Recruitment Test for Position:

CTO: Unity3D Senior Developer

The following test has been designed to measure 5 different components:

- Advanced software engineering concepts such as object oriented programming, MVVM, inheritance, interfaces, encapsulation, generics, delegates, multi-threading, unit testing etc.
- Attention to details in general and code quality (Backend and front-end)
- Comprehension and retention of received instructions by email
- Ability to learn new concepts
- Your management skills showing you are able to take decisions, ask when needed and plan your daily tasks

Please read this test completely before you start.
Follow all instructions described in the 5 points below.

Good luck!

---

1. **DISC test:** Please take this Personal Strengths Profile test and send your result to carolina@liv.it as soon as you complete it. Please type as well your own results below.

DISC Values

Decisive	**42**
Interactive	**53**
Stability	**63**
Cautious	**77**
Aesthetics	**35**
Economic	**25**
Individualistic	**55**
Political	**73**
Altruistic	**45**
Regulatory	**50**
Theoretical	**70**

2. **Development task:** We would like you to develop a small, clean and very organized Point n' Click Game with Unity3D using C#. Keep it simple and short, but show your strengths.

The game needs to have these features:

- Inventory System
- Save and Load

The game needs to follow these requirements:

- MVVM or MVC architecture
- Scalable Level design using XML as model

Provide below a brief and clear explanation (max 200 words) of how you solved this task, you can reference your code comments, the architecture, components and sources you used etc.

I wanted to display a wide range of programming methodologies, tool development and steps I learned while developing applications. I have written a Game Design document to help me in the process of developing the game.

All required features and requirements are implemented in addition to build a Maya Level Exporter (screenshots included in Documents folder on git) that can export all level information in XML so it can be read by the game and loaded on the fly inside Unity.

The code structure allows for reuse and extensibility of the game. No external code was used to demonstrate coding capabilities without the use of external libraries and every code line was created during my deadline.

3. **Bonus Task:** If you have more time before the deadline, complete as many as possible of the following 3 small tasks:
   - A unit test method using NUnit as a unit-testing framework.
   - Make your application able to run in webplayer and Android/iOS

[Click to type your response]

4. **Delivery:** Please upload your client application to a GitHub repository. Provide an easy way to follow the commits tree by creating clear commit messages. Make sure the application compiles and includes all references and external modules.

Link to repository:

https://github.com/joeespindola/labsterdemo

5. **Questions**: When working independently, it is good to ask things you want to clarify before making mistakes as well as to take decisions when we are sure about the answer. If you have any questions or need clarification, **you are always welcome to ask** Aditya Yuwono (aditya@labster.com) and Michael Bodekaer (michael@labster.com).