Design Team-Materials List, Spring 2015

Ewa Przybylko & Stephanie Sun

May 12, 2015

Table of Contents

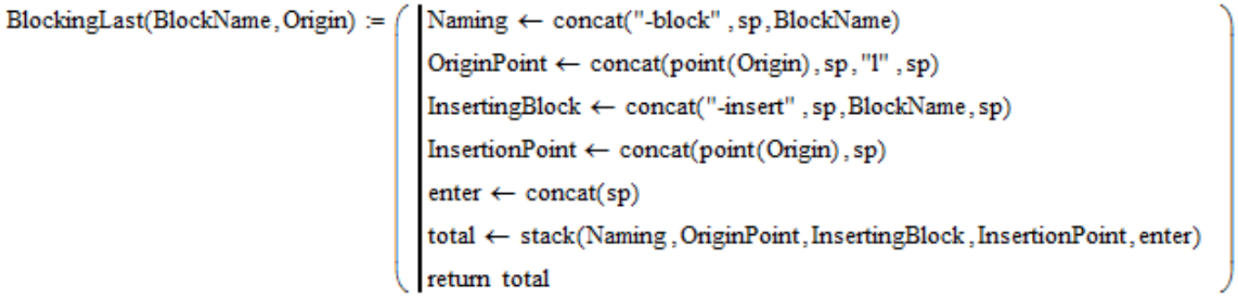
Part I: Problem Definition

Introduction

The Spring 2015 Materials List design task was a continuation of previous efforts to create an automated materials list that could readily be extracted from AutoCAD to facilitate the construction of AguaClara treatment plants. In past semesters, AguaClara Design Team members investigated converting plant components–namely pipes and accessories– to countable “blocks” in AutoCAD based on unique component specifications. This method allowed for the efficient extraction of a “materials list” from any given AutoCAD drawing, which listed the block types, their counts, as well as dimension specifications of the components contained within the drawing. The goal of the Spring 2015 Materials List team was to build on the idea of “blocking” proposed by the Fall 2014 Design Team. Specifically, the present team worked on revising blocking functions implemented in the past and creating new functions to block additional plant components. The team also investigated the possibility of extracting the materials list using the AutoCAD command line to make the data easily accessible to users abroad. If testing proved successful, the ultimate goal of the team was to incorporate and implement the blocking functions into all the official AguaClara drawing codes.

Design Details

The conversion of AguaClara plant components to “blocks” required the integration of MathCAD code with various AutoCAD functions. In fall of 2014, the Design Team wrote several blocking codes in MathCAD using the AutoCAD “-block” function. Their functions, namely “BlockingLast,” formed the basis for this present investigation (Figure 1).



```
BlockingLast(BlockName, Origin) := | Naming ← concat("-block" , sp, BlockName)
                                   | OriginPoint ← concat(point(Origin), sp, "1" , sp)
                                   | InsertingBlock ← concat("-insert" , sp, BlockName, sp)
                                   | InsertionPoint ← concat(point(Origin), sp)
                                   | enter ← concat(sp)
                                   | total ← stack(Naming, OriginPoint, InsertingBlock, InsertionPoint, enter)
                                   | return total
```

Figure 1: BlockingLast Function

These MathCAD codes were combined with previous plant drawing functions to allow for the simultaneous drawing and blocking of the individual plant components. The function “BlockingLast” used inputs of “BlockName” and “Origin” to carry out the AutoCAD “-block” function, concatenating the commands that would typically be manually inputted into the AutoCAD command line. Similarly, the function “copyblock” used inputs of “origin” and “insertionpoint” to replicate a previously drawn block component.

After examining and testing how the existing blocking functions were incorporated into the drawing code for the Entrance Tank by the Fall 2014 Design Team, the Spring 2015 Design Team continued to apply the blocking functions to the rest of the Entrance Tank. Through this process, constraints on the “BlockingLast” function were discovered. The team realized that AutoCAD objects cannot be subtracted from blocks, and therefore components of the plant with subtracted parts must be blocked after the subtraction process in order to be drawn correctly. In addition, when blocking components with subtracted parts, each object must be drawn in a separate layer as the “BlockingLast” function will not work correctly otherwise. The goal for this semester was to further test the blocking functions with all possible plant components and resolve any issues that may inhibit proper AutoCAD drawing and blocking execution. This objective was achieved as the blocking functions were incorporated into the entire Entrance Tank drawing code and a full Excel materials list was extracted.

Part II: Documentation

Week of February 23, 2015

Accomplishments

- The team explored the possibility of extracting an Excel spreadsheet containing all the blocking information directly from AutoCAD command line.
- A way to avoid the Data Extraction Wizard dialogue box when extracting the data to a previously created template was found, and existing .xls files (that are linked to the template) were updated by overwriting them.

Problems Encountered

- No way could be found to create a new .dxe template file path directly from the command line. The team then attempted to extract data from various AutoCAD drawings to a “template” file path created manually in a separate .dwg drawing.

