Computer Programming Project

Your final task for this course is to design and implement a project to be presented at the Design Show in May. Your project should fall under one of the following categories:

1. Games
2. Animations
3. Data Processing
4. Physical Computing (Arduino)
5. Other – your own ideas are welcome, talk to Mr. Alei

You may implement your project in any computer language you prefer, though I strongly suggest using JS Processing or Arduino, depending on the type of project you want to develop.

The project will have three phases that you will document with details to follow:

1. **Design**
2. **Implementation & Testing**
3. **Presentation**
4. **Reflection**

1. Design

a. Write a paragraph or two describing what your project is and generally what it will do.
b. Create a detailed description of the flow of the project, including the different screens, user inputs, and program outputs.
    i. You may use a variety of tools for this task but should include a flow chart. The goal is to communicate clearly and in detail precisely what your project will do (but not writing any code!).
    ii. You may need to write some simple code tests to see if you can implement certain ideas.
    iii. Note that this part will likely take ***more*** time than writing the code.
    iv. You will turn this in before writing any code and be assessed on how closely your code matches what you turn in.

2. Implementation & Testing

a. Write the code to implement your design. Break the code down into functional pieces, testing them as they are written. Keep a list of the functions that you write and document all your code clearly.
b. You should keep track of your work in a word processing file referred to as your "Project Notes" (like a "Process Journal"). Your notes should be updated regularly (minimum weekly) and should::
    i. Include detailed, dated notes documenting your progress.
    ii. Identify any changes that you make to the original design and why you made them.
    iii. Detail the functions that you write, their calling sequence, their return values and what they do.
    iv. Include links or references to other supporting files that you developed or used (flow charts, design specifications, sources, resources)
c. Your Project Notes should be kept in a Word or Google Docs document. It will be assessed as part of your project.

3. Presentation

a. You will be expected to present your project to parents and students at the Design Show on Wed, May 2nd.

4. Reflection

a. After the Design Show, you will make a final entry in your Project Workbook reflecting on the process and your result. The reflection should be thoughtful and detailed and address the following questions:
    i. Do you consider your project a success? Why or why not? How would you improve the design? What features would you add or remove?
    ii. How much did the project change from the original design? Why?
    iii. What was the most challenging part of the project to implement?
    iv. What surprised you about the whole process?
    v. What will you do differently on your next project?

Computer Programming Project Assessment Rubric

Name: **Score: _____/20**

Project Title:

Design	**4 marks**
1	• Creativity: Design shows little or no creative effort • Usability: Design does not consider ease of use or program flow • Documentation: Design specification is missing or incomplete, lacking detail
2	• Creativity: Design shows some creative effort • Usability: Design is somewhat easy to use, flow is somewhat clear and logical • Documentation: Design specification is partly incomplete, includes some detail
3	• Creativity: Design shows significant creative effort • Usability: Design is mostly easy to use, flow is mostly clear and logical • Documentation: Design specification is mostly complete, includes good detail
4	• Creativity: Design shows extensive creative effort • Usability: Design is very easy to use, flow very clear and logical • Documentation: Design specification is complete and concise, includes excellent detail
Implementation & Testing	**8 marks**
1-2	• Function: The project has numerous bugs or errors • Consistent with design: The project is not aligned with the design • Style: The code has little or no commenting, is not properly indented, choice of variable and function names hinders readability • Organization: The code is disorganized with little or no use of functions, loops, and branches to improve efficiency.
3-4	• Function: The project has some bugs or errors • Consistent with design: The project is fairly well aligned with the design • Style: The code has some commenting, is mostly properly indented, choice of variable and function names partly helps readability • Organization: The code makes some use of functions, loops and branching to improve efficiency
5-6	• Function: The project has a few bugs or errors • Consistent with design: The project is mostly aligned with the design • Style: The code has good commenting, is properly indented, choice of variable and function names helps readability • Organization: The code makes good use of functions, loops and branching to improve efficiency
7-8	• Function: The project has no evident bugs or errors • Consistent with design: The project is completely aligned with the design • Style: The code has thorough commenting, is properly indented, choice of variable and function names significantly helps readability • Organization: The code makes excellent use of functions, loops and branching to improve efficiency
Presentation	**3 marks**
1	• Presentation: The student is not prepared for or does not attend the presentation • Documentation: Project Notes are missing or incomplete, unreadable, or very disorganized
2	• Presentation: The student attends and is mostly prepared for the presentation • Documentation: Project Notes are mostly complete, readable, and well organized.
3	• Presentation: The student attends and is well prepared for the presentation • Documentation: Project Notes are complete, readable, and very well organized.
Reflection	**3 marks**
1	• Reflection is missing or very limited. Few or no suggestions for improvement.
2	• Reflection is somewhat meaningful and shows some thought. Some suggestions for improvements are given, but limited. Some challenges and solutions to them are mentioned but need elaboration.
3	• Reflection is meaningful and thoughtful. Suggestions for improvements are evident, challenges and solutions to them are described clearly.
Holistic	**2 marks**
1	• Overall project was weak, simplistic or not well executed
2	• Overall project was strong, creative and well executed

