COAL LAB Project

Procedures List

Sr.	Procedure Name	Parameters	Returns	Description
1	findRandom PROC	NULL	*rand_x to store the xCoordinate of array and rand_y to store the Y cooRdinate of the the array*	*This Function generates Random x,y coordinates of the array*
2	PlaceRandom PROC	NULL	*eax,ebx (roxIndex,colIndex)*	*The functions goes to the specified array index and check if there's a 0 or other value there.*
3	GameOverChecker	NULL	*Ret = { GAMEOVER = 1 when conditions meet else GAMEOVER=0}*	*The func is used to check if the game over conditions are meet*
4	GameOverCond1	NULL	*Ret = ebx = 1 means yes tiles can be moved still* *Ret = ebx = 0 means no titles can be moved*	*Checks if any row can be moved*
5	GameOverCond2	NULL	*Ret => ebx=0/ebx=1*	*Checks if any column can be moved*
6	GameOverCond3	NULL	*RET = GAMEOVER =1 WHEN Condition met*	*The func checks if any tile is equal to 2048 then sets GAMEOVER, YOU_WIN = 1 else leaves it unchanged*
7	DisplayGameOver	NULL	NULL	*The func simply displays Game Over on console*
8	DisplayGameWon	NULL	NULL	*Description:* The func simply displays Game Won on console
9	GetArrayValue_At	*Paras:* ESI = offset to 2D-Array EDI = rowIndex EDX = colIndex	RET = eax = value at GB_Arr[row][column]	*Func: Gives Value of 2D array at [EDI][EDX]*
10	printArrayValue	*Paras: eax [vlaue]*	NULL	*Func: Prints out the value being passed to it in eax*
11	printSpaces	*No paras : void*	NULL	*Func: Displays black spaces*
12	BorderOutline	*Paras: al , ah and ecx prints white color upto the ecx valued loop*	NULL	*Func : Adds a borderline when called*

13	DrawGameBoard	NULL	NULL	*Description:* The func simply displays the gameboard on console using the 2D-Array,its Rowsize and its Columnsize
14	DisplayScoreBoard	NULL	NULL	*; Description:* The func simply displays the score of the user on console
15	PlaceArrayValue_At	*Paras:* ESI = offset to 2D-Array EDI = rowIndex EDX = colIndex EAX = value to be placed	NULL	*Places value in the array at specified parameters(r,c)*
16	findRandom	NULL	*EAX = To store the random number* rand_x to store the xCoordinate of array and rand_y to store the Y cooRdinate of the the array	*This Function generates Random x,y coordinates of the array*
17	PlaceRandom	*row Index = 0* colIndex = 0 searchKey = searchKey Value counter ,0 ecx = number of rows	*RET ==> eax,ebx(roxIndex,colIndex)*	*The functions goes to the specified array index and check if there's a 0 or other value there.*
18	holdMoveKeys	NULL	NULL	*This fucntion record the movement keys of the game*
19	upWardMovement	NULL	NULL	*The func is provoked when movement called* is upWard Movement & uses uphelper to perfrom upward movement to each column
20	upHelper	*col number in eax*	NULL	*THIS FUNC IMPLEMENTS THE Down MOVEMENT* ON THE 2D ARRAY WHEN PROVOKED Column WISE
21	downwardMovement	NULL	NULL	*The func is provoked when movement called* is Downward Movement and uses downhelper to perform down movement
22	downHelper	*col number in eax*	NULL	*THIS FUNC IMPLEMENTS THE Down MOVEMENT* ON THE 2D ARRAY WHEN PROVOKED Column WISE
23	ColumChecker	*; Paras:* ESI = offset to 2D-Array	*eax = 1 or 0 true or false basically*	*Func: This fun checks if any column is non zero*

		EDI = rowIndex EDX = colIndex		or zero / it can be moved or not
24	RowChecker	*Paras:* Eax= rowNumber to be checked ESI = offset to 2D-Array EDI = rowIndex EDX = colIndex	*ebx = true or false 1 or 0*	*Func: This fun checks if any row is non zero* or zero / it can be moved or not
25	leftwardMovement	NULL/NONE	NULL/NONE	*The func is provoked when movement called* is left Movement and uses leftHelpher to perform right operation of movement to every row
26	leftHelper	NULL/NONE	EAX = ROW NO.	*THIS FUNC IMPLEMENTS THE LEFT MOVEMENT* ON THE 2D ARRAY WHEN PROVOKED ROW WISE
27	rightwardMovement	NULL/NONE	NULL/NONE	*The func is provoked when movement called* is right Movement and uses rightHelpher to perform right operation of movement to every row
28	rightHelper	NULL/NONE	EAX= ROW NO.	*THIS FUNC IMPLEMENTS THE RIGHT MOVEMENT* ON THE 2D ARRAY WHEN PROVOKED ROW WISE