UNIVERSITY OF OTTAWA
Faculty of Engineering

GNG1103 Design Project – Deliverable G

Group C2

Adrianna Chouliotis	300225514
Hunter Fleming	300213232
Minjung Gong	300080238
Toby Thai	300201869

Submitted to: Professor David Knox

Submitted on: March 13th, 2022

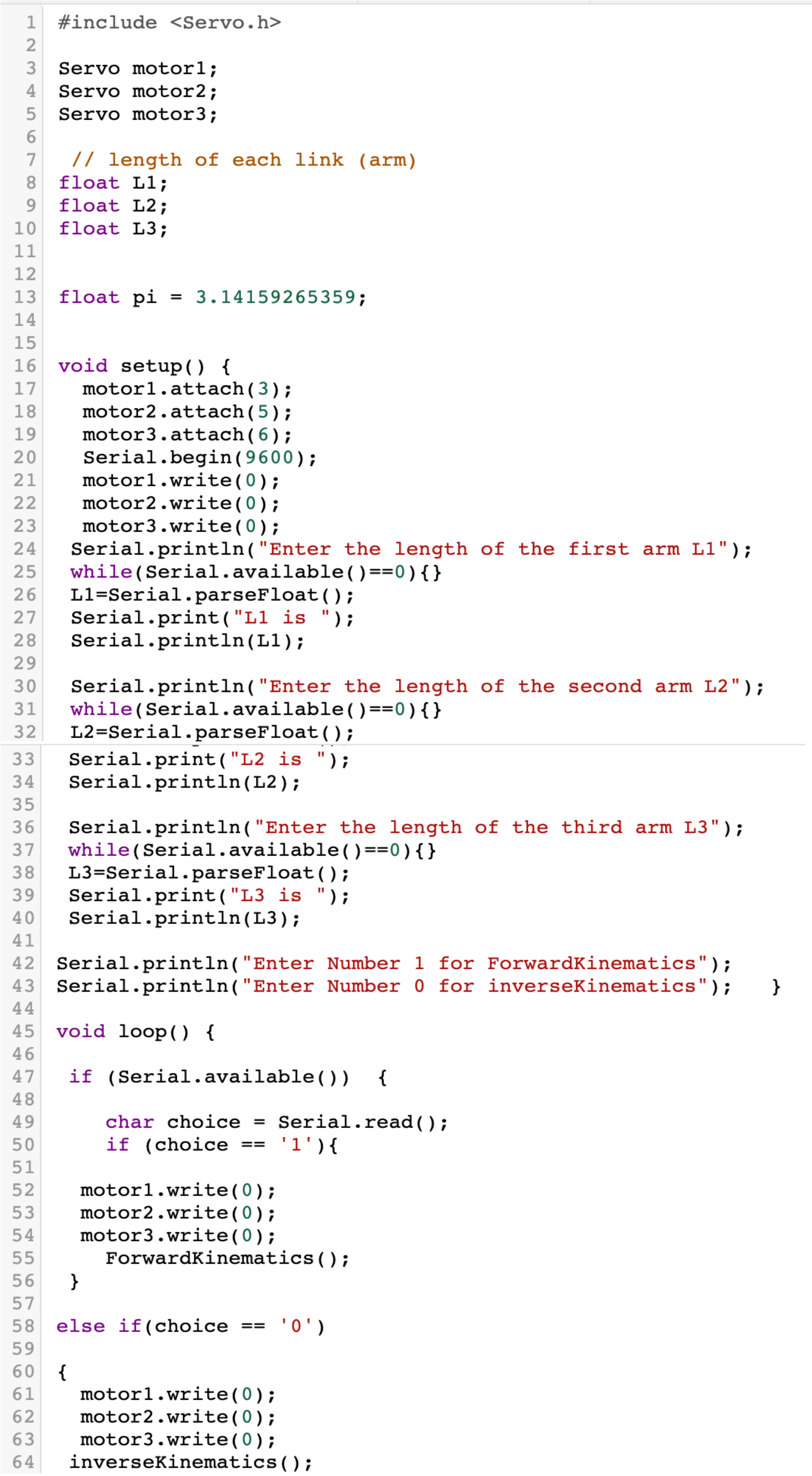
Wrike Snapshot:https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=8SmYo6oOuhHCyp5uQ66FdSaGgsV8qylD%7CIE2DSNZVHA2DELSTGIYA

TABLE OF CONTENTS

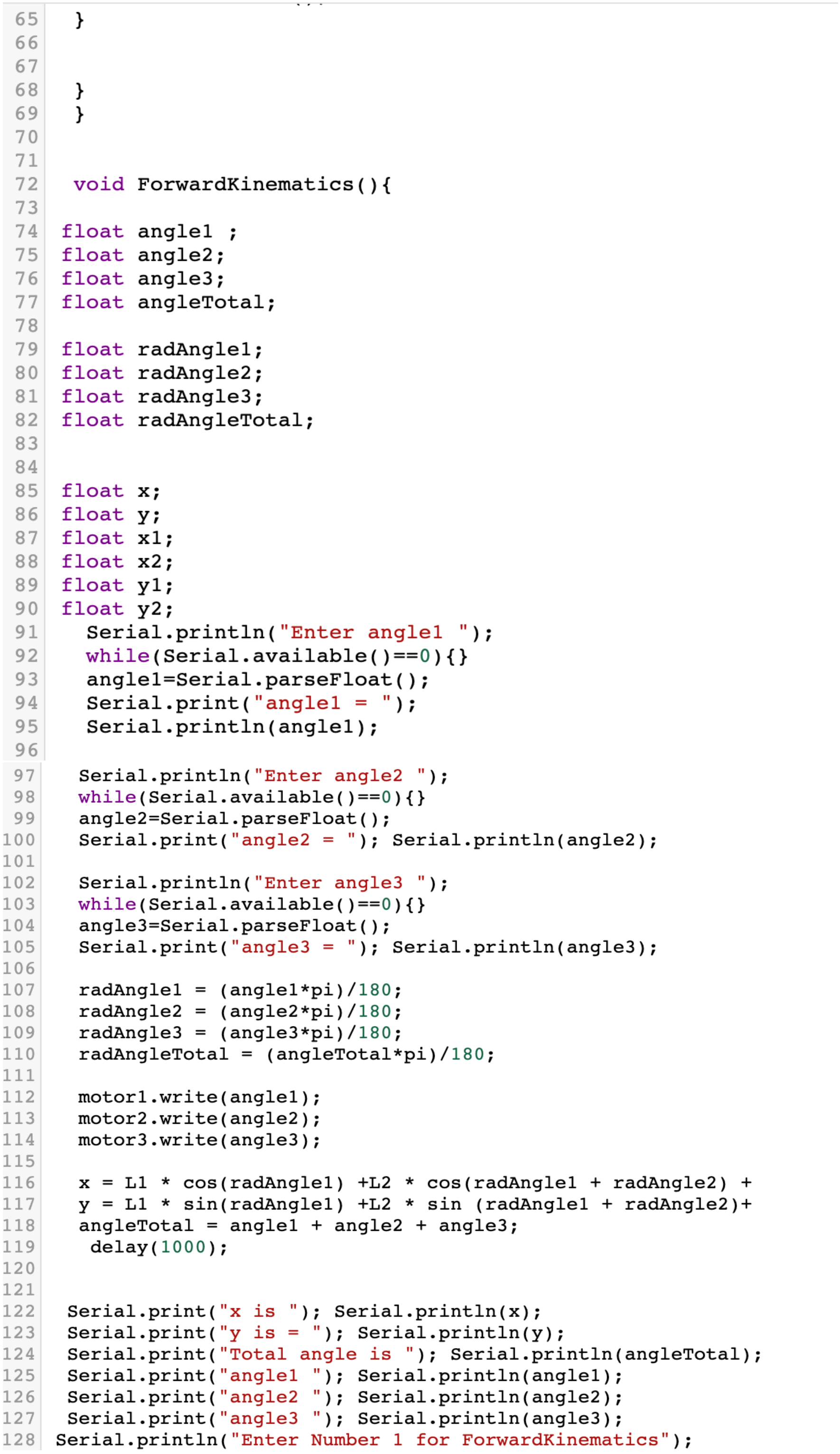
LIST OF TABLES

LIST OF FIGURES

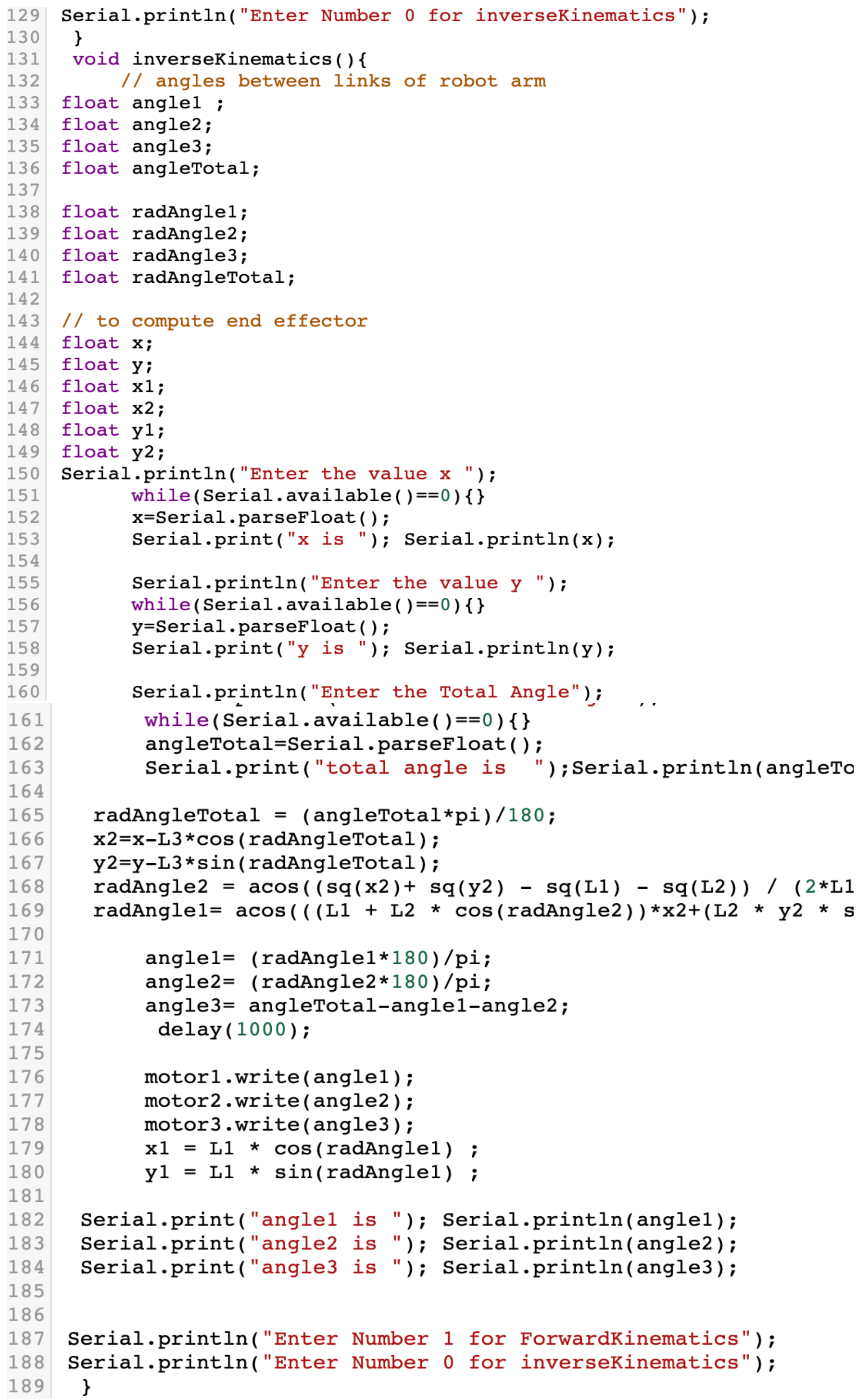
Updated Code



```
#include <Servo.h>

Servo motor1;
Servo motor2;
Servo motor3;

 // length of each link (arm)
float L1;
float L2;
float L3;


float pi = 3.14159265359;


void setup() {
  motor1.attach(3);
  motor2.attach(5);
  motor3.attach(6);
  Serial.begin(9600);
  motor1.write(0);
  motor2.write(0);
  motor3.write(0);
 Serial.println("Enter the length of the first arm L1");
 while(Serial.available()==0){}
 L1=Serial.parseFloat();
 Serial.print("L1 is ");
 Serial.println(L1);

 Serial.println("Enter the length of the second arm L2");
 while(Serial.available()==0){}
 L2=Serial.parseFloat();
 Serial.print("L2 is ");
 Serial.println(L2);

 Serial.println("Enter the length of the third arm L3");
 while(Serial.available()==0){}
 L3=Serial.parseFloat();
 Serial.print("L3 is ");
 Serial.println(L3);

Serial.println("Enter Number 1 for ForwardKinematics");
Serial.println("Enter Number 0 for inverseKinematics");   }

void loop() {

 if (Serial.available())  {

    char choice = Serial.read();
    if (choice == '1'){

  motor1.write(0);
  motor2.write(0);
  motor3.write(0);
    ForwardKinematics();
 }

else if(choice == '0')

{
  motor1.write(0);
  motor2.write(0);
  motor3.write(0);
 inverseKinematics();
```



```
 }


 }
 }


 void ForwardKinematics(){

float angle1 ;
float angle2;
float angle3;
float angleTotal;

float radAngle1;
float radAngle2;
float radAngle3;
float radAngleTotal;


float x;
float y;
float x1;
float x2;
float y1;
float y2;
  Serial.println("Enter angle1 ");
  while(Serial.available()==0){}
  angle1=Serial.parseFloat();
  Serial.print("angle1 = ");
  Serial.println(angle1);

  Serial.println("Enter angle2 ");
  while(Serial.available()==0){}
  angle2=Serial.parseFloat();
  Serial.print("angle2 = "); Serial.println(angle2);

  Serial.println("Enter angle3 ");
  while(Serial.available()==0){}
  angle3=Serial.parseFloat();
  Serial.print("angle3 = "); Serial.println(angle3);

  radAngle1 = (angle1*pi)/180;
  radAngle2 = (angle2*pi)/180;
  radAngle3 = (angle3*pi)/180;
  radAngleTotal = (angleTotal*pi)/180;

  motor1.write(angle1);
  motor2.write(angle2);
  motor3.write(angle3);

  x = L1 * cos(radAngle1) +L2 * cos(radAngle1 + radAngle2) +
  y = L1 * sin(radAngle1) +L2 * sin (radAngle1 + radAngle2)+
  angleTotal = angle1 + angle2 + angle3;
   delay(1000);


 Serial.print("x is "); Serial.println(x);
 Serial.print("y is = "); Serial.println(y);
 Serial.print("Total angle is "); Serial.println(angleTotal);
 Serial.print("angle1 "); Serial.println(angle1);
 Serial.print("angle2 "); Serial.println(angle2);
 Serial.print("angle3 "); Serial.println(angle3);
Serial.println("Enter Number 1 for ForwardKinematics");
```



```
Serial.println("Enter Number 0 for inverseKinematics");
 }
 void inverseKinematics(){
     // angles between links of robot arm
float angle1 ;
float angle2;
float angle3;
float angleTotal;

float radAngle1;
float radAngle2;
float radAngle3;
float radAngleTotal;

// to compute end effector
float x;
float y;
float x1;
float x2;
float y1;
float y2;
Serial.println("Enter the value x ");
      while(Serial.available()==0){}
      x=Serial.parseFloat();
      Serial.print("x is "); Serial.println(x);

      Serial.println("Enter the value y ");
      while(Serial.available()==0){}
      y=Serial.parseFloat();
      Serial.print("y is "); Serial.println(y);

      Serial.println("Enter the Total Angle");
       while(Serial.available()==0){}
       angleTotal=Serial.parseFloat();
       Serial.print("total angle is  ");Serial.println(angleTo

   radAngleTotal = (angleTotal*pi)/180;
   x2=x-L3*cos(radAngleTotal);
   y2=y-L3*sin(radAngleTotal);
   radAngle2 = acos((sq(x2)+ sq(y2) - sq(L1) - sq(L2)) / (2*L1
   radAngle1= acos(((L1 + L2 * cos(radAngle2))*x2+(L2 * y2 * s

       angle1= (radAngle1*180)/pi;
       angle2= (radAngle2*180)/pi;
       angle3= angleTotal-angle1-angle2;
        delay(1000);

       motor1.write(angle1);
       motor2.write(angle2);
       motor3.write(angle3);
       x1 = L1 * cos(radAngle1) ;
       y1 = L1 * sin(radAngle1) ;

  Serial.print("angle1 is "); Serial.println(angle1);
  Serial.print("angle2 is "); Serial.println(angle2);
  Serial.print("angle3 is "); Serial.println(angle3);


 Serial.println("Enter Number 1 for ForwardKinematics");
 Serial.println("Enter Number 0 for inverseKinematics");
  }
```

Figure 1. Arduino Code

Test Plan Results

Table 1. Test Plan for Prototype II

Test	Test Objective	Description of the Test Objective	Planned Start Date and Estimated Duration	Qualitative Results	Quantitative Results
1	To test that the assembly of the mechanical arm.	➢ Robotic arm can move flexibly within a one-meter range can bear the weight of the requested object.	March 7th, 2022 Test Duration: 5 minutes	The robot arm is not large enough for the original intended range; therefore a range was estimated through mathematical analysis based on maximum measured angle.	Maximum $\Theta_1 = 90^O$ Maximum $\Theta_2 = 150^O$
2	To test the code.	➢ A screenshot/recording will be taken and put in the Deliverable.	March 7th, 2022 Test Duration: 5 minutes	The code has been debugged and ran to give a result over simulations.	The code gives a precise answer matching inverse kinematics solvers.
3	To test the assembly of the mechanical arm with the code system.	➢ The robotic arm and the code system will be connected to test if the robotic arm can operate as required by the code system.	March 7th, 2022 Test Duration: 5 minutes	Results were not taken due to unavailability of arm + materials needed (Arduino, motors, etc)	N/A

Maximum Reach Calculations

(Based off Measurements)

X-Reach:

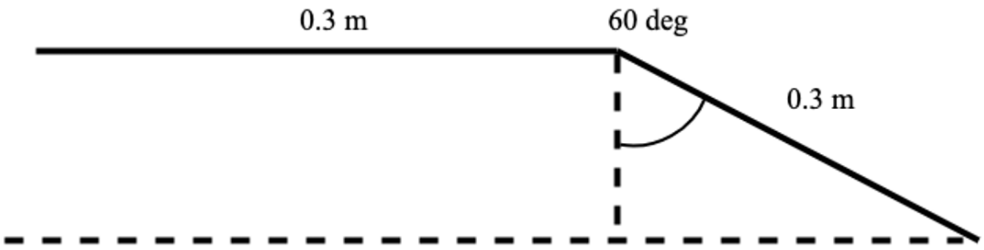


Figure 2. Schematic of Kinematics Problem

$$\sin(60) = \frac{x}{0.3\ m}$$

$$x = 0.26\ m$$

$$0.3\ m + 0.26\ m = 0.56\ m$$

X/Z-reach not needed due to 180 degrees turning.

UPDATED BILL OF MATERIALS

Table 2. Updated Bill of Materials

Name	Description	Dimensions	Quantity	Unit Cost	Total Cost
Software					
Python	End effector needs open-source Python to work. Python can be download for free from the Python site and is useful and easy for coding.	-	1	$0	$0
Hardware					
Laptop	To code, prototype, and test with Python, we need a laptop. But it is free because everyone on the team has their own laptop already.	-	1	$0	$0
Arduino Uno	The Arduino uno is an open-source microcontroller board. It is required to operate the robot arm using Python.	-	1	$0.00	$0.00
ABS Filament	ABS filament is 3d printer plastic. It is easy to process, has high impact resistance and good heat resistance. It is often used as a substitute for metal in industrial products such as automobile parts and electronic device parts because of its very high strength.	-	1	$0	$0
Screw	Screws are used to connect the arm to the end effector.	M5-0.8×45mm	3	$0.00	$0.00
Wire Set	Wires are used to connect the Arduino to the end effector, supply and operate power.	-	120 pcs	$0.00	$0.00
Velcro Industrial Roll	Velcro band is used in this project to strap the items together by attaching them to the End effector. This is suitable for strapping items as it has a good adhesion if only a certain amount of contact area is secured.	Length: 5m Width: 3cm	1	$20.99	$20.99
Servo Motor	Servo motor is a motor that can move exactly as much as inputted by the control and measurement circuit when movement is specified. This is where the movement of the end effector helps.	13.69×9.8×3.61 cm 50g	4	$0.00	$0.00
			TOTAL	$20.99	
			TAX (13%)	$2.73	
			TOTAL COST	$23.72	

