

MYP 5: DIGITAL DESIGN

UNIT 1: INTRO TO PROGRAMMING

ASSESSMENT OVERVIEW

INQUIRY

| | |
|-----------------------------|---|
| Key concept | Development |
| Related concepts | Function, Innovation |
| Global context | Personal & cultural expression |
| Statement of inquiry | Achieving function through computer programming is a developmental process that requires innovative methods to achieve new solutions to problems. |

Welcome to your assessment overview for our unit.

By now you should be quite used to the Design Cycle process. We will apply this in a computer programming context. In brief:

- You will find a client who has "a need or problem" that they think can be solved through software. You can use a friend or family member to act as your client. This will form the basis of your research for Criterion A investigation phase.
- You may **optionally** buddy up with a partner for the remainder of your design cycle. The two of you will ideally select one of your Criterion A problems to create a solution for.
- Criterion B will see you create success criteria, annotated screen prototypes, justify your design decisions, and document your program algorithm through a flowchart.
- You will then plan and create your product in Criterion C, culminating with DEMO DAY where you present your product to the class and invited guests.
- Finally you will evaluate and reflect upon your success, and obtain feedback from your client (Criterion D).

Some important points regarding method of submission:

- Submission of written work will be through Google Docs that the teacher will share via Google Classroom. You will receive one for your individual Criterion A investigation, and a separate one for your paired Criteria B, C & D portfolio.
- Due dates will be by Criterion and will be announced through Google Classroom and on Managebac.
- Criterion C will also require the upload of your product to Google Classroom and the creation of a short video in which you demonstrate the achievement of each of your success criteria. The video will be submitted via Google Classroom and can either be an MP4 file or YouTube link.

CRITERION A: INVESTIGATION

Now you have a problem to solve, it's time to do some research into it so your solution can be as meaningful as possible.

The bulk of your research should be the programming skills exercises we have conducted in class. The only additional research would be to briefly look at some existing products. This isn't intended to be much new work, mostly it should be a documenting of work already completed.

You should:

- Prepare a table based on the template following, outlining the questions you need answered in order to be able to produce your solution/product.
- Each programming skills lesson should be converted to a question that you can answer and listed under the category of "understanding the technology". Examples:
 - How to output to screen
 - How to read input from the user
 - How to perform calculations
 - How to selectively execute code based on different conditions (ie: if statement)
 - How to repetitively execute code based on different conditions (ie: for loop, while loop)
 - How to draw graphically on the screen (ie: pygame)
- Identify at least 3 question(s) you wish answered from similar existing software products. These could be about the design of the user interface, or it's functionality. List these with the category of "understanding similar products".
- For your programming skills questions, in "what I learnt / found out" you may refer to your programming notes or repl.it exercises as proof.

| Number | Category | Question | Priority | References | What I learnt / found out | How this will help me design or create my product |
|------------------|---|---|--|---|---|---|
| 1, 2, 3... | <i>understanding the problem, understanding the customer, understanding the technology, understanding similar products.</i> | <i>To produce my game I need to learn/find out about...</i> | <i>critical, important, useful, optional</i> | <i>For websites etc, include the relevance reference. For teacher provided in-class exercises, state the relevant lesson / exercise. For repl.it exercises, provide the exercise number</i> | <i>Not just "I learnt how IF statements work"! Detailed. Technical. Provide illustrations, references or an example. For instance explain the role of IF statements, the different forms it can take, the different queries you can ask of it etc</i> | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

CRITERION B: DEVELOP IDEAS

TASK 5: DESIGN SPECIFICATION

For unit 1, students will be given the option of working in pairs for criterion B, C and D. You may select the problem identified by one of you in Criterion A, or even select a completely new problem.

Task 5: Prepare a list of success criteria that will satisfy the problem they are creating a solution for. Success criteria should be written across the three categories of function, technical and aesthetics. Each success criteria should be specific / measurable and contain a statement of rationale to justify its inclusion. ~~The student should obtain feedback from their client before finalizing their success criteria.~~

The idea behind your success criteria is it provides a checklist that can be used at the end of the project to definitely declare whether you have achieved what you set out to achieve. It will form the basis of your Criterion D assessment.

Categories:

- Function: What will your program do?
- Technical: What tools will your program use?
- Aesthetics: What will your program look like?

Success criteria:

- Should be objectively measurable: Use specific terminology instead of vague or subjective language. In other words, it should be phrased so that different people will still interpret the statement the same way. Using language that can be measured is a key way of achieving this.
- Instead of "will load fast", use "will load within 2 seconds". Another common one to avoid is "will be user friendly" as this means different things to different people, instead make points that itemize how it will be user friendly (will exit when the escape button is pressed, will display instructions at the start showing what keys do what, etc)

Rationale:

- Why have you included this criterion? Ideally this should, where possible, link back to your relevant criterion A discussions on the nature of the problem and the things you learnt in research.

You should:

- Produce a table similar to the template provided, outlining the success criteria for your project.

| Number | Category | Success criteria | Rationale |
|--------|------------------|------------------|--|
| 1, | <i>Function</i> | | <i>Try to keep to a maximum of 50 to 70 words per response. If some are more detailed that's fine but this shouldn't contain essays.</i> |
| 2, | <i>Technical</i> | | |
| 3... | <i>Aesthetic</i> | | |
| | | | |

TASK 6: DESIGN IDEAS

Task 6: Prepare a series of screen mockups to illustrate the proposed computer program. Each mockup should include annotations detailing the function, technical and aesthetic qualities in the design.

You should:

- Produce a set of at least 2 to 3 quality, annotated diagrams of screen-mockups for your program.
- Your annotations should describe both functional and aesthetic qualities
 - eg: what will happen when a button is pressed, when a key is pressed, what images will appear, what movement/animations will occur
- If you don't use a ruler to make your diagrams neat and of a high standard, then don't bother submitting it.
- Scan and insert your diagrams into the relevant portion of the document.
- You can produce several diagrams illustrating different phases of the same program, or diagrams of alternative layouts/styles of creating your program, or a mix of both.

TASK 7: DESIGN JUSTIFICATION

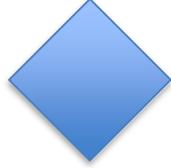
Task 7: Compare each of the screen mockups against your success criteria to determine which design best satisfies the clients requirements and why.

Expectations: A written statement or series of dot points (one per success criterion?) maximum 400 words.

TASK 8: DESIGN DIAGRAMS

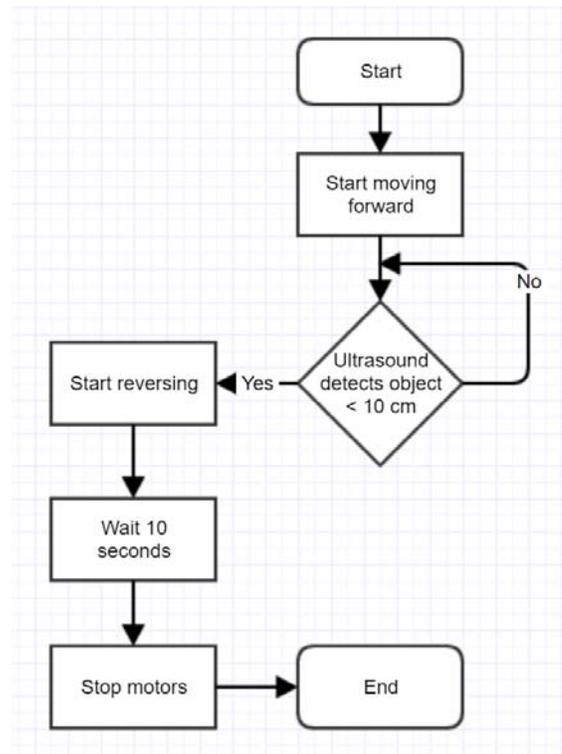
Task 8: Design an algorithm by way of a flowchart that illustrates the logic required to solve the problem. The flowchart should be sufficiently detailed for another programmer to be able to follow, and adhere to the norms of a flowchart diagram.

Flowchart norms:

| Symbol | | Notes |
|---------------|--|---|
| Program start |  | No arrows in, one arrow out. This is where everything begins. |
| Program end |  | One or more arrows in, no arrows out. All pathways terminate here. The program is finished. |
| Task |  | One arrow in, one arrow out. A statement inside the rectangle of what the program will do. |
| Decision |  | One arrow in, multiple arrows out. A measurable question inside the diamond Each outbound arrow should be labeled with the answers that correspond to that pathway. |

You can create your flowchart using either:

- Gliffy.com – online diagram drawing tool
- Draw.io – another online diagram drawing tool
- Paper and pencil



CRITERION C: CREATE THE SOLUTION

TASK 9: PLAN

Task 9: Students will identify the key tasks required for them to create their product and estimate the time required for each task. Students will then plot these tasks and their time requirements, against the time allotted by the teacher (plus whatever at-home time they consider necessary). This may be by way of a GANTT chart or a simple table. Their planning in this task should also identify resources they will require to create the solution (laptop, web server, Apple app developer account, robotics components etc).

