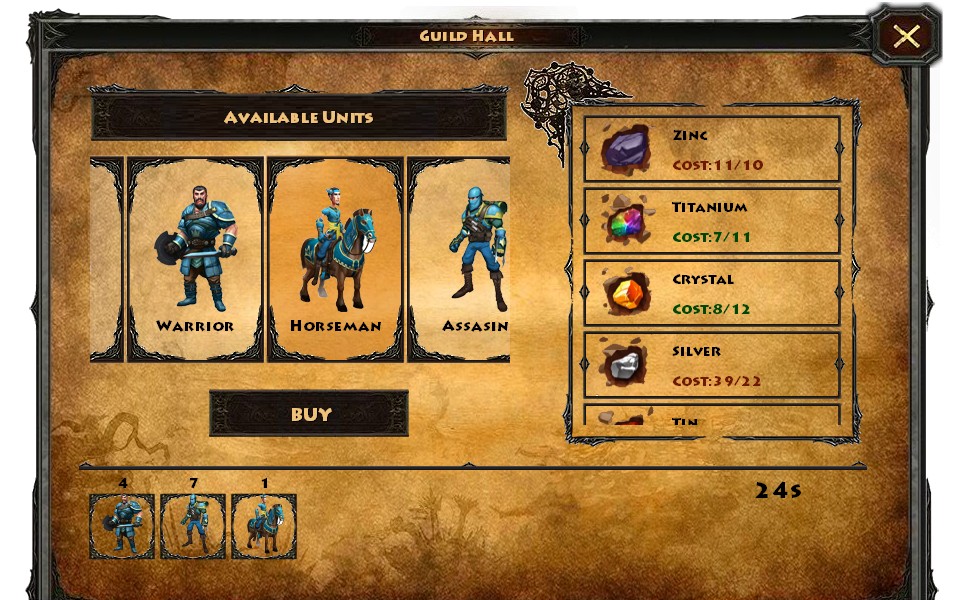
**Flash Developer Test**

– simple in-game market menu –

**Build a simple in-game store menu, based on the provided sample.**



**The resulting menu should function based on the following rules:**

1. You can open the menu by clicking a button (pop-up behavior).  
   You can close the menu by clicking the close button in the upper right corner.
2. On the left side there is the units list.   
   The list is entirely configurable from an external source, such as a XML file.
3. If there are more units than fits into the allocated space, the list becomes scrollable (horizontally).  
   The user should be able to scroll the list directly by performing a press-and-drag action, using the list itself as the interactive area.  
   Visual indicators for making this clear for the user (such as a scroll-bar) are a plus.
4. When selecting an item from the left the cost list is populated (configurable the same way as the units list) and the **unit's description\*** next to it. This list is also scrollable if necessary, but vertically.   
   Visual indicators for the costs (color, numbers) are required.
5. If the resources needed to buy an unit are available, the buy button is active and the user can buy the unit. After buying the unit, the cost is deducted from the user's resources, and the unit is added to the progress list in the bottom.
6. Each item in the bottom list should display the total number of items of the same type rather than be added as a duplicate. The bottom list is also scrollable when needed.  
   Items should also be removable one-by-one. A clear visual indication of this is a plus.
7. Each time the user adds or removes an item to/from the progress list, the user's resources get updated and the total manufacturing time for the all the bought items (bottom right) is displayed.
8. Add a box displaying the **available resources\*** (the player's current resources, taking into account the current purchases).

**Note**: \*This is a creative requirement, so the provided screen-shot above does not contain suggestions.

**Technical Requirements:**

1. Use Object Oriented Programming approach.  
   Reusing classes for interactive objects of the same type is a plus.
2. The targeted client is a mobile device (touch compatible).   
   Make sure the user can easily access any interactive object (always think about the ease-of-use).  
   Mouse events should be used instead of touch events, to simplify this requirement, as the test will run on a desktop computer only.
3. Reuse existing graphic assets to create new ones if needed, trying to keep close to the existing style.
4. Build your own components and avoid using 3rd party libraries.   
   Animation engines make an exception to this rule.
5. The XML file should contain the list of the units (with their names, IDs, image paths and a list of costs with resource IDs and value) and the list of spendable resources (with their names, image paths and available amount).  
   Adding more units or resources to the store should be possible without having to recompile the project.
6. Avoid navigation bugs and overlapping issues at all costs. An application without bugs and less features is better than buggy application with all the required features.  
   Lists should be able to contain more items than can be fitted on the screen.

**Additional requirements:**

1. The lists navigation should be fitted with smooth inertial movement animation.
2. All buttons and other interactive objects should present different states to emphasize the action the user is performing or can perform (such as a pressed state).
3. Animations can improve the ability to redirect the user's attention if used right. Use them whenever you think is necessary.

**Targeted platforms:** Windows, OSX or web application.

**Allocated time:** One week.