COMP 2710: Project 1 – Phase 1: Analysis

Dragons – A Simple Text-Based Game

Points Possible: **100 points**
Analysis Portion turned in via Canvas

No collaboration between students. Students should NOT share any project code with each other. Collaborations in any form will be treated as a serious violation of the University's academic integrity code.

Rating

- Design difficulty: 4/5
- Implementation difficulty: 2/5
- Time required: 4/5
- Fun: 3/5

Goals of Project 1:

- To perform object-oriented analysis, design, and testing
- To develop some proficiency with basic C++ syntax and semantics.
- To learn data flow diagrams.
- To learn use cases and use case diagram.
- To gain experience with unit and system testing.
- To perform separate compilation.
- **To use singly linked lists**
- To develop a reasonably user-friendly application
- To write a fun application!

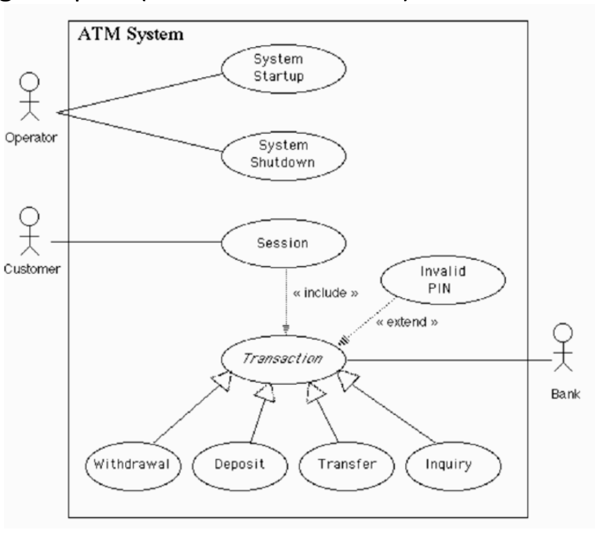
Goals of Project 1 - Phase 1:

- To design a use case diagram to capture the requirements of project 1.
- To use the argoUML tool to create a use case diagram and specify use cases.
- To use the argoUML tool to create a data flow diagram

1. Overview

1.1 (70 points)

1.1.1 **(30 points)** Design a **use case diagram** for a text-based game. The following is a sample use case diagram for an ATM system. Another sample use case diagram can be found at: "Sample Data Flow Diagram and Use Case Diagram.pdf" (available on Canvas).



1.1.2 (40 points) Write a few important **use cases**. Remember, these use cases describe how the user interacts with the text-based game (what they do, what the systems does in response, etc.). Your use cases should have enough basic details such that someone unfamiliar with the system can have an understanding of what is happening in the text-based game. They should not include internal technical details that the user is not (and should not) be aware of. Make sure that any special rules/features you plan to add are clearly described in your analysis section. Check out the following link for a use case example:

http://www.eng.auburn.edu/~xqin/courses/comp2710/useCases.ppt

1.2. (30 points) You must prepare a data flow diagram for your text-based game. Please refer to http://en.wikipedia.org/wiki/Data_flow_diagram for details on data flow diagrams. A sample data flow diagram can be found at: "Sample Data Flow Diagram and Use Case Diagram.pdf" (available on Canvas).

2. Background

A customer wants a cool new video game, similar in style to popular video games like *World of Warcraft* or *Bioshock*. However, the customer recognizes the lack of significant funds and only has a resource poor computer to play this game on. Hence, the game will be a simple text-based adventure concerning a graduate student trying to navigate his way down Shelby Center. In this project, you will help the customer to design and implement a simple text-based game.

3. Requirement Details

3.1. Player

The "player" is represented by *at least* three attributes: *intelligence*, *time*, and *money*. If the player runs out of intelligence, time or money, the player dies. The goal of the player is to survive to the end of the "hall" with the highest combined total of the attributes as possible. "Score" is determined as the three attributes multiplied together.

3.2. The Hall

The player starts the game at the beginning of a hall, which is linear.

