Cognitive Walkthrough:

Software Interface:

✓ To draw something with the software interface involves minimal steps, all of which is contained on one page, which is numbered from top to bottom so it is easy to follow.
✓ Only two steps are required to plot the image using the Graffiti Robot.
✓ First the user opens the image they would like to draw.
✓ Then they hit the “Go” button, which vectorizes the image and sends information to the robot hardware.
✓ No more interaction is required from the user, and when the robot is done painting the image a nice confirmation message will pop up.
✓ If the robot is unable to finish the design for some reason the program will say so and give information on how to fix it.

Hardware Interface:

✓ Drawing with the hardware interface uses a two button controller with a directional pad.
✓ This method uses a tried and true way of user input that has been around for many years.
✓ Pressing up on the directional pad will make the spray can move up, pressing left will make the can move left, and so forth. The two buttons on the controller are for spraying the different colors. Button A will be used for spraying color A and button B will be used for spraying color B.
✓ The buttons will be labeled on the controller so the wrong color isn’t accidentally sprayed, and since there are only two buttons, this implementation has a very low learning curve.

General:

✓ Anything can be drawn with either the software or hardware interface (assuming the input is a picture), but it is a given that the software would probably be faster, letting the computer do the work for you.
✓ The robot itself will be very mobile and scalable up to a point. The user can resize the vertical supports and also move the supports closer together, allowing for a wide range of sizes. From tall and narrow to short and wide, the user will pretty much be able to do whatever they want.
✓ When the user is done using the robot, it will be a matter of taking down the chain, quick connecting the two boxes and placing the spray nozzles and chain in the boxes, and folding up the vertical supports.
✓ The vertical supports will fold to a small size and will be light so the user can take it almost anywhere.
✓ The boxes will contain one motor each, along with circuitry to drive the motor. One of the boxes will contain a battery so the robot may be used while away from an electrical outlet. The battery will have a finite life, so it will have to be plugged in at some time to recharge, but the battery life will be sufficient enough so it will be able to be unplugged for a very long time.

Heuristic Evaluation:

1. Validity of system status
Are users kept informed about what is going on?
Users know what is happening because they can see the action. The robot’s movements are not subtle. Users may also watch the interface for system status messages such as “Painting a line” or “Painting a circle.”

Is appropriate feedback provided within reasonable time about a user’s action?
User input is in the form of a vector-based image. If the user provides a valid image, it is immediately accepted and displayed. If the image displays correctly, the user can tell the system to proceed with painting. If the image displays incorrectly, the user may cancel the process or choose another image. If the user provides an invalid image, it is immediately denied.

2. Match between system and the real world
Is the language used at the interface simple?
Yes, the interface is very simple. The interface has four buttons: an “Open Image” button, which brings up a file browser so that the user can specify a vector image; a “Paint Image” button, which, not surprisingly, paints the image; a “Help” button, which opens a helpful document; and finally, a “Close” button, which closes the interface.

Are the words, phrases and concepts used familiar to the user?
Yes, everyone should know what “Open Image,” “Paint Image,” “Help,” and “Close” mean. The “Open Image” dialog only shows and allows selection of the correct file types, so the user may only select compatible files.

3. User control and freedom
Are there ways of allowing users to easily escape from places they unexpectedly find themselves in?
We have a one-page interface, so this doesn’t really apply. If they are in the “Open Image” dialog and they wish to return, they may simply click “cancel.”

4. Consistency and standards
Are the ways of performing similar actions consistent?
Yes, only one thing is done: Carry the robot to suitable wall -> “Open Image” -> “Paint Image”

5. Help users recognize, diagnose, and recover from errors
Are user messages helpful?
Yes, the messages are all self-explanatory:
“Image successfully loaded.”
“Error: Image not loaded. Confirm file is a valid vector image.”
“Connection to robot lost. Please check battery.”
“Painting a line.”
“Painting a circle.”
Do they use plain language to describe the nature of the problem and suggest a way of solving it?
Yes, see above.

6. Error prevention
Is it easy to make errors?
No, if the image is valid and the batteries are charged, there shouldn’t be any problems.

7. Recognition rather than recall
Are objects, actions and options always visible?
Yes, the buttons and status (whether image is loaded or not) are always visible.

8. Flexibility and efficiency of use
Have accelerators (i.e. shortcuts) been provided that allow more experience users to carry out tasks more quickly?
No, because the interface is both as simple and efficient as it could possibly be.

9. Aesthetic and minimalist design?
Is any unnecessary and irrelevant information provided?
Nope.

10. Help and documentation
Is help information provided that can be easily searched and easily followed?
Yes, clicking the “Help” button will open a helpful document describing the vector image file types accepted, and with instructions for arranging the robot and connecting it to the computer interface.