MELANIE ZITO

Academic Portfolio

760 Stacy Oak Way
Millersville, MD 21108
410-991-0925
melliezito@gmail.com

Severna Park High School
60 Robinson Rd.
Severna Park, MD 21146

MAIN TABLE OF CONTENTS

PRINCIPLES OF ENGINEERING TABLE OF CONTENTS

Main TOC

COURSE DESCRIPTION

Principles of Engineering (POE) is a high school-level survey course of engineering. The course exposes students to some of the major concepts that they will encounter in a postsecondary engineering course of study. Through problems that engage and challenge, students explore a broad range of engineering topics, including mechanisms, the strength of structures and materials, and automation. Students develop skills in problem solving, research, and design while learning strategies for design process documentation, collaboration, and presentation

Main TOC POE TOC

KINEMATICS AND TRAJECTORY MOTION: BALLISTIC DEVICE

Important terms for unit:

-Mean: Arithmetic average. Sum of all data values divided by number of data values.

-Standard Deviation: A quantity calculated to find the extent of deviation in the group as a whole.

-X-Displacement: Horizontal distance traveled.

-Initial Velocity: Angular speed of a projectile at the start of its flight.

-Firing Angle: Angle at which the projectile left the ballistics device.

Constraints:

-Device had to fit into a 1 ft x1 ft box

-The device was required to have a mouse trap as a main component

-Students had to use mainly materials provided though were allowed to buy specific items

-Students given 5 build days of school time to complete

-A maximum of two students were in each group

Criteria:

-Students drew a random distance out of 5 -15 ft on testing day

-A ping pong ball must be shot into a 1 ft x1 ft box placed around the distance drawn for full points

-Students were graded on entries written about their build days and the performance of the device

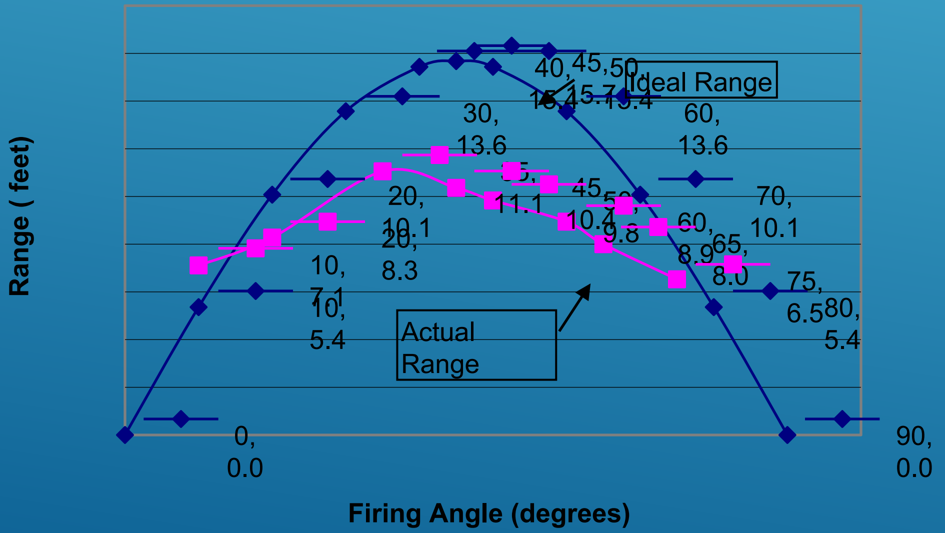
Main TOC

POE TOC

Angle (degrees)	Distance shot based on firing angle Range 1 (feet)	Range 2 (feet)	Range 3 (feet)	Average Range (feet)	Std Deviation (feet)
10	7.1666	7	7.1666	7.1	0.1
20	8.1666	8.3333	8.3333	8.3	0.1
35	11	11.0833	11.0833	11.1	0.0
45	10.4166	10.3333	10.3333	10.4	0.0
50	9.8333	9.75	9.9166	9.8	0.1
60	8.8333	9	9	8.9	0.1
65	7.9166	8	8.0833	8.0	0.1
75	6.5	6.5833	6.5	6.5	0.0

This ballistics device shot in a range of 6.5 to 11 feet based on the angles measured and then drilled.
Standard deviation: 0.1 feet
Angles used: 10, 20, 35, 45, 50, 60, 65, and 75

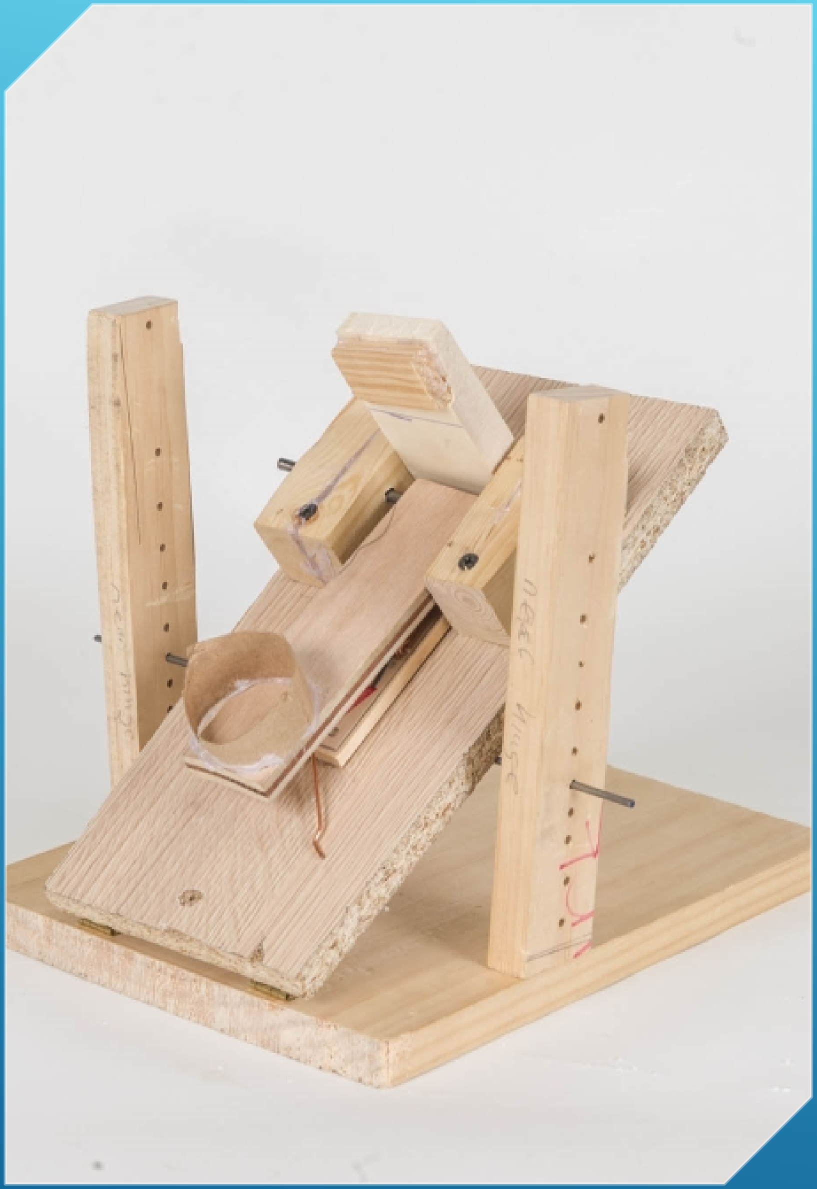
Ideal and Actual Range vs Firing Angle



The actual range of this device did not match the ideal range because of a bent mouse trap hindering the strength this device. It was only corrected on testing day. Had the data been accurate, this device would have shot farther and in more of an arc. This device shot closer to the ideal data on testing day.

Main TOC

POE TOC



This ballistic device was a catapult. A mouse trap for the force since it was required in the criteria. On the sides, two pieces of wood were placed with holes drilled at angles 10, 20, 35, 45, 50, 60, 70, and 75. A metal bar was then used to move the base board up and down and adjust the distance it shot.

Our team was one out of the only three teams in our class to successfully shoot a ping pong ball into a 1 foot cube around the distance drawn.

Issues:

-At first the catapult would only fire 3 feet maximum. The cardboard cylinder cut to hold the ball was blocking its trajectory and was later cut down.

-The original design had no way to adjust angle but the new idea of a bar to move the device worked immediately.

-On the testing day the mouse trap bent and it was changed out. This improved the device's shooting range.

MACHINE CONTROL DESCRIPTION

In this unit, students learned how to do basic programming with VEX which uses modified C++ and built a clawbot base which was used to complete challenges using programming. Programs were downloaded to the robot’s cortex, or its controlling unit, and were executed autonomously.

