

Team B2 – Leonardo Da Pinchy

Add your 10 slides after this slide... [remember, 10 min talk + 2 min Q/A]

For more information about formatting or importing slides see:

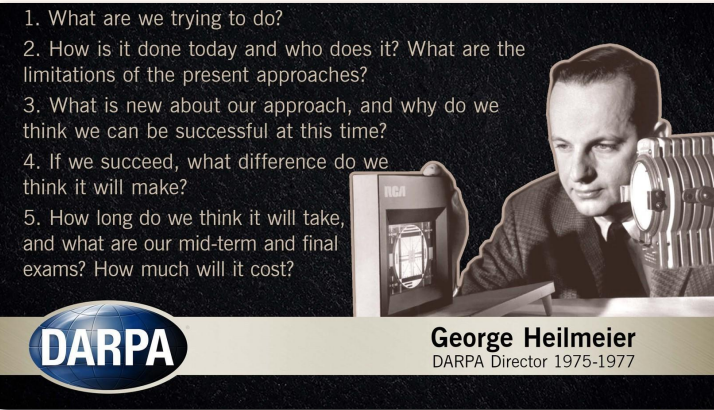
<https://gsuite.google.com/learning-center/products/slides/get-started/>

Make sure to cover

(refer to the Proposal Presentation Guidance):

- Use Case
 - Who is your customer, what product do they want to buy?
- Use-Case Requirements
 - What do they expect your product to do? [Answer this Quantitatively]
- Technical Challenges
 - What do you need to technically achieve so your product can meet these requirements?
- Solution Approach
 - Show us step by step what your proposed solution path(s) looks like – incorporate flow diagrams
- Testing, Verification and Metrics
 - Detail your testing & verification plan (and how you will measure & verify the results) of your product
- Tasks and Division of Labor
 - Technical Task List to build your proposed system, WHO is going to do WHAT in your team?
- Schedule
 - Gantt Chart (task timeline, demo system/components, MVP, stretch goal, include slack time)

Consider that this slide already works as a introduction slide so use your first slide wisely



Remember – Relevant figures (and tables) can be worth
“a thousand words”!

User Story

- Digital artist and business owner Leonardo creates custom ceramic mugs
- Leonardo makes his designs digitally
- Redoing every design on each mug is time consuming at scale
- **He wants a robotic solution that allows him to produce mugs en masse**



Use Case

- Reproducing digital art on non-planar surfaces where using standard CNC machines with 3 DoF is difficult

Often, problems with printing are:

- Lack of stroke depth, colors, and fluidness
- Robots have a difficult time adjusting to surfaces it's not familiar with

Our robot should successfully replicate a piece of submitted art on a fixed non-planar surface whose dimensions we know.



Use-Case Requirements (1/2)

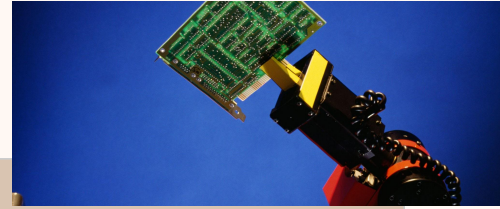
Requirement	Target	Justification
Target specifications	One surface; shape, curves, and dimensions known	Keeps the testing surface bounded.
Drawing area	Predefined, with front section reachable only	Avoid collisions and unreachable back surfaces.
Drawing surface alignment	Camera + fixture reference; set distance from target	Place the target consistently in a fixed location in front of the robot.
Working area	$0.4 \times 0.4 \times 0.4$ m	Allow space for the target, holder and pen motion.



Use-Case Requirements (2/2)

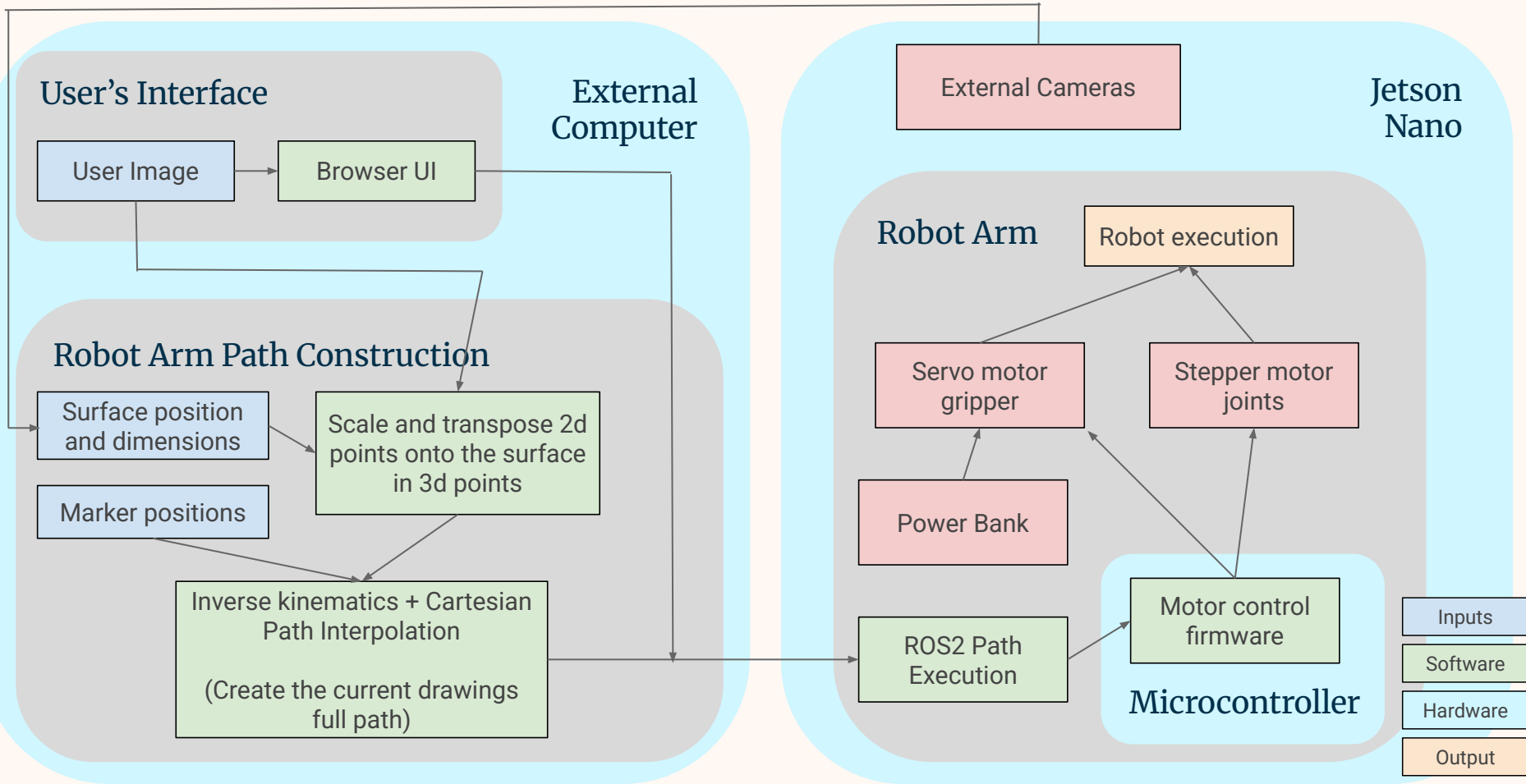
Requirement	Target	Justification
Drawing format	SVG vector paths	Preserve digital strokes for mapping onto the curved surface.
Line accuracy	Target $\leq 2.00\text{mm}$ surface-path error; statistic TBD	Keep outlines useful for subsequent carving or painting.
Input limits	2 colors; 50 strokes; 100 cm combined path	Support small patterns while bounding job complexity.
Pen orientation	Proposed 6-DOF arm	Adjust pen position and angle along the known curve.

Technical Challenges



Challenge	Requirement	Proposed response
Weight of arm	Arm needs needs to be able to draw without falling over	Install heavy counterweights at the base, or bolt the arm to a rigid fixture
Target position / rotation	Consistent design detection	Fixed location + April tag for manual calibration of (0, 0)
Marker contact	Maintain contact during curved motion	Spring-loaded holder; force sensing
Marker pickup	At least 90% swap success	Fixed rack; gripper; tip calibration
Reach / orientation	Reach / handle clearance	Front arc + collision checks; rotation later
Degrees of Freedom	6-DOF	Calculate curved path; adjust pen angle

