Construction Quotation Assistant

Overview Document

Software Engineering 3 (ENGR3791) Assignment A

Developed by:

Phil Lavender

Ben Peterson

Kane Stone

Peter Mitchell

The Construction Quotation Assistant (CQA) is an application that can be used to assist builders of retaining walls in providing a quote of the total cost of creating any number of retaining walls of differing lengths and heights using concrete sleepers. These walls can be planned out using the CQA and the quote of the cost of all the sleepers required is calculated, the builder can then add in a number multipliers that change the final cost of the project, these multipliers include the ease of access to the site and the type of material that the site is built on.

The CQA has three main panels which make up the user interface. The leftmost panel has buttons which allow the use of the functions required in the specification. The buttons available in this panel depend on whether the drawing space is in the top or side view. The central panel contains the drawing space which is the area in which the walls that have been created by the user appear in one of two views. The last panel of the wall shows the total cost of the wall which is based upon the total square meters of the wall and the additional cost of the multipliers. These multipliers are listed in the cost panel on the right hand side of the screen; the user can use the check boxes to select if the site has poor access or if the wall is built locally. There are also radio buttons with which the user can select one of: Normal, Sandy, Limestone and Bluestone. These multipliers are used to determine the total cost of the wall quote.

When the CQA opens the user is asked to enter their name, this name appears at the top of the drawing space and is used as the job identifier, which is then shown throughout the application. The CQA has two main views that appear in the drawing space, a Top view in which all of the walls in the project can be seen and a side view where the number of bays and the number of sleepers in each bay can be seen for an individual wall. There is different functionality depending on which view is currently active, in the top view the user deals with entire walls being able to add or remove them and reposition them within the drawing space. They can also choose a wall to edit which changes the view to the side view which allows the user to edit the bays that make up a wall. In the side view the user can add or remove bays from the wall and adjust the height of the bays thus determining the number of sleepers in a bay.

When the user creates a wall they receive a dialog box which asks them to enter the total length of the wall as well as the start and end height of the wall, the user can also set an initial rotation for the wall. Once the user has confirmed the selections that they have made in the dialog box then they are able to place the wall in the drawing space. If the user attempts to use another function before placing the wall they receive a warning letting them know that they must place the wall in order to complete any more functions. Once the user has clicked inside the drawing space the wall is centred on where it was clicked. Once a wall has been added to the drawing space then the cost panel on the side of the CQA calculates the cost of the wall including any multipliers selected and also lists the number of sleepers required and the total m^2 of the wall. After a wall has been placed in the drawing view the user can choose to change the rotation of the wall, once the user selects the wall they wish to change the rotation of they can set the new angle of the wall using a dialog box.

The user can also add bays to a wall when they are using the side view for the wall they wish to add bays to. The user receives a dialog box which lets them choose the number of bays they want to add, the height of the bays they wish to add and whether they wish to add the bays to the left or right hand side of the current wall. While in the side view the user can also edit the height of the bays that make up the view, this can be done using a button and selecting the bay you wish to change the height of or if you are not currently performing another action then you can click and drag the top of the bay in order to change the height.

The information about each wall is stored in an array in Top View. Each wall has the x and y coordinates of the centre of the wall, the height of the wall, which is rounded up to the increment of two metres, the number of bays within the wall and whether the wall is selected or not. The x and y coordinates of the four corners of the wall are also held in the information about a wall. When the CQA loads the side view the information stored in the element of the array related to the wall selected is focused on which provides all of the information that is held about the wall that is required by the side view.

Each of the classes that make up the CQA has a number of methods associated with them. The add bay class has a number of methods associated with it, these methods include the get and set methods for the Initial Bay Height, Location and the number of bays. The class also contains checks to see if the number of bays and bay height fall within the appropriate range which is defined by the type of variable that they are associated with. The class also contains a method called view dialog which causes the dialog box to appear when the user presses the button to add the bay. There is also a clear method which clears all of the fields within the dialog box.

The add wall class is very similar to the add bay class, it has get and set methods for the length, start height, end height, initial rotation of the wall. The class also contains checks for the length, start height, end height and the rotation of the wall. The class also contains the view dialog method which causes the dialog box to appear so that the user can add a wall.

The bay class contains two methods, a method to get the number of sleepers within a bay and a method to which converts either a specific height to a number of sleepers or to take a number of sleepers and store that in a variable called sleepers.

