

What Counts as Code to Criticize? *Interpreting Visual and Natural Language Programming*

Jeremy Douglass

Postdoctoral Researcher
Software Studies Initiative
University of California San Diego



Software Studies

// CRITICAL CODE STUDIES

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head profile="http://smil.org/xsmil/1.1">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Critical Code Studies</title>
<link rel="stylesheet" href="http://s.themes.wordpress.net/testrun/wp-content/themes/2433/style.css" media="screen"/>
<link rel="shortcut icon" type="image/ico" href="http://s.themes.wordpress.net/testrun/wp-content/themes/2433/favicon.ico" />
```

MENU CATEGORIES ARCHIVES RECENT METHODOLOGY

JOIN CCS

[Join the Critical Code Studies Blog]

email: mark + c +
marino
[at] g + mail [dot]
com

Login

RECENT COMMENTS

Mark Marino
: Wonderful articles,

CCS: COMING TO DIGITAL HUMANITIES 2009

≡ June 23rd, 2009 | → 0 Comments | V CCS, conferences |

This is just a head up about the next big even in Critical Code Studies and Software Studies: Wed. panel at Digital Humanities 2009:

Wednesday, June 24th
University of Maryland

9:00-10:30am
Charles Carroll Room
Critical Code and Software Studies
Mark Christopher Marino, Noah Wardrip-Fruin, Jeremy Douglass,
Elizabeth Losh, Stephanie August

ABOUT

Critical Code Studies

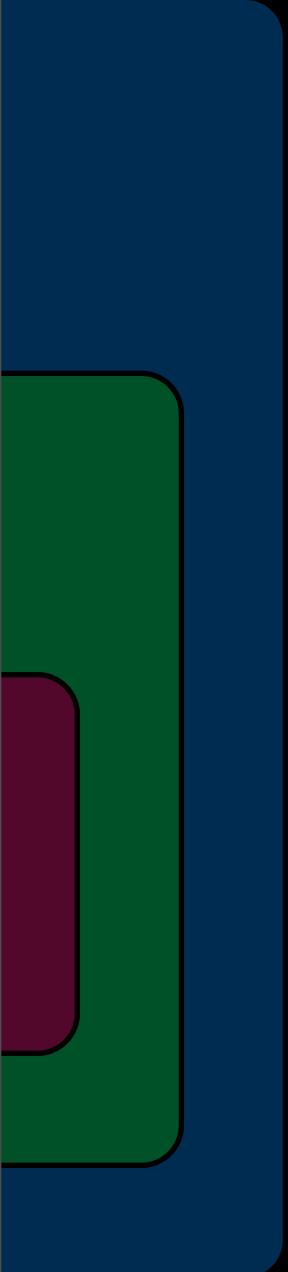
Critical Code Studies is a forum for resources, discussion, and demonstrations of the interpretation of computer code.

CATEGORIES

Digital Humanities

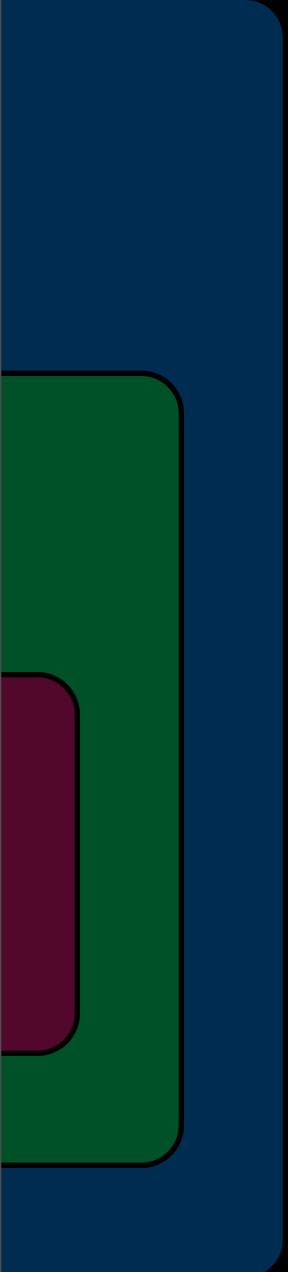
Critical Code / Software Studies

expanding our concept of
what code we critique & how

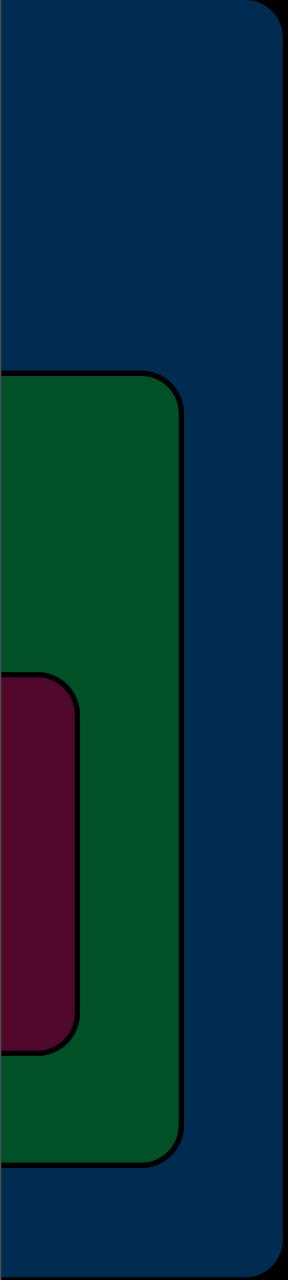


the question:

what does it mean to
study software and
criticize code?

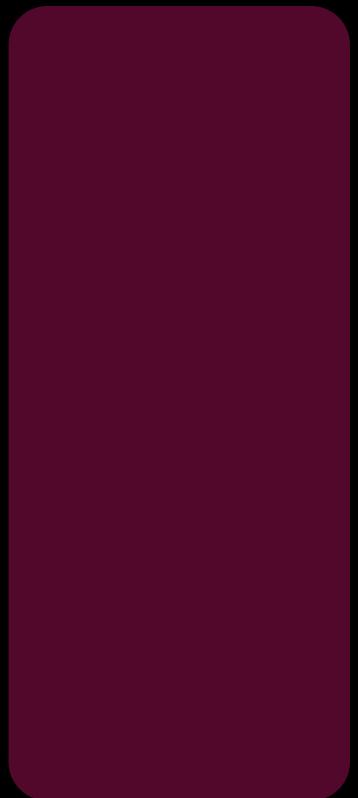


further,
what is possible when
performing critiques of
procedure &
specification?



let's consider a variety of
programmer-level
representations

many starting points, but let's begin with a
focus on three paradigms



SOURCE

IMPERATIVE
PROGRAMMING

NATURAL

NATURAL
LANGUAGE
PROGRAMMING

VISUAL

PATCH AND
FLOW CONTROL
PROGRAMMING

SOURCE

IMPERATIVE PROGRAMMING

