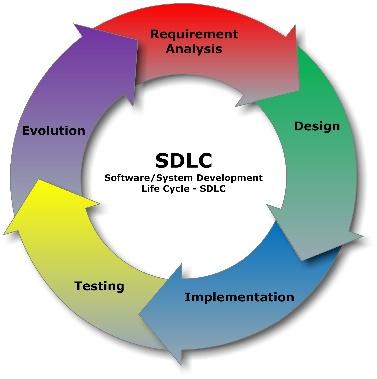
All graphics used are common license and have been pulled through MS Word or Google Doc’s common license images.



You only have 12 days in which to complete your final project in this course, **yes just 12 days!**

Some reading and research may be required outside class, depending upon your concept and the course work you have completed. All work **must** be your own original material OR **properly documented** when borrowed. You may borrow from all class notes, stating source, classmates code, stating source or other approved resources.

If you are unsure if you are allowed to borrow something, **ask, then log, source properly and date. Failure to properly source is plagiarism and earns an automatic zero for that piece of the project.**

This final project will give you an opportunity to demonstrate an understanding of the programming concepts learned throughout the course. You will be expected to follow the stages of software development (group and individual):

1. planning

2. design

3. coding

4. debugging

5. documentation

6. final testing

7. delivery

Note: These steps may worked on simultaneously in some cases, although this is the due date sequence in most cases.

Assessment of this programming project takes into consideration both the process and the product at both group level and individual level since Project Management is part of the Curriculum Expectations.

**Each step must be handed in on time**. A zero can be assigned on a section of the Final Task.

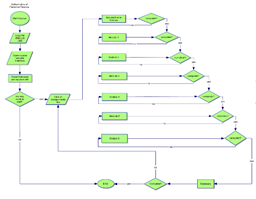


A Google Classroom Assignment Folder will be created for **each** major phase, but you will be working in Created folders in GitHub Classroom in your assigned teams. Teams will be created Monday January 7th to facilitate Team shares and all coding. Please ensure you hand your project pieces into the **correct folder or Team (Github Classroom)**. **Failure to hand into the correct folder will be considered NOT HANDED IN and not marked. You do not have time to be finding and resubmitting delayed work as it will impact your groups/games. Commit and push frequently. All Programming/coding/graphics must be completed at school. No exceptions on a culminating task.**

There will be both group component marks (very minimal, peer and collaborative pieces) and individual marks on this project (80%). Your contribution to the group will be assessed (20%), but the majority of the marks will be assigned to your tasks.

Your tasks will be determined by the Project job you will be completing as well as your part in the game you will be developing. Your preferences for game type will be taken into consideration and matched as accurately as possible as will the task within your group.

**#1 – Proposal and Problem Analysis – Step 1 Due: Tues. Jan 8/19 - Wed. Jan 9/19**



Your group mark and possibly your independent mark, will be based on this phase of the project.

Specific coding may not commence until this stage has been completed. The information and appropriate documents must be handed in.

The information required is;

1. Game name
2. Problem Definition including;
   1. Description of program
   2. Checklist of everything the program/game will do
   3. Rules for the program/game (must be unique, cannot be a game that already exists)
   4. List of screens
   5. List of tasks & people doing jobs as assigned
3. Problem Analysis and required tools needed – total 3 (2 required and 1 of 4 choices).
   1. Flow charts for main, each class, each function etc. (required)
   2. Gantt chart (required)
   3. Pseudo code (choice of 4) – check list will help with this
   4. IPO chart (choice of 4
   5. Checklists (choice of 4)
   6. UML/trace tables or other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (choice of 4) - must be approved.

# **#2 – Coding – Step 2 Dates: From step 1 completion (if # 1 complete, can begin Tuesday Jan. 8/19) See Gantt chart for your Game.**

# **Absolute Due Date: Tues. Jan 21/19**

# **Shared Work Due Date: Tues. Jan 15/19**

Two main pieces to keep in mind;

1. Classes that are shared class wide (like scoring) - Tues. Jan 15/19
2. Your individual work, the code, your main procedure and your group game.

As you know, from other projects, this can be long and involved, depending upon your task(s). It can also be frustrating.

You CANNOT leave the group tasks to the last minute because your games will not work without the group tasks and proper testing and consultation. Your games must work, they cannot crash. (Your group and possibly you, dependent upon your job, will be responsible for this phase of the project.)