- This method proved unsuccessful, as the “template” file path did not recognize the blocks contained in the various drawing files since it was not created specifically for those drawings.
- After extensive trial and error, the team came to the conclusion that a single template could not be used for multiple drawings if the .dwg was not explicitly linked to the .dxe template.
- Another method attempted was to create a new “dummy” template that was linked to every available blocked AutoCAD drawing. This was based on the idea that such a template would be able to recognize all the various blocks that might be contained in the drawing.
- However, when extracting the data from a single drawing using the “template,” the resulting Excel sheet listed ALL the blocks referenced in the template, rather than just those present in the drawing-in-question, and it did not update their counts (did not return zero if the blocks were not present).
- After repeated experimentation, it was determined that the only way to automate the data extraction process is to determine a way to create a new .dxe template from the command line.
- The team is considering reaching out to AutoDesk for additional assistance in this process, as this doesn’t seem to be a current AutoCAD capability.

Week of March 2, 2015

Accomplishments

- The team explored the MathCAD functions for blocking written by last year’s design team and recreated the blocked drawings for the Entrance Tank components to get a better understanding of how the blocking functions work.
- A blocking function, called “EtDrainPipes,” that would block the drain pipes for the entrance tank was created. Using the blocking script for “EtDrainCouplings” as a guideline, a script to draw, array, and block the 3 pipes was created.

Problems Encountered

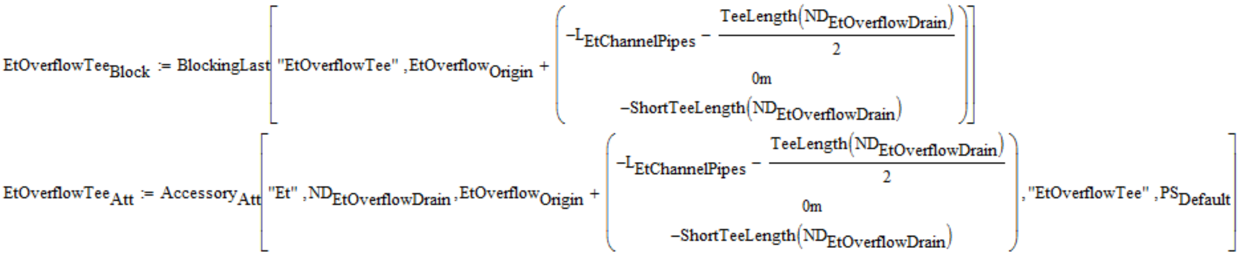
- When the script was run in AutoCAD, the team noticed that the pipes were being drawn, but not blocked.

- The team attempted to trace the error by copying parts of the text file output, section by section, into AutoCAD and noticed some potential bugs in the blocking code that were inhibiting AutoCAD from blocking new components.
- For example, the AutoCAD command line was expecting inputs such a scaling and rotation factor, which were not accounted for in the MathCAD blocking code.
- While examining the MathCAD code line by line to note differences between the blocking function and AutoCAD's expectations, instances in the code were found where an input of "enter" would be more appropriate than "space."

Week of March 9, 2015

Accomplishments

- The team successfully blocked the following Entrance Tank Plant Components:
  - EtDrainPipes
  - EtStopperPipes
  - EtOverflowTee (Figure 2 & 3)
  - EtOverflowCap1
  - EtOverflowCap2
  - EtOverflowPipeVert
  - EtOverflowChannelPipe1
  - EtOverflowChannelPipe2
  - EtInletElbow
  - EtInletPipe
  - EtInletElbow2
  - EtInletPipe2
  - EtFlowControlValve (Grit)



$$\mathrm{EtOverflowTee}_{\mathrm{Block}} := \mathrm{BlockingLast}\left[\text{"EtOverflowTee"}, \mathrm{EtOverflow}_{\mathrm{Origin}} + \begin{pmatrix} -L_{\mathrm{EtChannelPipes}} - \dfrac{\mathrm{TeeLength}\left(\mathrm{ND}_{\mathrm{EtOverflowDrain}}\right)}{2} \\ 0\mathrm{m} \\ -\mathrm{ShortTeeLength}\left(\mathrm{ND}_{\mathrm{EtOverflowDrain}}\right) \end{pmatrix}\right]$$

$$\mathrm{EtOverflowTee}_{\mathrm{Att}} := \mathrm{Accessory}_{\mathrm{Att}}\left[\text{"Et"}, \mathrm{ND}_{\mathrm{EtOverflowDrain}}, \mathrm{EtOverflow}_{\mathrm{Origin}} + \begin{pmatrix} -L_{\mathrm{EtChannelPipes}} - \dfrac{\mathrm{TeeLength}\left(\mathrm{ND}_{\mathrm{EtOverflowDrain}}\right)}{2} \\ 0\mathrm{m} \\ -\mathrm{ShortTeeLength}\left(\mathrm{ND}_{\mathrm{EtOverflowDrain}}\right) \end{pmatrix}, \text{"EtOverflowTee"}, \mathrm{PS}_{\mathrm{Default}}\right]$$

