CULMINATING ASSIGNMENT

**PART A: ANALYSIS**

Read the ***ENTIRE*** project and decide how much time you plan to spend on each section. As you work on each section, indicate the amount of time actually used, and finally indicate if the section was completed. Be sure to take into account the value of each component when allocating time:

Create a brief report that includes a description of the target group for **EACH** program. (Who is likely to use your program?) Describe your ideas for the animation, game, and application you will be creating. Explain why you think these programs will appeal to the target group.

**Time Management Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **PART** | **TIME ALLOCATED** | **ACTUAL TIME USED** | **COMPLETED (✓)** |
| 1. Analysis    1. Time Management Table |  |  |  |
| 1. Analysis    1. Report |  |  |  |
| 1. Design and Development    1. Game Planning Sheet |  |  |  |
| 1. Implementation    1. Game/Interactive Graphics Based Program |  |  |  |
| 1. Testing - Creating Error Handling Code |  |  |  |
| 1. Testing - Creating Error Handling Code |  |  |  |
| 1. Creating Test Data |  |  |  |
| 1. Maintenance – Releasing Your Programs |  |  |  |
| 1. Maintenance - Responding To User Feedback |  |  |  |

**PART B: DESIGN AND DEVELOPMENT**

1. **Game Planning Sheet**

Read over the requirements for the game program below. Design a game that meets the requirements by using the game planning sheet. Complete and include your [game planning sheet](02%20Game%20Planning%20Sheet.docx) in your culminating report in your culminating report.

**PART C: IMPLEMENTATION – CREATING YOUR PROGRAMS**

1. **Game/Interactive Graphics Based Program**

You will create a game or interactive graphics-based program. When selecting your idea, make sure that it is something that will appeal to your target audience.

Below are some ideas for your programs. You are ***not limited*** to these concepts; let your imagination run wild.

**Game Program Ideas:**

1. **Matching Game**

Create a game items are matched together. For example, an Animal Feeder program with MovieClips of a Monkey, Tiger, Giraffe, and three food items such as bananas, raw meat, and grass. The user will have identify the food to the correct animal to match them.

1. **‘Click To Destroy’**

The user has to click on a item to destroy it. You might create a “shooting arcade” where the user has to click on targets moving back and forth across the screen.

1. **Fishing Game**

The user raises or lowers a lure by clicking on buttons on the stage. As fish move back and forth across the stage, the program checks to see if there is a collision between the lure and the fish. If so, the fish is caught.

1. **Break Out**

Using a paddle, control the bouncing ball and break all the blocks.

1. **Space invaders**

A classic game …

1. **Hangman**

Guess the secret word …

1. **Shooting game**

Control a character using the keyboard and fire at a target. Keep a score.

**CHECKLIST**

Whatever idea you select, the program must contain the following elements:

**Include ALL of the following:**

|  |  |
| --- | --- |
| Proper naming conventions |  |
| Comments are included in the program on important lines of code |  |
| Semi-colons are used at end of every line requiring them |  |
| Final program matches the one proposed in the planning sheet |  |
| Buttons |  |
| Program includes User Input and Output |  |
| If- Selection Structure |  |
| For or While Loop - Repetition Structure |  |

**Include FOUR (4) of the following:**

|  |  |
| --- | --- |
| Functions that are passed data |  |
| Random numbers |  |
| Moving objects |  |
| Collision Detection |  |
| Timer |  |
| Adding/Removing objects from a screen |  |
| Counters |  |
| Accumulator |  |
| Changing object properties (x,y co-ordinates, colour, size, etc.) |  |

**PART D: TESTING - CREATING ERROR HANDLING CODE**

1. ***ALL*** the input components of your application should have error-handling if-statements to detect improper input.
2. Add code to the program that will detect ONE of the following user input errors.

Errors ***might*** include:

* Blank TextField
* Out-of-bounds input
* Wrong data type input
* Input that has too many or too few characters (eg. Password)
* Input that is missing a required character (eg. @ for email address)

**PART E: CREATING TEST DATA**

Create a set of test data for ***ONE*** application.

Your test data should include data that checks:

* Proper Data
* Out of Bounds Data
* Illegal Data Type Data
* Missing Data

**NOTE:** If your test data reveals any errors, make corrections to your program and resubmit it to your teacher.

In your culminating report, identify your test data and the reason you chose these data.

**PART F: MAINTENANCE – RELEASING YOUR PROGRAMS**

Tell your teacher when you have finished ***EACH*** program.

**PART G: MAINTENANCE - RESPONDING TO USER FEEDBACK**

Ask ***at least TWO*** students to try your program. Ask these students how you can improve your program. In your culminating report identify these changes recommended by the students.

Select ***ONE*** recommended improvement made by a student. Change your program to include the suggestion. Save the update under a new name. Tell your teacher the new file name. They will post the updated version of your program.