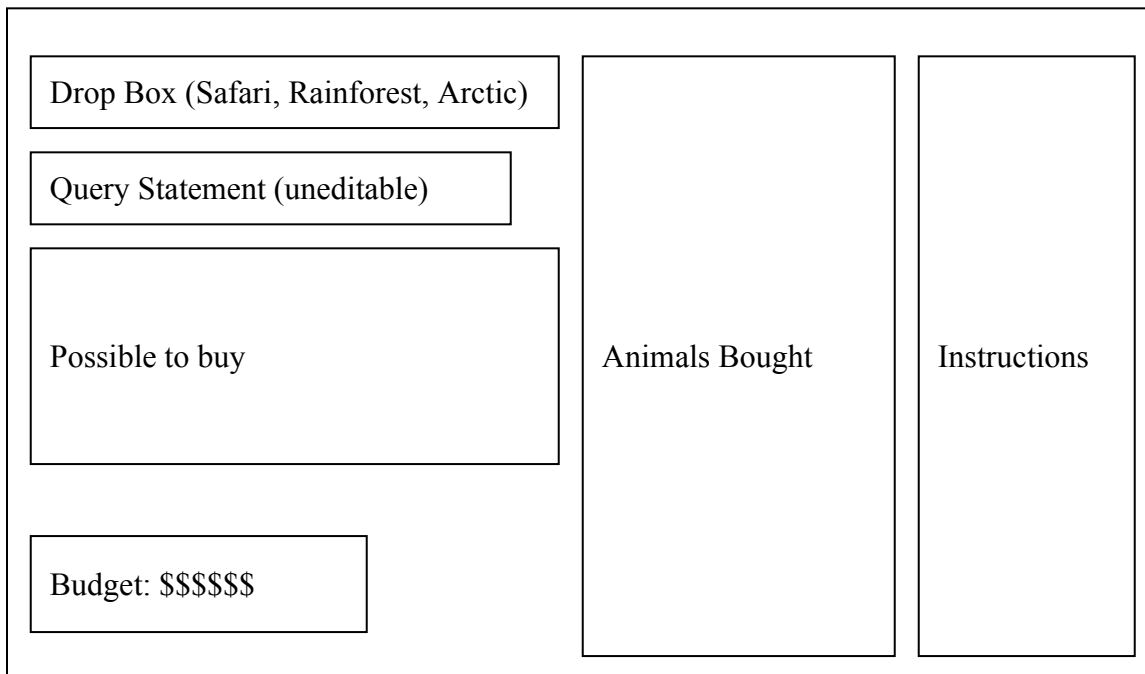


Figure 3: A visual display of Activity 2 under the HTML/Database section

HTML/Database: HTML/Database: Website for Zoo —Figure 4

- Creates a website for the Zoo.
- Right side has instructions.
- The left side has a miniature of website with clickable areas that can be altered. Clicking will bring up a screen over the left side (Referenced to the right side in below diagram).
 - New screen will have a combo box of possible query statements that user will decide which will apply best to the selected page.
 - Banner will bring up a screen for user to select from a few different banners.
 - Dynamic text will also be included so user can say something about the page, like “Check out our cool animals that live in the cold”
- When moved to area, must decide upon query, selected from a combo box, to decide which animals will be displayed there.
- The View Code button and a View Site button are first created. The View code button changes color when pressed and brings up a new screen from a shape tween that displays the HTML code that would be used to create a similar site. Lines such as body and font tags will be hard coded with static text and dynamic text will take the input from the previous screen and insert it into the correct space.
- The View Site button brings up a screen from a shape tween that displays what the site would look like if the code was implemented. Dynamic text will take the input from the previous screens