The graphics panel class contains the methods getClient and getSleepers which get the client name or number of sleepers respectively. The class also contains methods that allow the user to swap between the Top and Side view; these methods are called gotoSideView and gotoTopview. The graphics panel class also contains numerous set methods including setClient and setInitialClientName the setInititalClientName method is called at the beginning of the CQA and sets the name that is shown at the top of the screen, the setClient method allows the user to change the name that was set at the start of the CQA. There are also set methods which change the maximum value of the x and y sliders, the maximum value of the sliders are determined using the updateBounds method which determines whether the drawing is able to be completely shown within the main screen or whether it requires the sliders top be able to view the entire drawing. There is also a setMode method which exists as a public interface that the buttons make calls to in order to complete their function. The other set method is setTextHint which gives the user a hint of what function they are currently using. The graphicsPanel class also contains methods which update the x and y offsets of the Panel based upon the panel that is currently visible. The graphics panel class also contains the method getWallsSaceData which gets all of the data which relates to the walls of a particular job.

The main class has a number of methods which initialise the CQA. The main method for the class calls the initialise components method which through the use of the addButtons and initComponentsRight methods that are included in the main class adds the left and right panels of the CQA. The main class also contains a method which call the constructors of both the addWall class and the addBayClass. The swapButtons method is also contained within the class, this swaps which buttons are shown depending on the view that is currently in the Graphics Panel. The main class also contains the load and save methods which allow the user to save the current quotation with a file name or allows the user to load a specific quotation that has been previously saved.

The SideView class contains two methods called add bay, one of these methods opens a dialog box to allow the user to specify the information that is used by the second method to add the specified number of bays at the specified height to either the left or right of the wall depending upon what was selected in the dialog box. The class also contains a method called findBay which determines the bay the the mouse cursor is currently in based upon the x-coordinate of the mouse. The class also contains two methods which get the current maximum offset of the x and y axis. The method also contains a way for the user to change the x and y offsets, a method is also available to update the maximum y offset of the axis scrollbar. The paintComponent method displays the bays of the wall that are visible using a given offset. The SideView class contains a method to swap the wall which is currently focused on and a method to change the mode of the mouse which is based on the function that is currently being performed in the current view.

The TopView class is similar to the SideView class and it also contains two methods of the same name, in this case the addWall method, one of the methods opens the dialog box which allows the user to supply the information which is used by the other method to add the wall to the graphics panel. The TopView class also contains a method called cloneWall which allows the user to create a copy of a wall that is selected by the user is added to the graphics panel. The findClick method finds if the user has just selected a wall using the mouse. There are some get methods which gather the current X and Y offset, another method which gets the current bounds of the CQA. There is a method that gets the number of sleepers in each bay and there is a method which gets information about a selection if a wall is selected. There are also two methods which can be used to set the X and Y offsets based upon which method is chosen.There is a setMode method which dictates what action the mouse will currently perform. The rotateWall and removeWall methods alter a wall after it has been placed in the drawing place; the wall which is altered is selected by the user clicking on the wall. The resetSelection method deselects any object that is currently selected.

The Wall class contains methods which allow the user to change information related to the wall. The class contain get methods for the x and y coordinates of the centre of the wall, it also contains get methods for the current x and y offset of the wall as well as the current rotation of the wall. The class also contain a method to get the bounds of the graphics panel, a method that gets the number of sleepers that are contained in the entire wall and a method to get the number of bays in a specified wall which allows the bays to be accessed directly. A method called addBays allows the user to add a single bay to either the left or right of the wall. The Wall class contains a method called isSelected which checks which wall is currently selected; it also contains the changeSelection method which changes whether a wall is currently selected. The paintComponent method draws the rectangle within the drawing space, the pointInWall method detects if a specific set of coordinates lies within the bounds of the selected rectangle. The pointMatchs method determines whether the test point lays on the same side of the line which is defined by the endpoints a and b. The removeBay method removes the bay that is selected by the user. The wall class also contains set methods which can change the x and y offset of the rectangle, the centre point of the rectangle and the rotation of the rectangle, these methods then all update the polygon shown in the graphics panel. The updateRectangle method calculates the coordinates of each of the corner points of the rectangle. It factors in the centre point and then finds the points using offsets from the width and height. Then it applies a rotation to the points which transforms them to the correct locations. The transformX and transformY methods transform the respective coordinate using the current rotation. The slantIsPositive method determines whether the current slant of the rectangle is positive or not.

