# Functional Test Design
for
Multi-user Infinite Canvas Thingy (MICT)

Don Huckle, Ben Kaplan, Mark Wyrzykowski, Rob Wiesler

November 8, 2010

Contents

1 Client

Use Case	Initial System State	Input	Expected Output
Canvas.Connect	User starts off not connected to a server	User enters a server and login credentials, then clicks connect	User is connected to a server
Canvas.View	User does not see the canvas	User requests a section of the canvas to view	The user sees that section of the canvas
Canvas.Pan	User sees one section of the canvas	User selects the Pan tool, clicks in the canvas, and moves the mouse	The canvs moves so that the section of the canvas under the mouse remains the same point
Canvas.Jump	User is at one location on the canvas	User enters a new location and clicks the “jump” button	User is moved to the new location
Canvas.ListUsers	Initial State doesn’t matter	User requests a list of users logged into the server	User is presented with a list of users on the server
Canvas.JumpToUser	User is at a certain location in the canvas	User selects another user and clicks “jump to user”	User’s location is set to the same location as the other user’s location
Canvas.Mark	User is at a position in the canvas	User clicks the “Mark” button	The position is saved for the user. Clicking the “jump to mark” button will take them back to that position
Canvas.Select	Client is connected to a server	User clicks the “Select” button	The coordinates of the selected area of canvas is saved to memory
Canvas.Select.Copy	Client is connected to a server and has selected an area of canvas	User clicks the “Copy” button	The selected area of canvas is saved to the clipboard
Canvas.Select.Paste	Client is connected to a server and has selected an area of canvas and has an image saved to the clipboard	User clicks the “Paste” button	The selected area of canvas is filled with the image in the clipboard
Canvas.Select.Rotate	Client is connected to a server and has selected an area of canvas	User clicks the “Rotate” button	The selected area of canvas is rotated 90 degrees on the canvas
Canvas.Select.Scale	Client is connected to a server and has selected an area of canvas	User clicks the “Scale” button, and fills in a field specifiying how much to scale by	The selected area of canvas is scaled by the specified amount
Canvas.Select.Shear	Client is connected to a server and has selected an area of canvas	User clicks the “Shear” button, and fills in a field specifiying how much to shear by	The selected area of canvas is scaled by the specified amount
Canvas.Tool.Select	Client is connected to a server	User clicks a tool icon	The tool icon the mouse clicked on is made the active tool. The client state stores the active tool
Canvas.Tool.Draw	Client is connected to a server and a tool is designated as the active tool	User interacts with the canvas using the mouse in some way	The tool draws on the canvas, and the changes are propagated to the server and to the clients of all users viewing the affected area
Canvas.Undo	Client is connected to a server, and has changed the canvas in some way since connecting	User clicks the “Undo” button	the last action performed is undone, and the changes are propagated to the server and to the clients of all users viewing the affected area
Canvas.Redo	Client is connected to a server, and has undone a change made to the canvas since the last undone modification	User clicks the “Redo” button	the last action undone is redone, and the changes are propagated to the server and to the clients of all users viewing the affected area
User.Permissions.Check	Client is running	User checks his or her current or last known permission set on a server	The user’s current permission set, with group mask permissions included (but not marked as such)

2 Server

Use Case	Initial System State	Input	Expected Output
Server.Start	Server is not running	Admin starts the server with a configuration file as an optional parameter	Server starts with parameters identical to those in the configuration file
Server.Stop	Server is running	Admin stops the server, either from a superuser client or from the system console	The server stops
Server.MaxUsers.Set	(no initial conditions)	Admin either edits a configuration file or runs a command from a superuser client or the system console	The maximum number of users changes, and excess users are kicked from the server
Server.MaxUsers.Check	Server is running	Admin checks the current maximum nubmer of users by running a command from a superuser client or the system console	The maximum number of users
User.Permissions.Set	(no initial conditions)	Admin either edits a configuration file or runs a command from a superuser client or the system console	The user's permissions change, old restrictions are lifted, and new restrictions are effected
User.Permissions.Check	Server is running or client is running	Admin checks the current permission set for a user by running a command from a superuser client or the system console	The user's current permission set, with group mask permissions included and marked as such
Group.Permissions.Set	(no initial conditions)	Admin either edits a configuration file or runs a command from a superuser client or the system console	The user's permissions change, old restrictions are lifted, and new restrictions are effected
Group.Permissions.Check	Server is running	Admin checks the current permission set for a group by running a command from a superuser client or the system console	The groups's current permission set mask