Figure 2: EtOverflowTee blocking and attribute functions

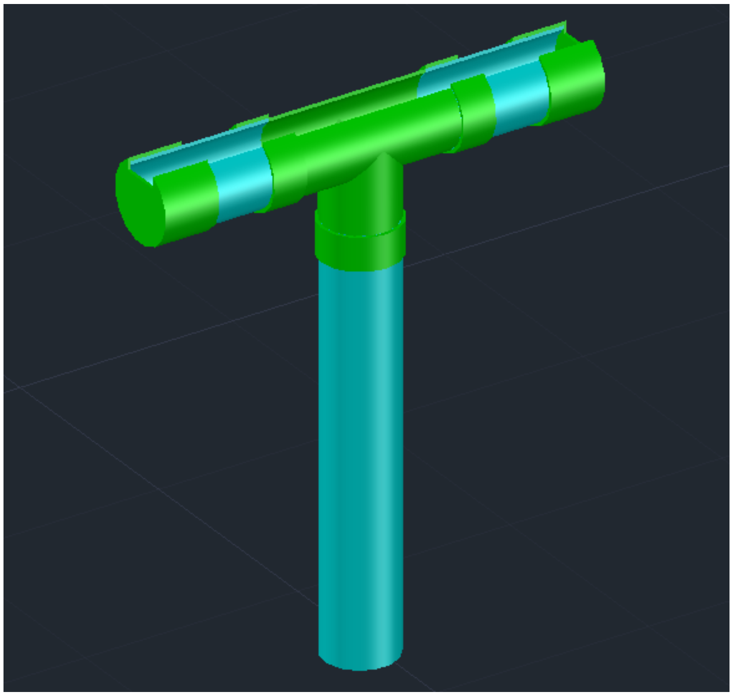
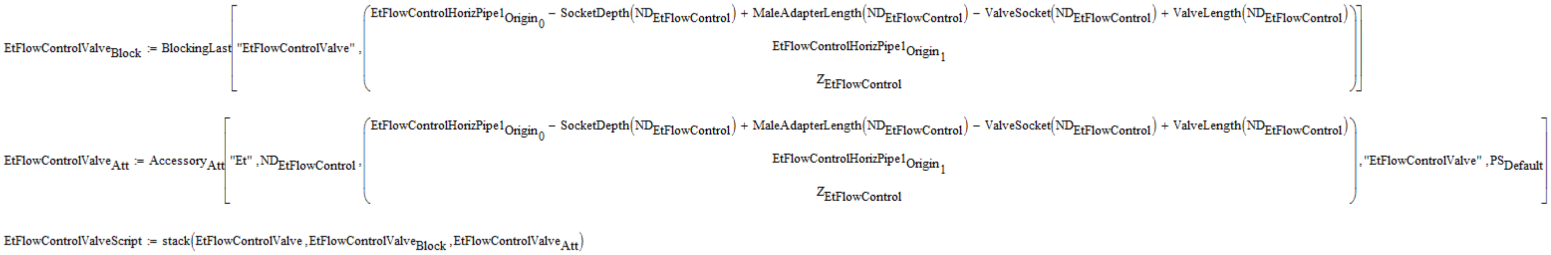


Figure 3: Entrance Tank Overflow Tee AutoCAD drawings

- For each individual part of the plant component, a separate blocking and attribute function was made, and then the drawing, block, and attribute functions were stacked into a script (Figure 4). These stacks were then stacked into a master stack for the whole component (Figure 5).



$$\text{EtFlowControlValve}_{\text{Block}} := \text{BlockingLast}\left[\text{"EtFlowControlValve"},\begin{pmatrix}\text{EtFlowControlHorizPipe1}_{\text{Origin}_0}-\text{SocketDepth}(\text{ND}_{\text{EtFlowControl}})+\text{MaleAdapterLength}(\text{ND}_{\text{EtFlowControl}})-\text{ValveSocket}(\text{ND}_{\text{EtFlowControl}})+\text{ValveLength}(\text{ND}_{\text{EtFlowControl}})\\ \text{EtFlowControlHorizPipe1}_{\text{Origin}_1}\\ Z_{\text{EtFlowControl}}\end{pmatrix}\right]$$

$$\text{EtFlowControlValve}_{\text{Att}} := \text{Accessory}_{\text{Att}}\left[\text{"Et"},\text{ND}_{\text{EtFlowControl}},\begin{pmatrix}\text{EtFlowControlHorizPipe1}_{\text{Origin}_0}-\text{SocketDepth}(\text{ND}_{\text{EtFlowControl}})+\text{MaleAdapterLength}(\text{ND}_{\text{EtFlowControl}})-\text{ValveSocket}(\text{ND}_{\text{EtFlowControl}})+\text{ValveLength}(\text{ND}_{\text{EtFlowControl}})\\ \text{EtFlowControlHorizPipe1}_{\text{Origin}_1}\\ Z_{\text{EtFlowControl}}\end{pmatrix},\text{"EtFlowControlValve"},\text{PS}_{\text{Default}}\right]$$

$$\text{EtFlowControlValveScript} := \text{stack}(\text{EtFlowControlValve},\text{EtFlowControlValve}_{\text{Block}},\text{EtFlowControlValve}_{\text{Att}})$$