The "Hall" is a path that is at least twenty (20) steps long. After a move, the user should be told how far away from the goal they are (in steps). If the player survives to the goal square without any of the attributes falling to 0, they win. Their score should be displayed with a simple ASCII victory message. If the player dies, a "You Lose" message should appear indicating the cause of death (for example, if money falls to zero, you can say that the player starved to death because of poverty). All the attributes should start in some random range (e.g. 8-25).

3.3. Turns

Every turn, the player has (at least) 5 options to choose from:

- **Move**: The player moves one step in the grid, but risks an Encounter or a Puzzle. Moving also takes time.
- **Read technical papers** (boost intelligence): The player loses a fixed amount of time, but increases intelligence by a random amount
- **Search for loose change** (boost money): The player loses a fixed amount of time, but increases money by a random amount.
- **View character**: A simple display should show the character attributes and current position in the hall (ASCII is fine)
- **Quit the game** (shows the "You Lose" screen – optional mockery- and exits the program)

3.4. Encounters

Encounters: Every time the character steps, there is a random change of various events happening. You are free to change the probabilities as you see fit for "game balance," but here are some suggestions:

- 25% chance: nothing happens, you just move forward.
- 30% chance: You encounter a Puzzle (see Puzzle below)
- 10% chance: Encounter a professor. This loses a random extra amount of time, but may slightly increase intelligence.
- 10% chance: Encounter another graduate student. This loses a random amount of time.
- 15% chance: Attacked by grunt work! Lose both time and intelligence.

- 10% chance: Grade papers. Lose time, but gain money.
- 0% chance: Get a huge raise, gain lots of money! (This never happens).

3.5. Puzzles

Puzzles: Puzzles are different from normal encounters since they require interaction from the user. These don't necessarily have to be brilliant, but riddles or even edutainment light puzzles are fine. Examples:

- "What is 2 + 2:" For a correct response, Money + 1. For an incorrect response, Money -20 (you idiot).
- "What can you put in a barrel to make it lighter?" For a correct response, int+2. For an incorrect response, int-2.

3.6. Other options

You are free to add more details and rules to your game, but you must have at least the above specifications. Feel free to be creative – there are many opportunities to do so.

4. argoUML

In this project, you must use argoUML to create a **use case diagram**, **data flow diagram** (i.e., **statechart diagram**), and specify all the use cases in your use case diagram.

4.1. Availability

The URL of the argoUML's website is: http://argouml.tigris.org/

4.1.1 Windows Users

Please click the following link to download argoUML for Windows.
http://argouml-downloads.tigris.org/nonav/argouml-0.34/ArgoUML-0.34-setup.exe

4.1.2 Mac OS Users

Do NOT download and install ArgoUML on your Mac OS.

You **must** launch ArgoUML via Java Web Start. The Google Chrome does not work for this application. Please use either Safari or Firefox as a browser. Click the following link to launch the latest stable release:
http://argouml-downloads.tigris.org/jws/argouml-latest-stable.jnlp

You may encounter the following two problems:

- How to open apps from an unidentified developer in Mac OS?
  The answer can be found here:
  http://www.imore.com/how-open-apps-unidentified-developer-os-x-mountain-lion

- How to fix Application Blocked by Security Settings in Mac OS?

The answer can be found here:
http://shellzero.wordpress.com/2014/01/24/how-to-fix-application-blocked-by-security-settings-mac-os-x/

In your java security setting, please allow your Mac OS to launch application from argoUML's website (i.e., http://argouml-downloads.tigris.org).

4.1.3 User Manual
The argoUML User manual can be found below:
http://argouml-stats.tigris.org/documentation/manual-0.34/

4.2. Create a Use Case Diagram using argoUML
Your use case diagram captures how use cases and actors interact. Click the following link to see an example of an ATM system's use case diagram.

http://argouml-stats.tigris.org/documentation/manual-0.34/ch04s03.html

In this project, you must use ArgoUML to draw a use case diagram. When you create a new project (see **Section 5** for "how to name your project") it has a use case diagram created by default, named use case diagram 1. Please click the following link for detailed instructions on how to use argoUML to create use cases in your first use case diagram.