```
#include <control.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "cold.h"
#include "joy.h"

char * patc="\n\nPress any key to continue

char move(char now){
    char k;
    if(Up[now]==0) Up[now]=now; //if key UP
    0 page
    while(k=joyinput()){
        //  if (now==0 && k==JOY_UP) return now
        if(k==JOY_A &&A[now]) return A[now]
        if(k==JOY_B &&B[now]) return B[now]
        if(k==JOY_SEL &&Select[now]) return
        if(k==JOY_STA &&Start[now]) return
        if(k==JOY_UP &&Up[now]) return Up[now]
        if(k==JOY_DN &&Down[now]) return Down[now]
        if(k==JOY_LF &&Left[now]) return Left[now]
        if(k==JOY_RT &&Right[now]) return Right[now]
    }
}

void clsO{
    waitblank();
    clrscr();
}
```

NATURAL

NATURAL LANGUAGE PROGRAMMING

The castle exterior is scenery in the drawbridge. The printed name of the castle exterior is "castle". Understand "tower" or "tower" or "drawbridge" or "bridge" as the castle exterior. The description is "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking east or west, you cannot see the furthest extent of the walls. An optical illusion: it is smaller inside.

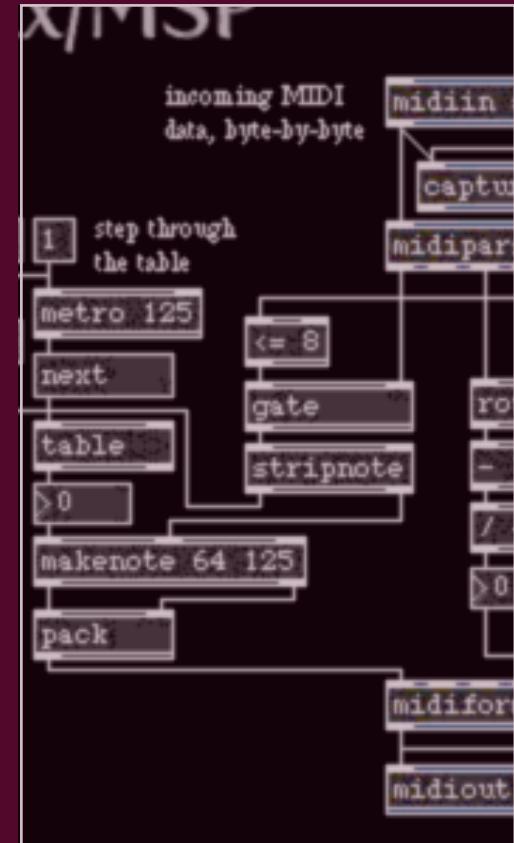
Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to

VISUAL

PATCH AND FLOW CONTROL PROGRAMMING



SOURCE

IMPERATIVE PROGRAMMING

```
#include <control.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "cold.h"
#include "joy.h"

char * patc="\n\nPress any key to continue

char move(char now){
    char k;
    if(Up[now]==0) Up[now]=now; //if key UP
    //page
    while(k==joyinput()){
        if (now==0 && k==JOY_UP) return now
        if(k==JOY_A &&A[now]) return A[now]
        if(k==JOY_B &&B[now]) return B[now]
        if(k==JOY_SEL &&Select[now]) return
        if(k==JOY_STA &&Start[now]) return
        if(k==JOY_UP &&Up[now]) return Up[now]
        if(k==JOY_DN &&Down[now]) return Down[now]
        if(k==JOY_LF &&Left[now]) return Left[now]
        if(k==JOY_RT &&Right[now]) return Right[now]
    }
}

void clsO{
    waitblank();
    clrscr();
}
```

NATURAL

NATURAL LANGUAGE PROGRAMMING

VISUAL

PATCH AND FLOW CONTROL PROGRAMMING

SOURCE

IMPERATIVE PROGRAMMING

```
#include <control.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "cold.h"
#include "joy.h"

char * patc="\n\nPress any key to continue

char move(char now){
    char k;
    if(Up[now]==0) Up[now]=now; //if key UP
    0 page
    while(k==joyinput()){
        //  if (now==0 && k==JOY_UP) return now
        if(k==JOY_A &&A[now]) return A[now]
        if(k==JOY_B &&B[now]) return B[now]
        if(k==JOY_SEL &&Select[now]) return
        if(k==JOY_STA &&Start[now]) return
        if(k==JOY_UP &&Up[now]) return Up[now]
        if(k==JOY_DN &&Down[now]) return Down[now]
        if(k==JOY_LF &&Left[now]) return Left[now]
        if(k==JOY_RT &&Right[now]) return Right[now]
    }
}

void clsO{
    waitvblank();
    clrscrO;
```

prototypical
or stereotypical
code

```
#include <conio.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "cold.h"
#include "joy.h"

char * patc="\n\rPress any key to continue ...";

char move(char now){
    char k;
    if(Up[now]==0) Up[now]=now; //if key UP not used ,use it to restart
    0 page
    while(k=joyinput()){
        //  if (now==0 && k==JOY_UP) return now; //patch for the 0 page
        if(k==JOY_A &&A[now]) return A[now];
        if(k==JOY_B &&B[now]) return B[now];
        if(k==JOY_SEL &&Select[now]) return Select[now];
        if(k==JOY_STA &&Start[now]) return Start[now];
        if(k==JOY_UP &&Up[now]) return Up[now];
        if(k==JOY_DN &&Down[now]) return Down[now];
        if(k==JOY_LF &&Left[now]) return Left[now];
        if(k==JOY_RT &&Right[now]) return Right[now];
    }
}

void cls(){
    wintvblank();
}
```

```
#include <conio.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "cold.h"
#include "joy.h"

char * patc="\n\rPress any key to continue ...";

char move(char now){
    char k;
    if(Up[now]==0) Up[now]=now; //if key UP not used
    if(k==JOY_UP && Up[now]) return Up[now];
    if(k==JOY_A && A[now]) return A[now];
    if(k==JOY_B && B[now]) return B[now];
    if(k==JOY_SEL && Select[now]) return Select[now];
    if(k==JOY_STA && Start[now]) return Start[now];
    if(k==JOY_UP && Up[now]) return Up[now];
    if(k==JOY_DN && Down[now]) return Down[now];
    if(k==JOY_LF && Left[now]) return Left[now];
    if(k==JOY_RT && Right[now]) return Right[now];
}
void cls(){
    waitvblank();
}
```

SOURCE

OPERATION / DATA

CONCISION

REPETITION

EMPHASIS ON SYNTAX

SPECIALIZED VOCABULARY

COMMENTS

