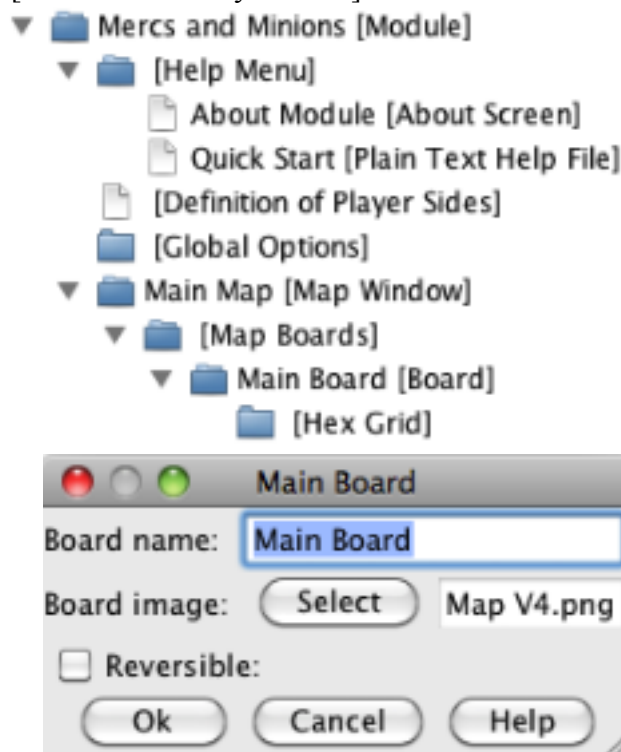


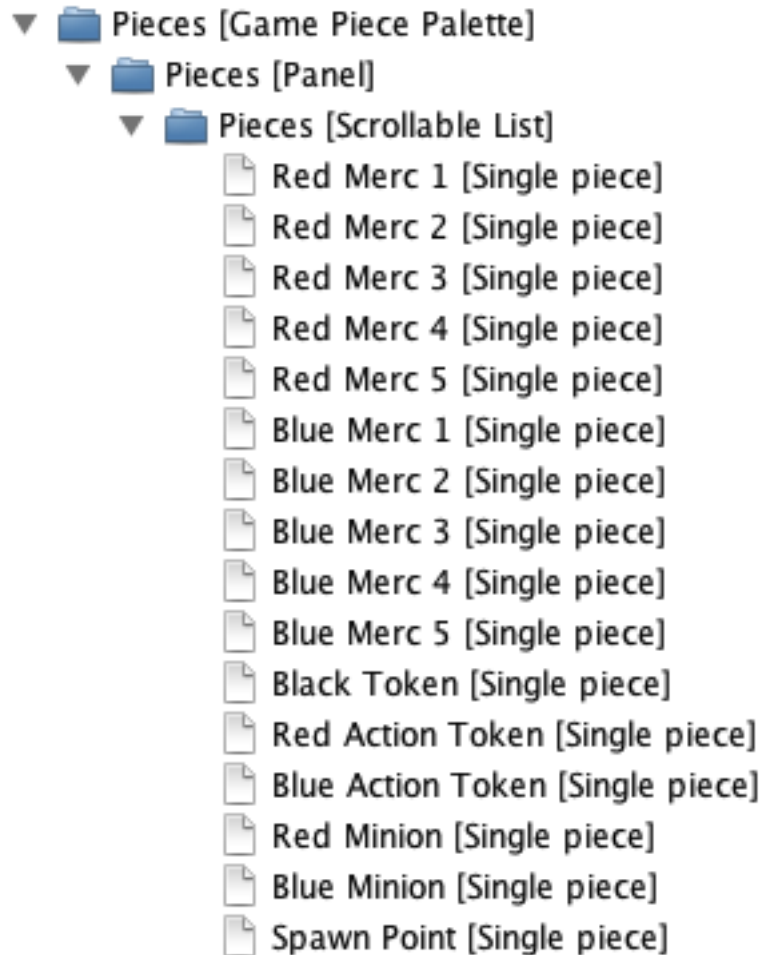
Vassal is an open-source program written in Java for the purpose of digitally implementing board games, primarily for online play with other people. This is done by the creation of Vassal ‘modules’ which represent different games. Over the course of the semester, I have attempted to learn how to use Vassal and determine whether or not it is an effective tool for making and testing prototypes of board games I’m designing. The first problem I ran into is that the tutorials found on the Vassal web site¹ are out of date and only teach you how to do a few very specific things. While there is a reference manual built into Vassal, it is often difficult to learn exactly what effects different properties and commands will have, or even what elements of the module will be affected by what commands and when. The best (and sometimes only) way to learn how to accomplish something is to either reverse-engineer another module or to experiment with different commands and see what happens.

I created a module for a game I am in the process of designing called Mercs and Minions. Modules in Vassal are created using the Vassal editor, which essentially involves creating lots of elements of various different types and changing their properties. The Mercs and Minions module has a number of elements put into it automatically, some of which I added more elements to after the fact. The ‘[Help Menu]’ element was created by the Vassal editor, as was the ‘[Global Options]’ element. I left both of these alone. The ‘Main Map [Map Window]’ element was also already present, but I defined what image it would use as well as adding a hex grid onto it. I also modified the ‘[Definition of Player Sides]’ element to define 2 sides, Red and Blue.



¹ <http://www.vassalengine.org/wiki/Tutorials>

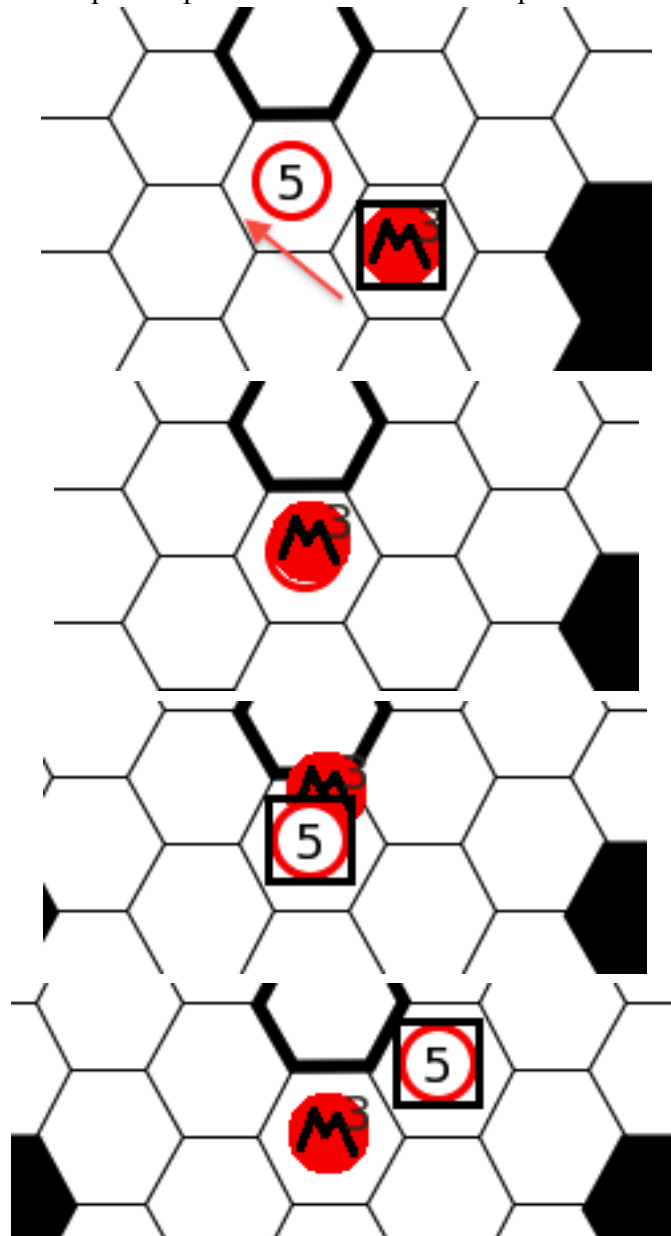
There were numerous other miscellaneous elements that I did not modify. The other major element which I did change was the '[Game Piece Palette]' element, which is where I added all of the various different pieces that Mercs and Minions requires: 5 Merc tokens per team, Black Tokens for tracking miscellaneous information, Action Tokens belonging to the two teams, Minions, and Spawn Points.



Each of these piece definitions allows any number of that kind of piece to be created on any board once the module is launched. The creation of pieces and boards is covered in the Vassal tutorials, so this process was fairly straightforward. The difficulties arose when I needed to give the pieces various different properties. Most of the names of the different properties are fairly self-explanatory. For instance, any piece with the 'invisible' property can be made invisible or visible with the press of a hotkey. I had to manually give each piece the 'delete' property if I wanted to be able to get rid of it once it was placed, which surprised me. Of the other three properties I used on pieces, one of them was described in the tutorial and two were not. The 'layer' property was described in the tutorial. Using this property allowed me to track the health of minions and change the color of spawn locations. The other two properties I used, 'does not stack' and 'restricted access' both required some experimentation on my part to understand. I placed the 'restricted access' property on the action tokens, preventing the other team from moving or modifying those pieces. This means that a player who starts the game as the red

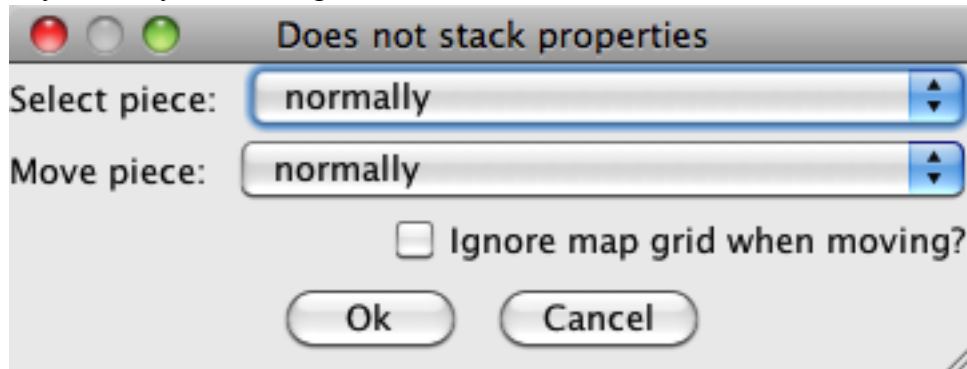
player is completely unable to even create a blue action token, much less move or modify existing blue action tokens. The other property I used, 'does not stack,' has to do with an interesting assumption about the behavior of Vassal pieces.

Vassal pieces, when placed on top of each other, automatically stack into a single group. Selecting a stack of pieces automatically selects all of the pieces of that stack, and you have to double click on the stack to split it up and select an individual piece.



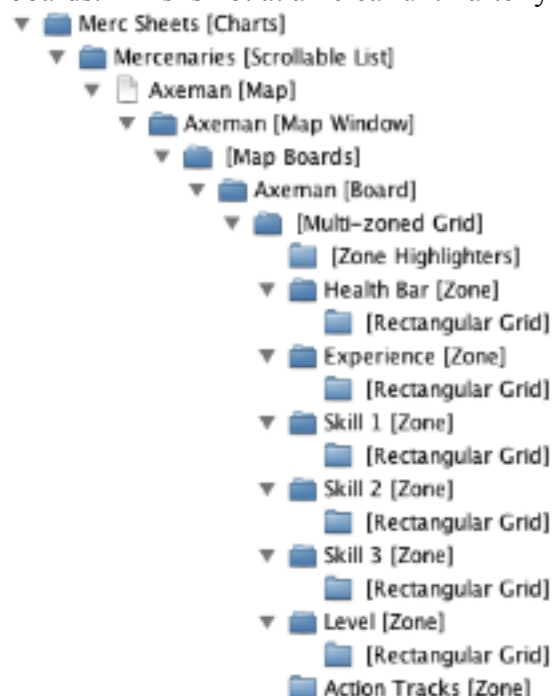
This is somewhat awkward for Mercs and Minions, since pieces frequently move onto and off of each other without stacking, so requiring extra effort to move pieces on and around each other was a little annoying. It does have the advantage of allowing you to select which piece you want to move, though. If there are two pieces of the same size placed on top of each other without stacking, it would be impossible to select the piece underneath. In the end, I decided to add the 'does not stack' property just to the spawn locations, since they are large enough to select

even if there is another piece on top of them. The really confusing this is what else the 'does not stack' property allows you to change.



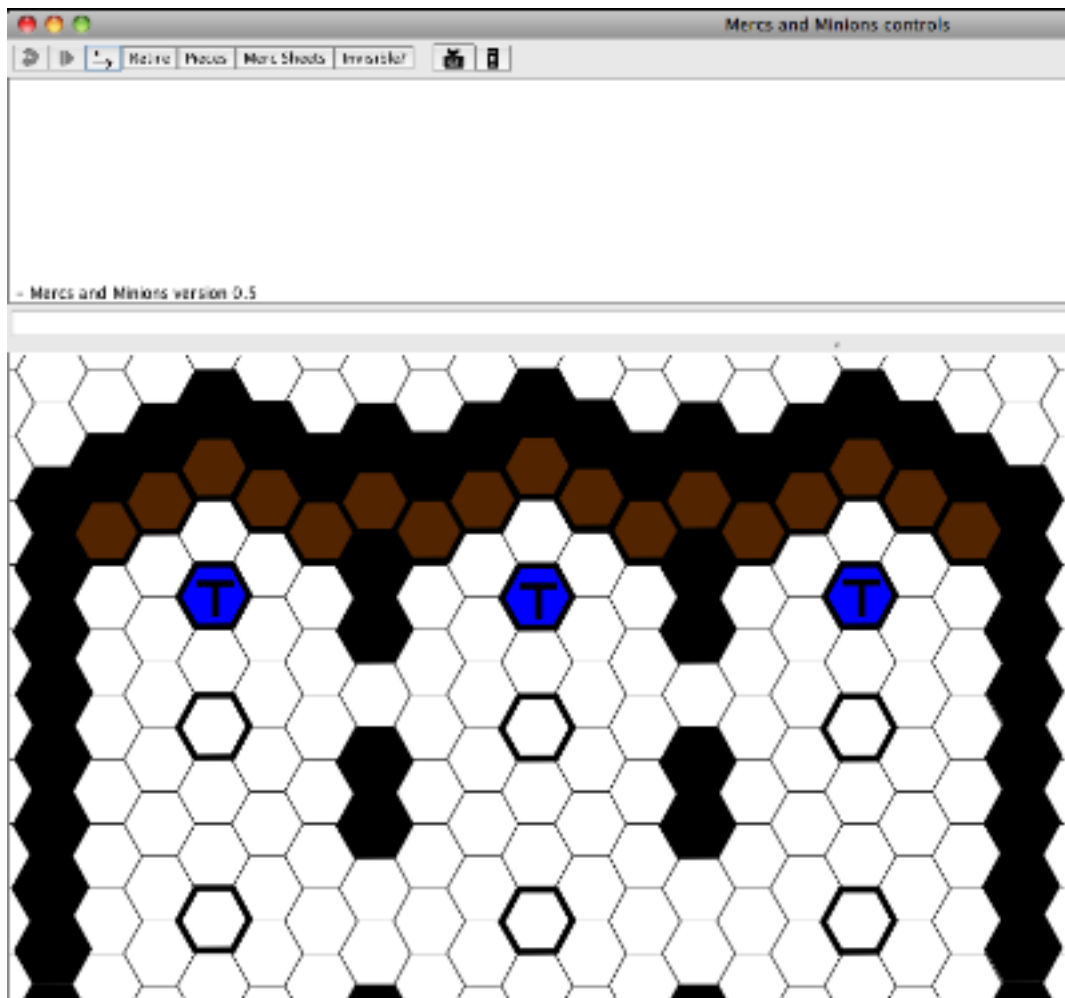
With the 'does not stack' property, you can in fact make it impossible to move or select that piece. Being able to select and move a piece seems very different than whether or not a piece stacks with other pieces, so why put these properties here? There are many things like this in Vassal which you have to learn simply by experience and which don't make any intuitive sense (at least, not to me).

Then there is the issue of boards. Mercs and Minions requires a series of different 'merc sheets' which store information about the different mercenaries in play as well as what actions they are going to take. Since these sheets are going to have pieces placed on them, I wanted to define them as boards. In addition, because there are so many of them, I wanted them to be contained in a scrollable list. Placing a board in a list is something that Vassal is not really set up to do. You can create map windows, but each such window creates a button on the toolbar. So, to create a different window for each mercenary would fill the toolbar with buttons. This is not an ideal solution. As it turns out, creating a 'charts' window allows you to create a list in that window which can contain boards. This is not at all clear until after you've already seen it done.

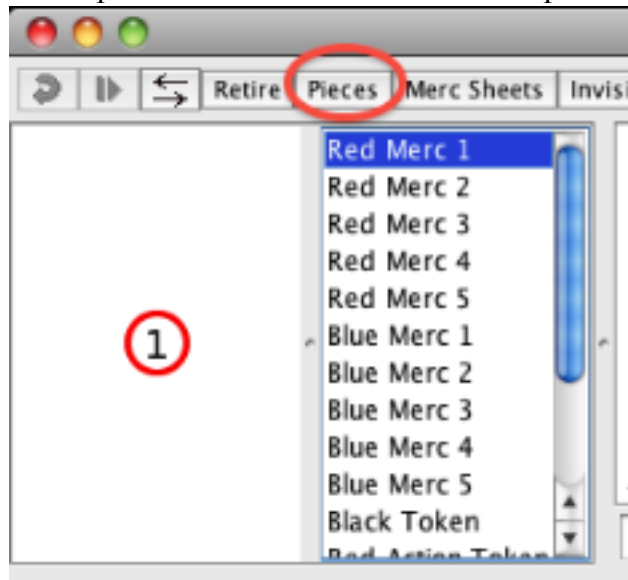


The last two elements that I added to the module were an '[Action Button]' and a '[Global Key Command]'. My purpose in adding these elements was to create a button which would make all of a given team's action tokens either visible or invisible. The invisible property on the action tokens allows them to be made visible or invisible by pressing CTRL+I. However, this only triggers if that piece is selected. While multiple pieces can be selected at once, it is awkward to select large numbers of pieces on different boards in order to turn them all invisible. I first tried to solve this problem with just an action button. However, Vassal's reference manual was very vague about the capacity of an action button, stating simply that one of its properties was the ability to send a hotkey, "to other components." When I created a button which supposedly sent the hotkey CTRL+I to other components, it did not trigger the invisibility action. So, I created a 'global key command' which sends the CTRL+I hotkey to all elements with the 'invisible' property. I then created an 'action button' which triggers that global key command, thus allowing the button to cause all of the pieces with 'invisible' property to toggle their visibility.

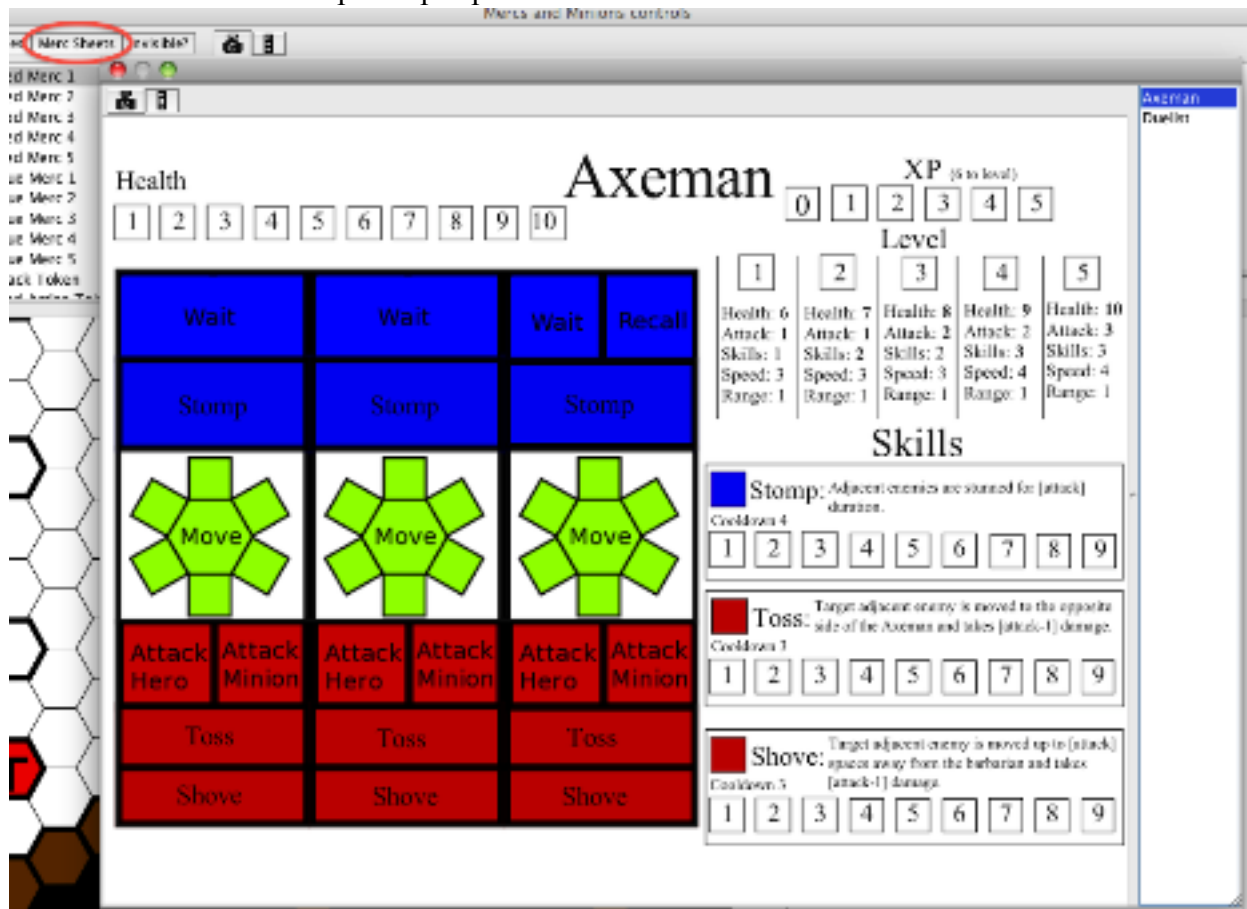
Actually playing the module is fairly straightforward. Once you've launched Vassal and opened the module, you need to go to the file menu and create a new game. You will then be asked to choose a side. Pick either Red or Blue and 'Finish'. You should then see a screen that looks like this:



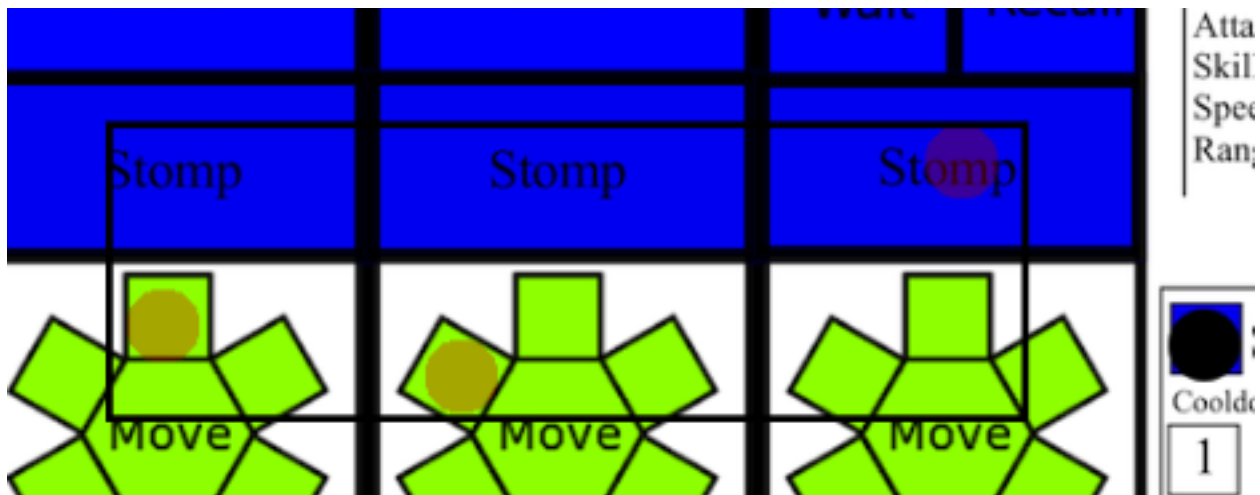
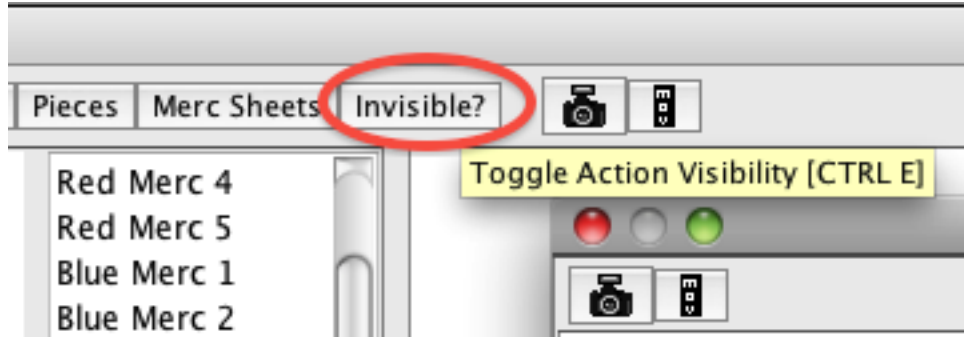
This is the main map window and is always open. By clicking on the 'pieces' button at the top of the screen, you can open a menu with a list of all the pieces I've defined.



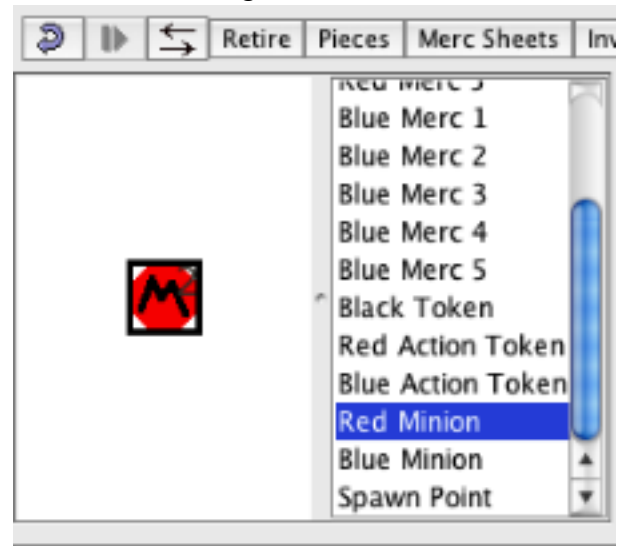
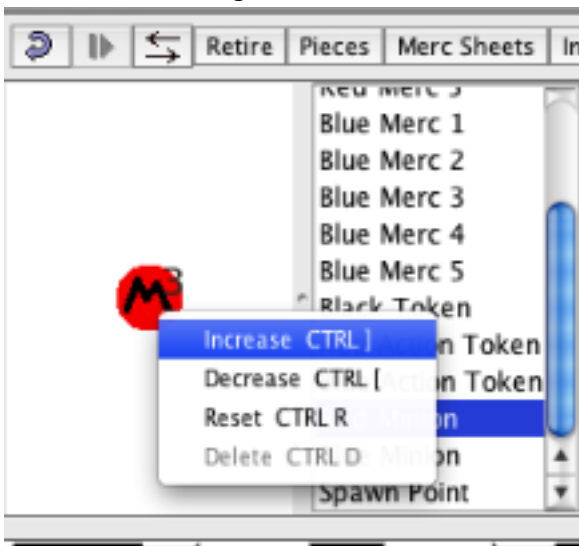
Selecting a piece from the list allows you to see an image of that piece on the left. You can click and drag that image to the board to create one of those pieces on the board. Clicking on the 'merc sheets' button opens up separate window with a list of all of the different mercenaries.



Each of these mercenary sheets is a board which may have pieces dragged to it from the pieces palette. If you drag an action token of your team's color onto any of these boards, you can turn it visible or invisible by selecting it and pressing CTRL+I. You may even make the piece invisible before dragging it off of the pieces palette. Pressing either CTRL+E or the 'invisible?' button, you can turn all of your team's action tokens visible or invisible at once.



Minion and spawn location pieces all have the 'layer' property. You can cycle between layers by selecting one of these pieces and pressing CTRL+[or CTRL+]. In the case of spawn locations, this changes their color. In the case of minions, this changes their health.



It is possible to play a module off-line on a single computer, but Mercs and Minions cannot effectively be played that way. In order to play on a single computer, you need to press the 'retire' button, which allows you to leave your current side and rejoin the game as another side. Unfortunately, the 'invisibility' property makes the token in question invisible to other players, not other sides. So even if you switch to the other side, you will still be able to see the other team's invisible tokens. Playing with multiple computer's using Vassal's server, however, does not have that difficulty.

On the whole, I think Vassal is a very useful program. Their tutorials, help menus, and documentation are significantly lacking, however, which makes learning how to design new modules very difficult. In terms of prototyping and playtesting, I can see Vassal being extremely useful for getting people to playtest digitally from all over the world. However, while playing in Vassal is a great test of some of the rules and mechanics, it cannot tell you what the feel of actually playing the physical game is like. In Vassal, you could move seventeen pieces at once very easily, while in a physical game that would be much harder. At the same time, with a physical game it is very easy to look at what your partners are doing while you're making a decision and quickly act on that information. In Vassal, that takes much more time with window-switching, scrolling through lists, and waiting for the internet to sync with the other player's actions. I can definitely see myself using Vassal in the future, but it certainly won't replace physical prototypes.