LogiMatch

A puzzle game for the Commodore C64.

# Description





LogiMatch is a puzzle finder game against the clock. You have one minute to find as many matching QR-Code like shapes as possible. But be aware, the shapes can be mirrored vertically or horizontally or even both. (If a shape is mirrored this is indicated by an arrow to the left or an arrow upwards.)

# How to play

Use the cursor keys to move the selecting colour frame. Push enter if you think that you found the match. That’s it!

# Emulator usage

Run the program with vice's x64.exe. (You may use drag and drop.)

# Variables and Data Structures

|  |  |
| --- | --- |
| Variable | Meaning |
| s | Screen base memory address |
| c | Colour base memory address |
| v() | Vertical move difference (depending on cursor keys) |
| h() | Horizontal move difference (depending on cursor keys) |
| t | Colour base memory address for solution field |
| u | Colour base memory address for play field |
| z | 40 (screen width; just used to save one bye somewhere else) |
| x, y | Position of player |
| a, b | Position of randomly chosen solution |
| h, v | Split horizontally / vertically (1..no, -1..yes) |
| l | Level reached |
| p$ | String to position the output of points and level |
| i, j | Used to iterate “rectangles” |
| w | 2 (for minor speedup) |
| f | 5 (for minor speedup) |

# Code

0?*"{white}{clear} {reverse on}logi{reverse off}match"*:pO53280,0:pO53281,0:s=1064:c=55336:dIv(s):dIh(s):v(145)=-1

1v(17)=1:h(157)=‑1:h(29)=1:t=s+9:u=s+2:z=40:w=2:l=0:ti$=*"235957"*:p$=*"{home}{right}{right}{right}{right}{right}{right}{right}{right}{right}{right}{right}{red}"*

2x=0:y=0:a=int(rN(0)\*3):b=int(rN(0)\*3):h=int(rN(0)\*1.8)\*2-1:v=int(rN(0)\*1.8)\*2-1

3f=5:fOi=1tof:fOj=1tof:pOs+i\*z+j,int(rN(.)+.5)\*80+80:nE:nE:l=l+1:pO1073,31-(v>0)

4fOi=-1to1:fOj=-1to1:pOs+9+(i+2)\*z+j,pE(s+w+(w+a+i\*h)\*z+b+j\*v):nE:nE

5fOi=1tof:fOj=1tof:pOc+i\*z+j,12+(aB(i-w-y)<waNaB(j-w-x)<w):nE:nE:pO1155,31+h

6gEd$:?p$;rI(ti$,2)*"{white}"*l:on‑(mid$(ti$,4,1)=*"1"*)gO9:on‑(d$=*""*)gO6:on‑(d$<>*"{return}"*)gO8

7fOi=-1to1:fOj=-1to1:on-(pE(t+(i\*h+2)\*z+j\*v)<>pE(u+x+(2+i+y)\*40+j))gO6:nE:nE:gO2

8x=x+h(aS(d$)):y=y+v(aS(d$)):x=x-int(x/3):y=y-int(y/3):gO5:rem1-liner compo pls!

9?*"{down}{down}{down}{down}{down}{down}{down}time up!{return}{return}you reached level"*;l:input*"{return}try again"*;x$:ifx$=*"y"*thenrun

# Code Description

* Line 0 clears the screen and shows the name of the game. It initializes many values.
* Line 1 and 2 also initialize many values. The coordinates of the solution are calculated and if the solution shall be mirrored (vertically (v) or horizontally (h)) or not.
* Line 3 fills the field with random shapes. Either empty or square (80 or 160). V-information (array left) is shown.
* Line 4 copies the solution.
* Line 5 sets the color frame, so you can see where you currently are. H-information (arrow up) is shown.
* Line 6 ready the users input (direction) and also checks the time.
* Line 7 checks if the solution is correct. If yes, you reach the next level.
* Line 8 moves the position of the selection.
* Line 9: Game over! Wanna try again?

## Line lengths