```
#include <conio.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "cold.h"
#include "joy.h"

char * patc="\n\rPress any key to continue ...";

char move(char now){
    char k;
    if(Up[now]==0) Up[now]=now; //if key UP not used use it to restart
    0 page
    while(k=joyinput()){
        // if (now==0 && k==JOY_UP) return now; //patc: up to 0 page
        if(k==JOY_A && A[now]) return A[now];
        if(k==JOY_B && B[now]) return B[now];
        if(k==JOY_SEL && Select[now]) return Select[now];
        if(k==JOY_STA && Start[now]) return Start[now];
        if(k==JOY_UP && Up[now]) return Up[now];
        if(k==JOY_DN && Down[now]) return Down[now];
        if(k==JOY_LF && Left[now]) return Left[now];
        if(k==JOY_RT && Right[now]) return Right[now];
    }
}

void cls(){
    wintvblank();
}
```

“source”
a point of
departure

SOURCE

IMPERATIVE PROGRAMMING

```
#include <control.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "cold.h"
#include "joy.h"

char * patc="\n\nPress any key to continue

char move(char now){
    char k;
    if(Up[now]==0) Up[now]=now; //if key UP
    //page
    while(k==joyinput()){
        if (now==0 && k==JOY_UP) return now
        if(k==JOY_A &&A[now]) return A[now]
        if(k==JOY_B &&B[now]) return B[now]
        if(k==JOY_SEL &&Select[now]) return
        if(k==JOY_STA &&Start[now]) return
        if(k==JOY_UP &&Up[now]) return Up[now]
        if(k==JOY_DN &&Down[now]) return Down[now]
        if(k==JOY_LF &&Left[now]) return Left[now]
        if(k==JOY_RT &&Right[now]) return Right[now]
    }
}

void clsO{
    waitblank();
    clrscr();
}
```

NATURAL

NATURAL LANGUAGE PROGRAMMING

VISUAL

PATCH AND FLOW CONTROL PROGRAMMING

SOURCE

IMPERATIVE PROGRAMMING

NATURAL

NATURAL LANGUAGE PROGRAMMING

The castle exterior is scenery in the drawbridge. The printed name of the castle exterior is "castle". Understand "tower" or "tower" or "drawbridge" or "bridge" as the castle exterior. The description is "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking east or west, you cannot see the furthest extent of the walls. An optical illusion: it is smaller inside.

Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to

VISUAL

PATCH AND FLOW CONTROL PROGRAMMING

The castle exterior is scenery in the drawbridge. The printed name of the castle exterior is "castle". Understand "tower" or "tower" or "drawbridge" or "bridge" as the castle exterior. The description is "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking east or west, you cannot see the furthest extent of the walls. An optical illusion: it is smaller inside.

Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to
open it, heavy as it is, but it
swings aside lightly at your

NLP

The castle exterior is scenery in the drawbridge. The printed name of the castle exterior is "castle". Understand "tower" or "tower" or "drawbridge" or "bridge" as the castle exterior. "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking east or west, you cannot see the furthest of the walls. An optical illusion: it is smaller inside.

Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to open it, heavy as it is, but it swings aside lightly at your

EXPRESSIVE POWER

ACCESSIBILITY

IMITATION OF COMPLEXITY

OVERDETERMINATION

the problem: abstraction and depth

The castle exterior is scenery in the drawbridge. The printed name of the castle exterior is "castle". Understand "tower" or "tower" or "drawbridge" or "bridge" as the castle exterior. "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking east or west, you cannot see the further extent of the walls. An optical illusion: it is smaller inside.

Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to open it, heavy as it is, but it swings aside lightly at your

which do we interpret?

the code?

or the compiler?

The castle exterior is scanning the drawbridge. The predominant name of the castle exterior is "castle". Understand "tower" or "drawbridge" or "bridge" as the castle exterior. "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking east or west, you cannot see the furthest extent of the walls. An optical illusion: it is smaller inside.

Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to open it, heavy as it is, but it swings aside lightly at your

implied reader vs. explicit reader

The castle exterior is scenery in the drawbridge. The printed name of the castle exterior is "castle". Understand "tower" or "tower" or "drawbridge" or "bridge" as the castle exterior. "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking across the castle exterior you cannot see the further parts of the walls. An optical illusion: it is smaller inside.

Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to open it, heavy as it is, but it swings aside lightly at your

SOURCE

IMPERATIVE PROGRAMMING

NATURAL

NATURAL LANGUAGE PROGRAMMING

The castle exterior is scenery in the drawbridge. The printed name of the castle exterior is "castle". Understand "tower" or "tower" or "drawbridge" or "bridge" as the castle exterior. The description is "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking east or west, you cannot see the furthest extent of the walls. An optical illusion: it is smaller inside.

Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to

VISUAL

PATCH AND FLOW CONTROL PROGRAMMING

SOURCE

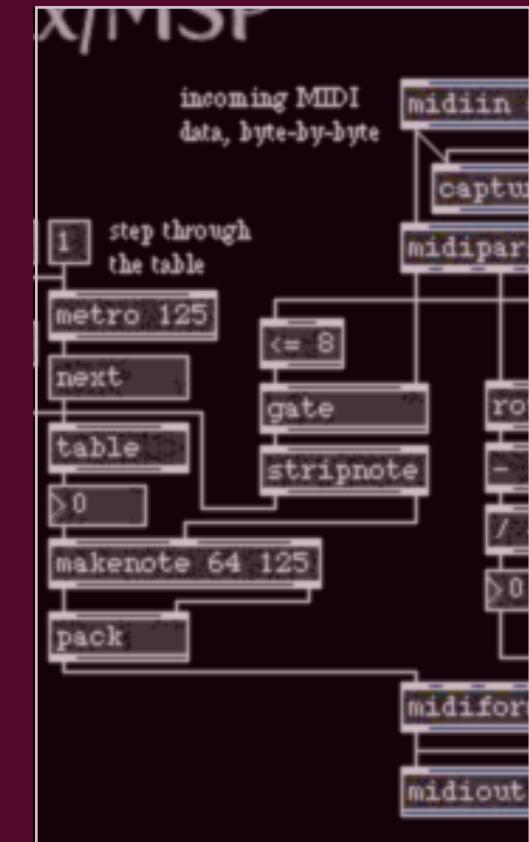
IMPERATIVE PROGRAMMING

NATURAL

NATURAL LANGUAGE PROGRAMMING

VISUAL

PATCH AND FLOW CONTROL PROGRAMMING



Max/MSP



Max/MSP



VISUAL

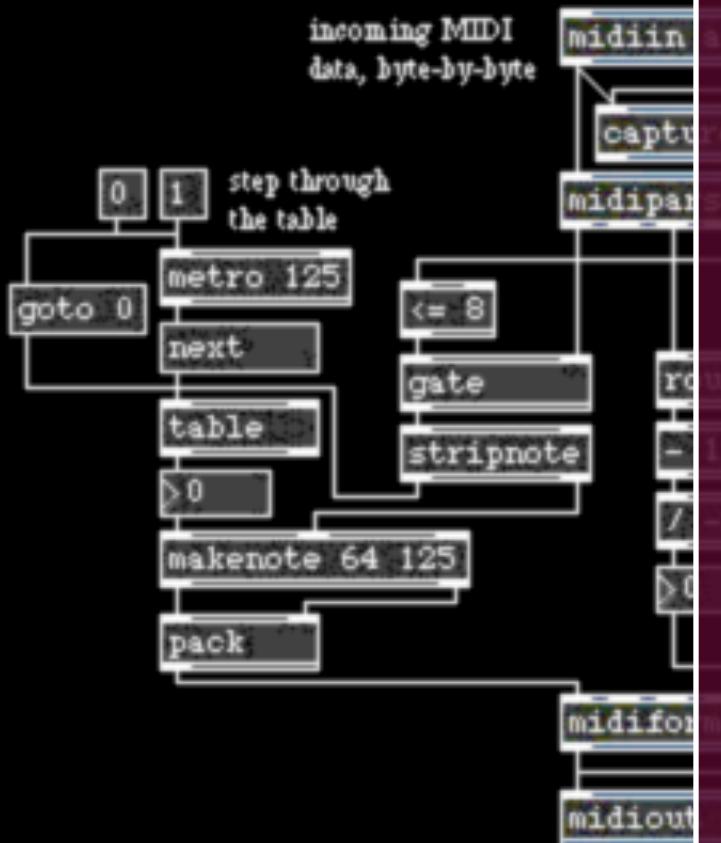
SPATIAL RELATIONSHIPS

SELF-REPRESENTING

TRACED RATHER THAN READ

A CLOUD OF BLACK BOXES

Max/MSP



how do we
“close read”
a graphic
specification?
visual
literacy?