Figure 4: EtFlowControlValve Stack

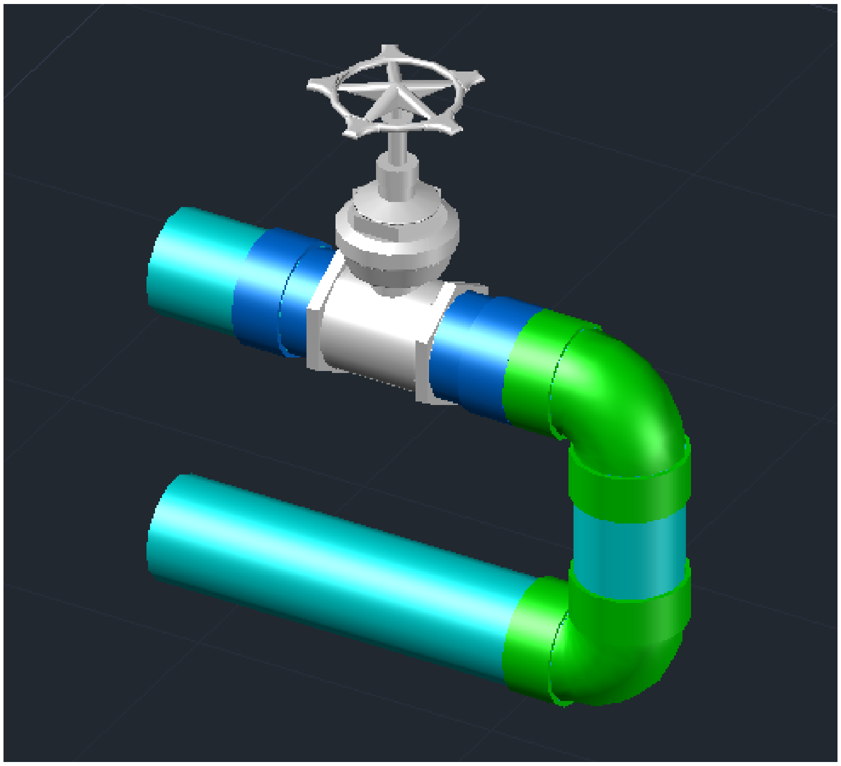


Figure 5: Entrance Tank Flow Control Drain Grit AutoCAD drawing

Problems Encountered

- In blocking the Entrance Tank Overflow Drain, there were problems when actually drawing the components in AutoCAD.
- The plant components were successfully blocked, but AutoCAD was not able to subtract blocks from other blocks or objects. Therefore, AutoCAD was simply drawing the whole pipes and tee without subtracting a rectangular weir slot from the top.
- The team explored whether or not there is a way to subtract blocks from objects, and considered writing a new subtract function for blocks.

Week of March 16, 2015

Accomplishments

- The team worked on completing a presentation about the Materials List for the AguaClara symposium and attended presentations by other subteams, and continued to explore the blocking/drawing problem with the Overflow Drain.

- The team tried exploding the blocks as well as rearranging the order of the stack to first draw, subtract, then block the components.

Problems Encountered

- When the blocks were exploded, AutoCAD no longer recognized them as such and therefore did not count them in the -bcount function.
- Blocking the components after drawing them and subtracting from them did not work because the BlockingLast function only blocks the last object drawn. Therefore, it only blocked ChannelPipe2, as that was the last object drawn.
- A solution to the problem of not being able to subtract blocks from other blocks and objects in AutoCAD in drawing the Overflow Drain has still not been found.

Week of March 23, 2015

Problems Encountered

- This week, the problem of blocking the EtOverflowDrain was further explored, focusing specifically on the tee.
- When selecting objects for blocking, a major issue was that selecting one object would select all the components that were housed in that layer.
- The team attempted to explode the tee and caps, then selected only the tee when trying to block it using window. This was successful, but it was very messy as the individual objects were no longer unioned.
- When blocking objects using the “window” or “box” inputs, it was discovered that blocking them from the top view selects unwanted objects that happen to lie on the axis when viewing the axis from above.

Accomplishments

- The team created separate layers for every hydraulic component of the OverFlow Drain, and developed a BlockingBox function to block the components.
- This function used the “box” input instead of “last” when prompted to select objects to block in AutoCAD.
- The BlockingBox coordinates for the EtOverflow Tee were found, and applied to the MathCAD code.

- The team also determined that the objects must be blocked while viewing the Overflow Drain from the front view (Figure 6).