Important concepts:

- Open systems have no feedback and only has one direction of control in which outputs are sent by the cortex.
- Closed systems have feedback and a closed loop of inputs and outputs being given back and forth between the sensors (input data to the cortex) and the motor (receives output commands from the cortex).
- Digital sensors give data that is either 1 or 0 (two possible options). e.g., button
- Analog sensors give data in an infinite number of values e.g., potentiometer
- Programs are organized in a similar way to flowcharts

CRITERIA AND CONSTRAINTS

Maze Challenge:

- Criteria:
  - Robot must drive through the course from start to finish while staying inside of the black lines marking the course
  - Course: left, then drive, then right, then drive, then right
- Constraints:
  - One and a half days given to complete
  - Must use time so as to learn why time is unreliable. Battery power affects the distance for the same amount of time.
  - Each team member must take part in coding
  - Three attempts were allowed

Sprint Challenge:

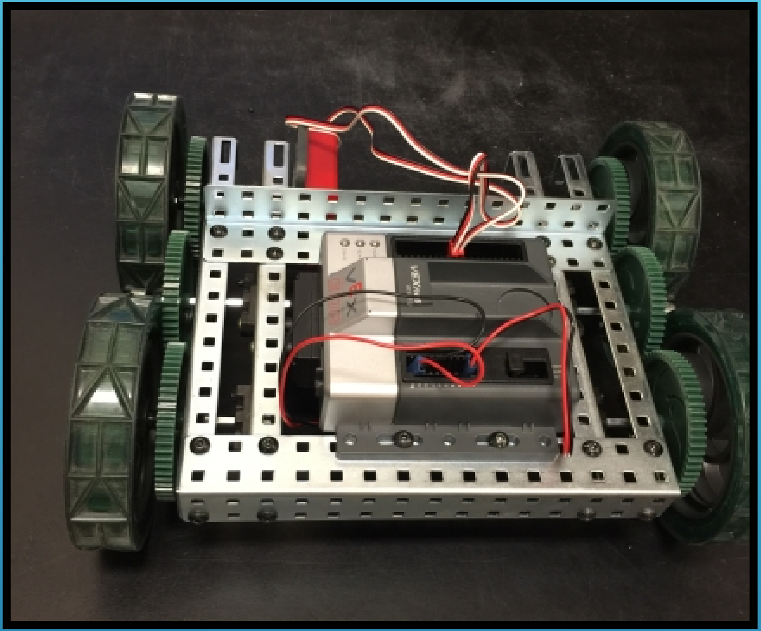
- Criteria
  - Robot must be on top of black line marking the correct distance of the four when making the 180 degree turn back to the start and be on top of the starting line at the end.
  - Course: drive forward, turn 180 degrees, return to the start and then repeat, going twice the distance the second time, etc., always returning to the start.
  - Must stay within black sidelines
- Constraints:
  - One and a half days given to complete
  - Must use encoder to measure wheel rotations and drive constant distances
  - Each team member does part of the coding
  - Three attempts allowed

Main TOC

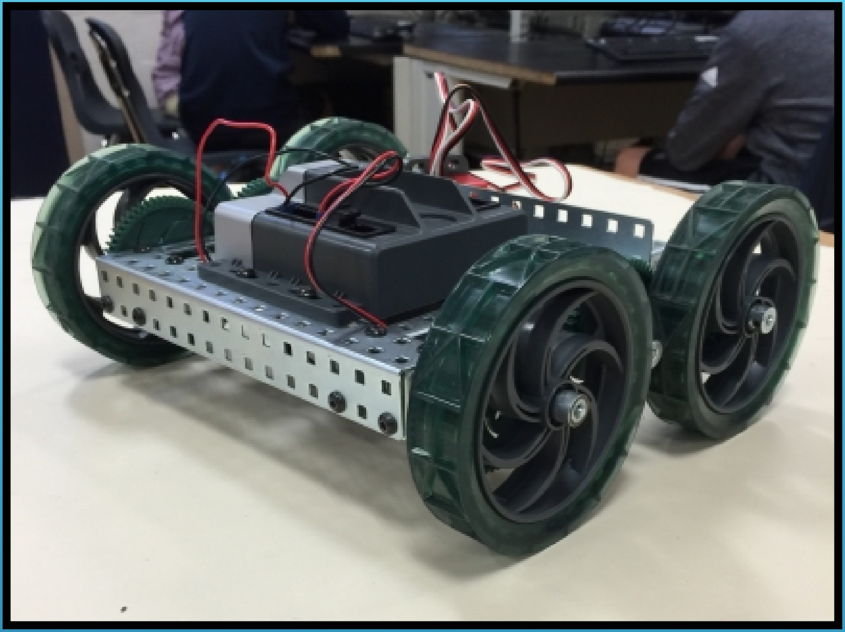
POE TOC

CLAWBOT CHASSIS PICTURES

Robot used to complete challenges



Top view



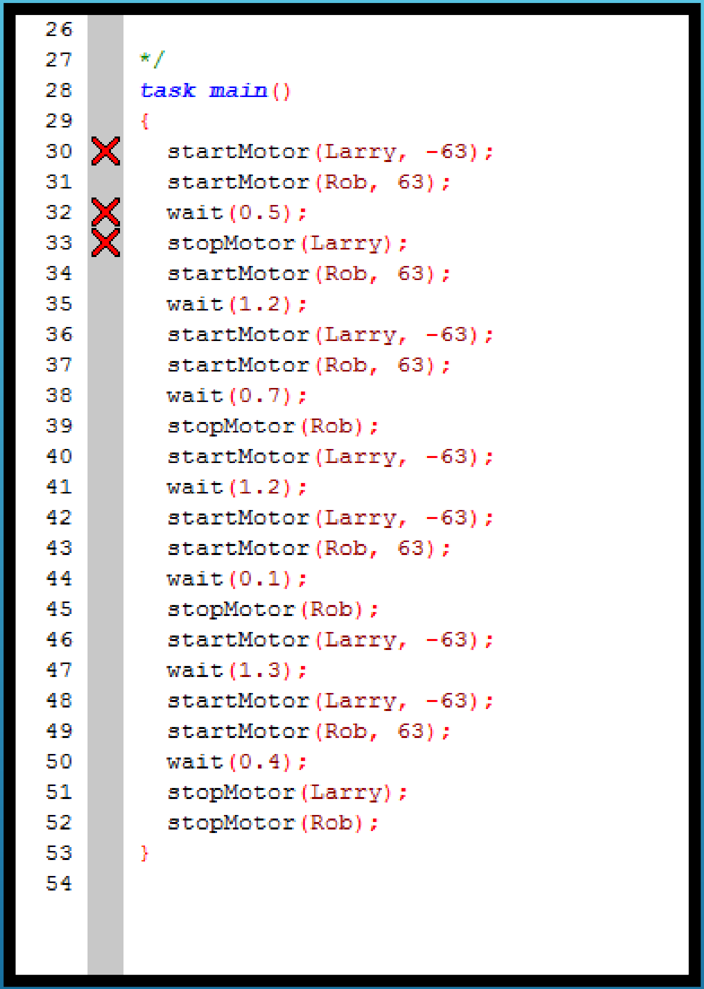
Side view

Components:

- Digital- none
- Analog-
  Encoder input and output
- Motors-
  Two 369s that drove each side
- Cortex- Robot's main controller

Main TOC

POE TOC

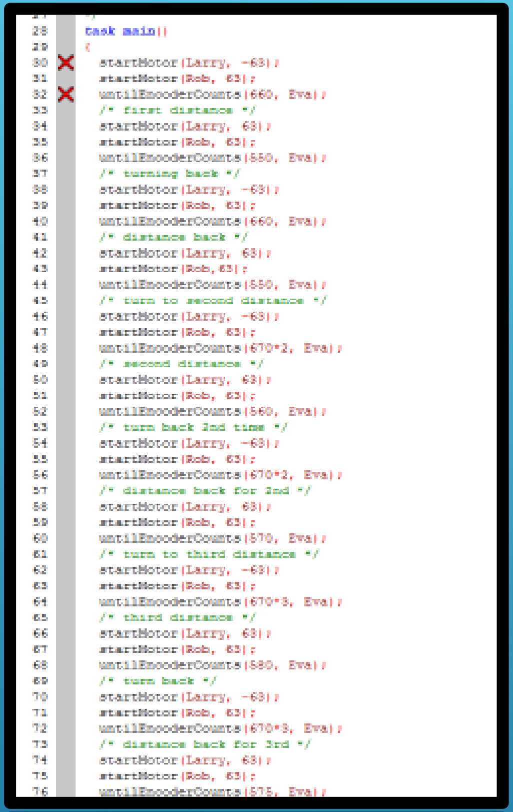

```
*/
task main()
{
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  wait(0.5);
  stopMotor(Larry);
  startMotor(Rob, 63);
  wait(1.2);
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  wait(0.7);
  stopMotor(Rob);
  startMotor(Larry, -63);
  wait(1.2);
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  wait(0.1);
  stopMotor(Rob);
  startMotor(Larry, -63);
  wait(1.3);
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  wait(0.4);
  stopMotor(Larry);
  stopMotor(Rob);
}
```

MAZE CHALLENGE

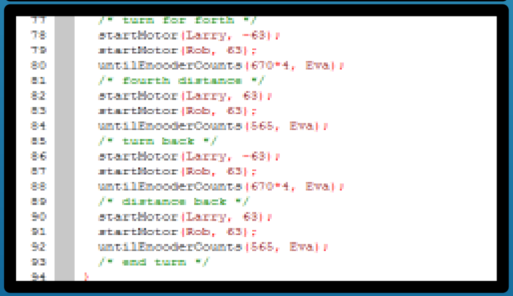
Because of the briefness of the challenge, only straight line coding was needed and the organization was relatively simple.

- Drive distances are less reliable on a run-down battery so a strategy was developed to avoid this. When programming began, the group made its best estimates on the time needed to both turn and drive forward. Later, the group tried to make most adjustments all at once so that the battery wasn't run down unnecessarily. By sticking to this strategy the frustration with the open loop system decreased.
- The motors were named Larry (left) and Rob (right) in the program.

SPRINT CHALLENGE



```
task main()
{
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  untilEncoderCounts(660, Eva);
  /* first distance */
  startMotor(Larry, 63);
  startMotor(Rob, 63);
  untilEncoderCounts(550, Eva);
  /* turning back */
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  untilEncoderCounts(660, Eva);
  /* distance back */
  startMotor(Larry, 63);
  startMotor(Rob,63);
  untilEncoderCounts(550, Eva);
  /* turn to second distance */
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  untilEncoderCounts(670*2, Eva);
  /* second distance */
  startMotor(Larry, 63);
  startMotor(Rob, 63);
  untilEncoderCounts(560, Eva);
  /* turn back 2nd time */
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  untilEncoderCounts(670*2, Eva);
  /* distance back for 2nd */
  startMotor(Larry, 63);
  startMotor(Rob, 63);
  untilEncoderCounts(570, Eva);
  /* turn to third distance */
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  untilEncoderCounts(670*3, Eva);
  /* third distance */
  startMotor(Larry, 63);
  startMotor(Rob, 63);
  untilEncoderCounts(580, Eva);
  /* turn back */
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  untilEncoderCounts(670*3, Eva);
  /* distance back for 3rd */
  startMotor(Larry, 63);
  startMotor(Rob, 63);
  untilEncoderCounts(575, Eva);
```



```
  /* turn for forth */
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  untilEncoderCounts(670*4, Eva);
  /* fourth distance */
  startMotor(Larry, 63);
  startMotor(Rob, 63);
  untilEncoderCounts(565, Eva);
  /* turn back */
  startMotor(Larry, -63);
  startMotor(Rob, 63);
  untilEncoderCounts(670*4, Eva);
  /* distance back */
  startMotor(Larry, 63);
  startMotor(Rob, 63);
  untilEncoderCounts(565, Eva);
  /* end turn */
}
```

Straight line coding was used due to limited time. On the left are screenshots of the group's code.