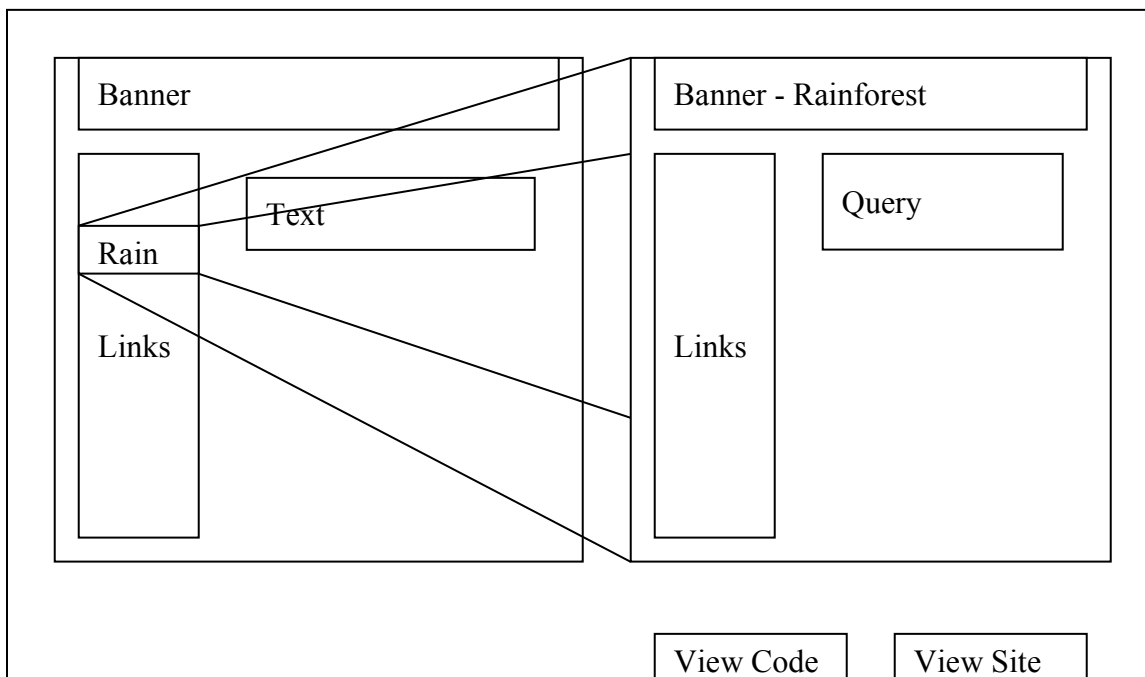


Figure 4: A visual display of Activity 3 under the HTML/Database section.

Security: Encryption: Animal message decoder —Figure 5

- When initiating the game, a random number between 1 and 26 is generated and is designated as the offset.
 - The offset is used to generate the code via a circular decoder. So if 5 is generated, the outside circle is turned 14×5 degrees ($360/26 = 14$) to show a correspondence between A and F ($A + 5$).
- Instructions explain that the message is encoded, and it is the task of the user to decode the message using the decoder.
- A random message is then selected from the list of messages and encoded using the offset. The encoded message is then displayed in the Encoded Message dynamic text field.
- Text Input is an input field that user types what they believe the decoded message is.
- A Show Answer button brings up a new display from a shape tween that displays the message correctly decoded in a static text field.
- Check Answer button checks text input and compares to the decoded message and displays a screen from a shape tween of either success or try again.

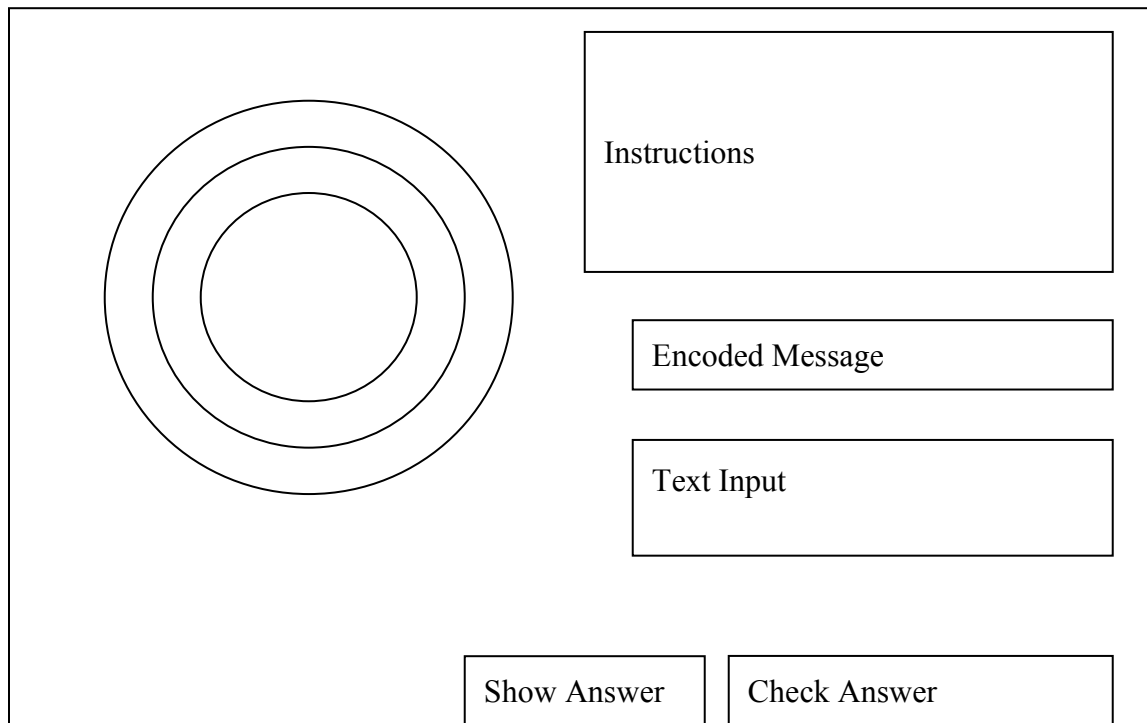


Figure 5: A visual display of Activity 1 under the Security section.

