Requirements Document
Bot Ross Painting Device

Patrick Frankiewicz (frankiep, 400032304),
Salik Hashmi (hashms2, 001400242),
Mariah Janet Lindsay (lindsamj, 001413072),
Winky Luo (luowq, 001403419),
Francesco Rao (raof, 001307671),
Marlee Roth (rothm1, 400021754),
Sophia Tao (taos1, 400019596)

October 28, 2019
Version 0

McMaster University

ENGINEERING

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Version	Date	Description
0	28/10/2019	Created document

1 INTRODUCTION

For thousands of years people have been creating art, and it is arguably one of the most important parts of our culture. Making art more accessible is such a valuable task because enabling an artist to express them selves is an essential part of the human experience.

Bot Ross is an image to paint robot that makes art more accessible. Bot Ross takes in a user provided **PNG** or **JPEG** file and converts it into brush strokes. The list of strokes are sent to the mechanical components of the device where a robotic head will apply paint onto a canvas. When the painting is complete, Bot Ross will play a sound signifying the completion of its painting.

As a result, the user will be able to enjoy a custom-painted version of an image they have chosen.

1.1 Purpose

The project Bot Ross will be a robotic painting device that takes in a digital image and recreates it on canvas with paint using brush strokes. This document will specify the requirements for creating a minimum viable product. These requirements are ones that are not likely to change, since they will be present for all versions of the product. This document will also specify the requirements that may be modified as the devices design becomes finalized. These requirements will be listed as requirements that are likely to change or may only be included in a future version of the device.

1.2 Scope

The scope of our project comes from two different factors: the types of images we are able to process, and the types of images we are able to create.

The scope for the minimum viable product includes being able to turn any image into a black-and-white piece of art of a fixed size. Paint will be applied in a solid black colour with crisp borders. We will not be able to accommodate for gray-scale tones, only solid black (which will be painted) and solid white (which will not be painted). Furthermore, we will not be able to accommodate any images that are not a *.jpg* or *.png* format.

An extended scope, for versions of the project beyond the minimum viable product, will include being able to turn any digital image into a coloured piece of art. Paint will be applied solidly and with crisp borders, but we will be able to mix paint to produce different colours. We will support up to 20 distinct colours for each painting. We will also support varied canvas sizes (up to a certain maximum dimension), so the user will be able to paint a smaller or larger image using the same device.

2 CONTEXT DIAGRAM SHOWING BOUNDARIES

Figure 1 depicts the context diagram that represents a top level view of the Bot Ross Printing Device. This diagram shows how the software and hardware systems interact.

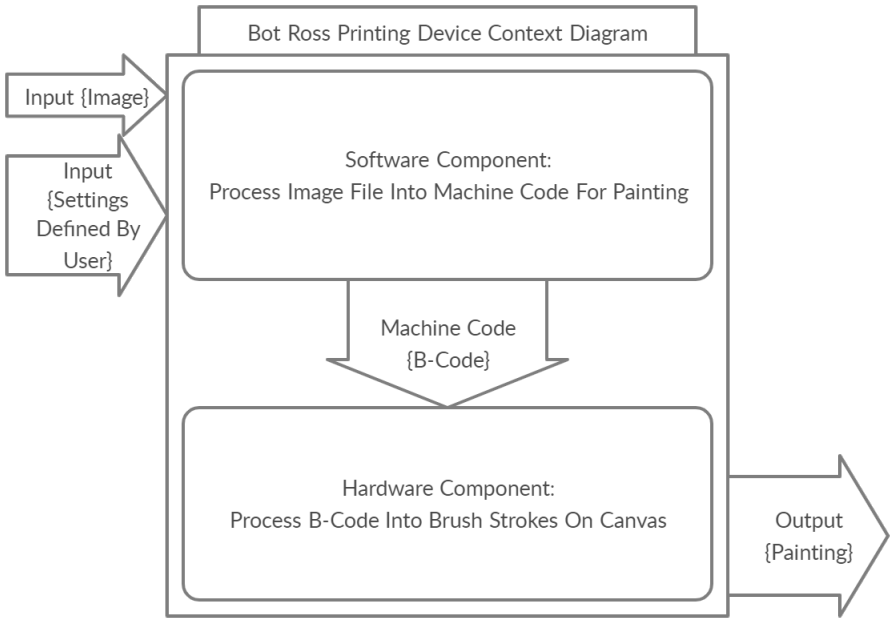

Figure 1: Context Diagram

3 MONITORED AND CONTROLLED VARIABLES

3.1 Monitored Variables

Monitored variables are variables that are dependent on the controlled variables. They may change over the course of the program's duration, and provide insight on whether requirements are met.

