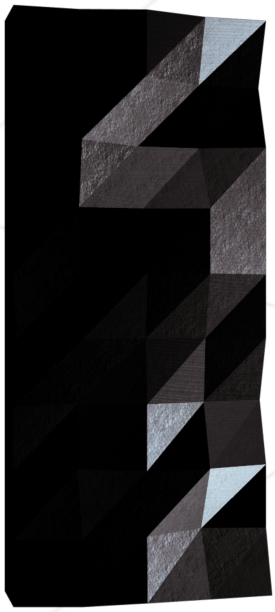

Yet another Advanced Dungeons and Dragons - YADD

Rush00

Staff 42 bocal@staff.42.fr

Summary: This document contains the subject for rush 00 of 42's Qt pool.

Contents

Chapter I

General instructions

Unless explicitely specified, the following rules will apply every day of this Piscine.

- This subject is the one and only trustable source. Don't trust any rumor.
- This subject can be updated up to one hour before the turn-in deadline.
- The assignments in a subject must be done in the given order. Later assignments won't be rated unless all the previous ones are perfectly executed.
- Be careful about the access rights of your files and folders.
- You must follow the `turn-in process` for each assignment. The url of your `GIT` repository for this day is available on your intranet.
- Your assignments will be evaluated by your Piscine peers.
- In addition to your peers evaluation, a program called the "Moulinette" will also evaluate your assignments. Fully automated, The Moulinette is tough and unforgiving in its evaluations. As a consequence, it is impossible to bargain your grade with it. Uphold the highest level of rigor to avoid unpleasant surprises.
- You must not leave in your turn-in repository any file other than the ones explicitly requested by the assignments.
- You have a question? Ask your left neighbor. Otherwise, try your luck with your right neighbor.
- Every technical answer you might need is available in the `mans` or in the Qt Documentation, which is available both in `QtCreator` IDE and on http://doc.qt.io.
- Remember to use the Piscine forum of your intranet and also Slack!
- You must read the examples thoroughly. They can reveal requirements that are not obvious in the assignment's description.
- Use the latest Qt version. Qt version 5.10 or newer is recommended.

- Many Qt classes are available as standard C++ classes. Qt classes should be preferred to standard classes, as you are supposed to learn Qt.
- Basic Qt coding style should be used, so read `http://wiki.qt.io/Qt_Coding_Style` before writing any assignments.
- Deprecated macros, functions or classes must not be used.
- Any build system, such as `qmake`, `cmake` or Qt Build System, can be used in the assignments. Instructions are based on `qmake`.
- Any editor or IDE, supporting Qt, can be used. However, `QtCreator` usage is strongly recommended.
- By Thor, by Odin! Use your brain!!!

Chapter II

Day-specific rules

- The UI must be based on Qt widgets. It is allowed to use HTML5, OpenGL or QML in widgets, but this is not required.
- Any Qt module can be used.
- 3rd party libraries may be used.

Chapter III

Foreword

Something clever should be written here.

Chapter IV

Subject

Your task is to implement yet another advanced dungeons and dragons game using Qt Widgets. To be more precise, you do not need a dungeon. However, you need some kind of a maze, where the player can move around, find precious treasures and fight against enemies. The maze can be based on tiles, for example. Some tiles represent paths, where the player can move, while other tiles represent buildings, trees or other obstacles, where the player cannot enter. A good selection of tiles and other game graphic assets can be found in
https://opengameart.org/content/side-scrolling-fantasy-themed-game-assets.
As the goal is to use Qt, drawing nice graphic assets yourself does not give you bonus score.

The goal of the game is to find a treasure - whatever that is - in the maze. You may decide, whether to show the whole maze to the player at the beginning or whether each tile is made visible, while the player enters the tile area or proximity. The maze size can be equal to a window size, which may be fixed. Scalable or scrollable maze gives you bonuses.

There should be enemies and collectable items in the maze. Enemies and items appear randomly in the random mazy location. Enemies try to destroy the player character. If they succeed, the player loses the game. Decide yourself, how enemies behave. Do they, for example, chase the player character or shoot against it. Decide also, how the character can defend: by shooting or escaping, for example.

Items are precious objects, what they player can collect and get score. An item may be also a bomb, which results losing the game. One item contains the treasure. Finding the treasure, results winning the game.

The essentials requirements for the game are:

- At least three separate forms: game settings form, maze form, high score form. No need to design forms with `Qt Designer`. They can be manually coded widgets.

- In the settings view, you must have at least a control to input player name, image, and game level, like rookie and advanced. The player must be able to take an image using the camera, if he or she has one or loading an image from the file. The level effects, how frequently enemies appear in the dungeon and how likely the treasure will be a bomb.

- The maze form must provide an area, where the player can move around. It may be a dungeon, park, city or whatever.

- The player can be moved using the keyboard, mouse or both.
- There must be areas in the maze, where player can go and areas, where the player cannot go.
- There must be randomly appearing enemies trying to destroy the player character.
- The player must have a mechanism to get rid of the enemies: shooting, escape, teleport, invisibility, for example.
- There must be collectable items, appearing randomly in the dungeon. Items may contain bombs or objects, increasing the score.
- The high score view shows player name, image, and score. The high score list must be persistent.

IV.1 Score

Rather than implementing the coolest YADD game in the world, the goal is to write a good Qt application. The following issues affect your score.

- Clear separation of UI and business logic
- Application structure
- Versatile use of Qt features
- Widget customisation and styling
- Use of model/view
- Responsive UIs, use of threading, if needed

If you have spare time to implement even more features, the following list provides ideas to get more bonuses. However, keep in mind that we'll consider them useless if you haven't done the other stuff before.

- Sound effects
- Scalable or scrollable play area
- 2-player mode over the network. Players may shoot each other or try to find the treasure together.