Eg:

| Number | Category | Task outline | Assigned to | Resources required | Time estimate | Time taken |
|--------|------------------|--------------|-------------|--------------------|---------------|------------|
| 1, | <i>Function</i> | | | | | |
| 2, | <i>Technical</i> | | | | | |
| 3... | <i>Aesthetic</i> | | | | | |
| | | | | | | |

TASK 10: DEMONSTRATE TECHNICAL SKILLS

Task 10: Students will use their technical expertise to create the solution.

This will be primarily assessed through teacher observation including student questioning, and partner feedback

TASK 11: PRESENT FUNCTIONALITY

Task 11: Students will present the functionality of their solution through completing the Criterion C Skills Self-Assessment checklist. .

TASK 12: JUSTIFY CHANGES

Task 12: Students will reflect on changes made during the design and creation process, explaining the rationale behind such changes.

As you go about the program creation process, it is expected you will make changes to your originally intended design. This is fine and perfectly normal. You do, however, need to explain/justify your changes. That justification happens in this section. For any variation from your original screen-mockups or flowcharts, provide an explanation justifying the change.

CRITERION D: EVALUATING

TIME ALLOTTED: ONE WEEK OF CLASS TIME

Students will create a narrated screen-cast video recording to demonstrate and evaluate the functionality of their final product.

Specifically the video should:

- Introduce your final product and its overall purpose (maximum 30 seconds)
- For each individual design specification success criteria (from Task 5 or Criterion B strand 1)... (about 30 seconds each?)
 - Introduce the success criterion
 - Demonstrate that the criterion has/has not been achieved on the video
 - Discuss any pertinent challenges faced in meeting that criterion
 - Discuss any possible improvements you could make to the product in relation to that criterion
- Conclude by discussing: (about 60 to 90 seconds)
 - Who your client / target audience for the product was (age/gender/income/culture/language/education...)
 - The overall usefulness/value of your final product to your client / target audience.
 - Either a quote or two from representatives of your client / target audience expressing their thoughts on your final product (either have them speak in the video or you read their quote to the video)
- Use a combination of voice over narration, text headings on screen, and screen recording video.

This is being assessed holistically against the entirety of Criterion D which consists of:

- Creating tests
- Conducting tests
- Discussing improvements
- The impact of the product on the proposed client/audience

Please refer to the marking scheme for Criterion D for more information.

DIFFERENTIATION

All students

- Self selection of the "learning challenges" to those that are a best fit for their group members.
- Working in pairs or small teams allows individual strengths and weaknesses to counter-balance each other

Learning support or EAL

Students with identified as having **English as an additional language** or **learning support needs** may qualify for task modification on a case-by-case basis. Provisions available include:

- Quality (Adapting the number of items that the learner is expected to learn or number of activities student will complete prior to assessment for mastery):
- Time (Adapt the time allotted and allowed for learning, task completion, or testing):
- Level of support (Increase the amount of personal assistance to keep the student on task or to reinforce or prompt use of specific skills. Enhance adult-student relationship; use physical space and environmental structure):
- Input (Adapt the way instruction is delivered to the learner)
- Difficulty (Adapt the skill level, problem type, or the rules on how the learner may approach the work):
- Output (Adapt how the student can respond to instruction):
- Participation (Adapt the extent to which a learner is actively involved in the task):
- Alternate Goals (Adapt the goals or outcome expectations while using the same materials. When routinely utilized, this is only for students with moderate to severe disabilities):

In order to qualify for this provision, students must registered with either the EAL coordinator or LS coordinator.

Learning extension

It is important that all students are suitably and rigorously challenged as part of the learning process.

Students who already have **demonstrably significant** prior experience in computer programming may request a modified assessment that takes this experience into account. The modified assessments will consist of concept relevant, in-depth tasks to challenge those students where they are at, to further develop their understanding of programming.

In order to qualify for this provision, students will need to present a portfolio of past programming work to the Computer Science teacher for evaluation and verification of purported skill levels.