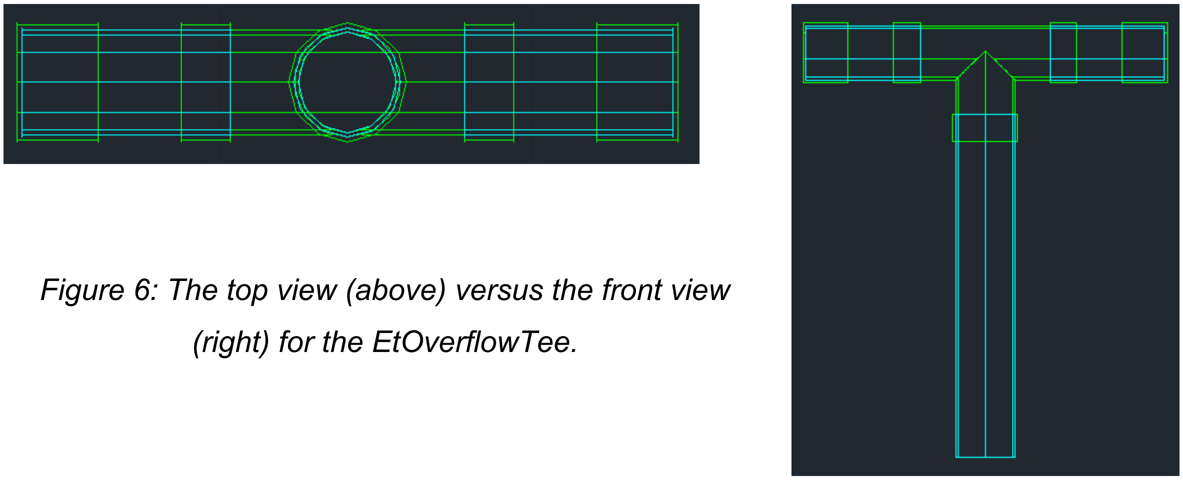
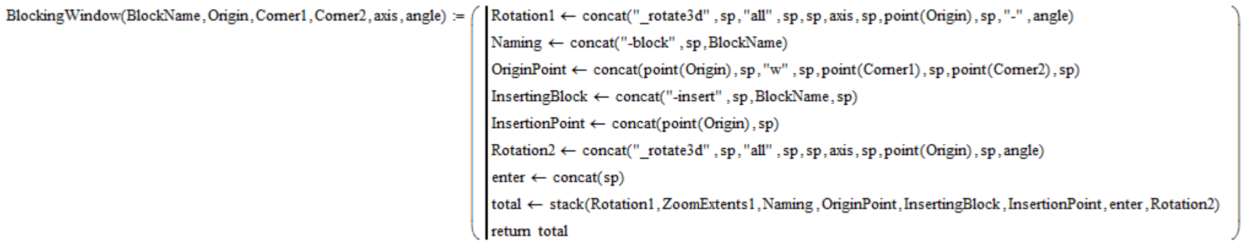


Figure 6: The top view (above) versus the front view (right) for the EtOverflowTee.

