Disney Streaming Services - Connected Device Native Client Engineering

Interview Candidate Assignment

Exercise #1: Take-Home Coding Assignment

Welcome to Disney Streaming Services,

We'd like to have you code an implementation of a common use case in the Disney Streaming Services Connected Device Engineering group. Please use the following technologies to complete the exercise.

1. C/C++ Codebase
    - You may use any version of C++, but be prepared to talk to the differences/features in more modern releases if you are familiar with them.
    - You may use 3rd party libraries or frameworks. You will need to include these libraries in source or object code form so we can compile your exercise locally on our machines. 3rd party libraries should be used as a *compliment* to the code that you write, and to speed work like drawing to screen or rendering UI.
    - For UI frameworks, please use an OpenGL style library for rendering (Examples, including but not limited to: SDL, GLFW, GLUT or Cocos2D). Do NOT use Qt or a similar library to render the UI. Our team is building a custom rendering engine, so showcasing your ability to contribute to that effort will be important.

Although the assignment is fairly limited in scope, treat it as a feature in a larger, real-world application, where architecture, design patterns and optimized code are important. Quality is more important than quantity, so good architectural and coding decisions are more important than completing any of the extra credit tasks. Once the assignment is complete, we'll review your work to assess the technical decisions made, and the completeness and quality of the work.

Our Native Client engineering teams focus on writing code on devices ranging from game consoles to very low powered set-top boxes. Be prepared to show and talk to techniques for writing optimized code in very CPU and memory constrained environments. Think 3000 DMIPS processors with very slow GPU's and limited memory bandwidth.

Here's the assignment:

1. Review the requirements below and ask any questions you have.
2. Then, provide an estimate of time to completion. Please estimate both total hours, and the projected completion date
    - Our intent is not to take too much of your time, so try to scope a solution that can be completed with a reasonable amount of effort
    - In the interview, we can talk about anything you might do differently if given more time

3. Code the example. Feel free to ask any additional questions during this phase.
4. Provide the code for review, either on a code repository (GitHub) or by email, DropBox, etc.
5. Provide instructions for running the app and include:
    - Instructions for building and compiling the source on Mac OS X or Windows
    - A compiled binary executable runnable on Mac OS X or Windows

Details

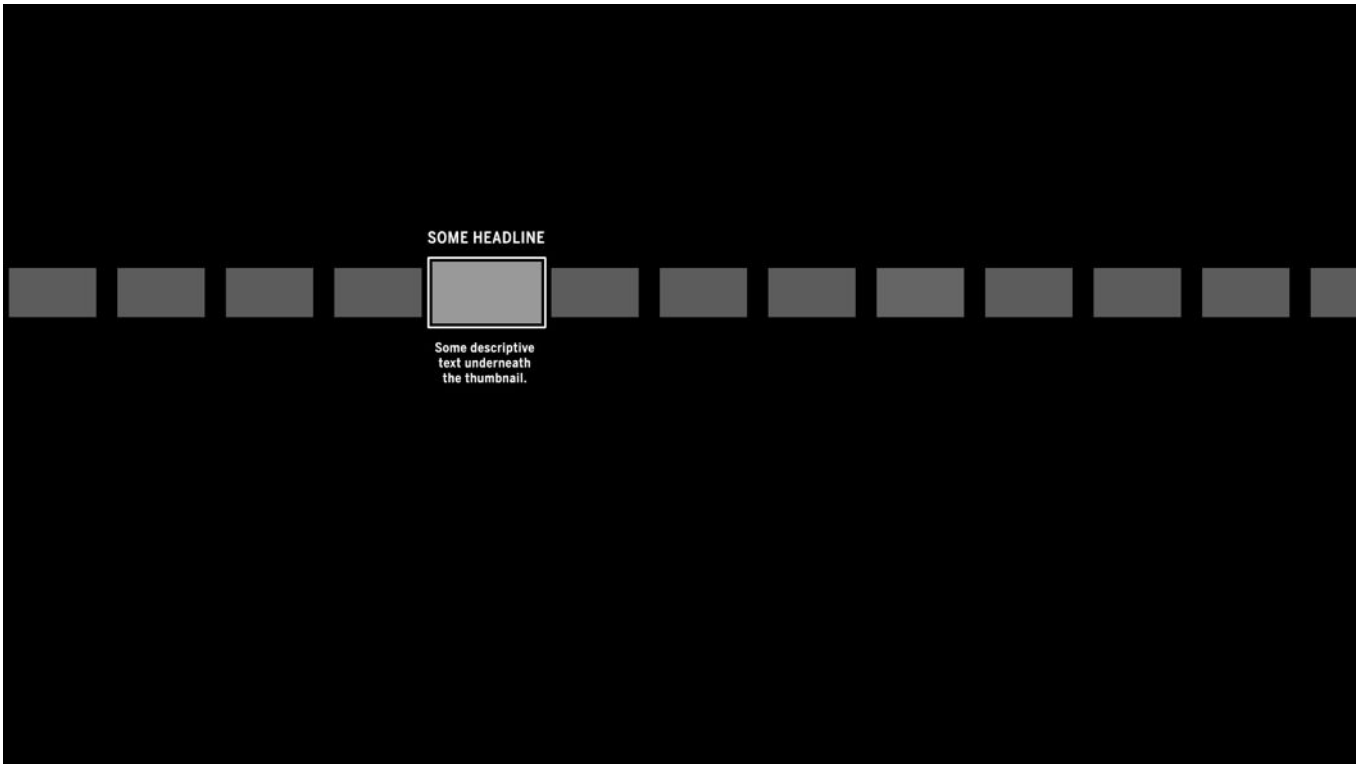
Parse out the data for each game in the JSON feed below, and construct a simple list that the user can navigate through with the keyboard (or remote/controller), left and right. Assume that a mouse will not be used for input. You'll find a number of MLB editorial recap content photo's, you may choose a size you deem appropriate for the design and this exercise.

http://statsapi.mlb.com/api/v1/schedule?hydrate=game(content(editorial(recap))),decisions&date=2018-06-10&sportId=1

Note that the above URL represents data for an example date. Feel free to code the example using a different date, or perhaps today's date.

When a game item is focused:

1. The element should scale up by 150% as shown in the wireframe
2. Display some descriptive metadata above and below the thumbnail



Use this as a background image:

http://mlb.mlb.com/mlb/images/devices/ballpark/1920x1080/1.jpg

For extra credit:

1. Instead of using the hard-coded JSON feed URL, make the URL dynamic to always show today's games
2. Support loading and displaying the http://statsapi.mlb.com/api/v1/schedule for adjacent days
3. Incorporate transitions, animations or any other visual aesthetics
4. Display a details screen or overlay when selecting an item with the Enter key or OK button on a remote