http://argouml-stats.tigris.org/documentation/manual-0.34/ch04s04.html

4.3. Create Use Case Specification in argoUML
Your must also use argoUML to document the behavior of each use case in your use case diagram. The specification of a use case should be described in ***the Documentation tab*** of the use case. The specification of each use case should contain the following items:

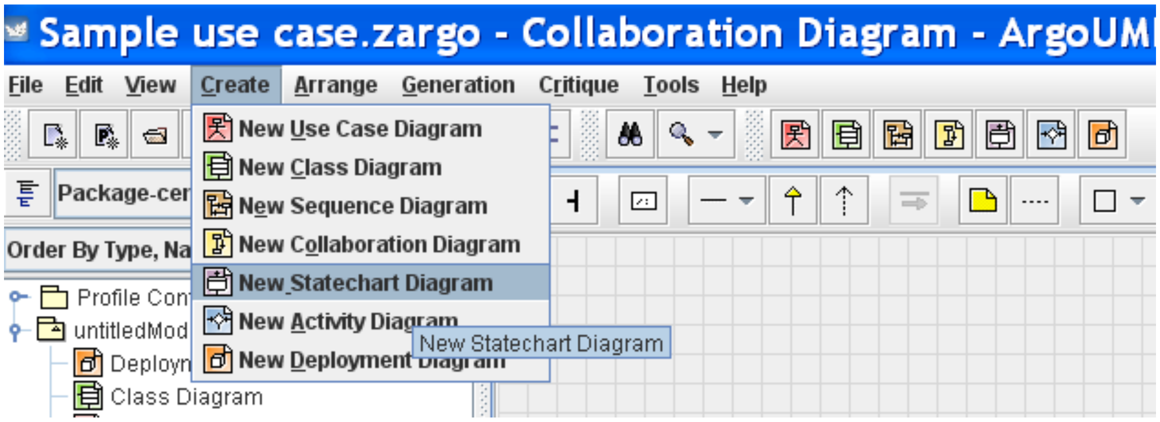
- **Name.** The name of the use case to which this relates.
- **Goal.** A one or two line summary of what this use case achieves for its actors.
- **Actors.** The actors involved in this use case, and any context regarding their involvement. Note: This should not be a description of the actor. That should be associated with the actor on the use case diagram.
- **Pre-condition.** These would be better named "pre-assumptions", but the term used everywhere is pre-conditions. This is a statement of any simplifying assumptions we can make at the start of the use case.
- **Basic Flow.** The linear sequence of steps that describe the behavior of the use case in the "normal" scenario. Where a use case has a number of scenarios that could be normal, one is arbitrarily selected.
- **Alternate Flows.** A series of linear sequences describing each of the alternative behaviors to the basic flow.
- **Post-conditions.** These would be better named "post-assumptions". This is a statement of any assumptions that we can make at the end of the use case.

- **Requirements.** In an ideal world the vision document, use case diagrams, use case specifications and supplementary requirements specification would form the requirements for a project.

Please click the following link for details on the use case specification:
http://argouml-stats.tigris.org/documentation/manual-0.34/ch04s03.html#d0e3323

4.4. Create a Data Flow Diagram using argoUML

Your can also use argoUML to draw data flow diagram for your system. To create a data flow diagram, you can click the "New Statechart Diagram" item in the in the "Create" menu (see the Figure below).



By default, the name of your statechart is "Unnamed StateMachine". Please change this default name to represent your own data flow diagram.

5. Deliverables

Please submit your project analysis through the Canvas system (e-mail submission will **not** be accepted). You just need to submit your analysis document as an ArgoUML compressed project file (*.zargo). The file name should be formatted as:

project1-analysis.zargo

Note: other format (e.g., pdf, doc, txt) will not be accepted.

6. Grading Criteria

6.1 (30 points) Use case diagram

1. (5 points) Actors
2. (10 points) Use cases in the diagram.
3. (5 points) Relations among actors and use cases.
4. (10 points) Relations among use cases.