Security: Protection: Trapped in a Lion Cage —Figure 6

- The scenario is that you got trapped in the lion’s cage. Lion is sleeping because of a tranquilizer. You have to get out before the lion wakes up. You have a note that has the password to the lock, but it is encoded so you have to decipher it when given a key word. The reason it is encoded is so others can’t get into the cage if they found the code.
- Instructions (static text) explain scenario and task.
- After user enters possible code, they click on the door.
 - The door will be button that checks possible answer and if it is true, the door will open and the user gets out of the lion cage, otherwise it will stay locked.
 - Door will also clear any numbers already entered into the LED.
- Timer is dynamic and will change counting down to the time the lion will wake up from his tranquilizer induced sleep
- If timer runs out, lion will wake up and jump at the screen. Failure splash screen will then be displayed
- User will enter the numbers in the LED pad. General touch pad phone layout will be used.
 - LED will change based on the numbers entered, moving previously entered numbers to the left of the LED screen. Dynamic text.

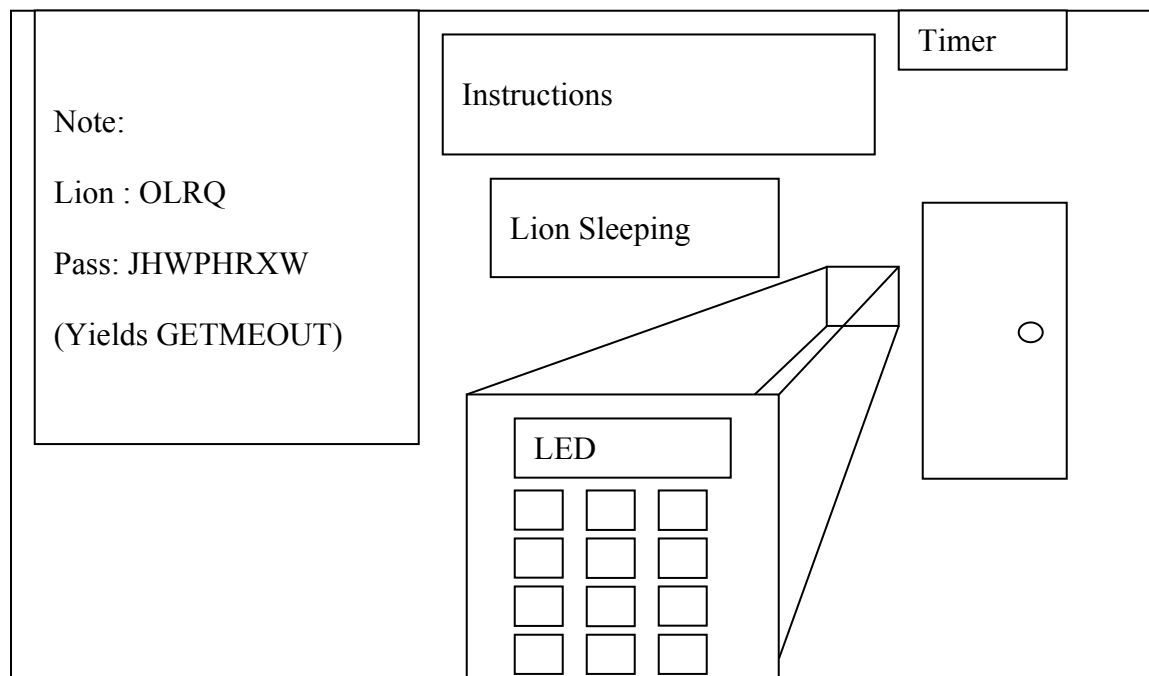


Figure 6: A visual display of Activity 2 under the Security section

Programming/Logic: Instructions: Create food for different types of animals —Figure 7

- Robot is to be instructed on how to make food for a certain animal.
- Instructions (static text) explain they are to make certain types of food and give it to the animals.
 - Animal picture will change based on what type of food you are making. Carnivorous -> Shark, Herbivorous -> Giraffe, Omnivorous -> Bear.
- Drop box of Types of food (Carnivorous Food, Herbivorous Food, etc.) will be displayed via a roll out from a movie clip.
- Possible instructions are used such as (add Chicken, add Grass, add Bananas, etc). Instructions will highlight after they have been selected so the user knows which one they chose.
- A Taste Test button is used to make robot assemble the food and give to animal to try. Robot will then move from left side of screen to right side of screen and give the food to the animal.
- There will be a visual display of the animal's reaction via a movie clip. Maybe hearts above head if they approve. Or they attack robot if they hate it (i.e. if robot does not interpret the instructions correctly).

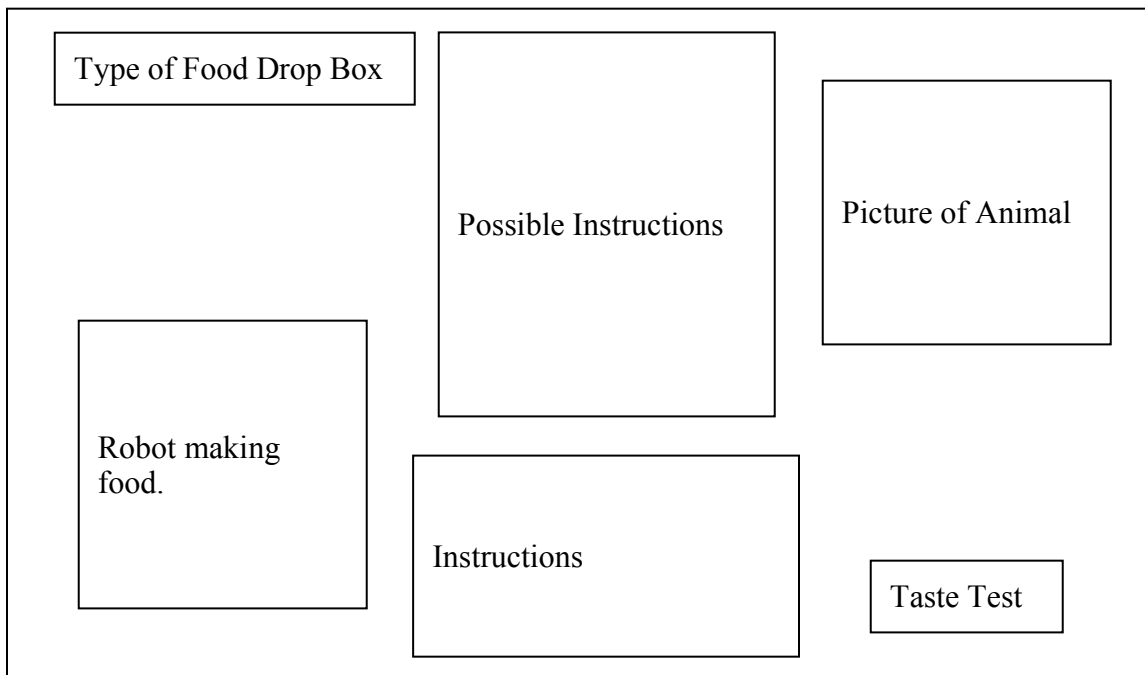


Figure 7: A visual display of Activity 1 under the Programming/Logic section.

Programming/Logic: Condition: Conditional Feeding —Figure 8

- Instructions (static text) explaining scenario and what to do.
- “If animal is a” (Herbivore, Carnivore, etc) combo box displays condition.
 - Combo box will be a movie clip that rolls out once it is clicked and allows for selection of possible animal types.
- Once an animal type is chosen, “Give it this food” (Chicken, Grass, Chicken/Grass, etc) combo box displays possible food to give it.
- A Feed Animals button has robot go and try to feed animal based on conditions set up. An animal will react to what food it is given.
- There will be a visual display of the animal’s reaction via a movie clip. Maybe hearts above head if they approve. Or they attack robot if they hate it.