- Comments helped to keep track of the organization.
- For turns, the motors were run opposite to each other so that the robot rotated around its center.
- Also, to ensure correct distances were driven, both motors were run until a specific, finely-tuned encoder, named Eva, reached the desired count. This created a closed loop system.
- Unfortunately, there was trouble setting up a variable for the distance or the turns so after the attempt only straight line coding was used.

MAZE CHALLENGE

Despite a slow start due to difficulty in assembly, once coding was started the trouble-shooting and adjustments were completed easily. The challenge was finished without any difficulty with the open loop system.

Issues:

- There was a mix up in the size of the beams originally, so after realizing that the larger size would increase the difficulty of the challenge (the turns would be very tight), the beams were replaced with smaller ones.
- During assembly, the left side was built identically to the right because of an error in communication and the robot was running in reverse. It was fixed by making all the left motor values negative.

In the future, more time spent checking the build instructions will ensure they are carried out correctly.

Main TOC

SPRINT CHALLENGE

Extensive fine tuning was used during the challenge. The robot would drive well but would cross the black line. After trouble-shooting it completed the challenge easily.

Issues:

- There was little trouble in driving the correct distances but the turns had to be adjusted multiple times to get an exact 180 degree turn. If the turn was incorrect the robot would run outside the black sidelines.
- Without the suggestion to have the motors run opposite to each other, every time the robot would turn it would be slightly to the left and so on the second turn it was outside of the black lines.
- Next time, the turn should be tested separately until it is an exact 180 degree turn instead of trouble-shooting it while trouble-shooting the rest of the errors. Also, variables should be used to cut down the size of the program.

POE TOC

INTRODUCTION OF ENGINEERING DESIGN TABLE OF CONTENTS

Main TOC

COURSE DESCRIPTION

Students will employ engineering and scientific concepts in the solution of engineering design problems. In addition, students use a state of the 3D solid modeling design software package to help them design solutions to solve proposed problems. Students will develop problem -solving skills and apply their knowledge of research and design to create solutions to various challenges that increase in difficulty throughout the course. Students will also learn how to document their work, and communicate their solutions to their peers and members of the professional community.

Main TOC IED TOC

BRAINSTORMING (CEREAL BOWL)

In this assignment students practiced brainstorming skills by working in groups to design a cereal bowl. The goal of the project was to create a cereal bowl that would be functional and transport cereal effectively. Working on this project demonstrated the purpose of brainstorming, to generate creative ideas without discarding any.

Main TOC IED TOC

BRAINSTORMING (CEREAL BOWL)

Requirements:

- Improve on basic cereal bowl
- Make a group presentation of the design in front of class
- Design must transport cereal effectively

Constraints:

- 3-4 group members
- One class period to complete
- 2 paper cereal bowls maximum, only a few sheets of construction paper, markers, tape, and staples to construct prototype

BRAINSTORMING (CEREAL BOWL)

Brainstorming Ideas:

- Lid
- Carrier for chest
- Collapsible bowl
- Smoothie bowl
- Cup
- Backpack container
- Gyro bowl
- Baseball hat
- Attachable to bike
- Two separate bowls
- Bowl attached to face
- Divider
- Premade mixer bowl
- Insulated bowl
- Bowl attached to Glove
- Tube shaped bowl
- Car shaped bowl

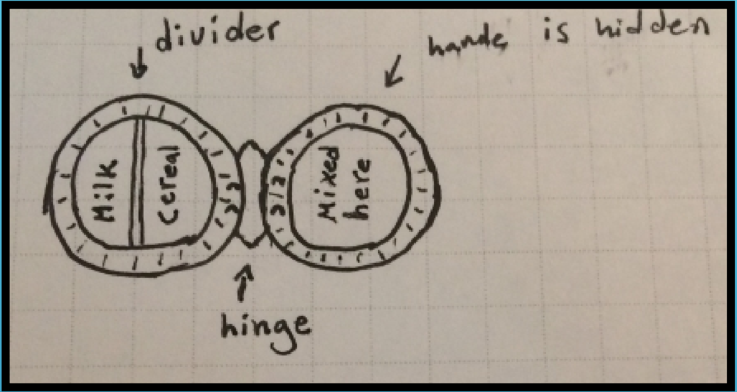
Final Bowl Design:

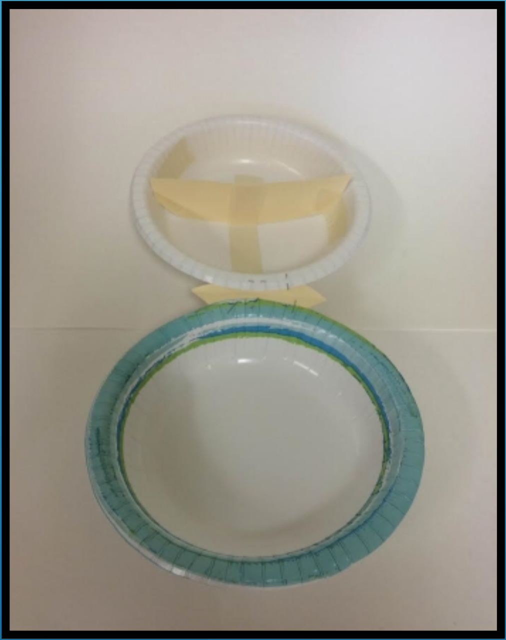
- Two separate bowls
- Mix cereal and milk when wanted
- Hinge to attach bowls together
- Handle
- Divider (only in one of the bowls)

BRAINSTORMING (CEREAL BOWL)

The prototype was built with construction paper, paper bowls, staples, and tape. The bowl is made to have one side unmixed with separate compartments for cereal and milk and then is attached by a hinge to the other side that will be mixed. In future versions a piece of plastic would separate the two bowls so they would stay separate until ready to be mixed.

Sketch and Picture:

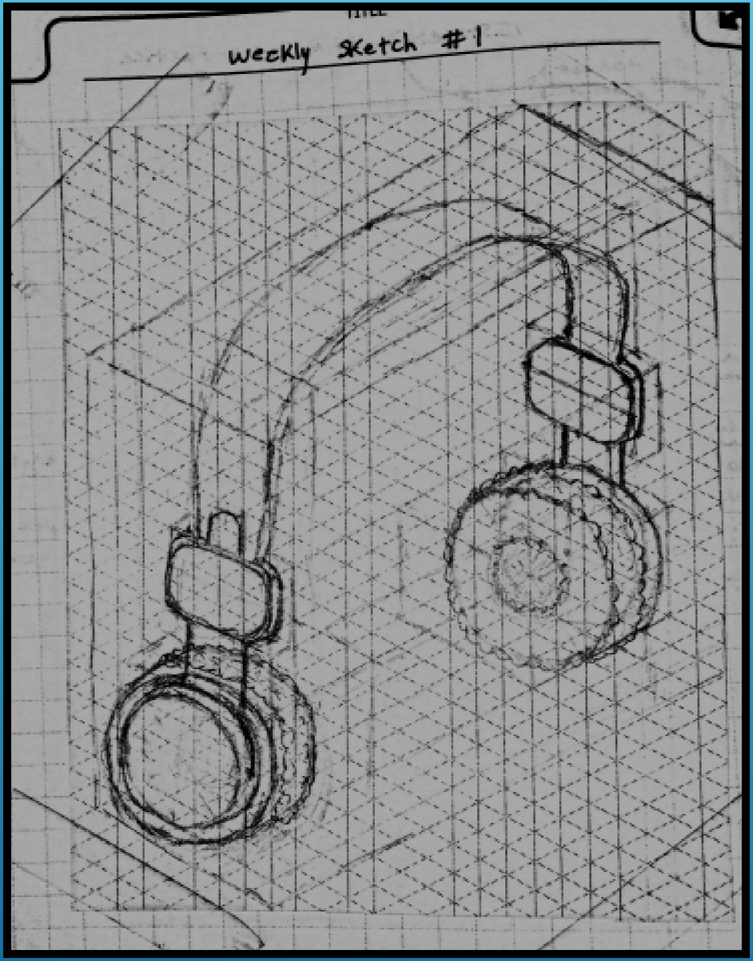




Main TOC

IED TOC

WEEKLY SKETCHES



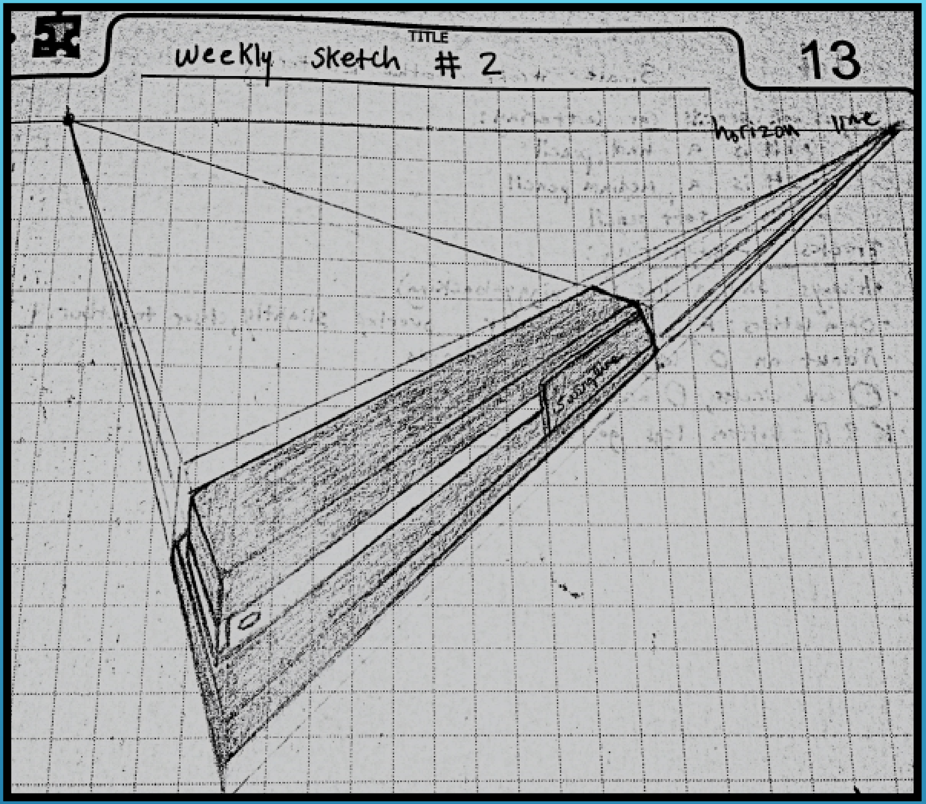
Weekly sketches allow students to polish up their technical drawing and rough draft sketches for future projects. These sketches should be pictorial sketches: perspective, isometric, or oblique.