Hints: Keep the problem and testing simple (KISS Principal), use versioning, do not continue until you have each small piece working, fitting together properly. Use your log to record what you want to do the next day, keep focused. Get help immediately from your group. Simple working code will earn a better mark than elaborate code that fails under certain conditions. (refer to Rubrics and job descriptions.) Ensure that your code follows all normal conventions and works for all game types where common elements are being used in classes. Code that does not work is before a Level 1.

**Communication Marks**: Your code MUST be commented appropriately including all sources including your classmates. Failure to document properly will earn a zero on this section and possibly other sections too – this is plagiarism if not documented appropriately.

# **#3 – Testing and Debugging Dates: From step 2 initiation and completion**

# **(An exception: if # 1 complete, test cases and error checking scenarios may be worked on by Code Testers while Step 2 is being completed.)**

Code testers will need to check Gantt charts for dates when they will be expected to work on specific testing tasks. Gantt charts will be shared after Step 1 has been completed by all groups (or Wed. Jan. 10th). Note: Gantt charts do change and this is expected. (Shared Gantt Charts will help Liaison persons who may be testing and will need to see that their code fits into the other projects deadlines.

# **Absolute Due Date: Thu. Jan 17/19**

As you know, no program works the first time. Debugging is critical. After groups believe their code is complete (and within your deadlines – see Gantt chart), you will be testing your game project. You will also be testing other games if you are a Liaison person. All testing logs will be kept and handed in as part of your role.

The code tester will evaluate, suggest and possibly correct code (correct code only for your group.) The tester receives marks for knowing how your game program works and how all collaborative classes used in all programs work shown through the test routines AND logs. This person also fills in logs for test data and provides recommendations, ratings, improvements suggestions implemented or requested depending upon deadlines.

This will also be recorded in your daily logs.

**#4 – Document, Deploy and Demo to Class From step 2 & 3 completion.**

# **Absolute Due Date: Tues Jan 22/19 (Demo to class, Tuesday and Wednesday.)**

The final stage. 

A group version of everything zipped and named appropriately will be handed in. This includes;

* the original version of Gantt chart, a mid-stream modified version, and the final version of the Gantt chart (though Ms. Harris will collect many more versions of Gantt charts, any major changes should be handed in..)
* the original version of flow chart, a mid-stream modified version, and the final version of the flow chart
* the original version, a mid-stream modified version, and the final version of the optional tool your group chose
* The first version tested of your game (working), a mid-stream version (working, when rotation changes) and the final version of your game. {Note: each pre-final version must be clearly identified and in appropriate folders, so no mistake is made. Suggested naming conventions *game\_name-*alpha-*group*.zip, *game\_name-*beta-*group*.zip, *game\_name-*final-*group*.zip} HINT: watch comments and sources.
* Trouble shooting logs from all testers and your group notes on testing and approvals.
* All documentation, preliminary and final
* Individual work that must be handed in includes;
  + All daily logs (12 days marked, 2 worst removed for 10 logs) – no absences allowed.
  + All coding/work for your job. {Your game and other games worked on if you are a tester.}
  + All Reports

# **#5 – Self Evaluation – Google Form.**

This may be completed when project has been handed in and not before. This will be live Wed. Jan 23rd or earlier if needed.

# **Additional Details**

## **Programming**

A. I know you can program. This is the end of ICS3U. What you need to show me is what you have learned since the beginning of this course. I know that everyone can sometimes hit the “wall”. If this happens, you may temporarily add hardcode and work on another piece. Do not stop. Log it in today’s log so you do not forget. Tonight or later today, check your resources. Give credit where due. If unsolved, tomorrow return and ask for help, try again. If you cannot solve it in the first ½ of the class get more help. Your first resource are the people on your game project. Log this and give them credit or state you are still stuck. (Do not procrastinate, especially where help is concerned.) You know what happened in the small project we did in class, this is bigger.) Your second resource is group person assigned as a floater. The people doing the same job in other games in the class. Log this and give them credit or state still stuck. The third group of people are your job testers. Log this and give them credit or still stuck. Be open to other ways of solving your problem, other interpretations. If absolutely stuck, and you’ve tried all resources, submit a “Help request” (subject - EMail) to your teacher on the second day, include all details from your logs of what you have done/tried/who you asked for help etc. Your teacher will look at it overnight, excluding Monday evenings.

You may come to me **3 times**, so work quickly and use all your resources first.

B. There is no time for “bad habits” during class and no time for absences. Every time you are not engaged your Learning skills and work habits will be reduced and your final mark could be lowered a partial level. You must be present in class or have a **validated absence**. This is your final for the course (no exam).

Example: Your testers can set up and create a test set of data ready to go, but cannot really test code until code has been written. When you are at a point like this, help your game team with whatever is required.