Week of April 6, 2015

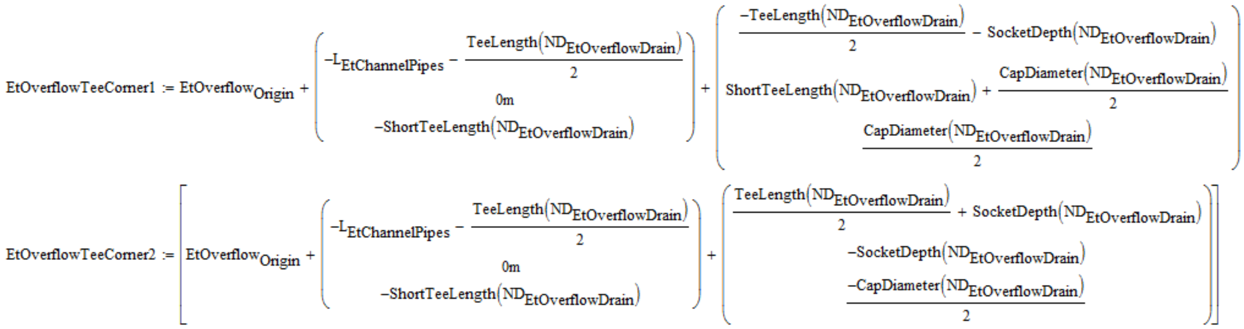
Accomplishments

- The team realized that the using the “window” input in AutoCAD was more successful than using the “box” input, and therefore created a BlockingWindow function adding the “_rotate3d” command at the beginning and end of the function (Figure 7).
- The team found that the command “_rotate3d” was useful when choosing which axis to rotate the drawing around.
- The team also added the “ZoomExtents1” function to the BlockingWindow function in order to ensure the object is in view when blocking.
- The coordinates for BlockingWindow for the EtOverflowDrainTee were determined and integrated into the blocking code (Figure 8 & 9).



```
BlockingWindow(BlockName, Origin, Corner1, Corner2, axis, angle) := | Rotation1 ← concat("_rotate3d" , sp, "all" , sp, sp, axis, sp, point(Origin), sp, "-" , angle)
                                                                   | Naming ← concat("-block" , sp, BlockName)
                                                                   | OriginPoint ← concat(point(Origin), sp, "w" , sp, point(Corner1), sp, point(Corner2), sp)
                                                                   | InsertingBlock ← concat("-insert" , sp, BlockName, sp)
                                                                   | InsertionPoint ← concat(point(Origin), sp)
                                                                   | Rotation2 ← concat("_rotate3d" , sp, "all" , sp, sp, axis, sp, point(Origin), sp, angle)
                                                                   | enter ← concat(sp)
                                                                   | total ← stack(Rotation1, ZoomExtents1, Naming, OriginPoint, InsertingBlock, InsertionPoint, enter, Rotation2)
                                                                   | return total
```

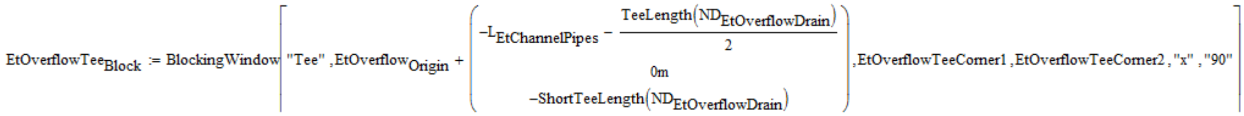
Figure 7: BlockingWindow Function



$$\text{EtOverflowTeeCorner1} := \text{EtOverflow}_{\text{Origin}} + \begin{pmatrix} -L_{\text{EtChannelPipes}} - \dfrac{\text{TeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right)}{2} \\ 0\text{m} \\ -\text{ShortTeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right) \end{pmatrix} + \begin{pmatrix} \dfrac{-\text{TeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right)}{2} - \text{SocketDepth}\left(\text{ND}_{\text{EtOverflowDrain}}\right) \\ \text{ShortTeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right) + \dfrac{\text{CapDiameter}\left(\text{ND}_{\text{EtOverflowDrain}}\right)}{2} \\ \dfrac{\text{CapDiameter}\left(\text{ND}_{\text{EtOverflowDrain}}\right)}{2} \end{pmatrix}$$

$$\text{EtOverflowTeeCorner2} := \left[\text{EtOverflow}_{\text{Origin}} + \begin{pmatrix} -L_{\text{EtChannelPipes}} - \dfrac{\text{TeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right)}{2} \\ 0\text{m} \\ -\text{ShortTeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right) \end{pmatrix} + \begin{pmatrix} \dfrac{\text{TeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right)}{2} + \text{SocketDepth}\left(\text{ND}_{\text{EtOverflowDrain}}\right) \\ -\text{SocketDepth}\left(\text{ND}_{\text{EtOverflowDrain}}\right) \\ \dfrac{-\text{CapDiameter}\left(\text{ND}_{\text{EtOverflowDrain}}\right)}{2} \end{pmatrix}\right]$$

Figure 8: BlockingWindow coordinates for Tee



$$\text{EtOverflowTee}_{\text{Block}} := \text{BlockingWindow}\left[\text{"Tee"}, \text{EtOverflow}_{\text{Origin}} + \begin{pmatrix} -L_{\text{EtChannelPipes}} - \dfrac{\text{TeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right)}{2} \\ 0\text{m} \\ -\text{ShortTeeLength}\left(\text{ND}_{\text{EtOverflowDrain}}\right) \end{pmatrix}, \text{EtOverflowTeeCorner1}, \text{EtOverflowTeeCorner2}, \text{"x"}, \text{"90"}\right]$$

Figure 9: BlockingWindow function applied to the EtOverflowTee

Problems Encountered

- The team members realized that the view of the drawing was a factor when selecting a window for the BlockingWindow function. When creating a window in the top view (which is what the drawing was originally drawn in), both the tee and the vertical pipe would be selected.
- The team tried to control the view by using the “ -view” function, but found that the UCS would also change when the view was changed.
- The team then attempted to rotate the figure instead; used “-rotate” and “-3drotate” but both did not work as needed (“-3drotate” did not allow us to choose a rotation axis when viewed in top view).
- It was determined that the objects could not be blocked unless they were in view on the AutoCAD screen when the function was being applied.
- After writing the BlockingWindow function, problems were encountered while testing the code due to misplaced spaces and enters.

Week of April 13, 2015

Accomplishments

- The team continued to investigate the concept of blocking the EtOverflowDrain using the BlockingWindow function and successfully identified the window coordinates for the following hydraulic components:
  - EtOverflowCap1
  - EtOverflowCap2
  - EtOverflowPipeVert
  - EtChannelPipe1
  - EtChannelPipe2

Problems Encountered

- There were challenges when calculating the corners for the BlockingWindow function for the other parts of the EtOverflowDrain.