UPDATED DESIGN SPECIFICATIONS

Table 3. Updated Target Design Specifications

	Design Specifications	**Relation**	**Value**	**Units**	**Verification**	**Prototype 1: Specs Met?**	**Prototype 2: Specs Met?**
	Functional Requirements						
1	3D Printed	-	Yes	>	Design	N/A	N/A
2	Coded off Python	-	Yes	-	Design	Yes	Yes
3	Open Source	-	Yes	-	Uploading	Yes	Yes
4	Ground Mounted	>	Yes	-	Design	Yes	Yes
	Constraints						
1	Cost per Arm	≤	50	$ CAD	Cost Calculations	Yes	Yes
2	Temperature to Withstand	Range	-30 - 60	∘C	Testing/Analysis	N/A	N/A
3	Degrees of Freedom	=	3	-	Design	Yes	Yes
4	Weight	≤	9	Kg	Weighing	Yes	Yes
5	Size (L x H x W)	≤	1	m^2	Calculation	Yes	Yes
6	Payload	≈	1	Kg	Testing	N/A	N/A
7	Pressure to Withstand	≤	8	Bar	Testing/Analysis	N/A	N/A
8	Range of Motion	≥	±0.5	m	Testing/Calculations	N/A	Yes
9	Time to Paint 4 ft^2	≲	4	Hours	Testing	N/A	N/A
	Non-Functional Requirements						
1	Ease of Use	≤	6	Hours	Training Testing	Yes	Yes
2	Pinch Points	≥	4	-	Design	Yes	Yes
3	UV/Corrosion Resistance	-	Yes	-	Testing	N/A	N/A
4	Life Span of the Product	≥	1	Year	Use	N/A	N/A

TEST PLAN UPDATE

Table 4. Test Plan for Prototype III

Test	**Test Objective**	**Description of the Test Objective**	**Planned Start Date and Estimated Duration**
1	To test that the assembly of the mechanical arm.	➢ The 3 dots of freedom will allow the arm to move flexibly while having a blaster/painter strapped into the end effector within a one-meter range. ➢ A screenshot/recording will be taken and put in the Deliverable.	March 21st, 2022 Test Duration: 15 minutes
2	To test the code.	➢ A screenshot/recording will be taken and put in the Deliverable.	March 21st, 2022 Test Duration: 15 minutes
3	To test the assembly of the mechanical arm with the code system.	➢ The robotic arm and the code system will be connected to test if the robotic arm can operate as required by the code system.	March 21st, 2022 Test Duration: 15 minutes

Risk and Contingency Plan

Table 5. Risk and Contingency Plan for Prototype III

Risk	Risk Management	Contingency Plan
The maximum range is not like what we calculated	➢ Different calculations done by different members will be gathered to minimize the error.	➢ If the difference between the realistic range and the calculated one is negligible which is still in the estimated range, we can assume the value to be accurate. ➢ Otherwise, the group will overestimate the assumed value, if possible, to provide for a margin of error.
Operational/ Technical Failure	➢ The testing results done in the last prototype will reduce the probability of operational failure in this prototype phase.	➢ If the problem cannot be resolved, then the product will not include that aspect.
Scope Creep	➢ We will seek regular customer/client feedback as well as the project manager's and TAs' opinions to minimize the probability of this risk.	➢ The requirement is expected to be integrated successfully in the final product. ➢ If the solution is not possible, the final product will not include the aspect.
Performance Risk	➢ Tests will be done on this prototype so if there is any performance problem, it will be determined before the final presentation	➢ The team will do some extensive research to define and resolve the problems. If the team cannot resolve the problem, we will ask the TAs and/or project manager for assistance. ➢ If it still does not help resolve the problem, the final product will not include the aspect.