```
#include <conio.h>
#include <string.h>

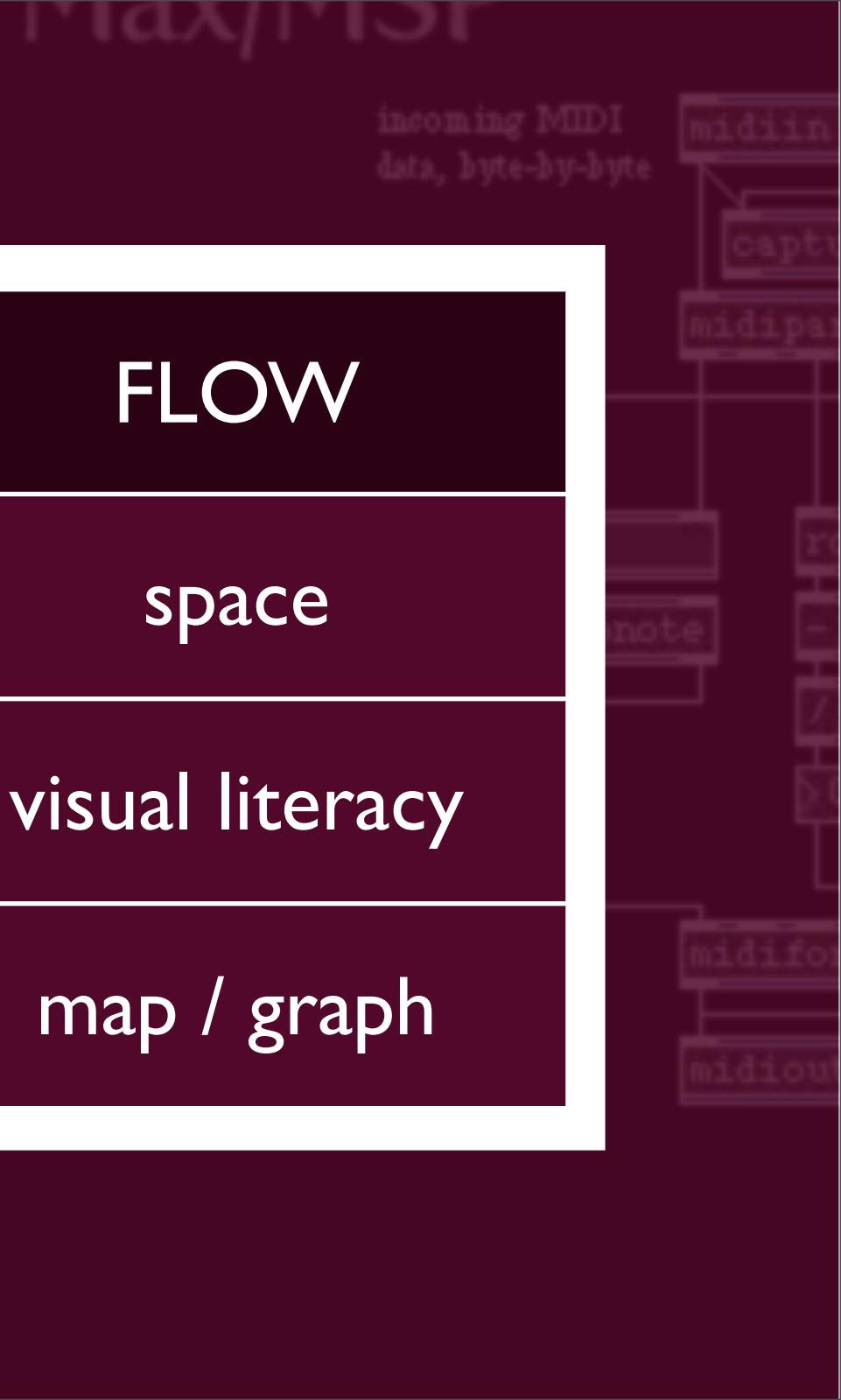
#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "col.h"
#include "joy.h"