As shown on this page, isometric sketches have edges in the front face that appear as 30 degrees from the horizon while side face edges are at 120 degrees from it.

Main TOC

IED TOC

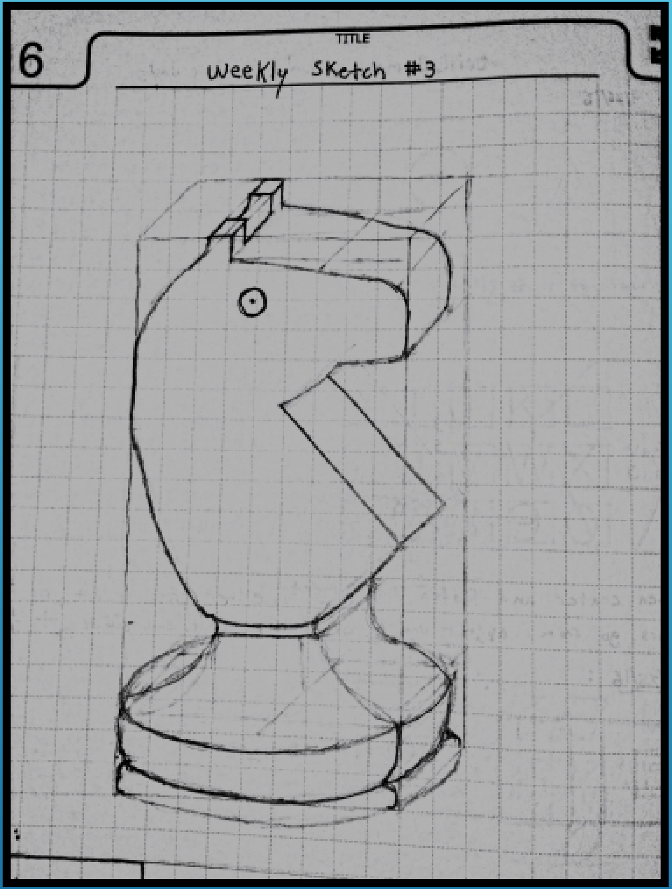
WEEKLY SKETCHES



This sketch is a second perspective sketch. Perspective sketches offer most realistic three-dimensional view of the pictorial sketches. The horizontal line that represents the horizon has one or more vanishing points. A series of lines drawn from points on the object to the vanishing point(s) make the object appear from a different view point.

Main TOC IED TOC

WEEKLY SKETCHES

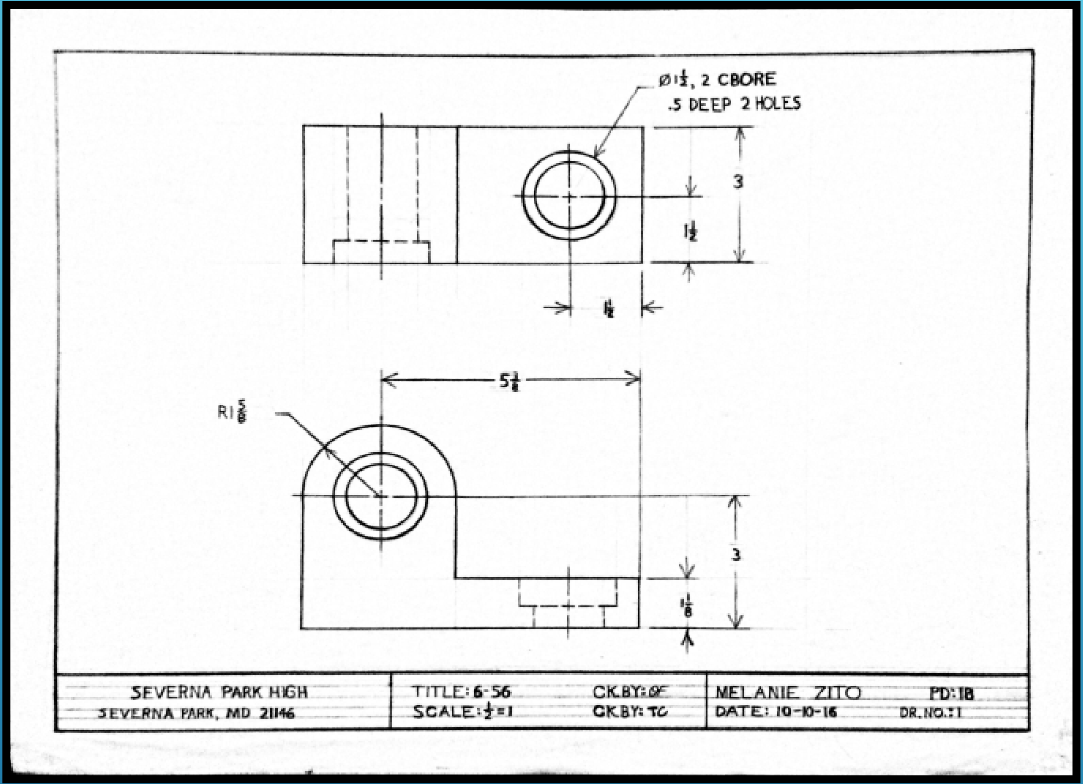


Oblique sketches emphasizes the front of an object. Also, cavalier (rather than cabinet) obliques such as the one shown, emphasizes the depth by twice it's actual dimension. The other faces are set at 45 degrees from the front face.

Main TOC

IED TOC

BOARD ORTHOGRAPHICS



SEVERNA PARK HIGH
SEVERNA PARK, MD 21146
TITLE: 6-56
SCALE: ½=1
CK.BY: [illegible]
CK.BY: TC
MELANIE ZITO
DATE: 10-10-16
PD: 1B
DR.NO.: 1

Board orthographics are technical drawings drawn by hand on a flat surface. This drawing (6-56) was completed to learn proper dimensioning and professional technique.

These skills later aided students in computerized drawing when they had to use advanced dimensioning and features.

Main TOC

IED TOC

PUZZLE PROJECT (DESIGN PROCESS)

In this project, students designed a puzzle cube and then used Inventor, an adaptive 3D modeling program, to create a computerized model of the design. Each of the puzzle pieces were created and then assembled using Inventor.

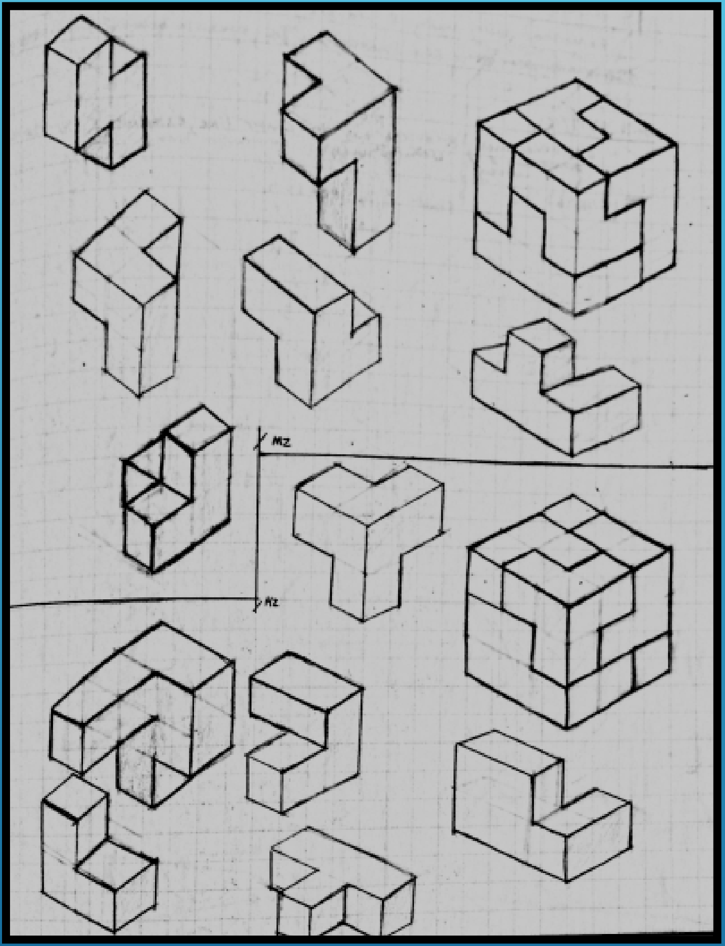
Constraints:

- 24 cubes that are 0.75”x0.75”x0.75”
- Only one puzzle piece may be duplicated
- Must have at least 4 different pieces
- Each piece must have components on the x, y, and z axes.

