Game Developer test for King (C++)

Purpose

The goal is to provide a relevant, fun and inspirational foundation for a technical conversation in an upcoming interview. The submission itself together with the interview will provide the reviewer with an understanding of the applicants coding style and skills.

At least the features specified in this document shall be error free when used correctly, how everything else is handled is subject to creativity and ambition.

We will evaluate your code, your game and your delivery.

Game Concept

- The game consists of a 8x8 grid with different colored objects
- The objects can swap place as long as they are positioned next to each other and on the same row or column (up, down, left, right)
- If a switch result in three or more object of the same color in a row or column, these objects are removed from the game
- If a switch does not result in a match, the objects are automatically switched back
- When objects are removed from the game, the game board collapses so that object positioned above the removed ones falls down to take their place and new objects falls in from the top to fill the board.

Facts & Features

- 1 minute long
- 5 colours
- User can drag or click objects to swap them (both dragging and clicking should work)
- Use a game like Midas Miner for reference: http://www.royalgames.com/games/puzzle-games/miner-speed/

Tools & Frameworks

- You should develop the game in C++.
- If you're using Visual Studio (2012 and later versions), you can use the predefined project in this archive to help you creating the game. It provides a simple game engine with a small interface to display images and text on screen.
- If you're using a Mac or if you want to create the project from scratch you can and should use libSDL2 and its official extensions to create your game.
  The library itself is found at http://www.libsdl.org/.
- Using STL is ok no need to re-invent the wheel.
- If you believe that you can create a better game with an engine you wrote yourself – you are free to do so. Note! It's not ok to claim that you've developed Cocos2DX or Unity3D :)
- Please remember that we want to be able to test your game without having to install a bunch of obscure libraries...

Evaluation

- We will look at your source code to evaluate
  - Code structure and design choices
  - Usage of language specific features
  - Readability and reusability
  - Performance & memory considerations
- Evaluate how the test performs against the specification

- Evaluate the ”feeling” when playing the game i.e. Attention to transitions, animations and other details that might enhance the user experience

- Look at the delivery itself with regards to structure, size, and whether or not it runs out of the box.

Submitting

- Please send us a .zip archive or a link to where we can download one. It should contain your source code as well as an executable that runs out of the box on Windows or OSX.

- Visual Studio or XCode project files (or a way to generate them) are a welcome but not required addition to your submission.

- A readme.txt file to explain any architectural/design decisions is helpful.

Timeframe

- Please do your best to return the test within 7 days from receiving it, and if you need more time please send us a notification email.