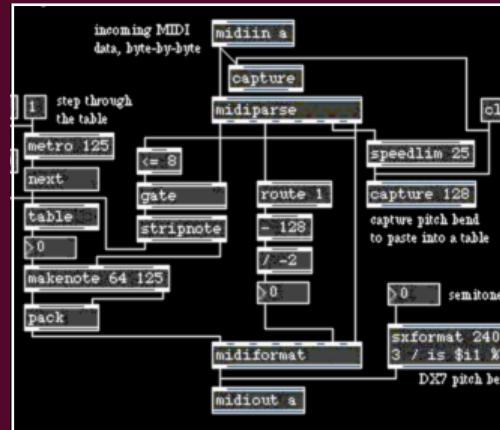
char * patc="";

char move(char k);
char k;
if(Up[now]==0) page++;
while(k!=j)
// if (n==0) {
if(k==1) if(k==2) if(k==3) if(k==4) if(k==5) if(k==6) if(k==7) if(k==8) if(k==9) if(k==10) if(k==11) if(k==12) if(k==13) if(k==14) if(k==15) if(k==16) if(k==17) if(k==18) if(k==19) if(k==20) if(k==21) if(k==22) if(k==23) if(k==24) if(k==25) if(k==26) if(k==27) if(k==28) if(k==29) if(k==30) if(k==31) if(k==32) if(k==33) if(k==34) if(k==35) if(k==36) if(k==37) if(k==38) if(k==39) if(k==40) if(k==41) if(k==42) if(k==43) if(k==44) if(k==45) if(k==46) if(k==47) if(k==48) if(k==49) if(k==50) if(k==51) if(k==52) if(k==53) if(k==54) if(k==55) if(k==56) if(k==57) if(k==58) if(k==59) if(k==60) if(k==61) if(k==62) if(k==63) if(k==64) if(k==65) if(k==66) if(k==67) if(k==68) if(k==69) if(k==70) if(k==71) if(k==72) if(k==73) if(k==74) if(k==75) if(k==76) if(k==77) if(k==78) if(k==79) if(k==80) if(k==81) if(k==82) if(k==83) if(k==84) if(k==85) if(k==86) if(k==87) if(k==88) if(k==89) if(k==90) if(k==91) if(k==92) if(k==93) if(k==94) if(k==95) if(k==96) if(k==97) if(k==98) if(k==99) if(k==100) if(k==101) if(k==102) if(k==103) if(k==104) if(k==105) if(k==106) if(k==107) if(k==108) if(k==109) if(k==110) if(k==111) if(k==112) if(k==113) if(k==114) if(k==115) if(k==116) if(k==117) if(k==118) if(k==119) if(k==120) if(k==121) if(k==122) if(k==123) if(k==124) if(k==125) if(k==126) if(k==127) if(k==128) if(k==129) if(k==130) if(k==131) if(k==132) if(k==133) if(k==134) if(k==135) if(k==136) if(k==137) if(k==138) if(k==139) if(k==140) if(k==141) if(k==142) if(k==143) if(k==144) if(k==145) if(k==146) if(k==147) if(k==148) if(k==149) if(k==150) if(k==151) if(k==152) if(k==153) if(k==154) if(k==155) if(k==156) if(k==157) if(k==158) if(k==159) if(k==160) if(k==161) if(k==162) if(k==163) if(k==164) if(k==165) if(k==166) if(k==167) if(k==168) if(k==169) if(k==170) if(k==171) if(k==172) if(k==173) if(k==174) if(k==175) if(k==176) if(k==177) if(k==178) if(k==179) if(k==180) if(k==181) if(k==182) if(k==183) if(k==184) if(k==185) if(k==186) if(k==187) if(k==188) if(k==189) if(k==190) if(k==191) if(k==192) if(k==193) if(k==194) if(k==195) if(k==196) if(k==197) if(k==198) if(k==199) if(k==200) if(k==201) if(k==202) if(k==203) if(k==204) if(k==205) if(k==206) if(k==207) if(k==208) if(k==209) if(k==210) if(k==211) if(k==212) if(k==213) if(k==214) if(k==215) if(k==216) if(k==217) if(k==218) if(k==219) if(k==220) if(k==221) if(k==222) if(k==223) if(k==224) if(k==225) if(k==226) if(k==227) if(k==228) if(k==229) if(k==230) if(k==231) if(k==232) if(k==233) if(k==234) if(k==235) if(k==236) if(k==237) if(k==238) if(k==239) if(k==240) if(k==241) if(k==242) if(k==243) if(k==244) if(k==245) if(k==246) if(k==247) if(k==248) if(k==249) if(k==250) if(k==251) if(k==252) if(k==253) if(k==254) if(k==255) if(k==256) if(k==257) if(k==258) if(k==259) if(k==260) if(k==261) if(k==262) if(k==263) if(k==264) if(k==265) if(k==266) if(k==267) if(k==268) if(k==269) if(k==270) if(k==271) if(k==272) if(k==273) if(k==274) if(k==275) if(k==276) if(k==277) if(k==278) if(k==279) if(k==280) if(k==281) if(k==282) if(k==283) if(k==284) if(k==285) if(k==286) if(k==287) if(k==288) if(k==289) if(k==290) if(k==291) if(k==292) if(k==293) if(k==294) if(k==295) if(k==296) if(k==297) if(k==298) if(k==299) if(k==300) if(k==301) if(k==302) if(k==303) if(k==304) if(k==