Week of April 20, 2015

Problems Encountered

- The team tested the new blocking method by creating a file output for the entire Entrance Tank drawing code using the modified scripts that included the blocking functions.
- The first problem encountered was the presence of attribute titles that showed up in the AutoCAD Entrance Tank drawing.
- Another major problem was the incorrect blocking of the EtOverflowDrain. In the final drawing, the EtOverflowTee was in the wrong layer and not all the blocks showed up in the final block count.
- The team also encountered a problem with the attribute definitions for the EtOverflowDrain. When the entire Entrance Tank was drawn, error messages would display every time AutoCAD attempted to execute the attribute definition part of the code.

Accomplishments

- The team addressed the issue of the attribute titles displaying in the drawing by using the -attdisp function in the AutoCAD command line. The drawing code was then augmented to include this function, which automatically turned the attribute display text off.
- The team also investigated using BlockingLast to block the components of the EtOverflowDrain. Using a separate script for each component, the team stacked a series of functions to successfully draw and block each part of the Overflow drain. First, a new layer was created and the component was drawn in full. The layer was then frozen and the weir-slot was drawn and then subtracted from the component, where applicable. The original component layer was then selected and the component was then blocked using the BlockingLast function and then assigned attributes. Finally, the layer was frozen and the process was repeated for the next component.
- The team then applied the BlockingLast function to other parts of the Entrance Tank, and tested the method by drawing the entire Entrance Tank using the augmented scripts. This method was found to be successful ONLY when all the components were drawn in separate layers (Figure 10).

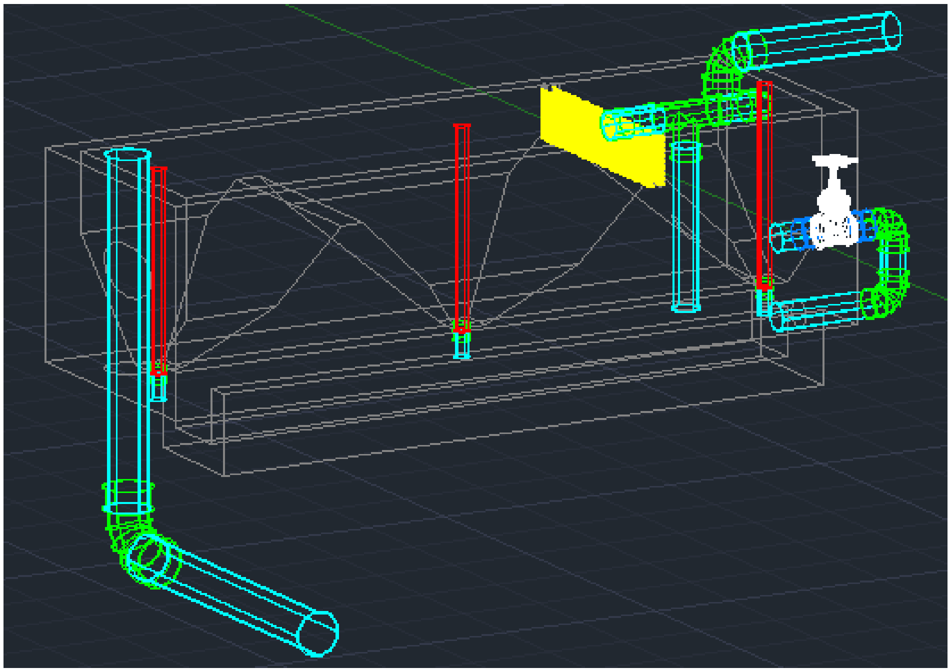
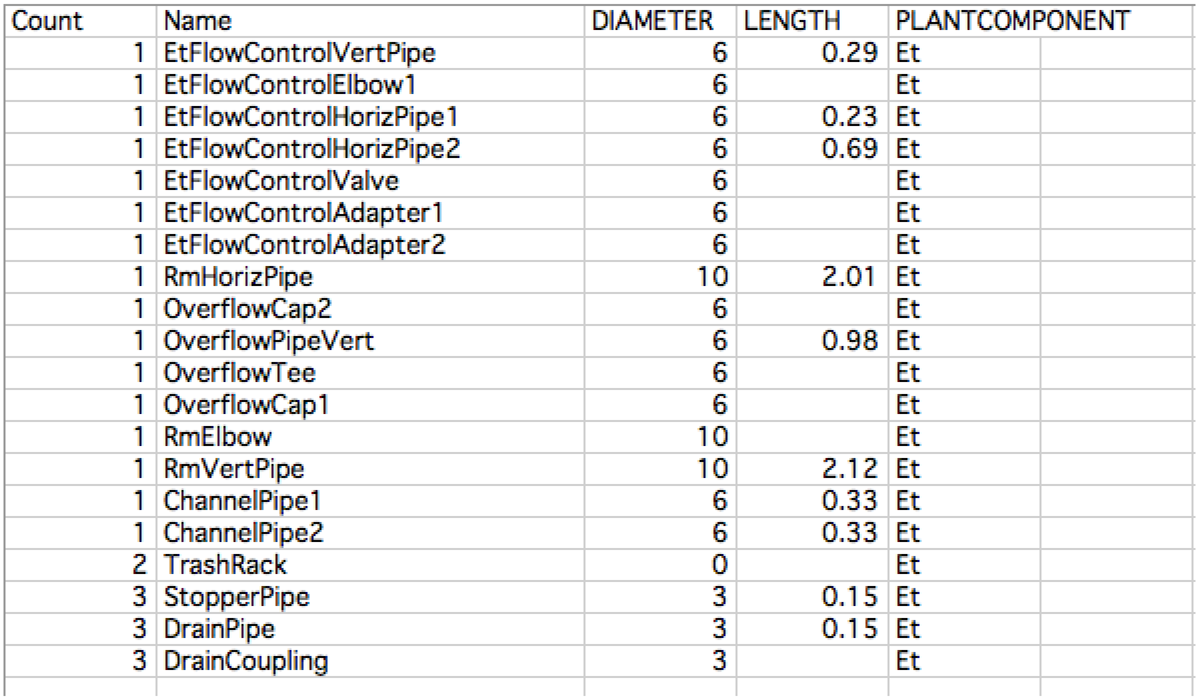