Table 2: **Monitored Variables**

Variable	Unit	**Description**
VM1 Paint Used	(Liters)	The total quantity of paint consumed for a single painting.
VM2: Render Time	(minutes)	The total time it takes for our algorithm to convert the input image into machine code (**B-Code**) for painting.
VM3: Painting Time	(minutes)	The total time required to produce a painting, not including image rendering time.
VM5: Number of generated strokes	(integer)	The number of strokes generated by the software.
VM6: Colours generated	(**RGB**)	The specific colours generated by the software.

3.2 Controlled Variable

Table 3: **Controlled variables**

Variable	Unit	**Description**
VC1: Stroke Length	(mm)	The specific length of a single brush stroke made on the canvas.
VC2: Stroke Width	(mm)	The specific width of a single brush stroke made on the canvas.
VC3: Stroke Angle	(degrees)	The specific angle of a brush stroke in relation to the home point (0,0) on the canvas.
VC5: Number of Colors	(integer)	The number of different paint colors used by the device.
VC6: Canvas Size	(mm x mm)	The length and width dimensions of the canvas in use.

4 CONSTANT

Table 4: **Constants**

Constant	Unit	**Description**
CS1: Budget	(dollars)	The budget is set to $750 for this project.
CS2: Maximum processing time	(sec)	The time to process the image into grid will be no more than 10 seconds
CS3: Maximum canvas size	(cm)	The maximum allowable side length of canvas that the machine can paint is 50cm.
CS4: Maximum paint time	(minutes)	The maximum time the machine will take to paint an image is 240 minutes.
CS5: Maximum colours	(integers)	The number of colours that the machine can paint the picture in, set to 2 for the minimum viable product, and 20 for versions beyond the minimum viable product.

5 BEHAVIOUR OVERVIEW INCLUDING NOTATION

- User inputs image into the User interface which sends the image to the **API** for processing
- User specifies settings of the print job (i.e. canvas size, homing position, etc)
- Software translates image into **B-Code** that correspond to a colour and strokes
- Brush stroke are executed as follows:
    1. travel to start position
    2. lower brush to canvas
    3. travel to end position
    4. lift up
- The device will **re-home** the brush head to the bottom left corner (0,0) at the beginning of every paint job

6 NORMAL OPERATION

6.1 Description

During normal operation, our product will take in a digital image as input from the user through a user interface. The image will then be processed into **B-Code** and the device will recreate the picture on canvas using brush strokes

6.2 Use Cases And Scenarios

6.2.1 User wants to upload an image

The devices user interface will prompt the user to input an image file when the system is ready for a new file to be uploaded. The user can then upload an image using a familiar file explorer interface.

6.2.2 User wants to change the canvas Size

There will be form fields for the user to enter the canvas size, either as an absolute measurement (mm) or as a percentage of the maximum size. The user will have the choice to keep the **aspect ratio** the same or change it. After the user uploads an image, the form will auto-populate with a suggested size based on the image that was uploaded.

6.2.3 User wants to begin painting the image

Once the user has uploaded an image, the system will display a button to begin the painting process. After the user clicks on this button, the robot will begin painting, which involves the following steps:

1. If the feature to have variable canvas sizes is enabled, the robot will adjust for the canvas size according to the user input
2. The software will process the image into a list of brush strokes
3. The robot will begin reproducing the image on canvas
4. The robot will alert the user via an auditory signal when the painting is done.

6.2.4 User wants to pause the painting

During any time in steps 1-4 in section 11.2.3, the user may choose to pause any progress via the user interface. The robot will stop immediately and retain state in a way such that they can resume.

6.2.5 User wants to cancel the painting

Any time while the painting is paused, the user can choose to cancel the painting entirely. In this scenario, the robot will reset its brush and canvas sizes into the original position and return to its initial state. It is now ready to begin a different painting if the user desires it.

7 DIAGRAM SHOWING FUNCTIONAL DECOMPOSITION

Figure 2 depicts the functional decomposition diagram that shows the separation of the program into separate functional modules. The only external control in our design is the **UI**. This is where the user will select an image for input and can specify settings for how the image will be painted. The two major elements of our system are the software and hardware subsystems, where each system will process the information relevant to its subsystem.

The software component converts the inputted image into a list of **B-Code** instructions and passes that information to the hardware component once it's done compiling. The hardware component is responsible for converting the **B-Code** into motor movements which are controlled by one motor for the X, Y, and Z axes. The final result is painted onto the canvas in the bed of the device.

