Analysis of Evaluation

Strengths:

HagIt Praise	Analysis
Visibility and System Status: “It is clear how to switch modes easily after you learn how to do so with the rollerball.”	The switching of the modes (ink/color) is visible and thus well placed.
Match between real world / intuitive “Intuitive for anyone who has used any paint program. Understanding and using the commands is very intuitive that you porting their knowledge will be a breeze.”	In our design we tried to use existing standards, basing some of our interface design on existing paint programs. The evaluators were pleased with our accustomed design.
User Control and Freedom “ The system supports undo, redo and erase”	In our final system undo and redo will be fully implemented. There will be a history of 10 last undo functions which will ensure the user has good control and freedom in the program.
All menus and pop ups are consistent	We tried to make sure that our design is consistent trough out.
Flexibility and Efficiency “The program was pretty intuitive and didn’t take much time to learn!”	We were hoping that the learning time will be minimal and that the program will be intuitive to use; we are happy to receive positive feedback on this heuristic. However it’s been noted that the program is not highly efficient, at least not at the prototype stage. We are hoping this will improve with further development.

Weaknesses:

HagIt Critique	Analysis
There is no practical way to start a new canvas while the program is open.	There will be a practical way to do so; it just hasn’t been implemented yet. There is going to be a menu on top of the screen (File, edit and Help). In the Alpha system, new Canvas option will be part of the File menu (just as in our paper prototype)
It is still unclear how a user will be able to open a saved file.	Similarly, there will be an Open option in the File menu.
We can not figure out which is the right hand and which is the left hand status	That is an interesting observation. Since right (dominant) hand is using a mouse and left (non dominant) hand is operating a rollerball (or for

	now the keyboard), we are considering implementing cursor feedback. That is the cursor will change depending on its position and system mode. Also, only the dominant hand mouse has a cursor display, so when a user moves that mouse, they should immediately be able to discern which hand is controlling the cursor. Similarly for the trackball or non dominant mouse, when a user moves up or down, a different colour /pen tip/ etc is highlighted.
We are also unable to identify the current Nib selected and its colour.	While it hadn't been implemented in our prototype, in the Alpha version, the drawing cursor will show the current pen (or nib) tip as well as the current colour.
You can not select the color while you are in the Nib selection mode or vice versa. Which makes switching modes a hassle.	These 2 modes should not be combined. The pen will always be in black. This should be explained in detail in the user manual to assure there is no frustration while trying out the program.
It is unclear from the program that you have to use Rollerball to switch modes.	This is a very useful observation. Aside from clearly stating this in the user menu, there should definitely be some clue in our interface that lets users know to user Rollerblade to navigate between the modes. We understand that not knowing this may cause lots of frustration. We are considering implementing a start up diagram/note explaining the use of two mice in the program. We are hoping to include this in our Beta System.
Switch hands is not a necessary feature. As far as we understood in our tests we found out that there is no difference but mirroring the interface. If the mouse buttons are not switched this function should be removed.	This function is implemented for left handed users. The interface is preset for a right hand user, so that features/options requiring precise manipulations would be done with the dominant (right) hand. For the left hand user, the switch option is necessary as they need to perform high precision tasks with their left hand. The usefulness of this feature will become apparent when interfacing with 2 mice or with a mouse and a rollerball
The bottom tabs are not consistent in the sense that they keep on modifying depending on the button clicked. Eg. Ink mode has "Shapes" as its third menu and Color Mode has "Color" instead. Furthermore in the color chooser mode you	By design, the *Ink* mode has shapes and the *Color* mode has color. This is because as previously stated, our application is targeted for collaborative cartoon creating, where one person would only ink and the other only color. The colorist does not need to draw shapes,

choose the color with the mouse. While in nib chooser panel mode you have to choose it with Rollerball. It is hard to map such differences.	while the inker does not need to color. There seems to be a little bug in the prototype allowing user to select color in the ink mode. (This seems to be only possible if the color mode was switched to ink after while in the color mode of the rollerball menu). This bug will be fixed. The ink mode will have pen, shape, undo and zoom, while the color mode will have color, undo and zoom. Also as stated previously for the purposes of the alpha system, where collaborative aspect of the program will not be taken into consideration, ink and color modes will be grayed out and all 5 options will be combined on the rollerball menu.
The colour and shapes tabs at the bottom replace each other when we click the ink or the colour mode. The program doesn't give any feedback of this replacement of tabs that is a sound, highlight or a pop-up. A new user might wonder what changed when he clicked the ink or colour mode. He will realize this only after looking at the bottom.	This is a valid point. However if we fix the ink mode to have pen, zoom, shapes and undo, and the color mode to only have color, zoom and undo, we believe the change will be noticeable enough. Considering the user is familiar with the purpose of the program, we believe this feedback is sufficient. Notifying the user each time he switches the modes of the change does not seem very elegant, but rather annoying. Again as mentioned above this will only be an issue for the collaborative aspect.
Since color is not quantized (i.e. you can not enter the values manually) if you mistakenly change the color. You have no way to get back to the previous color. Also any user may assume that the bottom button-like tabs are mouse clickable. We are not sure if it is a matter of practice but for a first comer it is a frustrating problem.	The color menu has been completely redesigned in the alpha. We will also be adding a colour status code at the bottom of the colour screen which will give users the exact colour value so they can reuse it later on. Users will also be able to enter a specific RGB value there to get a desired colour. This should be implemented by our beta version. The mode selection on the bottom needs to be redesigned/improved too (as mentioned above) to avoid the frustration.
The system does force you to memorize how to select colors! In a normal HVS color selection model you have 2 tools to select from. One of them is for the Hue, the other is for Saturation and Value. In your chooser you only have one! This makes us to guess the value of the Hue.	The color menu will be completely redesigned