6.2 (40 points) Use case specification

5. (10 points) Name, goal, actors in each use case

6. (10 points) Pre-condition/post-condition in each use case
7. (20 points) Basic Flows/Alternate Flows in each use case

6.3 (30 points) Data flow diagram

8. (5 points) External entity
9. (10 points) Processes
10. (10 points) Dataflows
11. (5 points) DataStore

7. Late Submission Penalty

- A ten-point penalty per day for late submission. For example, an assignment submitted after the deadline but up to 1 day (24 hours) late can achieve a maximum of 90% of points allocated for the assignment. An assignment submitted after the deadline but up to 2 days (48 hours) late can achieve a maximum of 80% of points allocated for the assignment.
- Assignment submitted more than 3 days (72 hours) after the deadline will not be graded.

8. Rebuttal period

- You will be given a period of 72 hours to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

9. Hints

- Start early, you have a good deal of time but you may need it to learn how to use the argoUML tool. Although the following timeline is not mandated, it is a suggestion of milestone:
    - 1/4 time: Download the argoUML tool. Get familiar with the tool.
    - 2/4 time: Read the project specification (i.e., this document). Finish process planning.
    - 3/4 time: Finish the use case diagram; complete important use cases.
    - 4/4 time: Prepare the data flow diagram and finish final documentation using argoUML.
- If you bring your design documents by early, I will give you comments and help point you in the right direction on this project.

10. Sample Usage

```
What’s your name? Bob
```

```
         ============================================================
         |                   Welcome, Bob!                          |
         ============================================================

    1) Start a New Game of Dunstan and Dragons!
    2) View top 10 High Scores
    3) Quit

         Please choose an option: 2

    The top 5 High Scores are:

    Win 1337
    CaseyZZZ 625
    JonnieKill 400
    Bob 75
    Daisy 33
    -no more scores to show-


    1) Start a New Game of Dunstan and Dragons!
    2) View top 10 High Scores
    3) Quit


         Please choose an option: 1

Entering the Dungeon…

You have:

intelligence: 20
time: 25
money: $11.00

You are 20 steps from the goal.  Time left: 25.

          1) Move forward(takes time, could be risky…)
          2) Read technical papers (boost intelligence, takes time)
          3) Search for loose change (boost money, takes time)
          4) View character
          5) Quit the game


      Please choose an action: 4



You have:

intelligence: 20
time: 25
money: $11.00
```

```
You are 20 steps from the goal.  Time left: 25.

         1) Move forward(takes time, could be risky…)
         2) Read technical papers (boost intelligence, takes time)
         3) Search for loose change (boost money, takes time)
         4) View character
         5) Quit the game


      Please choose an action: 2

You read through some technical papers.  You gain 3 intelligence, but
lose 2 units of time.

You are 20 steps from the goal.  Time left: 23.

         1) Move forward (takes time, could be risky…)
         2) Read technical papers (boost intelligence, takes time)
         3) Search for loose change (boost money, takes time)
         4) View character
         5) Quit the game


      Please choose an action: 1

You move forward one step, and…

NOTHING HAPPENS!

You spent one unit of time.

You are 19 steps from the goal.  Time left: 22.

         1) Move forward (takes time, could be risky…)
         2) Read technical papers (boost intelligence, takes time)
         3) Search for loose change (boost money, takes time)
         4) View character
         5) Quit the game


      Please choose an action: 1

You move forward one step, and…

YOU FIND SOME PAPERS TO GRADE.

You spent two units of time, but gained $3.00!

You are 18 steps from the goal.  Time left: 20.  You can move forward
or backward.

         1) Move forward(takes time, could be risky…)
         2) Read technical papers (boost intelligence, takes time)
```

```
          3) Search for loose change (boost money, takes time)
          4) View character
          5) Quit the game


      Please choose an action: 1

You move forward one step, and…

PUZZLE: It’s a riddling imp.  I hate riddling imps.  But fine, he asks:
“Find the product of 8 and 8!”

   1) 16
   2) 64
   3) 256
   4) Uh…uh… no?

Choose wisely: 4

The imp cackles “Oh yes.  Yes indeed.  Now you die.”

TIME HAS FALLEN TO ZERO.  YOU DIE.

<Print Score, adjust high scores>
```