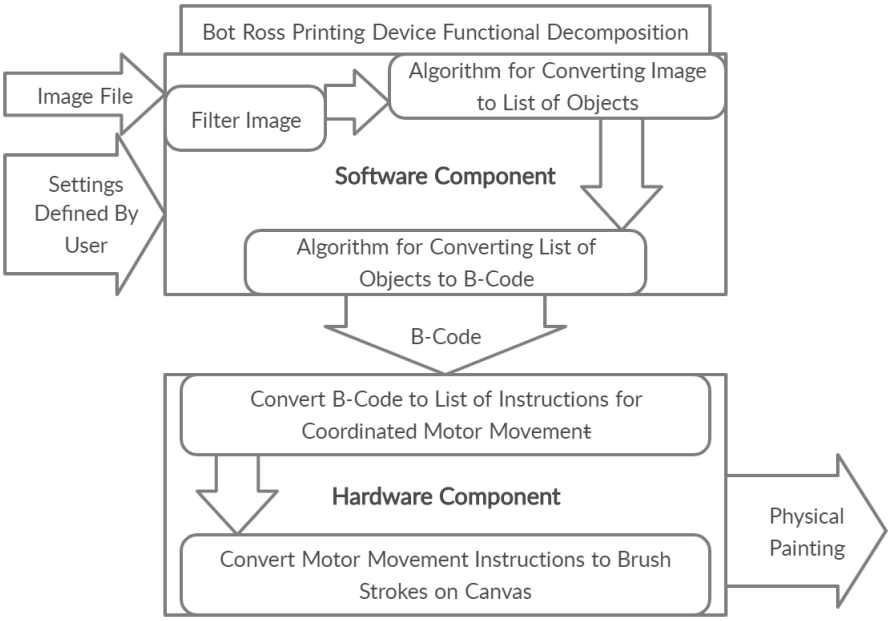

Figure 2: Functional Decomposition Diagram

8 REQUIREMENTS

8.1 Software Requirements

8.1.1 Software requirements not likely to change

Software inputs

- RSNI1: Program will be compatible with **JPG** and **PNG** formatted image files. The rationale is that **JPG** and **PNG** are the most common digital photo image formats, and our program should support the most common formats as input.
- RSNI2: Program will accept a file upload from the user's computer.
- RSNI3: Program will accept custom user inputs as parameters for the painting.
- RSNI4: Program will validate user inputs before presenting the button to begin the painting. The rationale is that the program should not begin painting if any inputs are invalid.

Software outputs

- RSNO1: Program will scale the image to fit a fixed canvas size. The rationale is that the painting must fit entirely on the canvas provided
- RSNO2: Program will present the user with a preview of what the painting will look like after the software has processed the image
- RSNO3: Program will prompt for user approval of the preview image before painting begins. The rationale is that the user must accept the rendering of the painting before the machine starts, since it is less time and resource intensive to cancel an unwanted painting before the physical painting begins.
- RSNO4: Program must be able to transform image a list of brush strokes, as specified by its start and end coordinates The rationale is that this interface will make it easy for the hardware to interpret the painting as a list of brush strokes.
- Program must be able to communicate to hardware via **B-Code**.

8.1.2 Software requirements likely to change

Software inputs

- RSLI1: Program renders a sample of the imputed image on screen for user to see
- RSLI2: Program allows the user to preview and accept the painting before machine starts to paint
- RSLI3: Program must resize the file according to the canvas size inputted by the user

Software outputs

- RSLO1: Program must be able to transform image a list of brush strokes and corresponding colours as an **RGB** value, as specified by its start and end coordinates. The rationale is that this interface will make it easy for the hardware to interpret the painting as a list of brush strokes.
- RSLO2: Program will determine the sequence of which color to paint first, grouping similar colours together. The rationale is that this will allow the program to paint similar colours together, without needing to clean the brush as much in between (as is necessary when painting very different colours)

8.2 Hardware requirements

8.2.1 Hardware requirements not likely to change

Hardware inputs

- RHNI1: Machine must read **B-Code**. The rationale is that the **B-Code** will contain the information needed for each stroke to paint the picture
- RHNI2: Machine shall be able to take in, hold, and apply paint. The rationale is that the machine will use acrylic paint for the painting, and it is thus necessary to be able to store and use the paint effectively.
- RHNI3: Machine checks dimension of the canvas and entered dimension are correct. The rationale is that it is desirable to perform an additional check to ensure that the dimension that were used for the rendering are the same as the actual canvas, since misalignment can lead to an inaccurate painting.

8.3 Hardware outputs

- RHNO1: Machine must be able to apply paint to a canvas
- RHNO2: Machine must move to end position when the painting is completed. The rationale is that this will prevent any water or paint from dropping onto finished painting, and to reset the position for the next painting.