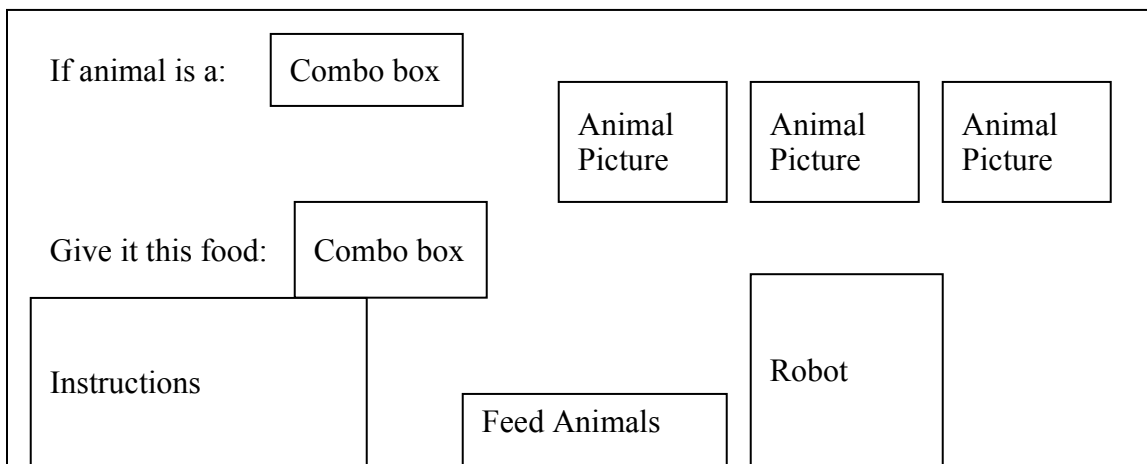


Figure 8: A visual display of Activity 2 under the Programming/Logic section.

Programming/Logic: Functions: Feed the Zoo animals —Figure 9

- Make robot go around feeding the Zoo animals on its own via specifications you have given it.
- Instructions (static text) explain scenario and what to do.
- “If animal is a” movie clip rolls out combo box, with selections such as Monkey, Shark, Lion, Bear, etc.
- “And it is” movie clip rolls out combo box, with selections such as night, day, evening, morning.
- “Call Robot Function:” movie clip rolls out combo box, with functions such as Feed Carnivore or Feed Herbivore.
- “Feed animals” button makes robot go around and feed animals. Four cycles will be used, one for morning, one for day, one for evening, and one for night. Robot will make 4 passes at the animals, and if all conditions are true, functions are called to feed the animal.
- There will be a visual display of the animal’s reaction via a movie clip. Maybe hearts above head if they approve. Or they attack robot if they hate it.

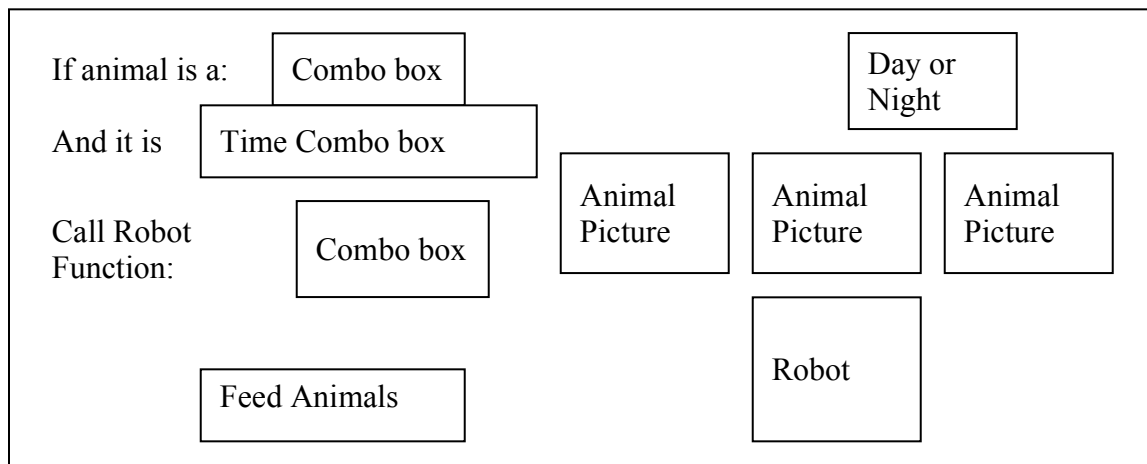


Figure 9: A visual display of Activity 3 under the Programming/logic section

Conclusion:

The method of implementation we chose for this project—flash—proved to be a bit of a challenge for us. The initial plan was to implement eight activities, but as we progressed with the project it became apparent that it would be infeasible to implement all eight of those activities. Therefore, we fully implemented one activity as well as created designs, block diagrams, and layouts of the remaining seven activities. It is most likely that another group in the near future will be asked to take this project further.

The group that proceeds with the project should be able to pick up where we left off fairly easily.

REFERENCES

[1] W3Schools Online Web Tutorials
<http://w3schools.com/>, accessed June, 2007.

[2] Flash Kit
<http://www.flashkit.com/tutorials>, accessed June, 2007