

## **Defining the Problem**

Client – “Trucidation”, a fellow guild member.

Advisor – Mr. Balcombe, computer science teacher.

The client and users of my proposed solution are members of the Black Desert Online community. In this game, there exists an “Amity mini-game”, through which one is able to unlock special features the more Amity one has in-game.

The mini-game has a series of three “conversation” rounds: by choosing a selection (3-7) of conversation topics a fixed pool of topics (10-12), the player can gain different amounts of Amity depending on the order and combination of topics chosen whilst fulfilling a given success condition (randomly chosen; 5 conditions exist).

A route optimization problem exists when attempting to find the best combination and the order of said topics of conversation for the possible success criteria.

It is extremely time-consuming to calculate and go through all the possible combinations in order to find the optimal route for gaining the highest amount of Amity.

I interviewed [See Appendix 1] various guild members who also complained of this same problem. When I suggested to make a program for solving this time-consuming dilemma, they were in full agreement that such a creation would make their gameplay experience both simpler and more enjoyable.

The thoughts and suggestions of my fellow guild members shall be aiding me throughout this creation process.

## **Rationale for Proposed Solution**

A java application would be perfect for this problem. Through its platform independent, users will be able to run the program on a wide range of platforms that support the Java RE, input the necessary data required for calculating the best route, and the program will be able to calculate, through self-made route-optimization algorithms, the best route, and display it to the user, simpler and faster than if the user were to calculate the best route by hand.

I’ve chosen Java for the following reasons:

- Part of the IB CS Curriculum
- Platform Independence
- Easy to use NetBeans GUI creation kit
- Easy to input and submit data through Java’s GUI Capabilities
- OOP capabilities to reduce redundancy of information and data

## **Stating Success Criteria**

1. User will be able to input the necessary data: interest rates, favor level, percentage to spark interest; needed for route optimization calculations.

2. Program will have a simple, user-friendly UI, with buttons and text fields, to be judged and approved by users through feedback.
3. Program will have input data checks to alert the user in cases of invalid data entry.
4. Program will be able to correctly calculate the optimal route for a given set of user-inputted data – I shall test this myself with a selection of real test data multiple times.
5. Program will be able to output its proposed solution.
6. Program will provide a selectable list of most commonly used NPCs with default data values already stored.
7. Program will provide a selectable list of all possible success conditions.
8. Program will provide a selectable list of all possible constellations (route structure).
9. If the NPC that the user wants to converse with isn't in the program, the program will provide an interface to allow a user to create their own default settings and values for a custom NPC with custom conversation topics with the needed data values [1] required for the calculation of the optimal route.

(Word Count: 547)