Figure 10: AutoCAD drawing of Entrance Tank

- A data extraction was performed on the final drawing, and an Excel data table was generated for the entire entrance tank (Figure 11).



Count	Name	DIAMETER	LENGTH	PLANTCOMPONENT
1	EtFlowControlVertPipe	6	0.29	Et
1	EtFlowControlElbow1	6		Et
1	EtFlowControlHorizPipe1	6	0.23	Et
1	EtFlowControlHorizPipe2	6	0.69	Et
1	EtFlowControlValve	6		Et
1	EtFlowControlAdapter1	6		Et
1	EtFlowControlAdapter2	6		Et
1	RmHorizPipe	10	2.01	Et
1	OverflowCap2	6		Et
1	OverflowPipeVert	6	0.98	Et
1	OverflowTee	6		Et
1	OverflowCap1	6		Et
1	RmElbow	10		Et
1	RmVertPipe	10	2.12	Et
1	ChannelPipe1	6	0.33	Et
1	ChannelPipe2	6	0.33	Et
2	TrashRack	0		Et
3	StopperPipe	3	0.15	Et
3	DrainPipe	3	0.15	Et
3	DrainCoupling	3		Et

Figure 11: Excel output of all the blocks found in the Entrance Tank, as produced by the Data Extraction function in AutoCAD.

Week of April 27, 2015

Problems Encountered

- A few of the definitions in the Entrance Tank Flow Control code were initially undefined after adding it into the Entrance Tank Materials List file due to insufficient referencing.
- All the components of the Entrance Tank Flow Control were drawn in one layer, which differed from the established convention of drawing the accessories and pipes in different color-coded layers.

Accomplishments

- The team successfully blocked the Entrance Tank Flow Control components using BlockingLast.

- The team also created layers for the different components. Accessories were drawn in a green-colored layer, while pipes were drawn in a blue-colored layer.

Week of May 4, 2015

Accomplishments

- The team organized the Entrance Tank Materials List file and deleted any unnecessary code (such as the three other Flow Control sections that are replaced with the new Entrance Tank Flow Control).
- The team integrated the Entrance Tank Flow Control code into the overall Entrance Tank output.
- The block names were changed to follow the convention of beginning with “Et” (the plant component) followed by what component of the Entrance Tank it is (such as RM for Rapid Mix) and the name of the part (for example, Elbow1).
- The team tested the complete Entrance Tank Materials List file with the integrated blocking functions using the beta server.

Problems Encountered

- When testing the Entrance Tank Materials List file with the beta server, some of the layers were named incorrectly (not using the established convention) and a few diameters were incorrect when extracting the data file (perhaps due to editing the Flow Control code to match established naming conventions).

Future Work

Future tasks for Design Team members include applying the blocking methods developed during the Fall 2014 and Spring 2015 semesters to the remaining components of AguaClara plants. The Spring 2015 Materials List team noticed a major shortcoming of the BlockingLast function, namely its inability to subtract AutoCAD objects from blocks. The team solved this problem in the context of the Entrance Tank Overflow Drain by drawing each pipe and accessory in separate layers (following appropriate naming conventions). Future teams should use this method to resolve conflicts with the LFOM, since the orifices are not being properly subtracted from the blocked pipe. Future teams can also investigate the possibility of creating a new blocking function that makes it possible to subtract objects from blocks. Additionally, design team members should explore a new way to import valves for each plant component, since the current method assigns a name to the valve which cannot be changed for different plant components. Finally, the design team should explore a way to automate the materials list data extraction process in AutoCAD. Currently, block data has to be manually extracted from the AutoCAD drawings by typing “dataextraction” into the command line and then selecting appropriate settings in a dialogue box. In past years, teams have been able to contact AutoDesk representatives to aid them in creating custom AutoCAD functions if they do not already exist. The Design Team can request a function that will allow them to avoid a dialogue box when extracting data, effectively automating the extraction process.