## **Handing in Early**

Yes. Please do. If you get everything finished before another group, or even tied with another group you have an advantage in the market place. For each 1 school day your project is early, your mark goes up a partial level. You can only hand in your project early to a maximum of 3 school days (1 complete level). Remember your code must work to pass.

You cannot earn the complete level, above, if you have missed any of the time lines for handing in any of the parts you were responsible for because the group may have completed parts on your behalf. Partial levels possible, but watch your deadlines carefully, particularly testers as you have multiple deadlines to handle.

## **How do I earn a 4+**

A. See above, handing in early.

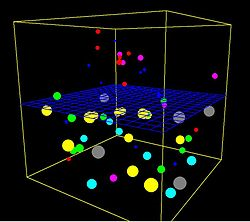
B. Package up your program to work on ANY PC/laptop without Python, (or your stick) or GibHub – you may test this out on any PC in our classroom with your stick removed.

C. Put your group program on Android/Blackberry and/or iOS (due date for this piece is end of Class Jan 23rd). Note: This only applies to marks level 3+ and above earned on Jan 23/19.

## **Other evaluations tools that will be used:**

- Spot checks

- Self-assessment (see # 5)

- Demonstration (last 2 days)

- Check lists

- Discussions

- Observations (learning skills and work habits) – may be check lists.

# **General Specifications to watch for;**

1. Be user friendly;

a. Clear instructions for the user.

b. “Help” section is always useful and available.

2. An Open/Welcome screen, Game screen, scoring, and conclusion screen. Feedback is useful if your game is trivia/learning

3. Follow proper naming conventions for variables, classes, functions etc.

4. Create good quality code. (comments, good use of symbols, graphics etc.)

a. The following used properly; variables and constants, selection and repetition, functions and classes, GUI, IO, sharable classes

b. To achieve the best marks, you should also include some more advanced structures like; arrays, user defined classes and functions, string manipulation, events, time, motion, images, tables, sound etc.

5. Look at the rubric for marking, checklists, Gantt chart, flowchart, reread your logs - daily (stay on top)

6. Don’t wait for someone – move ahead and return if necessary – help your teammates on the game if you are waiting for a key person (see Gantt chart.)

7. Be careful – help not “do”.

**Rough Mark breakdown – {refer to rubric too}**

## **Group Marks: (20%)**

Completeness (5%)

Your part in Step 1 (10%) - beginning

{More, a larger percentage of your mark will be assigned to Step 1 if your role is Program management and Gantt charts}

Your contribution - Step 2 – 4 reports, code & design, logs, help (10%)

See each Rubric below for contributions to Group marks, not all may apply.

## **Job Marks: (50%)**

Your job;

* Project Manager (2 people - approx. ½ project - roles switch)
  + Gantt charts, motivating your team – ongoing leadership, organizing and planning, programming, group meeting with Project Managers and your game group, seeking help and finding solutions, coding, documentation – ie milestones (met, not met etc.), help where required most {other jobs as needed}
  + Products responsible for: 1st, mid, and final versions of Gantt chart, flow chart and optional tool, checklist, Rules, list of screens, daily logs.
* Programmer/Coder/GUI Design (2 people - approx. ½ project - roles switch)
  + Coding and Design, meeting deadlines, making sure code works and is documented properly, versioning, your logs, responding to tester requests, fixes and modifications, comments (dates, initials, and modifications), seeking and finding solutions, classes, modularity, readability of code, reports and documents as required, help as required, passing work to tester(s).
  + Products responsible for: Code including, readability, comments, selections and repetition, Classes and adoption of classes, error proofing code, advanced procedures and functions. Also daily logs.
* Tester (2 people - approx. ½ project - roles switch)
  + All functions related to testing of your game first, and secondary testing of other group games. Test each class/function/array for my game and other’s when requested (2nd priority), test GUIs, develop test case data and scenario’s to crash proof, error checking to verify fixes, report on all work and results. You will also test the final game for each group and rate as appropriate. Integrity, honesty and accuracy matters. You will be busiest during phase2, 3, and 4.
  + Products responsibly for: Testing logs, daily logs.
* Document Expert AND Team Liaison (this job is for entire project - does not have another rotation.)
  + Coding and design Class elements that are shared, scoring, good with Turtle, pyGame, meeting deadlines, making sure group programming and GUI code works and is documented properly with other Team Liaisons, versioning, your logs, responding to tester requests, fixes and modifications, comments (dates, initials, and modifications), seeking and finding solutions, working closely with Coder(s) and group for initial splash and intro. Screens including rules which would potentially be subclasses. Screens including win/loss at end of game.
  + Checklists, Rules, Screens
  + Products responsible for: all class elements.
* Everyone - seeking/finding solutions helping my group
* One group has one fewer person: You may decide what role is being split, when and by whom excluding the Team Liaison piece, this must remain constant. (IE: This will be determined by your group strengths and what will need more effort to achieve and must be clearly communicated to your teacher and Liaison to the other group Liaisons)