305) if(k==306) if(k==307) if(k==308) if(k==309) if(k==310) if(k==311) if(k==312) if(k==313) if(k==314) if(k==315) if(k==316) if(k==317) if(k==318) if(k==319) if(k==320) if(k==321) if(k==322) if(k==323) if(k==324) if(k==325) if(k==326) if(k==327) if(k==328) if(k==329) if(k==330) if(k==331) if(k==332) if(k==333) if(k==334) if(k==335) if(k==336) if(k==337) if(k==338) if(k==339) if(k==340) if(k==341) if(k==342) if(k==343) if(k==344) if(k==345) if(k==346) if(k==347) if(k==348) if(k==349) if(k==350) if(k==351) if(k==352) if(k==353) if(k==354) if(k==355) if(k==356) if(k==357) if(k==358) if(k==359) if(k==360) if(k==361) if(k==362) if(k==363) if(k==364) if(k==365) if(k==366) if(k==367) if(k==368) if(k==369) if(k==370) if(k==371) if(k==372) if(k==373) if(k==374) if(k==375) if(k==376) if(k==377) if(k==378) if(k==379) if(k==380) if(k==381) if(k==382) if(k==383) if(k==384) if(k==385) if(k==386) if(k==387) if(k==388) if(k==389) if(k==390) if(k==391) if(k==392) if(k==393) if(k==394) if(k==395) if(k==396) if(k==397) if(k==398) if(k==399) if(k==400) if(k==401) if(k==402) if(k==403) if(k==404) if(k==405) if(k==406) if(k==407) if(k==408) if(k==409) if(k==410) if(k==411) if(k==412) if(k==413) if(k==414) if(k==415) if(k==416) if(k==417) if(k==418) if(k==419) if(k==420) if(k==421) if(k==422) if(k==423) if(k==424) if(k==425) if(k==426) if(k==427) if(k==428) if(k==429) if(k==430) if(k==431) if(k==432) if(k==433) if(k==434) if(k==435) if(k==436) if(k==437) if(k==438) if(k==439) if(k==440) if(k==441) if(k==442) if(k==443) if(k==444) if(k==445) if(k==446) if(k==447) if(k==448) if(k==449) if(k==450) if(k==451) if(k==452) if(k==453) if(k==454) if(k==455) if(k==456) if(k==457) if(k==458