Requirements:

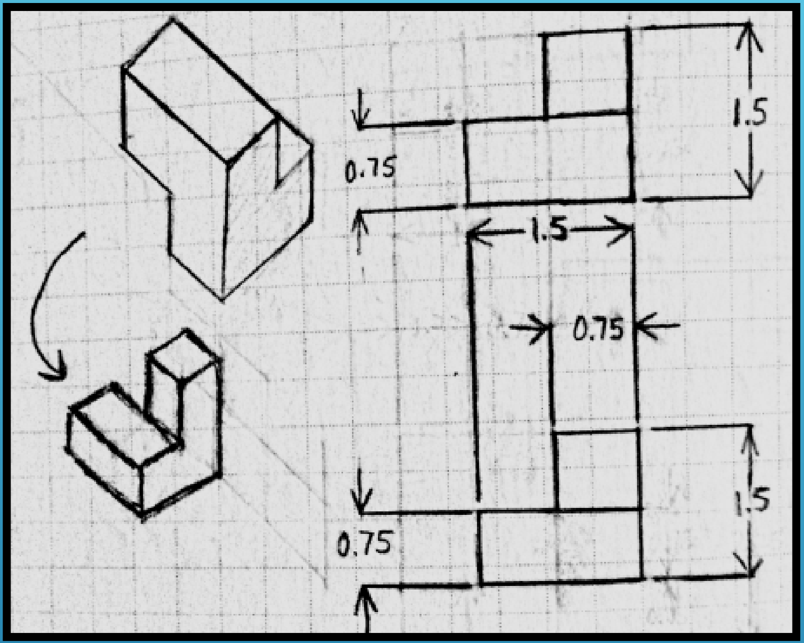
- All puzzle pieces must assemble together in a puzzle cube

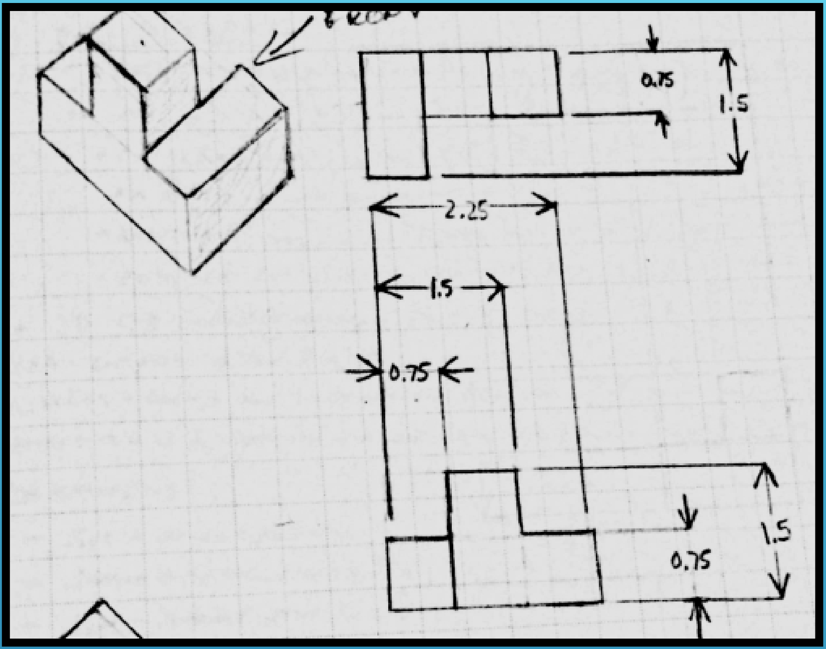
PUZZLE PROJECT (DESIGN PROCESS)



Thumbnail Sketches were drawn to express students' original design ideas for their puzzle cubes. These isometric sketches let them compare two possible options for the cube while also visualizing how the individual pieces fit together.

PUZZLE PROJECT (DESIGN PROCESS)

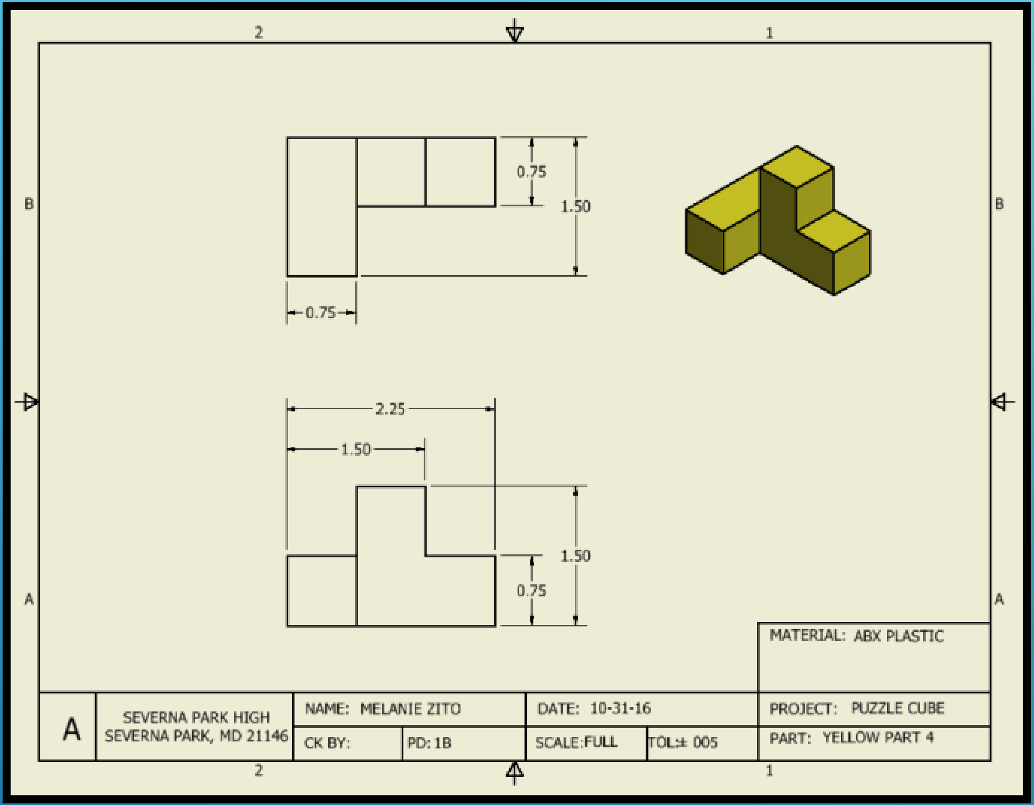




After deciding on their first puzzle cube idea from the thumbnails, students drew rough drafts for each of the puzzle pieces and dimensioned the multi-view sketches. Later, these sketches allowed them to easily recreate the pieces in Inventor.

Main TOC IED TOC

PUZZLE PROJECT (DESIGN PROCESS)



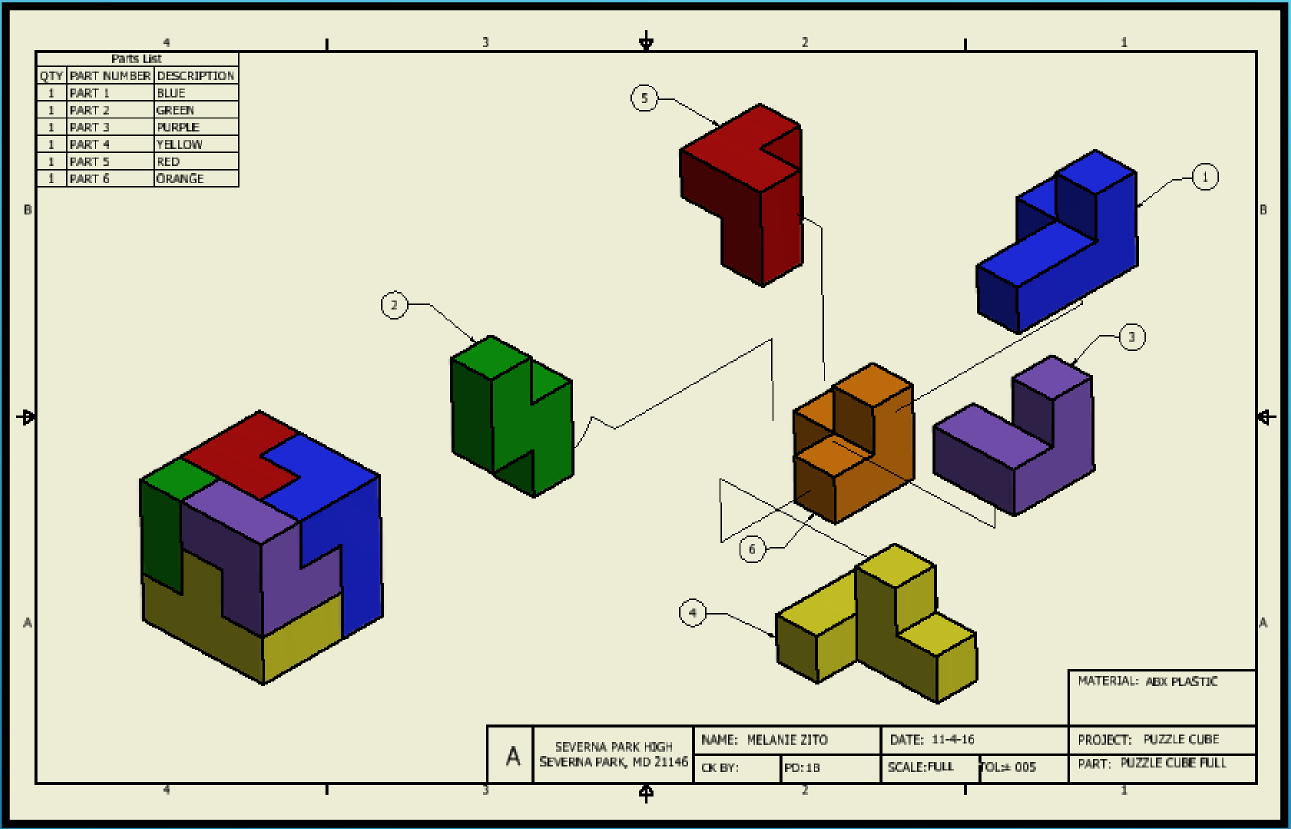
MATERIAL: ABX PLASTIC

A	SEVERNA PARK HIGH SEVERNA PARK, MD 21146	NAME: MELANIE ZITO	DATE: 10-31-16	PROJECT: PUZZLE CUBE
		CK BY: PD: 1B	SCALE:FULL TOL:± 005	PART: YELLOW PART 4

Once the puzzle pieces were created in Inventor, students used a template to dimension each of their pieces, similar to the technique used for the original rough drafts in their engineers notebooks. The isometric view is included for clarity.