## **Reporting Marks: (20%)**

Through daily logs and project documentation, depending upon your role, your effort and contribution will be included to determine your mark in this area. Honesty is important, especially in the logs so that your group will assist as needed to ensure group success. Spelling and grammar on final products will be accessed as well. Keep it honest and accurate and everyone should be able to reach a level 4 for Reporting.

# **Ministry Curriculum Expectations – examples in this project**

The course has touched upon curriculum expectations many times, specifically the final project touches on all areas and specifically addresses;

A1 – arrays, constants, variables, assignment statements, string expressions, and Boolean operators.

A2 – user input, processing, screen output, sequence, selection, repetition, nested structures, and arrays.

A3 – functions, random numbers, absolute values, parameters, local and global scope.

A4 – identify and correct syntax, logic and run-time errors, naming conventions, internal documentation, writing, compilation, testing, tracing, program flow, range of test cases

B1 – problem solving strategies, solve problems independently and as part of a team, and IPO.

B2 – design programs, appropriate vocabulary, program design, modularity, classes, structure control, flow charts, data flow diagrams, pseudocode, and user-friendly software interfaces.

B3 – simple algorithms, remove date in the middle of an array, solve common math problems, and handle exceptions (e.g. division by zero).

B4 – phases and products (problem definition, design, writing code, testing, implementation, maintenance, milestones, program specifications, products, bug reports etc., techniques to clarify specifications, Gantt and PERT chart, milestones, test programs with suitable data, debug programs, and status of project.

C1 – peripheral devices required, USB flash drive, mouse, keyboard, monitor, and possibly speakers.

C2 – file maintenance, versioning and backing up projects.

C3 – source code, debugger, IDE help, tutorials, websites, user documentation, source code, machine code, interpreter and compiler, functions of applications, programming language (Python), and operating systems (PC, Mac, iOS, Android).

D1 – negative effects of computer use (eye strain, musculoskeletal disorders).

D2 – Exploring Computer science not covered in the Final Project.

D3 – Postsecondary Opportunities, not covered in the Final Project.

**Rubric**

Levels

4- Demonstrates a strong understanding.

3- Demonstrates a good understanding

2- Developing an understanding.

1- Shows no evidence of understanding

0- Not handed in, not completed, not working. (Individual mark – not group mark.)

Other team member completed your work. (Assistance would, depending upon the issue, not necessarily earn a zero. Level 3 to 1 depending upon issue – log properly please).

**1.** PROPOSAL AND PROBLEM ANALYSIS # 1 and # 4 (**Project Management)**

Gaming world job titles include: (Assistant Producer/Audio Engineer/Creative Director/Dev. Ops Engineer etc.)

**Rough marking Group:**

|  |  |  |
| --- | --- | --- |
| **Evaluation Item** | **Level** | **KTCA** |
| Group Mark: Time Management (whole proj.) | 4 3 2 1 0 | A |
| Group Mark: Gantt chart (version 1) | 4 3 2 1 0 | T |
| Group Mark: Pseudo/IPO/checklists (v. 1) | 4 3 2 1 0 | K |
| Group Mark: Problem definition | 4 3 2 1 0 | C |
| Group Mark: Flowchart | 4 3 2 1 0 | C |
| Group Mark: Rules (version 1) | 4 3 2 1 0 | A |
| Group Mark: Final Documentation | 4 3 2 1 0 | A |
|  |  |  |
| Individual Mark: your daily logs | 4 3 2 1 0 | C |
| Individual Mark: your help to the game | 4 3 2 1 0 | K |
| PM:Individual Mark: your reports – milestones met {Completeness \_\_\_\_%} | 4 3 2 1 0 | C |
| PM:Individual Mark:- mid, or final versions of Gantt chart ( | 4 3 2 1 0 | C |
| Individual Mark: mid, and final versions of flow chart | 4 3 2 1 0 | A |
| Individual Mark: 1st, mid, and final versions of tool | 4 3 2 1 0 | K |

**2. Coding and Design (Programmer and Liaison)**

Gaming world job titles include: Animator, Game Programmer, Games Artist, Graphic Designer, sometimes Audio Engineer or Security Engineer, Researcher.

