Requirements Documentation

Introduction

1.2 Scope of the Product:

**What type of platform (Windows, Macintosh, UNIX, etc.) must the software work with?**

Windows

**Will the software function as a standalone application on a given computer, or will it function over a network connection?**

Standalone application

**What other software, if any, must the software interact with? For example, you might be building a subsystem component that will be integrated into a larger system. In such a case, it’s important that you don’t duplicate functionality provided by existing subsystems.**

Not applicable

**If you are building a game, how many players will be supported? Will there be a computer player? If so, will it use artificial intelligence?**

Tetris - One player, no AI.

**What programming language will be used for the project?**

C++

**Will the software use a graphical interface or a command line interface? Note that these are also part of the software requirements, and should therefore be included in your requirements document.**

Graphical interface

1.4 References: The Tetris trade dress is owned by Tetris Holding. Licensed to The Tetris Company. Tetris ® & © 1985~2018 Tetris Holding.

General Description

2.1 Product Perspective: We are building Tetris the game for the challenge of creating a classic game. Although the game has been around for many years there is still a challenge to creating a game from nothing that you already know what the final product is. It is a useful piece of software because it can provide the user with hours of entertainment.

2.2 Product Functions: The product will create a user interface that allows the user to play the game. Once the interface is running, the user will be able to interact with the Tetris game by using the keyboard to position the generated piece.

2.3 User Characteristics: The end user will be anyone who might enjoy playing Tetris and have free time to play it.

2.5 Assumptions and Dependencies: Windows, 2 GB RAM, 1 GB HDD/SSD, Dual core processor.

Specific Requirements

1.0 Get code on Qt framework running.

* 1. Make sure some test code compiles and runs after implementing some C++, QML, and JavaScript so we know we have all necessary dependencies.
  2. Implement all the front-end types (QML).
  3. Implement backend logic (C++).
  4. “Connect” front-end with back-end.
  5. Implement code to listen for user input.
  6. Send data from user input to back-end.
  7. Perform necessary functionality in back-end using data received.