8.3.1 Hardware requirements likely to change

Hardware Inputs

- RHLI1: Machine must receive feedback from motors on location of paintbrush
- RHLI2:Machine must check the level of the paint in the repository. The rationale is that the machine should make sure there is enough paint for the project, and if not, inform the user

Hardware Outputs

- RHLO1: Machine must be able to switch out brushes. The rationale is that the machine will be able to adopt paintings of different styles depending on different brushes used.
- RHLO2: Machine must be able to mix paint and create new colors, up to 20 colours at a time per painting
- RHLO3: Machine must be able to clean the brush. Rationale: Upon completion, the brush will be cleaned to prevent paint from drying on the bristles. Furthermore, between painting dissimilar colours, the machine should clean the brush to prevent any unwanted colour mixing.
- RHLO4: Machine will be able to paint on any canvas size less than the max dimensions
- RHLO5: Machine must be able to paint various thicknesses of strokes with the same brush. Rationale: This will allow adding style and precision to the painting.

8.4 Non-functional requirements

8.4.1 Appearance requirements

- NA1: The program has a clean and organized interface
- NA2: The device will look polished with no rough parts sticking out

8.4.2 Usability requirements

- NU1: The program instructions and text will be easily understandable to English speakers
- NU2: All electronic components can be powered on or turned off
- NU3: The brush head is able to move in the directions (Z axis) up, down, and (X,Y axis): up, down, left, right, and diagonally at any angle

8.4.3 Privacy requirements

- NP1: Program will not save any uploaded photos or data submitted by the user

8.4.4 Installability requirements

- NI1: The software will work on all operating systems
- NI2: The user will be able to mount a canvas onto the device
- NI3: The final device assembly will only require fasteners
- NI4: The device will be able to be placed on any level surface provided there is enough space for the platform
- NI5: The user will be able to replace brush head when needed
- NI6: The user will be able to refill paint repository if necessary

8.4.5 Portability requirements

- NP1: Program only requires a URL to access the interface
- NP2: The product will have a freestanding frame.
- NP3: The product will weigh under 20lb so that it may be carried.
- NP4: The product will be able to fit through a doorway after it has been rotated.

8.4.6 Learning requirements

- NL1: Program has a simple UI that only allows the user to input a photo and user settings.
- NL2: Users with basic computer experience should understand interface instinctively.

8.4.7 Safety requirements

- NS1: The user interface will blink or animate excessively. Rationale: such behaviours may cause user discomfort or trigger epilepsy.
- NS2: The machine will not spill any paint outside of the canvas area
- NS3: The machine will not have sharp edges that can endanger the user
- NS4: The machine will not have loose small parts that will be a choking hazard for small children

8.4.8 Performance requirements

- NPE1: The user interface will load in 1 second or less.
- NPE2: The user interface will respond to any commands in 1 second or less.
- NPE3: The software processing will process the image in 10 minutes or less.
- NPE4: It is unfavourable for the user to wait a long time before seeing the hardware begin progress.
- NPE5: The setup, including operations like resizing the campus, getting the brush ready, and preparing the colour pumps, will take 2 minutes or less.
- NPE6: The picture will be painted in 4 hours. Rationale: The robotic painter to paint it in the average time that human painters take to paint an image and not significantly more.
- NPE7: The cleanup, including operations like resetting the position of the brush, cleaning the paint brush, and resetting the canvas size, will take 5 minutes or less.

9 UNDESIRED EVENT HANDLING

9.1 Software Handling

- If the image processing fails, there will be a software reset to restart to process.
- In case of a software failure, the user will need to re-upload the image and start the process again.

9.2 Hardware Handling

- To ensure a safety critical system, upon a detection of the machine leaving the desired painting area the power will be cut off to the motors until system is reset.
- If a hardware failure is detected (pump, motor, etc.) operation is suspended until it is corrected and system resets.
- There will be an Emergency Stop button to halt the operation and cut off electrical power to all components at any time if a user deems it necessary.
- Upon a reset, the machine will clean the brush and return to a predefined resting area.

10 REFERENCES

History or Art Origins and Evolution of Visual Arts. (2019, October 25). Retrieved from http://www.visual-arts-cork.com/history-of-art.htm

11 APPENDIX

11.1 Glossary of Terms

Table 5: **Glossary of Terms**

Term	**Description**
API	Application Program Interface
Aspect Ratio	the ratio between height and width of the image
B-Code	A custom standard developed by our team for the machine code that is generated by the image conversion algorithm. The B-Code is the set of instructions that can be understood by our device to produce the required paint strokes.
Computer Numerical Control (CNC) Machine	A machine capable of manufacturing physical components by using a microcontroller to process input instructions into cutting operations.
G-Code	A standard for machine code used by CNC machines to define how a machine should manufacture a component.
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
Rehome	Position the brush head to the user specified home position of the canvas. Usually coordinates (0,0)
RGB	A way to specify colours using the amount of red, green, and blue as parameters. Each red, green, or blue value is between 0 and 255.
UI	User interface where the user inputs the desired image