Father Michael McGivney
Catholic Academy
November, 2015 – January, 2016
Computer Science (ICS4U1)

Final Project:
FMM Escape

Group Members:
Chun Kit (Calvin) Li
Miao Wang
Jasmine Xiao

Index

1. Analysis Phase ____ 2

1.1 Description ____ 2

1.2 Rule of Game ____ 3

1.3 Target Audience ____ 3

1.4 Objectives ____ 3

1.5 Questionnaire ____ 4

1.6 Other Games Considered, Selection Criteria ____ 4

1.7 Timeline and Duties ____ 4

1.8 Riddle and Trivia Questions ____ 4

1.9 Annotated Bibliography ____ 4

2. Design Phase ____ 5

2.1 Required Elements ____ 5

2.2 Major Pieces of Code ____ 7

2.2.1 Swing Version ____ 7

2.2.2 Console c Version ____ 8

2.3 Pseudocode of Controller ____ 9

2.3.1 Swing Version ____ 9

2.3.2 Console Version ____ 10

2.4 Test Cases ____ 11

2.5 User Interface Design ____ 11

2.6 UMLs ____ 11

3. Implementation ____ 12

3.1 Coding Log ____ 12

4. Testing and Installation ____ 12

4.1 Testing Sample ____ 12

4.1.1 Swing Version ____ 12

4.1.2 Console c Version ____ 13

1. Analysis Phase

1.1 Description

This project is about escape room game.

This game will locate in Father Michael McGivney Catholic Academy ("**FMM**"). The player need to finish an assignment from each classroom and find hints in order to escape the school. It will be fun for especially student who always want to escape from the school or generally classes every day. It also highlights the importance of completing assignments as it is the only way for you to win the game.

It is not a common escape room game – each classroom will have 6 trivia questions about the subject, and the player needs to answer all of the questions. Each classroom represents one of the school departments, Maths, English, Science, Business, Physical education and Art. This means this escape game can give *education* and *entertainment* at the same time. The questions are probably multiple-choice, to make it easier for the player. After they finish all the rounds with collected clues (if they pass each one, they might possibly get a clue, or just a motivation saying), they have to solve the final riddle with the help of those hints. It will be GAME OVER if they fail any course! They can restart after losing the game but it will be recorded down as how many years (times) they need for graduation, or if they give up their name will be recorded, but they will never graduate!

1.2 Rule of Game

You are a FMM student. Today is exam day! NO WAY!!!

You will be tested on each of the 6 basic courses in the order of Math, English, Science, Business, Physical Education, and Art.

You will not be able to skip any course, and must pass the first course to allow access to the next course.

In each round, you will be required to answer 6 trivia questions and if you get at least 4 out of them right, you pass that course.

Each time you pass a course, you will obtain an “item”, which is a hint for the final riddle or simply a motivation quote.

Once you fail any course it is game over.

You can restart but it will start from the first class (Math) again, and the years you take to graduate will increment by 1.

Be careful, once you press the "give up" button you will become a permanent student!

The “escape” button brings you to the riddle page, where you will get an unlimited number of tries to input the answer.

After you pass all courses and answer the final riddle, you will finally be able to escape from FMM!

1.3 Target Audience

Young highs school people, especially grade 12 students at FMM. Anyone who wants to escape from school by answering trivia questions! Questions’ level of difficulty will be between easy and hard for most high school students (some are very easy while some can be considerably difficult).

1.4 Objectives

The program will involve many programming skills learned in the past along with new components that have not been taught yet. It should be both challenging and fun. This game will be unique compared to other traditional, common games like card game or chess game. All group members should have fairly equal contribution to the project and timeline should be followed.

1.5 Questionnaire

1. What subjects do you prefer?
2. Do you prefer multiple choice or short answer questions?
3. Would you like to see teacher's faces in the game?
4. Would you like a game that will help you prepare for exam or simply have fun?
5. Which teacher do you want to see in the game?
6. How many rounds, how many questions in each round and how many correct answers do you think are reasonable?
7. Would you prefer real or virtual backgrounds?
8. Would you mind entering your name for record keeping?
9. Would you like a certificate once you finished the game?
10. Do you think having a button for clearing all records of scores is reasonable?

1.6 Other Games Considered, Selection Criteria

The other two major options are Chinese chess and role-playing game.

Chinese chess:
It will be too complicated to keep track of fouls, and records as we need to keep track of every piece on the board and it is a little boring to be honest.

RPG game:
We are not good writers and it would not be challenging to the players.