Proposed Solution



Testing and Verification

```
243 void showNote(int mm){
244     FILE *fp;
245     int i = 0, isFound = 0;
246     system("cls");
247     fp = fopen("note.dat","rb");
248     if(fp == NULL){
249         printf("Error in opening the file");
250     }
251     while(fread(&R,sizeof(R),1,fp) == 1){
252         if(R.mm == mm){
253             gotoxy(10,5+i);
254             printf("Note %d Day = %d: %s", i+1, R.dd, R.note);
255         }
256     }
257 }
```

1

Object/Surface Calibration

Procedure: Use AprilTags (QR codes) and external camera for calibration

Metrics:
Calculate error (cm) between calculated origin and intended origin

Ideal error ≤ 2 mm

2

Marker ID and Selection

Procedure:
Use external camera + CV model to pick out desired marker color

Metrics:
Calculate accuracy of selecting correct color marker

Ideal accuracy $\geq 90\%$

3

Coordinate Algorithm Testing

Procedure:
Have robot draw over predetermined printed image

Metrics:
Calculate maximum error (mm) between printed and drawn image

Ideal error ≤ 2 mm

4

Pen Orientation

Procedure:
Test roll, pitch, and yaw DOFs while drawing on flat surface

Metrics:
Calculate maximum roll, pitch, and yaw angles

Ideal yaw $\approx 60^\circ$
Ideal pitch $\approx 60^\circ$
Ideal roll = 360°

Tasks & Division of Labor

Owner	Workstream	Deliverables
Jonathan Waller	Embedded / hardware	Arm, mug holder, gripper, controls; rotary stage later
Bryan Hong	Circuits	Power, drivers, sensor circuits, hardwired stop
Aaron Tang	Software / modeling	Mug alignment, curved SVG paths, kinematics, web UI
All three	Integration / tests	Mug drawing, placement and contact tests

Schedule & Demonstration Milestones

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Leonardo Da Pinchy Project Schedule														
2	September 13–December 5, 2026 Known dimensions; fixed front section first; indexed rotation is a stretch goal.														
3	Task	Owner	Done?	9/13– 9/19	9/20– 9/26	9/27– 10/3	10/4– 10/10	10/11– 10/17	10/18– 10/24	10/25– 10/31	11/1– 11/7	11/8– 11/14	11/15– 11/21	11/22– 11/28	11/29– 12/5
4	Hardware & Circuits														
5	Define mug dimensions, front arc and test limits	All	☐												
6	Select arm for mug reach and handle clearance	Jonathan Waller	☐												
7	Design motor power, force sensing and stop	Bryan Hong	☐												
8	Build drivers and test power / stop circuits	Bryan Hong	☐												
9	Assemble arm; build repeatable mug fixture	Jonathan Waller	☐												
10	Build compliant pen holder and marker rack	Jonathan Waller	☐												
11	Wire joint references and contact sensors	Bryan Hong	☐												
12	Verify homing, overload and USB-loss stops	Jonathan Waller	☐												
13	Software & Modeling														
14	Model known mug; check joint reach and angles	Aaron Tang	☐												
15	Calibrate camera, fixture zero and pen tip	Aaron Tang	☐												
16	Map SVG lines to measured mug radius / height	Aaron Tang	☐												
17	Run timed joint commands over USB	Jonathan Waller	☐												
18	Tune pen contact on curved test strokes	Jonathan Waller	☐												
19	Build upload, height / angle placement preview	Aaron Tang	☐												
20	Plan curved moves, pen angles and safe lifts	Aaron Tang	☐												
21	Implement two-color swaps and tip offsets	Jonathan Waller	☐												
22	MVP: upload, align and draw fixed front arc	All	☐												
23	Testing & Demonstration														
24	Verify stop circuit and sensor fault response	Bryan Hong	☐												
25	Test mug reload alignment and front-arc reach	All	☐												
26	Measure curved-path error vs. 2 mm target	Aaron Tang	☐												
27	Verify two-color swaps vs. 90% target	Jonathan Waller	☐												
28	Measure contact force, gaps and surface marks	Bryan Hong	☐												
29	Stretch: indexed rotation and section-joint test	All	☐												
30	Fix issues; rerun full mug-drawing tests	All	☐												
31	Rehearse mug demo; finish artwork and report	All	☐												
32															
33	Jonathan Waller: embedded / hardware. Bryan Hong: circuits. Aaron Tang: software / modeling. All: shared work.														