Yes it is easy to paint a few shapes. It's just that we don't have any feedback regarding the size of the shapes when they are being drawn. For example there should a dotted line showing the size and placement of the shape when we drag the mouse to make one; like in paint. Maybe you intend to do this but haven't implemented it yet, so we hope to see this feedback in future.	The shape menu definitely needs to be improved to make sure there is feedback regarding the size and position of the shape. The rubber band effect should be implemented by either alpha or beta which will show a dotted line while trying to make a shape.
We have timed ourselves drawing the same shapes with Microsoft's paint and using collaborative paint. The one handed paint was faster.	This is an interesting test and useful results. However considering that Microsoft paint is a complete program and that the user was proficient in it versus our incomplete prototype, the comparison was perhaps a bit premature. However, one conclusion that could be drawn is that Collaborative paint might take some getting used to in order to achieve comparable efficiency
Overall we did not find the program visually appealing for a couple of reasons.	We hope this opinion will change once the program is more complete.
No help menu were implemented	The help menu should appear in the top menu and will be partially implemented in the alpha system.

General Suggestions and Improvements

HagIt Suggestions	**Analysis**
Undo is an action and not a tab. The undo tab should be renamed to History tab and there should be a button that undo's one change.	That is an interesting suggestion. However we do not fully agree. In the later versions we are indeed planning to implement history of undos, which will permit redos. However we do not think it's beneficial to rename undo into history as it will confuse the user when he/she will wish to undo something, since he might not associate pressing history to undo. We did however add an undo function to the edit menu which also works with the normal undo shortcut of Control- Z. This should be useful for more advanced users.
The collaborative paint team should have specified an existing system for comparison with their system, for example, Microsoft paint or flash. And the team should also have done a lab experiment with that system. This way we could compare the collaborative paint with that program. We ourselves had to use an experiment with paint for comparison.	It is a good suggestion and we are glad such test was run, however as explained above, we felt like it may have been too early to compare. In addition while the system is somewhat similar to existing paint programs it targets different consumer (cartoonists) and pursues different goals than existing paint software.
The user wants to get exactly the same colour he used some time before in the sketch. You program should have a function that would allow the user to select exactly the same colour he used before. For this you can either keep a history of previous colours used or simply the user can click on the portion of the sketch where the colour was used, and then get the same colour back(just like in Microsoft paint).	This is a very good suggestion and has actually been discussed by the Collaborative Paint team at the time of paper prototype. It was decided to have a certain number of previously used colors displayed (and dynamically changing) in the color menu. However as this is a more advanced feature it has not been implemented yet and may not be implemented until the Beta System.
You must quantize the HSV values	As mentioned above, the color menu will be completely redesigned. It will resemble MS Paint color selection, however for each of the main colors it will be possible to change color We are also considering to add a separate method for color selection where the user can input the RGB color value and obtain the desired color this way.
If both mouse buttons will provide the	Each input device (ideally a mouse and a roller

same function in the final system then remove it otherwise you should state clearly the state the user is in using either a viewable option or a reachable hidden one	ball) has unique functions. They are not interchangeable. We understand that we need more feedback as well as documentation to clarify the functions of each device.
Add a quick color switching toolbar that is visible and easily accessible when you are painting	Color menu will be complete redesigned.
The program should be more consistent from the sense that some buttons are clickable others are not! The non clickable items should be accessed with a rollerball! If they are visible and easily accessible with a mouse I do not see reason why they should not be clickable with the mouse.	This is a very good point. We will try to modify the right hand side buttons to appear more clickable and hence separate their shape from the rollerball buttons. In addition, as mentioned above, we will implement cursor feedback to further guide the user.
The bottom tabs should have all the possible values to help the user develop muscle memory. That means remove color and ink mode, and add a color tab in the bottom. This will decrease the number of buttons by one and still have the same functionality. This will also eliminate the feedback problem stated earlier.	As previously mentioned, we can not remove ink/color selection on top since it was designed to facilitate collaborative work of inker and colorer. Also as mentioned above, these options will be greyed out and all options will be displayed on the bottom.
The HSV is usually a two selection model. One from a xy grid and the other is linear. You correctly implement it but do not give a clue to the user of the Hue value (usually the linear one). The user will be forced to remember where certain colors like different shades of green fall on the HSV table. You should add another linear Hue selection bar.	The color selection will be redesigned (see above)
The lack of efficiency is due to the fact that we have to keep on switching between modes to select different tools. More tools should be viewable in each mode. There are some suggestions earlier that will facilitate that e.g. add a quick color selection toolbar.	We agree that the design itself is not very intuitive. We hope that by fixing some of the things described above, the program will seem more intuitive, at least with some practice. We also have to keep in mind that it was originally designed for collaborative use of cartoonists: a colorist and an inker. In such a context we believe the program might seem more intuitive.

The design is not very pleasing to the eye due to the fact that the GUI does not make efficient use of iconic buttons in the toolbars. The fix is simple use iconic buttons.	This is definitely something to consider. We might implement iconic buttons further in our development.
I don't think the help is necessary for this program though it might be useful.	Minimal help shall be available in the help menu.
You should also allow the cursor to change when in different modes. The cursor should depict the color and the nib style.	That is a good idea and will provide good feedback. For example we could try to distinguish between mouse cursor and rollerball. Some of this functionality shall be implemented in the alpha system
Though it might be an implementation specific problem I think you guys should use the rubber band effect when drawing shapes.	This actually will be implemented in the alpha/beta system.

We would like to thank HagIt team for their extremely valuable evaluation. They brought up certain important aspects that were new to us and made us rethink parts of our design.