1.7 Timeline and Duties

Appendix A – Timeline and Duties

1.8 Riddle and Trivia Questions

Appendix B – Riddle and Trivia Questions

1.9 Annotated Bibliography

Appendix C –Annotated Bibliography

2. Design Phase

2.1 Required Elements

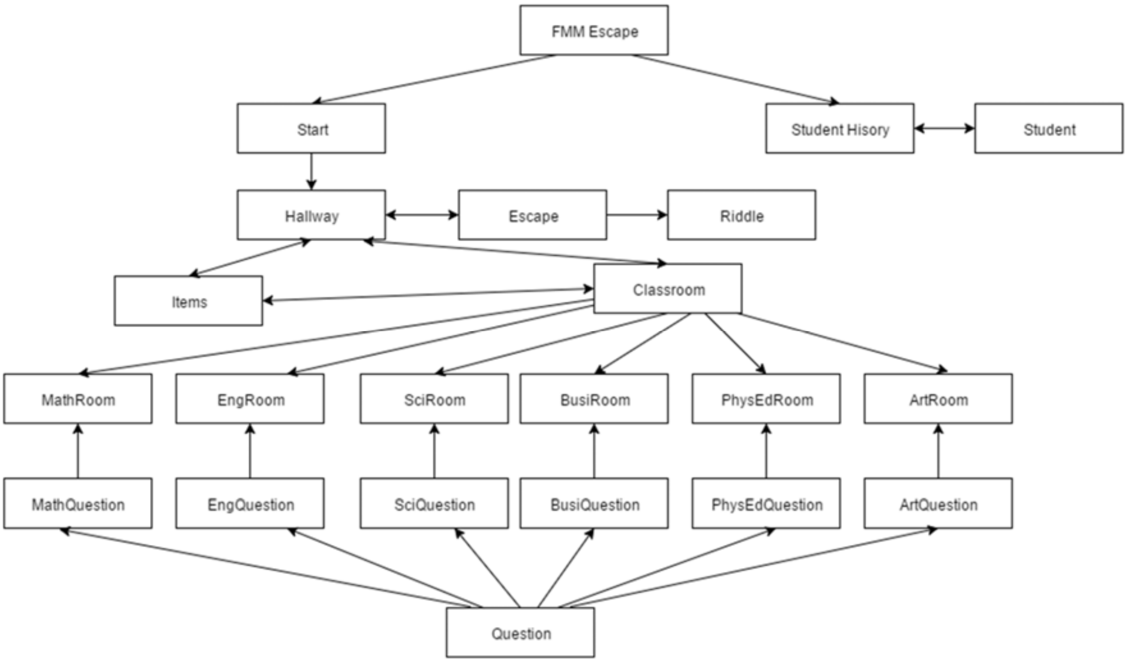
1	Two-dimensional array
	- Store multiple choice answers for the questions of each course - i.e. questions [0] is associated with choices[0][0], choices[0][1], choices[0][2], choices[0][3] in place of its a, b, c, d choice respectively
2	One-dimensional array*
	- Many, such as questions[], answers[] in Questions class - itemNames[], itemExplain[] in Items class - JButton[] in Hallway class
3	String
	- Many, such as name in Student class - questions[], answers[], choices[][] in Questions class - answer in Riddle class - Strings are compared with equals(), compareTo() and compareToIgnoreCase() methods in StudentHistory class for the binarySearch() method
4	Methods - passing parameters
	- Many, such as passing Container object to setContentPane() method in FMMHomePage class - Passing a Student object to writeStudentToFile() method in StudentHistory class
5	Multiple classes
	- Many, such as MainClass, FMMHomePage, StudentHistory, Hallway, Riddle, Items, Questions, Classroom

6	User-defined objects and classes*
	- Student objects are stored in ArrayList objects such as studentHistory, graduates, permanent in StudentHistory class - Many classes are called by instantiating objects from them (eg. Items, FMMHomePage)
7	Inheritance (subclasses and superclass)*
	- Question class to MathQuestion, EnglishQuestion, ScienceQuestion, BusinessQuestion, PhysEdQuestion, ArtQuestion - Classroom class to MathRm, EnglishRm, ScienceRm, BusinessRm, PhysEdRm, ArtRm
8	Private & public
	- Many, such as private variables like name, years, graduate in Student class - Private methods include mergeByYear() and mergeByName() in StudentHistory class - Many public variables (many Swing components are public by default) and methods (almost all methods are public)
9	Error handling (try/catch)*
	- Used when reading files and writing to files (specific files refer to element #11)
10	Search*
	- Binary search (used recursion in the process) for a Student object's name in StudentHistory class
11	I/O files including appending the file*
	- Read files from text files like FMM Instructions, Student History, math, english, science, business, physEd, arts - Write to Student History text file

12	2 of: Sorting, Recursion, Animation, Threads, Interface
- Merge sort with the use of recursion, or the use of the interface ActionListener for Swing (implements ActionListener)	
13	Something new that we haven’t learned before
- GUI - Swing, ArrayList (as an object), Swing JScrollPane, play audio files - Play music – using throws IOException and sun.audio.AudioPlayer to play music	

2.2 Major Pieces of Code

2.2.1 Swing Version



Who will code what? (Job allocation)

Calvin Li	Miao Wang	Jasmine Xiao
- Music (same as Console c version)	- Classroom - MathRoom - EngRoom - SciRoom - BusiRoom - PhysEdRoom - ArtRoom	- MainClass - FMMHomePage - Student - StudentHistory - Hallway - Items (changes from Console c version)