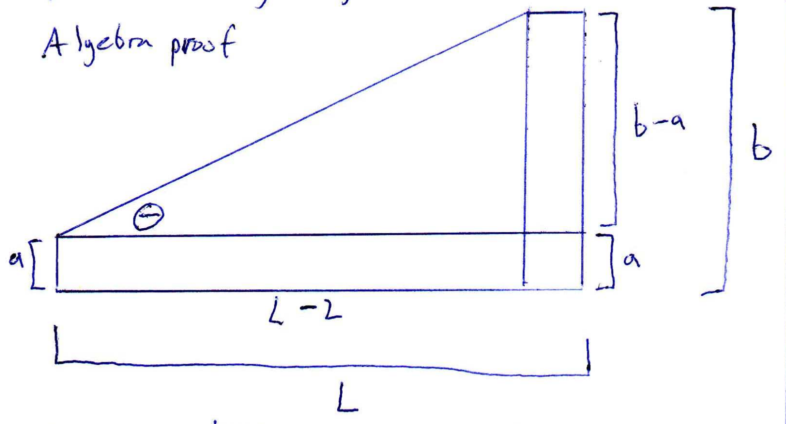
Proof of bay height calculation:

Disputing that $(L=10,\ a=1,\ b=2)$ gives result: 5|6|7|9|10

Instead claiming it gives: 5|6|8|9|10

a = startHeight
b = end Height
L = Length

Algebra proof



$$\tan\theta = \frac{b-a}{L-2} \quad \ldots (1)$$

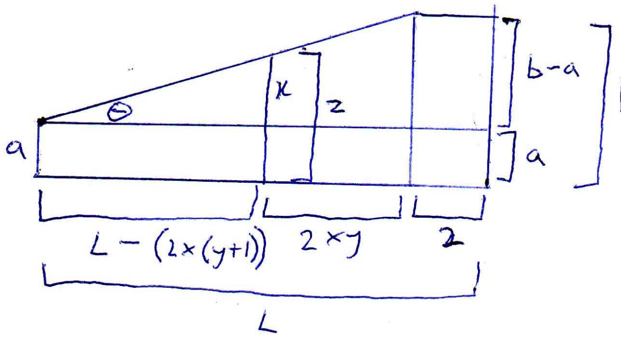
$$\therefore\ \theta = \tan^{-1}\left(\frac{b-a}{L-2}\right)$$

Height for any point:

x = Height in triangle

y = number of bays done (minus end height bay)

z = total height



$$\frac{x}{L-2xy} = \tan(\theta)$$

$$\therefore\ x = \tan(\theta) \times (L-2(y+1))$$

$$\therefore\ z_y = a + x$$

$$= a + \tan(\theta) \times (L-2\times(y+1))$$

$$= a + \left(\frac{b-a}{L-2}\right) \times (L-2\times(y+1)) \quad \ldots \text{using}(1)$$

Let:

$L=10,\ a=1,\ b=2$

$$\tan\theta = \frac{2-1}{10-2} = \frac{1}{8}$$

For $y=1$:

$$z_1 = 1 + \tfrac{1}{8}(10-2\times(1+1))$$
$$= 1+\tfrac{1}{8}(6)$$
$$= 1.75$$

For $y=2$:

$$z_2 = 1+\tfrac{1}{8}(10-2\times(2+1))$$
$$= 1+\tfrac{1}{8}(4)$$
$$= 1.5$$

For $y=3$:

$$z_3 = 1+\tfrac{1}{8}(10-2(3+1))$$
$$= 1+\tfrac{1}{8}(2)$$
$$= 1.25$$

For $y=4$:

$$z_4 = 1+\tfrac{1}{8}(10-2(4+1))$$
$$= 1+\tfrac{0}{8}$$
$$= 1$$

∴ • Heights of the 5 bays are:

1, 1.25, 1.5, 1.75, 2.0

• Rounding (0.2 increments):

1, 1.2, 1.6, 1.8, 2.0

• Dividing by 0.2 to get sleepers:

5, 6, 8, 9, 10

Where did the 7 come from? Inaccurate numbers:

$$\theta = \tan^{-1}\left(\frac{2-1}{10-2}\right) \cong 1.1243549948$$

using this number as θ instead of $\tan\theta = \frac{1}{8}$: $\tan\theta = 0.1249999994$

which then gives a fractionally smaller answer than 1.5, so rounds down to 1.4, which is 7 sleepers.