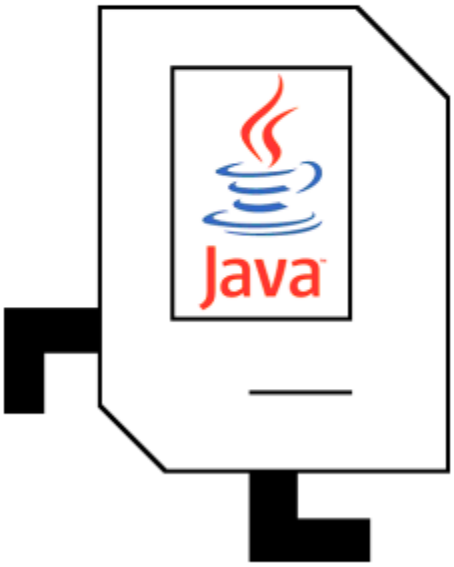
KAREL THE ROBOT



Assignment karel

Khaled rajaie yaser abdullah

Assignment :

Divide a given map into 4 equal chambers – take into consideration the special cases such as small maps that can't be divided into 4 chambers equally and divide them into the maximum possible number of equal chambers (1, 2, or 3)

functions:

1_ getColumn(int column):

I used while loop while(frontisclear()) and inside it I called function move() to move to block and every time a function move() is called the variable a column will increase by one and when the function frontisclear() return false the function getColumn(int column) will return the value of the variable column

2_getRow(int row):

First thing I called function turnLeft() because the front is block then, I used while loop while(frontisclear()) and inside it I called function move() to move to block and every time a function move() is called the variable a row will increase by one and when the function frontisclear() return false the function getRow(int row) will return the value of the variable row

3_RowOrColumnEqual1(int MoveRowOrColumn, Boolean checkIfWillTurnLeft):

First thing make sure if I'm going to call a function turnLeft() or not by mean of the variable checkIfWillTurnLeft If the value of the row is 1 then I will call it otherwise I don't need to call it, after that I will check if the variable MoveRowOrColumn is divisible by 2 , I will use a while loop while(frontisclear()) and inside it I called function move() to move to block And I defined a variable named walk to calculate the number of steps If the value of walk is equal to the value of The variable MoveRowOrColumn/2 or MoveRowOrColumn/2-1 I'll call a function putBeepers() then,if the variable

MoveRowOrColumn is not divisible by 2, I will use a while loop while(frontisclear()) and inside it I called function move() to move to block And I defined a variable named walk to calculate the number of steps If the value of walk is equal to the value of The variable MoveRowOrColumn/2 I'll call a function putBeepers()

4_IfTurn(string checkturn):

I get the function variable checkturn if chekturn equal left I'm going to call function turnleft() else if checkturn equal right I'm going to call function turnright() else checkturn equal around I'm going to call function turnaround(), I create this function to make the code more effective.

5_MoveIfRowEqualsColumn(int MoveRowOrCol, boolean checkIfWillputbeepers, boolean checkifwilturnleft, boolean usebreak):

I used while loop to get a certain position whose distance is stored in the variable **MoveRowOrCol** also inside the while loop I used if statement in case I want to put beepers and to make sure I used the variable **checkifwillbeepers** in case the condition is correct I will call the **putbeepers()** and also use if statement to check if front is blocked so I call function turnaround() to I can go back and then, I will call the function **move()** so the **karel** can moves to the desired position. When the condition loop return false so karel got to the position and I use variable **checkifwillturnleft** to check if the **karel** need to turn left so I will call function **turnLeft()**

6_MoveIfRowNotEqualsColumn(int MoveRowOrCol, boolean checkIfWillbeepers, boolean CallIfTurn, string checkifwillturn, string checkifwillturn1, boolean checkIfWillputbeepers1):

I used while loop to get a certain position whose distance is stored in the variable **MoveRowOrCol** also inside the while loop I used if statement in case I want to put beepers and to make sure I used the variable **checkifwillbeepers** in case the condition is correct I will call the **putbeepers()** and also use if statement to check if front is blocked so I use break to get out of while loop, I will call the function **move()** so the **karel** can move to the desired position and I will check if the **karel** position at need beepers or not then, I call function **Ifurn(string checkturn)** to change **karel** direction, also I will check if I need to change **karel** direction or not. I will use variable **callIfturn** to see if I need to change **karel** direction.

