**St. David Catholic Secondary School  
ICS 4UI Final Evaluation Project Assessment  
Due Date: Wednesday January 31st, 2017**

**Learning Goals:**The student can:

* demonstrate the ability to use different data types and expressions when creating computer programs (objects, data structures, strings, data comparisons)
* use modular programming concepts and principles in the creation of computer programs (classes and methods, inheritance, polymorphism.)
* design and write algorithms and subprograms to solve problems (use of file input/output, recursion, sorting techniques)
* use proper code maintenance techniques (formal testing, java documentation)
* apply standard project management techniques in the context of a student - managed project (project stages - planning, development, production and closing)

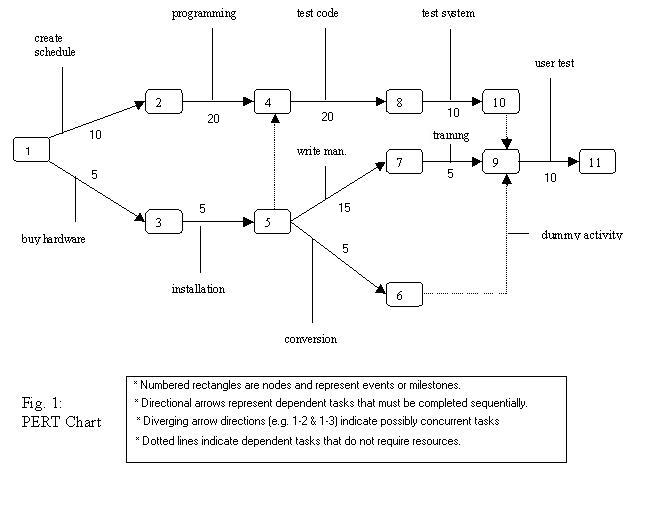
Students are to choose a programming project that interests them and that meets the overall expectations listed above. Previous project have included but not limited to:

* board games
* dice games
* card games
* video games

The project will follow the stages:

**Stage 1 Analysis/Design**Requirements:

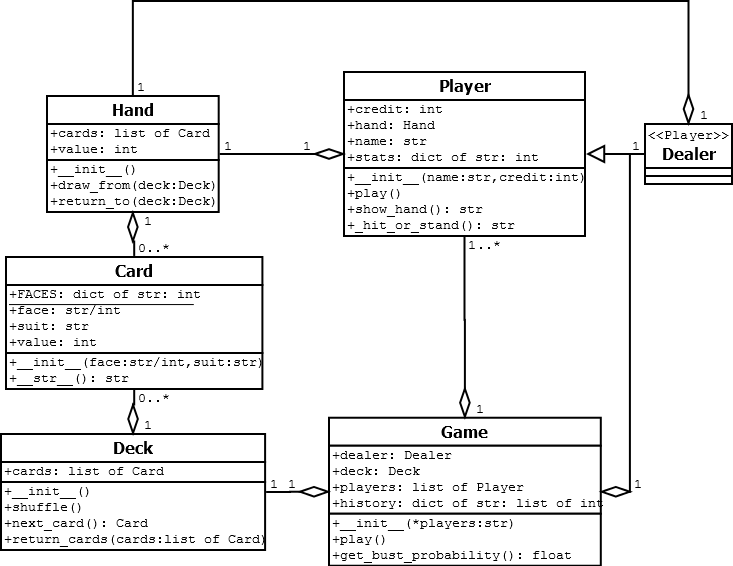
* your name, project title
* **project log/tracking (**Overall planning guideline (see demo below))



* Day by day **log** of both accomplishments as well as goals for each partner in the team

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Team Member** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |
| Angela | Planned game structure  Began GUI  Programmed user movement of box | Finalized game design | Abstract classes for user movement | Continued working with character movement | Implemented characters into GUI  shooting |
| Jack | Planned game structure | Enemy abstraction class | Different enemy classes | Implementing user and enemy to GUI | Working on enemy movement |
| **Team Member** | **Day 6** | **Day 7** | **Day 8** | **Day 9** | **Day 10** |
| Angela | Continued with shooting | Intersection between enemy and bullet | Scoring class / end of game | Improved overall appearance and run | Fixed bugs |
| Jack | Working on enemy attack | Flow control inside of game | Implementing enemy move and attack into GUI | Improved overall appearance and run | Fixed bugs |

* **project analysis (outline)** 
  + a detailed **UML** design of all classes:
    - Be specific about all **instance variables** and **methods** contained in each class
    - Are you implementing the three guidelines of **stepwise refinement**?
  + a detailed **mapping** of the relationships between **all classes** (inherited methods, static methods, public and private methods, etc.)



* + possible **future enhancements** to project
  + are there any topics that you will need to research to aid in the development of your project?

**Stage 2 Development and Implementation**

This is the **coding** and **testing** phase of the project. At all times, you should have a portion of your project working and tested. It will save a great deal of time if you do bottom-up development and test each class's methods as you write them. You must also document thoroughly, this is best done as you go, not left to the end. Remember to follow all coding standards. Your project must be ready to be tested by your classmates on the day of the final evaluation. Also be prepared to answer any questions regarding your code.

**Hand in:** Hand in all source code electronically.

**Final Evaluation Rubric Name(s):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category/Criteria** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| **Application** of programming concepts (**Objects**) defined under the overall learning goals of the project | program meets project expectations in a limited manner | program meets project expectations with some effectiveness | program meets project expectations with considerable effectiveness | program meets project expectations with complete effectiveness |
| **Thinking**  **design** principles and methodologies (classes and methods) of project (see project analysis)  level of **complexity** of project  error checking/debugging | limited understanding of design principles and methodologies  project is very simplistic  limited error checking | some understanding of design principles and methodologies  some degree of project complexity  some error checking | considerable understanding of design principles and methodologies  considerable project complexity  considerable error checking | thorough understanding of design principles and methodologies  project is highly complex  complete error checking |
| **Communication**  Documentation follows style guide of Java documentation  comments communicate purpose of code | code follows style guide limited accuracy  comments communicate with limited clarity | code follows style guide some accuracy  comments communicate with some clarity | code follows style guide considerable accuracy  comments communicate with considerable clarity | code follows style guide complete accuracy  comments communicate with a high degree of clarity |

Comments: