Dominic Tippabattuni, Ovidiu Mura **Total:** **93** / 100

CS420/520 Object-oriented Programming
Assignment 5: A Morphic User-interface

	Criteria	Comments	Pts
Mechanics /6	Monticello version can be loaded. Instructions on how to run the code included in commit message or email		6
Tests /10	You have included tests, and the tests run green!		10
Method-level design /8	Method are composed from smaller methods with intention-revealing names (Beck p.21)		8
No Magic Numbers /8	Methods doing morphic layout use a minimum of magic numbers.		8
Morphs move together /10	If I move the outermost morph of your GUI across the screen, its submorphs move with it, and stay in the same relative positions.		10
Morphs resize together /6	If I resize the outermost morph of your GUI its submorphs resize with it.	I can't resize (or even select) PokerGameView, because you've overridden perform:withArguments:. If I remove this method, then resizing works properly.	3
Model-View Separation /8	The model of the game is separate from and independent of the GUI (the view.) They communicate using the observer pattern.		8

Naming /8	Classes and methods are named to be meaningful to their *clients*. Names follow Smalltalk's capitalization conventions.		8
Only once /6	Common code is factored out and reused, not copied.		6
Once /6	Everything important about the GUI is captured in your code.		6
Method Comments /5	If the action of a method is not clear from its name and parameters, see if you can fix them. Only then add a method comment		5
Blindingly obvious code /5	Clear code and class comments are all that you need		5
Methods are classified /6	No '*as yet unclassified*' methods. Methods are in appropriate protocols (look at other implementors of that message)	The "add" methods in the "user interface" protocol would be better put in "initialize" or maybe "drawing".	4
Quality /8	The ineffable	Nice work overall, but be careful when you override system methods!	6