Orthographic drawings shows the manufacturer the true dimensions, layouts, and anything else the manufacturer needs to reproduce the part.

PUZZLE PROJECT (DESIGN PROCESS)



Parts List

QTY	PART NUMBER	DESCRIPTION
1	PART 1	BLUE
1	PART 2	GREEN
1	PART 3	PURPLE
1	PART 4	YELLOW
1	PART 5	RED
1	PART 6	ORANGE

MATERIAL: ABX PLASTIC

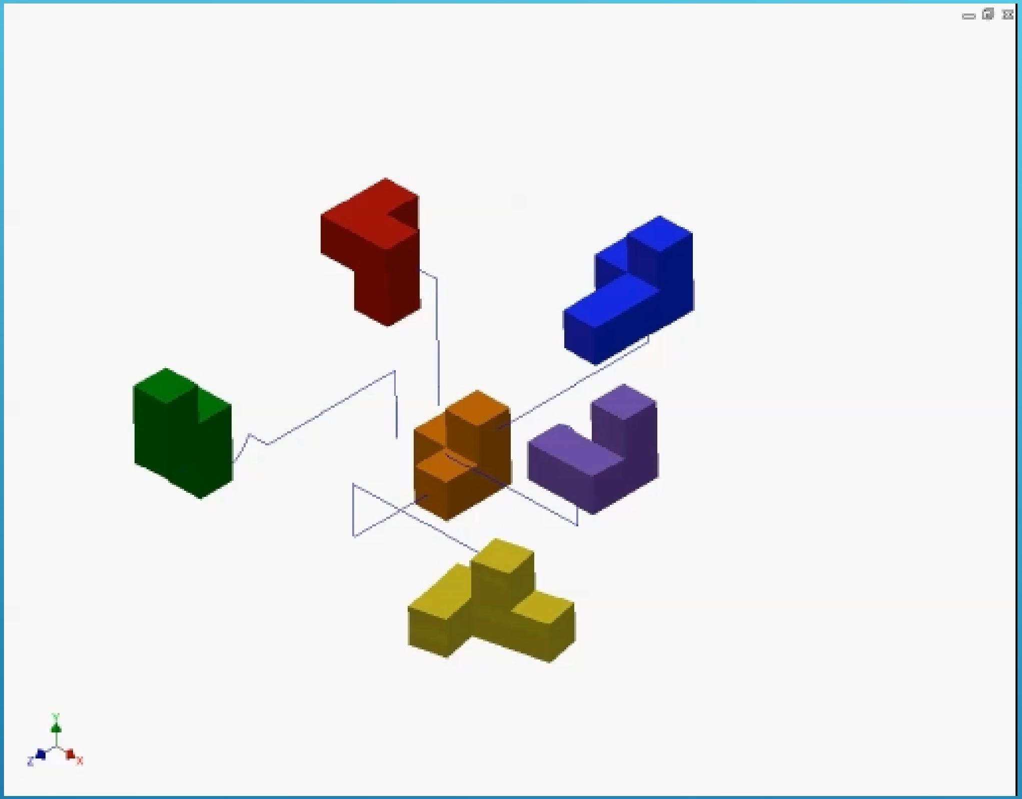
A	SEVERNA PARK HIGH SEVERNA PARK, MD 21146	NAME: MELANIE ZITO		DATE: 11-4-16		PROJECT: PUZZLE CUBE
		CK BY:	PD:1B	SCALE:FULL	TOL:± 005	PART: PUZZLE CUBE FULL

After the pieces were created, students could reference their original thumbnails to assemble their cube together and place it on the template as well.

Main TOC

IED TOC

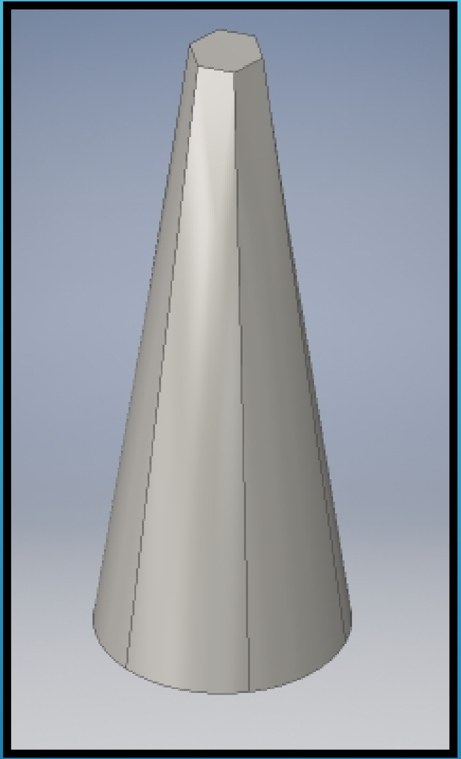
PUZZLE PROJECT (DESIGN PROCESS)



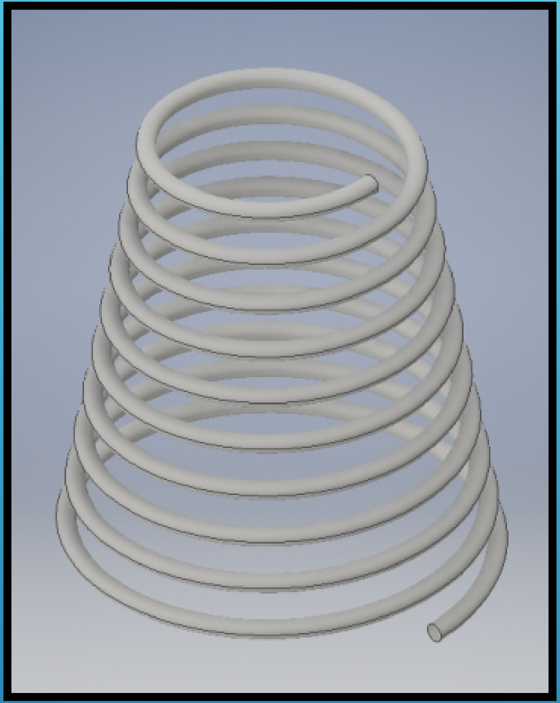
Finally, students used Inventor's presentation file to animate their puzzle pieces fitting together into their puzzle cube design.

Click twice on the image or click the play button on the bottom left part of image to play the animation.

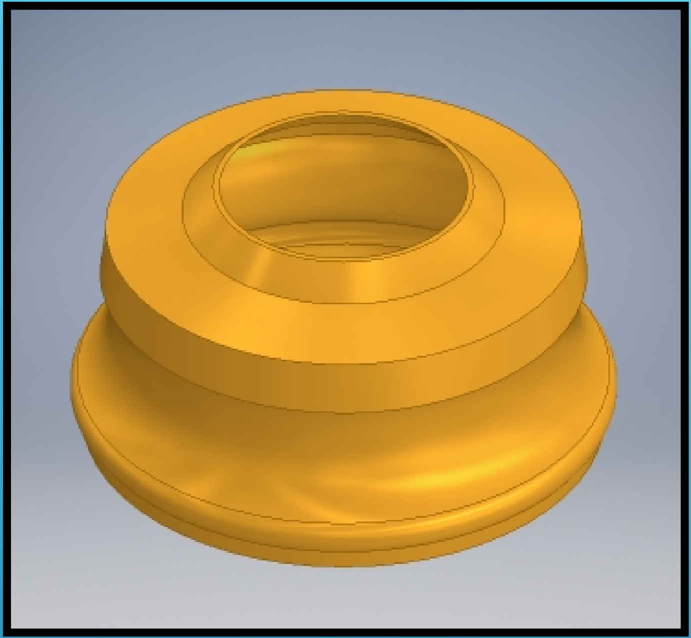
INVENTOR FEATURES



This design (above) was created using the loft tool. The loft tool creates a shape between two different work planes.

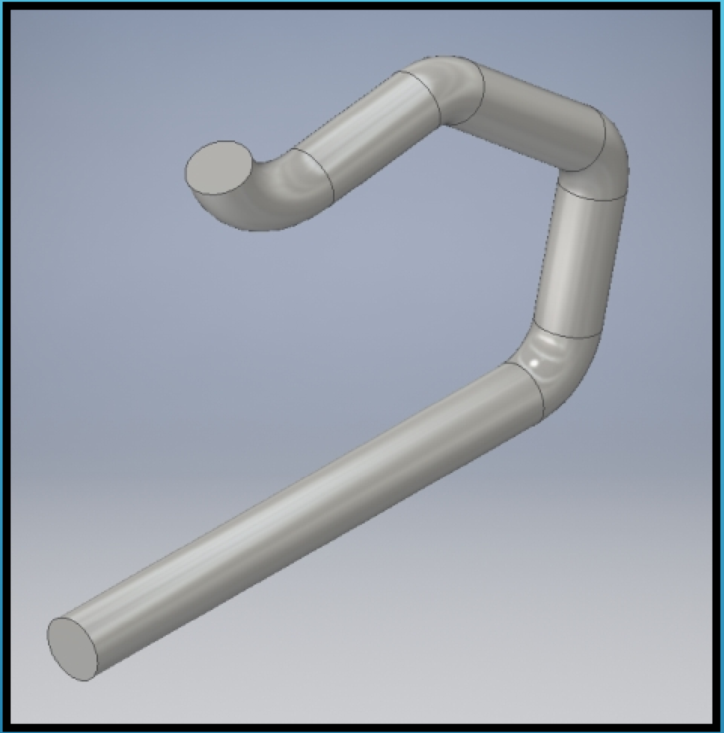


The above design was made by the coil tool from a sketch of a circle and a chosen axis. One can decide the specific pitch, radius, thickness of the coil, and number of revolutions needed.

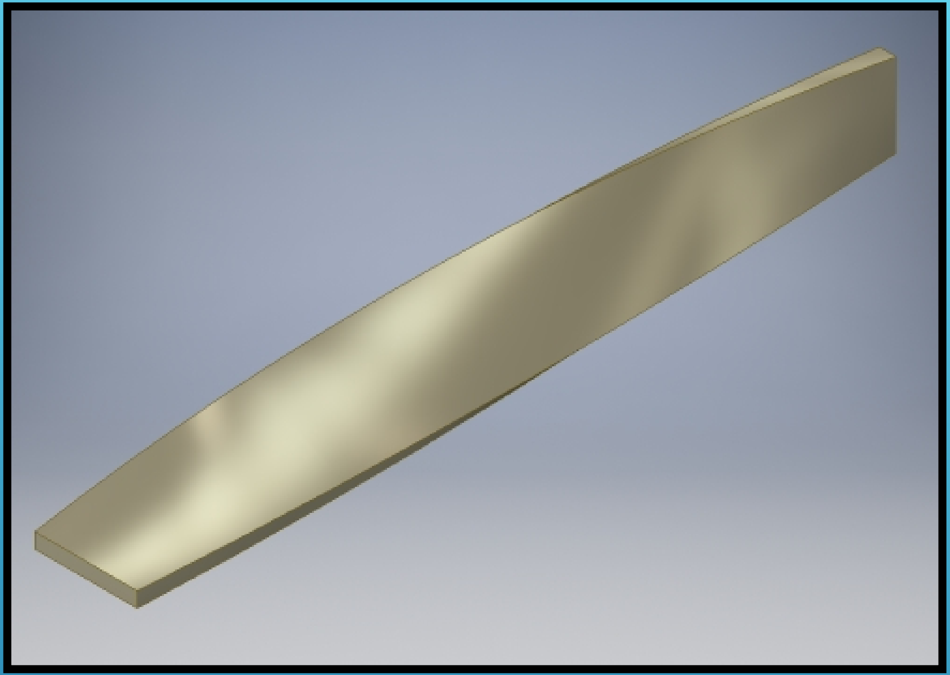


The revolve tool creates designs such as the one above by revolving a plane around an axis of the user's choosing.

INVENTOR FEATURES



This design (above) was created using the sweep feature. First, make a 2D circle on the XZ plane and then make another 2D sketch on the XY plane of the desired path one wants the sweep to take. Finally, use the sweep feature to create it.



The twist feature created the above design through sketches on three different work planes. These sketches were rotated so that the design would be twisted.

Main TOC

IED TOC

PARAMETRIC CONSTRUCTION (CHESS PIECE)

For this project, students designed a chess piece in Inventor as part of a themed chess set. Each piece and the chess board were designed by a different group member.

Constraints:

- 5-6 members in a group
- All pieces must be in metric
- Piece utilizes at least 2 different Inventor features
- Chess piece must be completely parametric
- Each of the chess pieces must less than or equal to 45mm long by 45mm wide
- Rook size (my piece) must be 38.1-44.5mm tall

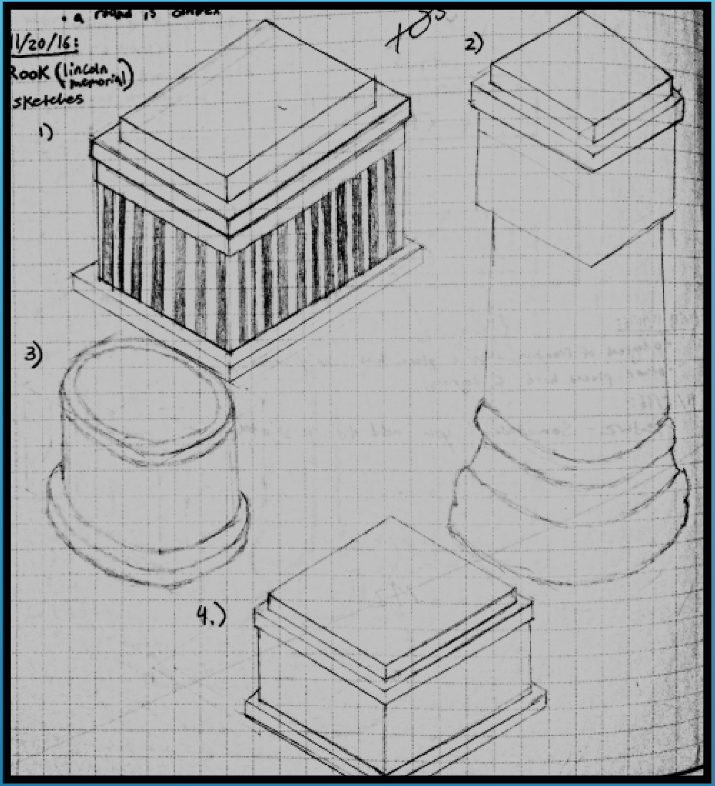
Requirements:

- All of the group's chess pieces must fit a theme of their choosing

Main TOC

IED TOC

PARAMETRIC CONSTRUCTION (CHESS PIECE)

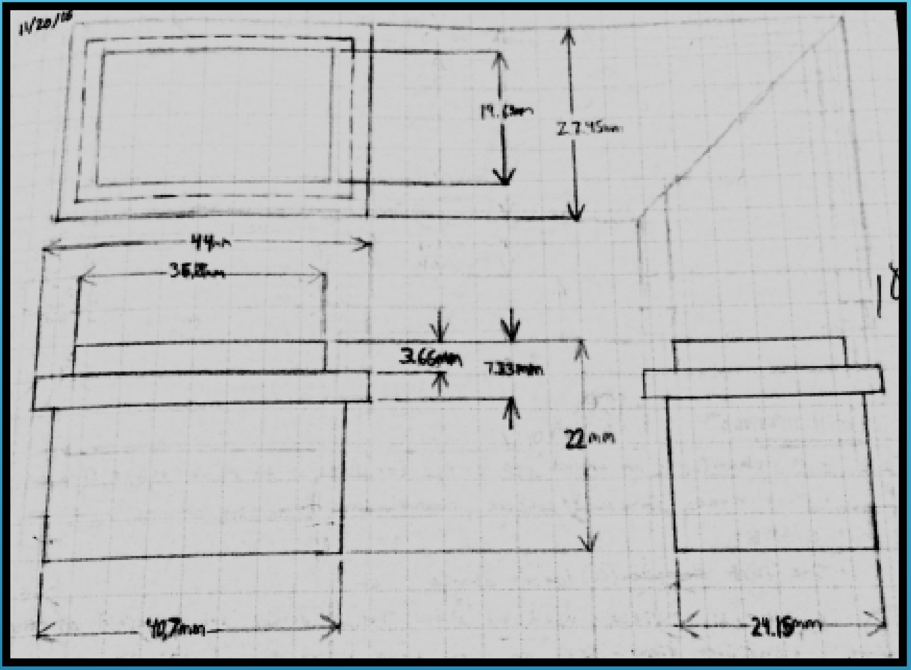


The team drew 4 different thumbnails to visually depict their design ideas for their chess pieces, in this case, the rook. Since the group's theme was Washington monuments, the Lincoln memorial was chosen for the rook. Although the first and fourth option were both good candidates, the first option expressed the Lincoln memorial more realistically and was chosen.

Main TOC

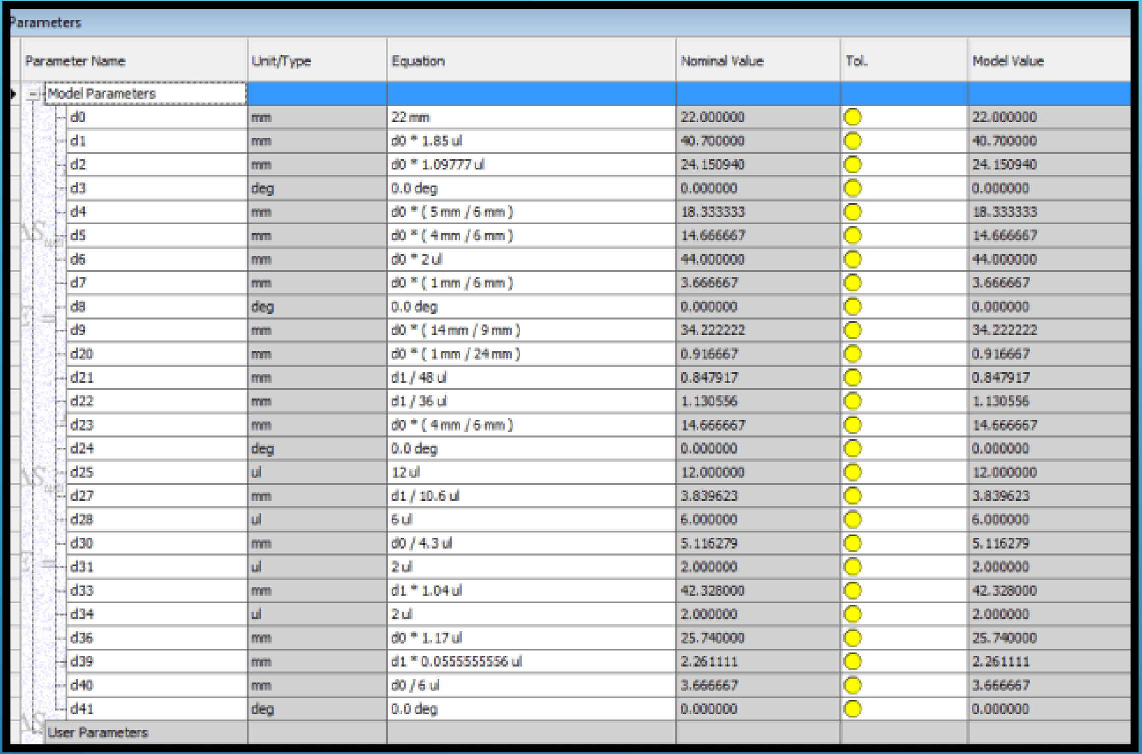
IED TOC

PARAMETRIC CONSTRUCTION (CHESS PIECE)



The rough draft for the chosen design was drawn and dimensioned to create a reference for when this process was repeated using Inventor. Initially, the idea was for a pattern of pillars on the side of the chess piece but this was impossible with the tools provided. So, in the later Inventor design, actual pillars were extruded and the dimensioning for the final orthographic was adjusted.