	- Question (changes from Console c version) - MathQuestion - EngQuestion - SciQuestion - BusiQuestion - PhysEdQuestion - ArtQuestion	- Escape - Riddle

2.2.2 Console c Version

Calvin Li
- Main - HomePage - Game - School - Item - Music - Question - MathQuestion - EngQuestion - SciQuestion - BusiQuestion - PhysEdQuestion - ArtQuestion - Classroom - MathRoom - EngRoom - SciRoom - BusiRoom - PhysEdRoom - ArtRoom - Riddle - Student (same as Swing version) - StudentHistory (changes from Swing version)

2.3 Pseudocode of Controller

2.3.1 Swing Version

```
if “start” is pressed
    prompt for player’s name
    if name is valid (does not repeat what is on the text file)
        display instructions
        display doors to classes
        if “escape” is pressed
            if all classes are passed, display the riddle question
                if answer is correct, show the escape message and record the student object on text file as graduate, show the certificate
            else display the principal’s message
        else if “give up” is pressed
            give up message displayed
            student object information saved to text file as permanent student
        else if “restart” is pressed
            go back to instructions page, student object’s years variable is increased by one
        else if “items” is pressed
            the items gained from the classes are displayed
        else if any door to any class/subject is pressed
            if all previous doors/classes are passed
                display 6 random questions and the multiple choices for each one
                if 4 or more choices are correct, item obtained
                else game over, player can restart or give up
else if “student history” is pressed
    display list of graduates and list of permanent students
    if “search” is pressed
        name of student is searched and if found, information about it is displayed
    if “clear all records” is pressed
        text file becomes empty
```

2.3.2 Console Version

```
if “start” is chose
    prompt for player’s name
    if name is valid (does not repeat what is on the text file)
        display instructions
        display game menu
        if “escape” is chose
            if all classes are passed, display the riddle question
                if answer is correct, show the escape message and record the student object on text file as graduate, show the certificate
            else display the principal’s message
        else if “give up” is pressed
            give up message chose
            student object information saved to text file as permanent student
        else if “restart” is chose
            go back to instructions page, student object’s years variable is increased by one
        else if “items” is chose
            the items gained from the classes are displayed
        else if any classroom is chose
            if all previous doors/classes are passed
                display 6 random questions and the multiple choices for each one
                    if 4 or more choices are correct, item obtained
                    else game over, player need to restart
else if “student history” is chose
    display list of graduates and list of permanent students
    if “search” is chose
        name of student is searched and if found, information about it is displayed
    if “clear all records” is chose
        text file becomes empty
```

2.4 Test Cases

These are important test cases that we will be careful to keep in mind as we are coding our game:

1. General:
   - User exits without finishing the game.
     (Name not recorded in the student list)
2. Main game:
   - User inputs a name that already exists when prompted to enter their name.
     (Return "need to enter another name" message)
   - User chooses to proceed without entering a name.
     (Return "invalid name" message)
   - User attempts to "escape" before finishing all the classrooms. (Principal stopping them from trying to cheat)
   - User attempts to skip levels and proceed without finishing the previous one. (Nothing happens until they chose the correct one)
   - User chooses to restart the game.
     (Increase the "year" player stayed in school)
   - User presses the submit button without selecting an answer in class.
     (Count as a wrong answer)
   - User attempts to go back to a completed level.
     (Do not allow them to return)
   - User fails to pass the test. (show "Stay another year" message and return to hallway to start over)
3. Student history
   - User search for a non-existent name.
     (Return "name not found" message)
   - The student history becomes overcrowded.
     (Give user the clear history option)

2.5 User Interface Design

Appendix D – Storyboard

2.6 UMLs

Appendix E – UMLs

3. Implementation

3.1 Coding Log

Appendix F – Weekly Coding Log

4. Testing and Installation

4.1 Testing Sample

4.1.1 Swing Version

Input	Location	What is tested	Works?
Vert	Enter Name Page	User enter name	Yes
Alvin Li	Enter Name Page	User enter name will a space	Yes
(Press Button)	Anywhere with button	Press the button and check any action after it	Yes
keyboard	Riddle Page	Correct answer	Yes
Apple 56125	Riddle Page	Incorrect answer and ask the user guess again	Yes
Press "Clear All Records"	Student History Page	Clear all the student's records	Yes
N/A	Student History Page	The name we entered before have displayed	Yes
Alvin Li	Student History Page	Search student's information	Yes
Kris	Student History Page	Search student's information which does not in the record	Yes
(nothing)	All the Pages	Play music when the program start	Yes

4.1.2 Console c Version

Input	Location	What is tested	Works?
a	Classroom page	Answer the question	Yes
F Fanowidnow	Classroom page	Invalid input, ask the user enter the answer again	Yes
leave	Riddle Page	Leave the riddle page	Yes
keyboard	Riddle Page	Correct answer	Yes
Apple 0156	Riddle Page	Incorrect answer and ask the user guess again	Yes
a ,s ,d, w keys	Main Menu Page Game Menu Page Items Page	Move the green box	Yes
<Space> key	Anywhere with "Please Press any key to continue"	Press the key and continue the game	Yes