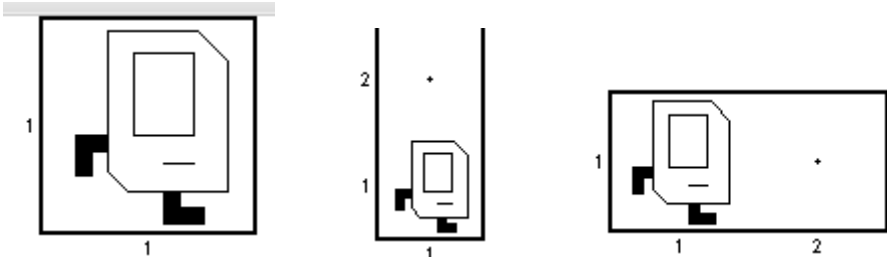
7_Move(int movekarel, boolean checkIfWillputbeepers, string choose):

I used while loop to get a certain position whose distance is stored in the variable **MoveRowOrCol** also inside the while loop I used if statement in case I want to put beepers and to make sure I used the variable **checkifwillbeepers** in case the condition is correct I will call the **putbeepers()** and also use if statement to check if front is blocked so I use break to get out of while loop, I will call the function **move()** so the **karel** can move to the desired position then, I use variable choose to determine which direction karel direction will change.

Test Case:

1_if row equal 1 and column equal 1 or row equal 2 and column equal 2 or column equal 2 and row equal 1:

I divided into 1 because the number of room inside map is less than the possible number to divide the map int 2,3 or 4

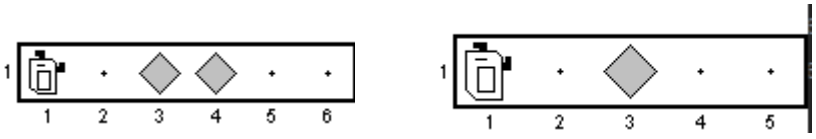


2_if row equal 1 and column greater than 2:

I will call a function **RowOrColumnEqual1(column,false)** and I will give it the value of the variable column and false because the front is clear after that, the map will be divided into 2

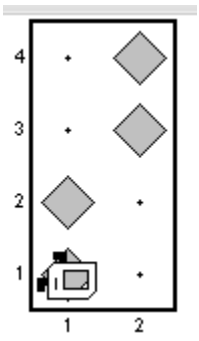
3_if column equal 1 and row greater than 2:

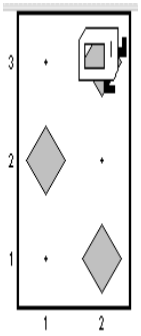
I will call a function **RowOrColumnEqual1(row,true)** and I will give it the value of the variable row and true because the front is block after that, the map will be divided into 2

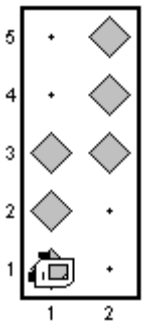


4_if row equal 2 and column not equal 2 or column equal 2 and row not equal 2:

If column it is divisible by 2 I will call function **Move(column/2 or row/2 ,true,"I")** and I will give it column/2 to move to the mid point and true to putBeeper After that I need to turn left to divide the map in equal area After that I will call **function Move(column/2, or row/2 ,true,"I")** again to complete the division process and it will be divided into 2 but if column equal 3 I will use for loop **For(int i=0;i<row*column;i++)** and if variable i it is divisible by 2 I will call function **putBeepers(),** if front is block I will call function **turnleft()** and I will call function **move()**

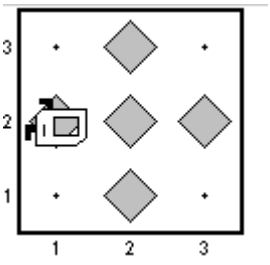






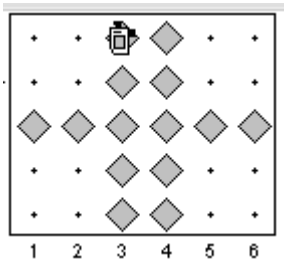
5_ if row odd and column odd:

I called function **Move** 7 times, one time I moved to mid point, second time I moved to row/2 (mid point) and put beepers, third time I will turn karel to left and moved to column/2 (block) and put beepers, fourth time I will back to mid point, fifth time I moved to the end of the row, sixth time I backed to mid point, seventh time I moved to the end of the column.