PARAMETRIC CONSTRUCTION (CHESS PIECE)



Parameters

Parameter Name	Unit/Type	Equation	Nominal Value	Tol.	Model Value
Model Parameters					
d0	mm	22 mm	22.000000	●	22.000000
d1	mm	d0 * 1.85 ul	40.700000	●	40.700000
d2	mm	d0 * 1.09777 ul	24.150940	●	24.150940
d3	deg	0.0 deg	0.000000	●	0.000000
d4	mm	d0 * (5 mm / 6 mm)	18.333333	●	18.333333
d5	mm	d0 * (4 mm / 6 mm)	14.666667	●	14.666667
d6	mm	d0 * 2 ul	44.000000	●	44.000000
d7	mm	d0 * (1 mm / 6 mm)	3.666667	●	3.666667
d8	deg	0.0 deg	0.000000	●	0.000000
d9	mm	d0 * (14 mm / 9 mm)	34.222222	●	34.222222
d20	mm	d0 * (1 mm / 24 mm)	0.916667	●	0.916667
d21	mm	d1 / 48 ul	0.847917	●	0.847917
d22	mm	d1 / 36 ul	1.130556	●	1.130556
d23	mm	d0 * (4 mm / 6 mm)	14.666667	●	14.666667
d24	deg	0.0 deg	0.000000	●	0.000000
d25	ul	12 ul	12.000000	●	12.000000
d27	mm	d1 / 10.6 ul	3.839623	●	3.839623
d28	ul	6 ul	6.000000	●	6.000000
d30	mm	d0 / 4.3 ul	5.116279	●	5.116279
d31	ul	2 ul	2.000000	●	2.000000
d33	mm	d1 * 1.04 ul	42.328000	●	42.328000
d34	ul	2 ul	2.000000	●	2.000000
d36	mm	d0 * 1.17 ul	25.740000	●	25.740000
d39	mm	d1 * 0.0555555556 ul	2.261111	●	2.261111
d40	mm	d0 / 6 ul	3.666667	●	3.666667
d41	deg	0.0 deg	0.000000	●	0.000000
User Parameters					

Parametric construction creates each dimension off of a single dimension (d0). Then, when building objects parametrically, the entire object size can be changed by changing the original d0 dimension.

Example: the dimension for the height, d0, is 22mm and the dimension for the length, d1, needs to be 44mm. So the dimension for d1 is entered as d0*2

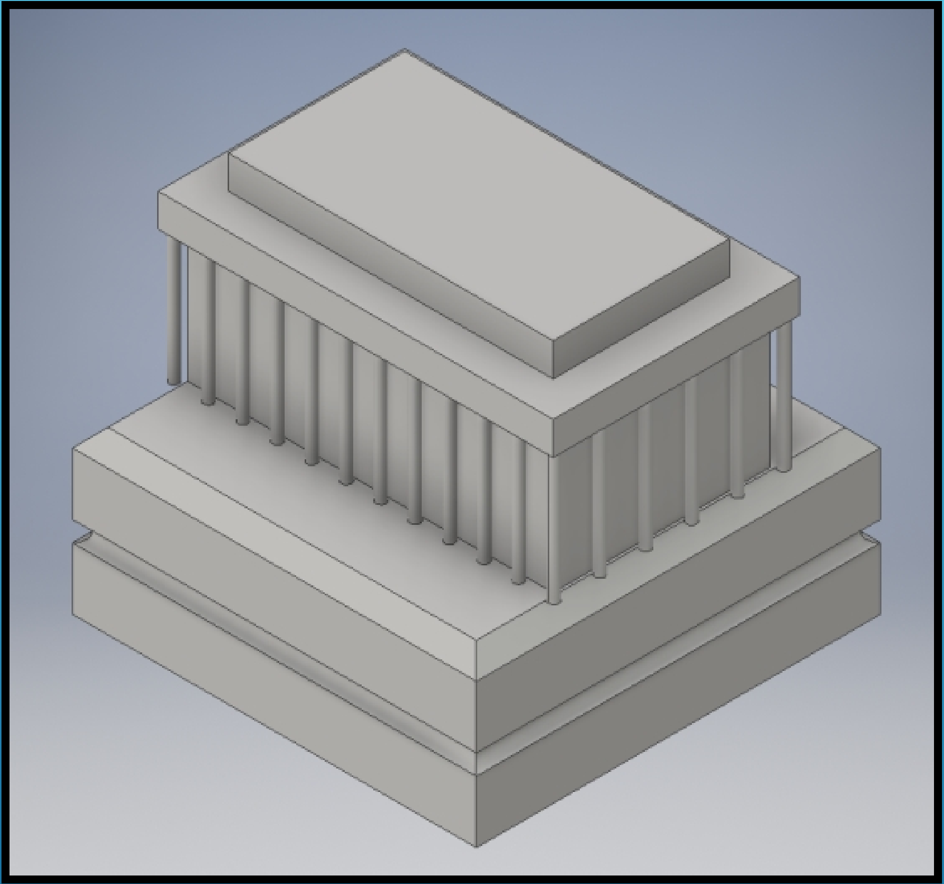
Main TOC

IED TOC

PARAMETRIC CONSTRUCTION (CHESS PIECE)

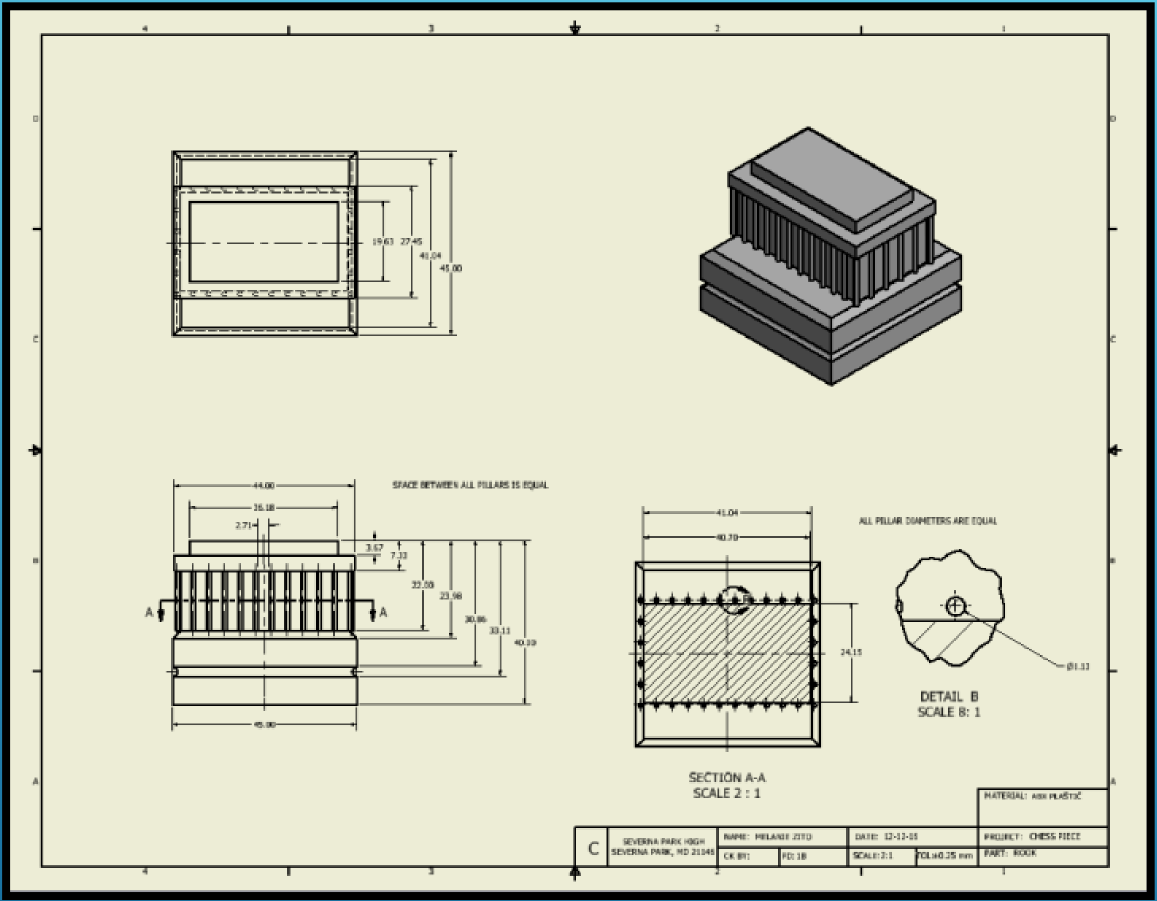
After creating the rook design the group base design had to be attached. To do this the chess piece was derived into the file for the base, creating a separate part. A derived part is a new part that references other existing parts to create a new piece. In this case, the new derived part used the parts and parametrics from the original files for the chess piece and the base to place them together into the full piece.

PARAMETRIC CONSTRUCTION (CHESS PIECE)



The finished derived part used a symmetrical 45mm by 45 mm base and the rook design. As this advanced model view shows, the final chess piece used the original thumbnail with extruded pillars.

PARAMETRIC CONSTRUCTION (CHESS PIECE)



Similar to the final isometric piece, the Inventor orthographic was adjusted to dimension to the pillars added. A section and detail view were included to show the dimensions of the pillars in greater detail.

Main TOC

IED TOC