Project Ideas and Hints

Some project ideas

1) **Games**

Pong	Tetris	Asteroids	Tic-Tac-Toe	Tag
Dodger	Worm	Connect Four	Mancala	BlackJack

2) **Animations**

Let your imagination go. Consider creating a ***really*** cool fish tank. A 3D shape that grows out of nothing. Create an animated scene of some sort. Construct an animation of a geometric proof like the Pythagorean Theorem. Animate the construction of various geometric ideas using a virtual straightedge and compass. Construct a fractal like the Sierpinski Triangle or a fern.

3) **Data Processing**

A pig latin translator
Sort a series of numbers, using and comparing different sort algorithms

4) **Physical Computing (Arduino)**

Run a stepper motor fan
Write a user defined scrolling ticker on an LCD display
Monitor the temperature of a cup of tea as it cools, collect the data in an Excel file
Connect or disconnect the school's intercom speakers with the push of a button
Set up a camera and Arduino to take a picture of every person who enters the Red Lab

5) **Other** – If you don't like what you see or ar having trouble coming up with an idea, see Mr. Alei

I suggest you do your work on Khan Academy using their "Create a Program" tool at https://www.khanacademy.org/computer-programming/new/pjs. There you can name and save your program under your user account.

There are ProcessingJS functions that are not implemented by the Khan Academy IDE. So, for more flexibility but a somewhat less "user friendly" platform, try Sketchpad or another Proecssing.JS IDE of your choice.

Some hints for a successful project

1. Be creative, but don't bite off more than you can chew. I can help you with this – please ask.
2. Get the design right ***before you start coding.*** The more detail your design documents include, the easier and more effective your coding will be. Know your destination before you start your journey.
3. Document ***as you go***, keeping your Project Notes file up to date every time you make significant progress.
4. Use loops, functions and conditionals to make your code efficient and "***elegant***".
5. Fully comment your code. Use proper indentation. Use meaningful names for variables and functions.
6. Organize your code into functional parts. Follow your design spec and flow chart. If you need to make changes to the design, document them clearly in your Project Notes file.

Design Specification Outline

When you write a significant piece of computer code (program), you will inevitably go through a cycle of steps that is represented by the MYP design cycle. A diagram of the process you will use is shown below. You start by **investigating** ideas, then **design** the flow of your program and what it will do, which also involves **planning** the user's experience – what they will see, what responses the program will give to various inputs, etc. The design and planning phases of a project are typically the steps that take the most time. Doing them well allows you to efficiently create the program, during and after which you will **evaluate** its performance against your design. Inevitably, some things won't go as expected and you will need to investigate why, design and plan a correction, create a solution to the problem, evaluate again and continue the cycle.

A Design Specification is:

> *A detailed description of the conditions, requirements and restrictions with which a design must comply. This is a precise and accurate list of facts, such as conditions, dimensions, materials, process and methods, that are important for the designer and for the user. All appropriate solutions will need to comply with the design specification.*

For a software project such as this one, the specific components of a Design Specification include, at a minimum:

- Goals – brief description (you've already written this, tighten it up to be very specific
  - Specific as possible
- User Interface – what will the user see?
  - Screen shots, etc. that the user sees
- Functionality – how will it work?
  - What happens when the user interacts with the program?
  - Flow chart
- Milestones – what intermediate steps will you take as you develop the program?
  - How you will break the code into parts
  - What will you create first?
  - How will you test the parts of your code?
  - How will you test the overall program?
  - Target dates

You can read a lot more about these ideas at https://www.toptal.com/freelance/why-design-documents-matter