# Step 1: Proposal

## Overview:

Provide a description of what product(s) you intend to create in this project

## Objective:

Provide a description of what skills and understandings you intend to gain in completing this project:

## Connections:

Describe how this project connects with two other CTS courses that you have taken:

## Resources:

Provide a list of resources, outside of what is available in class, that you will require to complete this project

## Risks:

Identify any risks to health and/or safety, and how you intend to mitigate these:

## Indicators for Success:

How will you know that you have achieved your objectives? Provide a rubric which describes what a successful project looks like.

1. On the left hand side, list the various components (i.e. parts) of your product (e.g. if you are writing a game, list the GUI, Game Play, Scoring, Sounds, etc). On the right hand side, describe what these elements *should* look like for this product to receive a Mastered.
2. Then, also consider the process. What does good work look like? What would you look for in a teammate?

|  |  |
| --- | --- |
| **Component** | **Mastered (5/5)** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Code Management** |  |
| **Time Management** |  |
| **Problem Solving** |  |

## Approval

At this point, submit this document to your teacher to obtain feedback and approval on your proposal.

# **Planning**

Gant Chart

Break your project into distinct phases. Then, provide estimates for how long each phase will take. You have already completed the first phase, which is to create a proposal. You are currently on the second phase, which is planning.

Design

Consider your IPO.

* What are the inputs into your program?
* What processing needs to be done? i.e. which algorithms do you need?
* What are the Outputs

For graphical projects, consider what your project would *look* like. How does it interact with the user? Provide a series of mock-ups which clearly show the flow of execution. These can be either drawn or designed using WindowBuilder.

For console-based projects, provide a “script” for how the program would accept input and provide output. These can be text-based, but should be thorough. Consider cases (and show examples of) where the user provides improper input. Also, provide an explanation / flowchart / pseudocode of the algorithm(s) that you are using to process the input.

## Journal

Provide a journal for the time that you worked on this project. You do not need to provide an entry for each period or session that you worked on the project, but your entries should cover all time spent. This is not meant to be a simple accounting of your time. Rather, this is a living document that records the process of building your projects and demonstrates your learning. Entries can contain your thoughts on

* Work completed
* Victories achieved
  + getting something to work
  + understanding a concept
* Challenges encountered
* Resources used
* Meta-cognition
  + What seems to be working well

## Reflection Questions

These are to be completed along with your final submission of your code.

1. Provide an updated Gantt chart

2. Refer to your original specification. How does the planned duration of each task compare with the actual time it took to complete?

3. Did you need to deviate from your original specifications? If so, describe which and why.

4. What would you change about your process (i.e. how you approached the creation of your product)?

5. What worked well in your process?

6. What part of the finished product are you most proud of, and why?

7. If you were to create a version 2.0 of your product, what part of the product would develop further?

8. Are there any specific bugs or functional shortcomings in your product? Be honest – being aware of these and recording them is a good thing!

9. Which specific skills and understandings did you gain while working on this project?