6_if row odd and column even:

I called function **Move** 9 times, one time I moved to mid point (col/2-1), second time I moved to row/2 (mid point) and putBeepers, third time I turned to left and moved to column/2 (block) and put beepers, fourth time I backed to mid point , fifth time I moved to the end of the row, sixth time I moved to next row and I moved to row/2 (mid point) , seventh time I moved to the end of the column, eighth time I backed to column/2 (mid point) ,ninth time I moved to the end of the next row



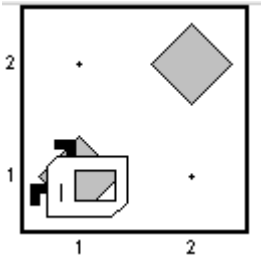
7_if column odd and row even:

I called function **Move** 8 times, one time I moved to mid point (col/2), second time I moved to row/2-1 (mid point) and putBeepers, third time I turned to left and moved to column/2 (block) and put beepers, fourth time I moved to next column and moved to column/2 (mid point), fifth time I moved to the end of row (block), sixth time I backed to the mid point(row/2) o, seventh time I move to the end of the column (block) ,eighth time I backed to mid point(col/2)

8_ if row even and column even:

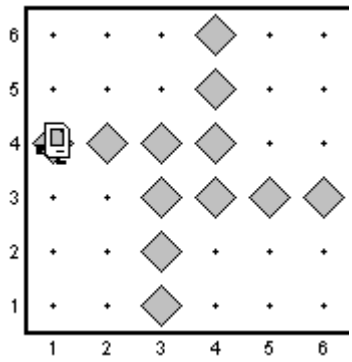
- ## If row equal 2 and column equal 2:

  I moved one steps and put beeper then I turned left and move one steps and put beepers



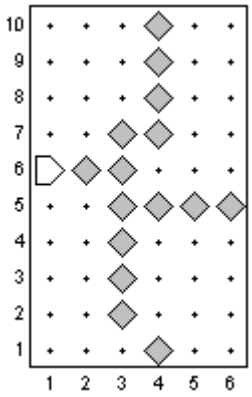
- **If row equal column :**

  I called function **MoveIfRowEqualsColumn** 7 time, one time I moved to mid point (column/2-1),second time I moved and put beepers to row/2 (mid point), third time I moved to column/2 and putbeepers, fourth time I backed to mid point(column/2-1) ,fifth time I moved and putbeepers to row/2(blocked),sixth time I return to mid point (row/2-1),seventh time I moved and putbeepers to column/2(blocked).



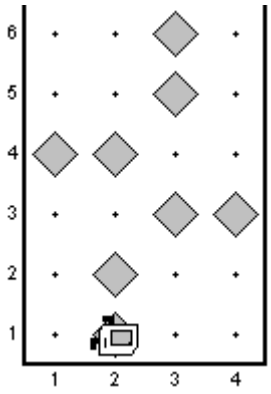
- **If (row*column)-(row+column)%4 equal 0:**

  I called function **MoveIfRowNotEqualColumn** 8 time, one time I moved to mid point(col/2-1), second time I moved to mid point (row/2-1) then I turned right then turn left ,third time I moved to (row-row/2-1)-1 then I turned left then turn right, fourth time I moved to row/2-1 then I turned around to return ,fifth time I moved to row/2-1,sixth time I moved to column/2 then I turned around, seventh time I moved to column/2 then I turned right , eight time I moved column/2 (blocked).



- **If (row*column)-(row+column)%4 not equal 0 and row+column-2/4<=column/2 or row+column-2/<=row/2:**

I called function **Move** 7 times, one time I moved to mid point(column/2-1) then I turned left, second time I moved and putbeepers to (row/2-1) and turn right and move one step more, third time I moved and putbeepers to column/2 then I turned around to return , fourth time I moved to column/2-1 then I turned right and move 2 steps more, fifth time I moved and putbeepers to column/2,sixth time return to mid point (col/2),seventth time I moved to row/2-1(block).



- **If column greater than row** :

I called function **Move** 9 times, one time I moved to mid point (col/2-1), second time I moved to row/2-1 (mid point) and putBeepers, third time I turned to left and moved to column/2-1 (block) and put beepers, fourth time I moved to next column (column/2-1) , fifth time I moved to row/2-1 and turned right , sixth time I moved to next row and I moved to row/2-1 (mid point) , seventh time I moved to column/2-1 (next column) , eighth time I backed to column/2-1 (mid point) ,ninth time I moved to the end of the next row.