|  |  |  |
| --- | --- | --- |
| **Evaluation Item** | **Level** | **KTCA** |
| Group Mark - Liason: Classes/Subclasses/Overloading/Function (\_\_\_\_\_%) | 4 3 2 1 0 | A |
| Group Mark - Liason: Modularity (Break larger problems up into smaller functions/objects/symbols) | 4 3 2 1 0 | A |
| Group Mark - Liason: Advanced coding | 4 3 2 1 0 | A |
| Group Mark - Liason: Error checking and trapping | 4 3 2 1 0 | T |
| Group Mark - Liason: GUI | 4 3 2 1 0 | T |
| Group Mark - Liason: Testing | 4 3 2 1 0 | K |
| Individual Mark: your daily logs | 4 3 2 1 0 | C |
| Individual Mark: your help to the game | 4 3 2 1 0 | K |
| Individual Mark: your reports – milestones met {Completeness \_\_\_\_%} | 4 3 2 1 0 | A |
| Individual Mark: Code header-comments (Description, name, date, source…)  In every section where you have code. | 4 3 2 1 0 | A |
| Individual Mark: seeking/finding solutions for other’s outside my job | 4 3 2 1 0 | T |
| Individual Mark: error proofing code | 4 3 2 1 0 | A |
| Individual Mark: advanced procedures and functions. | 4 3 2 1 0 | A |

**3. Testing & Debugging/ Liason**

(Editor/Tester)

|  |  |  |
| --- | --- | --- |
| **Testing and Debugging** | **Level** | **KTCA** |
| Group Mark: Implement suggestions made by group members or Liason | 4 3 2 1 0 | A |
| Group Mark: Help with testing as time allowed, depending upon job requirements(PM’s, extra person, include documentation.) | 4 3 2 1 0 | A |
|  |  |  |
| Individual Mark: your daily logs | 4 3 2 1 0 | C |
| Individual Mark: your help to the game | 4 3 2 1 0 | K |
| Individual Mark: Test each small function/class my | 4 3 2 1 0 | K |
| Individual Mark: Test small function/class other game | 4 3 2 1 0 | K |
| Individual Mark: Develop setup test module (hard coded where applicable to test each scenario ) | 4 3 2 1 0 | T |
| Individual Mark: Test data developed for my game - include docs/logs | 4 3 2 1 0 | T |
| Individual Mark: Test data developed for other game – include docs/logs | 4 3 2 1 0 | T |
| Individual Mark: Test conditions for classes my game | 4 3 2 1 0 | A |
| Individual Mark: Test conditions classes other game | 4 3 2 1 0 | A |
| Individual Mark: Error checking– my game | 4 3 2 1 0 | T |
| Individual Mark: Error checking– other game | 4 3 2 1 0 | T |
| Individual Mark: Reporting testing results | 4 3 2 1 0 | C |
|  |  |  |

**4 Reporting & Liason (Documentation, Programmer and Researcher)**

Gaming world job titles include: Animator, Graphic Designer, sometimes Audio Engineer, Document Specialist, Code Analyst, Report Writer, Development Specialist and more

|  |  |  |
| --- | --- | --- |
| **Evaluation Item** | **Level** | **KTCA** |
| Group Mark: GUI Classes/Subclasses/Function | 4 3 2 1 0 | A |
| Group Mark: GUI Modularity Splash screen | 4 3 2 1 0 | A |
| Group Mark: Overall Readability GUI - Instructions | 4 3 2 1 0 | K |
| Group Mark: GUI Selection and Repetition | 4 3 2 1 0 | A |
| Group Mark: GUI Error checking and trapping | 4 3 2 1 0 | T |
| Group Mark: GUI Advanced: arrays, String manipulation, timers, etc. | 4 3 2 1 0 | T |
|  |  |  |
| Individual Mark: your daily logs | 4 3 2 1 0 | C |
| Individual Mark: your help to the game | 4 3 2 1 0 | K |
| Individual Mark: your reports – milestones met {Completeness \_\_\_\_%} | 4 3 2 1 0 | A |
| Individual Mark: Code header-comments (Description, name, date, source…)  In every section where you have code. | 4 3 2 1 0 | A |
| Individual Mark: Splash screen | 4 3 2 1 0 | A |
| Individual Mark: Instructions screen | 4 3 2 1 0 | A |
| Individual Mark: final reporting (everything combined into one document – group version) | 4 3 2 1 0 | A |
| Individual Mark: Win/loose screen | 4 3 2 1 0 | A |