) if(k==459) if(k==460) if(k==461) if(k==462) if(k==463) if(k==464) if(k==465) if(k==466) if(k==467) if(k==468) if(k==469) if(k==470) if(k==471) if(k==472) if(k==473) if(k==474) if(k==475) if(k==476) if(k==477) if(k==478) if(k==479) if(k==480) if(k==481) if(k==482) if(k==483) if(k==484) if(k==485) if(k==486) if(k==487) if(k==488) if(k==489) if(k==490) if(k==491) if(k==492) if(k==493) if(k==494) if(k==495) if(k==496) if(k==497) if(k==498) if(k==499) if(k==500) if(k==501) if(k==502) if(k==503) if(k==504) if(k==505) if(k==506) if(k==507) if(k==508) if(k==509) if(k==510) if(k==511) if(k==512)
```

SOURCE	FLOW
sequence	space
literacy	visual literacy
write	map / graph



VISUAL



putting visual programming in context:
procedural flow in visual art

INFOVIZ

VISUAL

MAPS

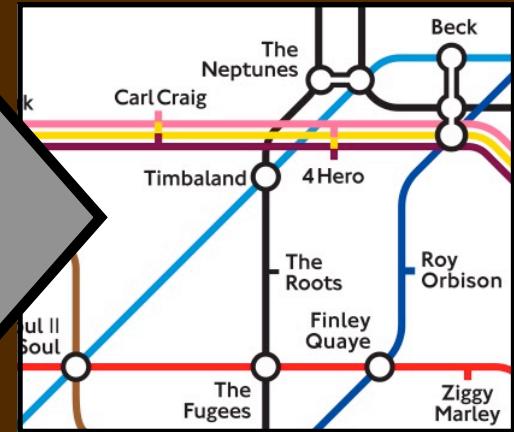
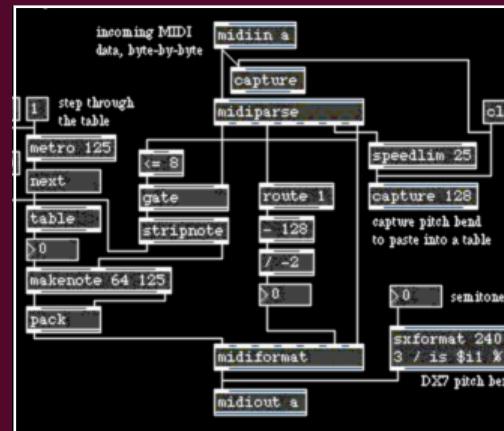
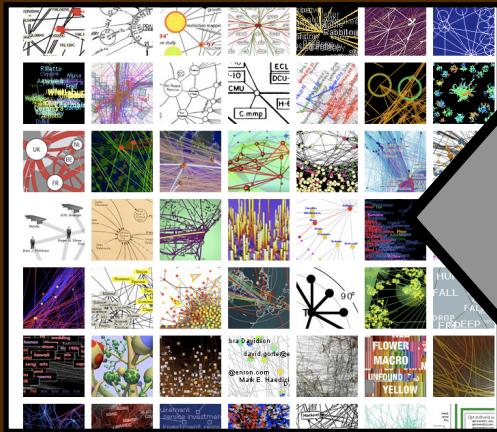
SCRIPTS

COMICS

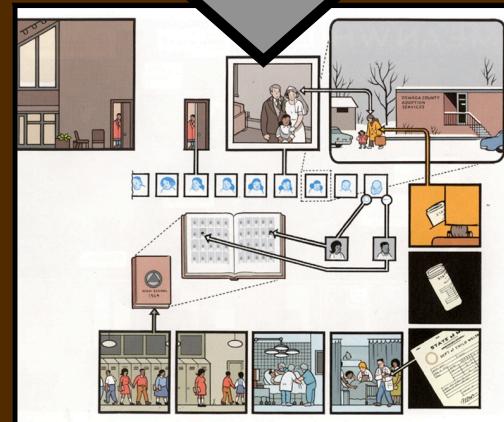
GUIs



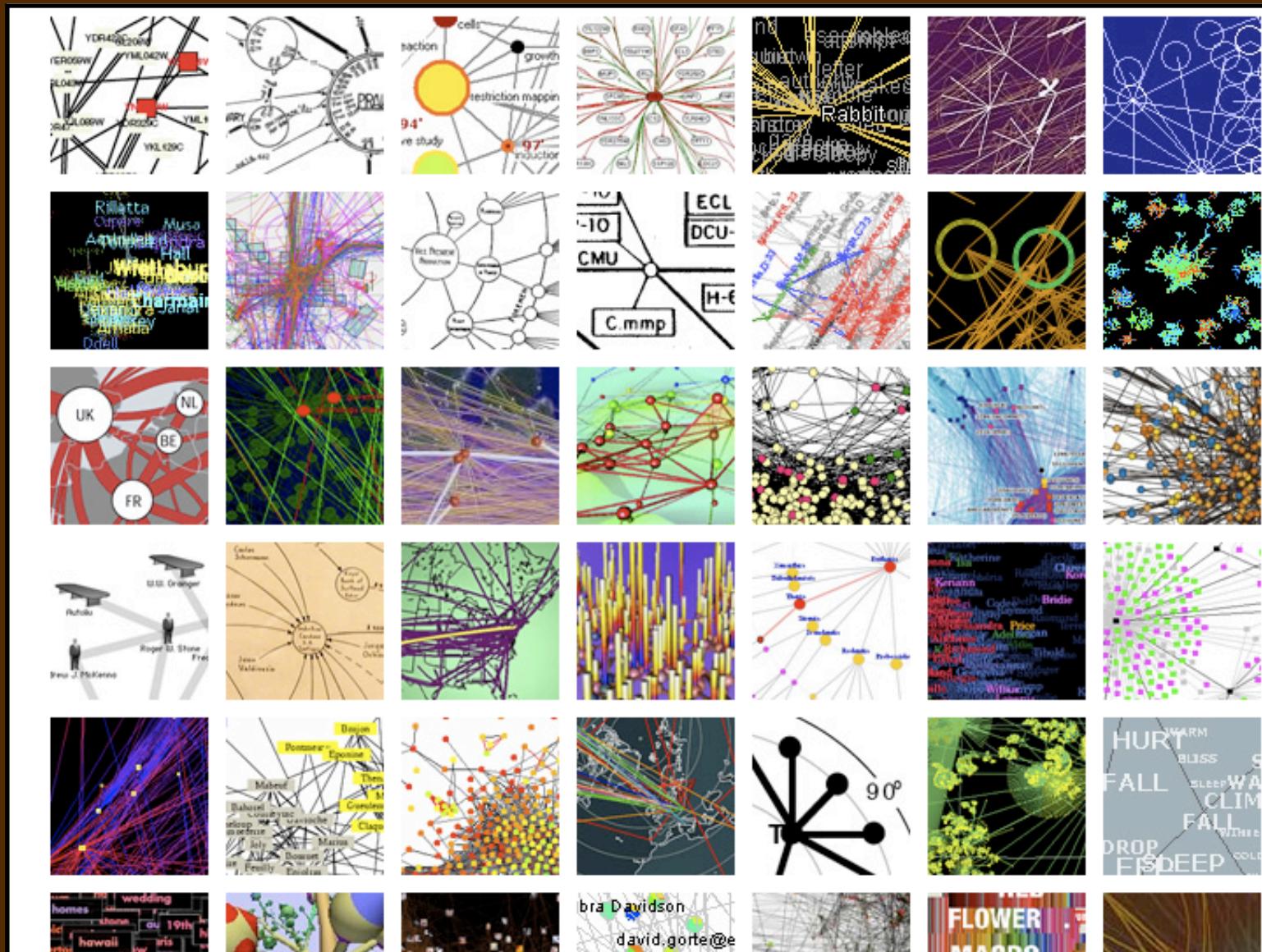
VISUAL



↓
YOU CAN'T MOVE.
↓
YOU FEEL PEACEFUL.
↓
YOU STARE INTO THE DARK.
↓
YOUR HEART IS POUNDING.
↓
YOU CAN'T SCREAM.
↓
THEN YOU LIE STILL.
↓
YOUR MOUTH IS DRY.
↓
YOU JUMP UP.
↓
YOU STARE INTO IT.
↓
AND YOU'RE FALLING.
↓
THE SUN IS SHINING IN YOUR EYES.
↓
YOU REALIZE YOU ARE NOT AWAKE.

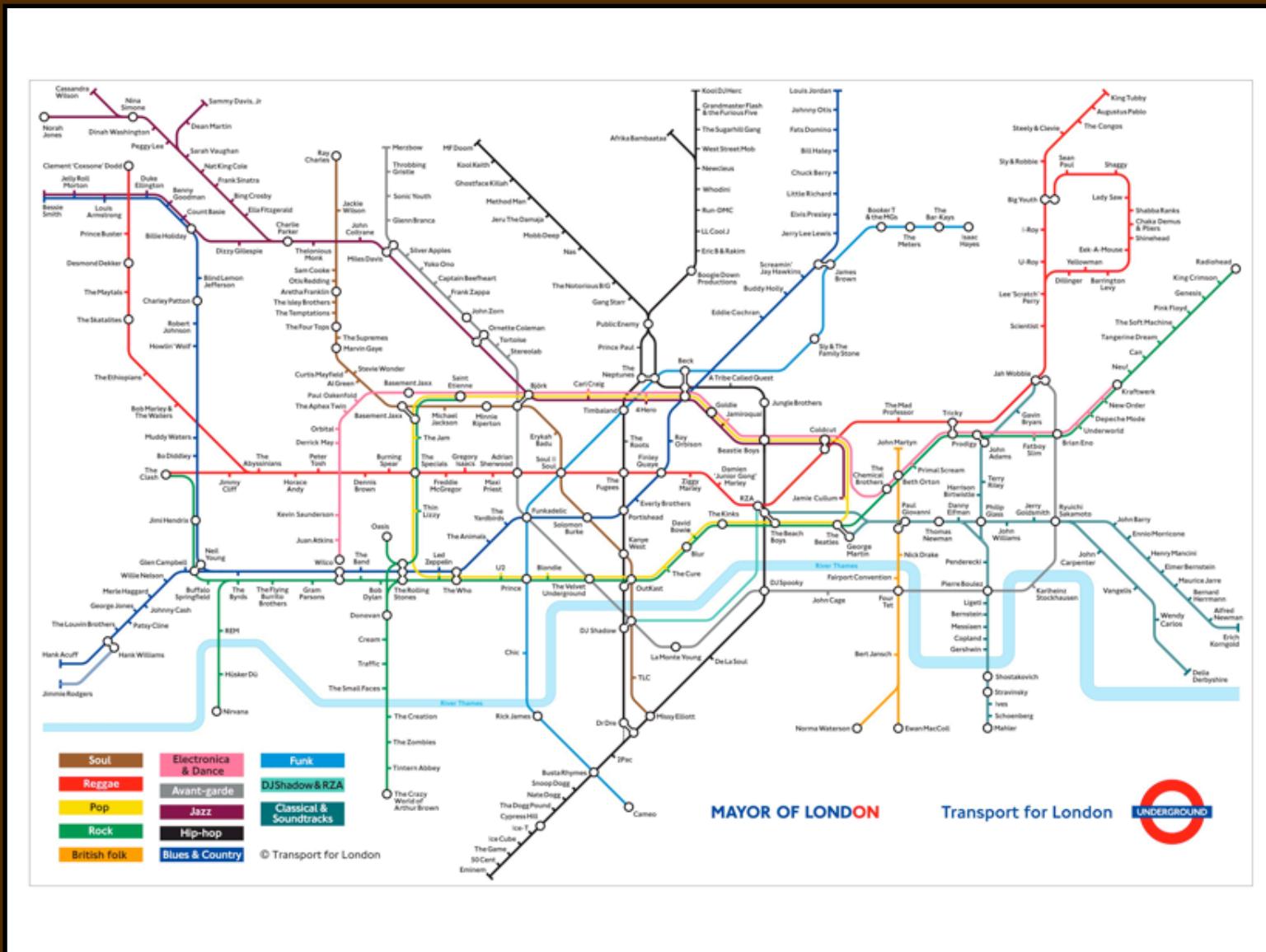


prevalence of nodemaps in visualization art

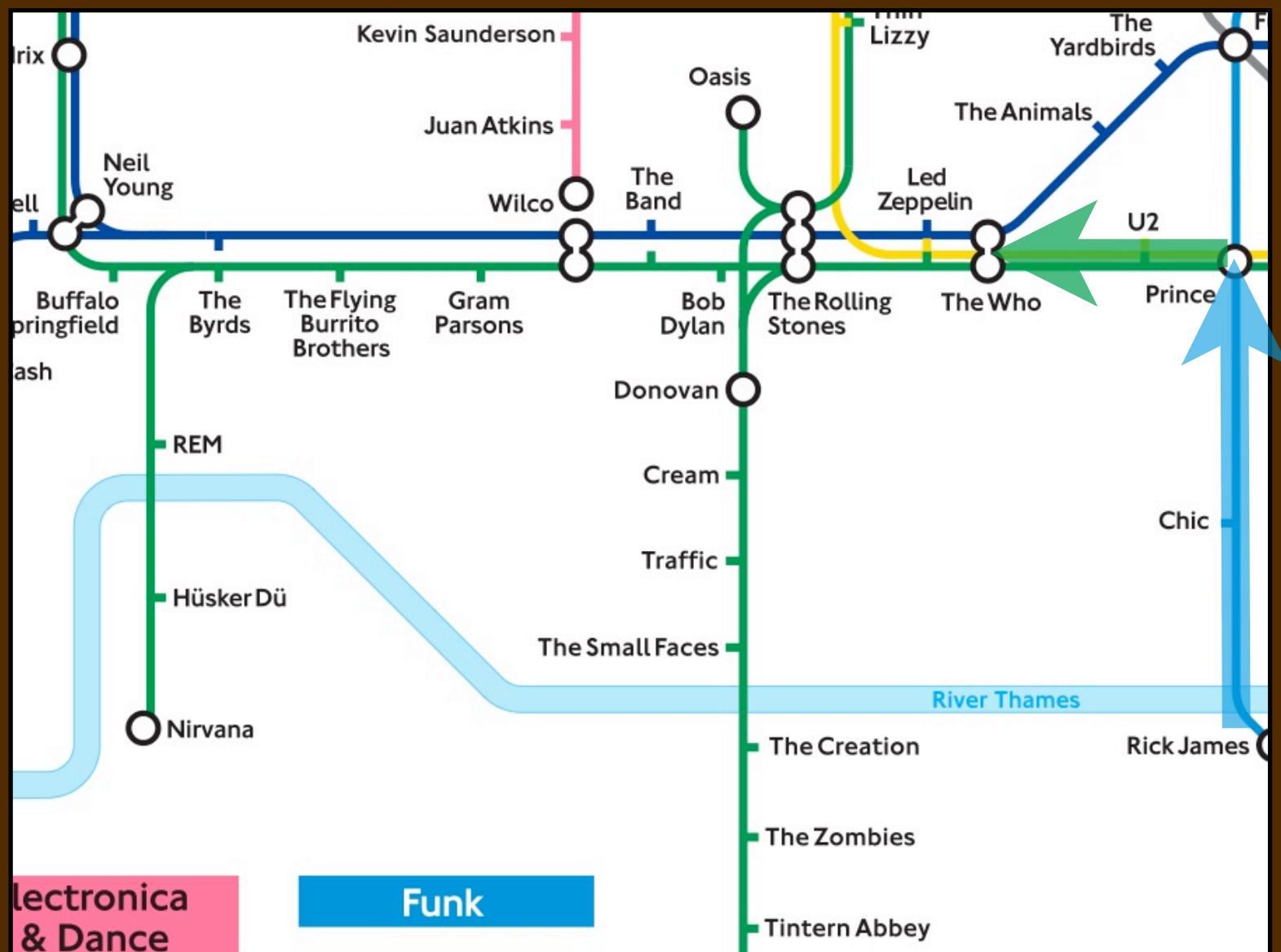


VisualComplexity.com (2008)

subway map remix art: specifying transitions

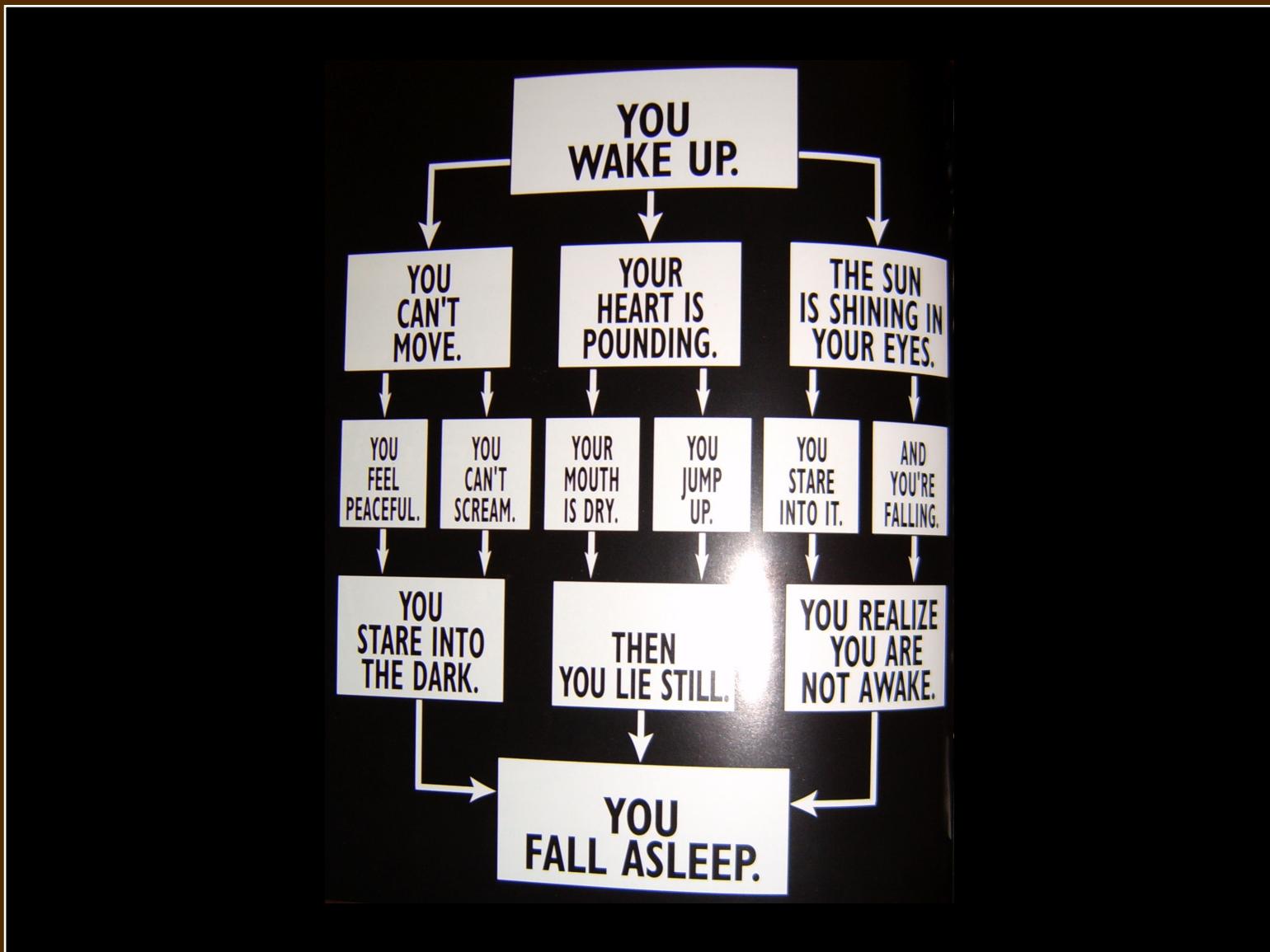


Dorian Lynskey's “Going Underground” (2006)



Dorian Lynskey's "Going Underground" (2006)

flowchart / directed graph as CYOA narrative



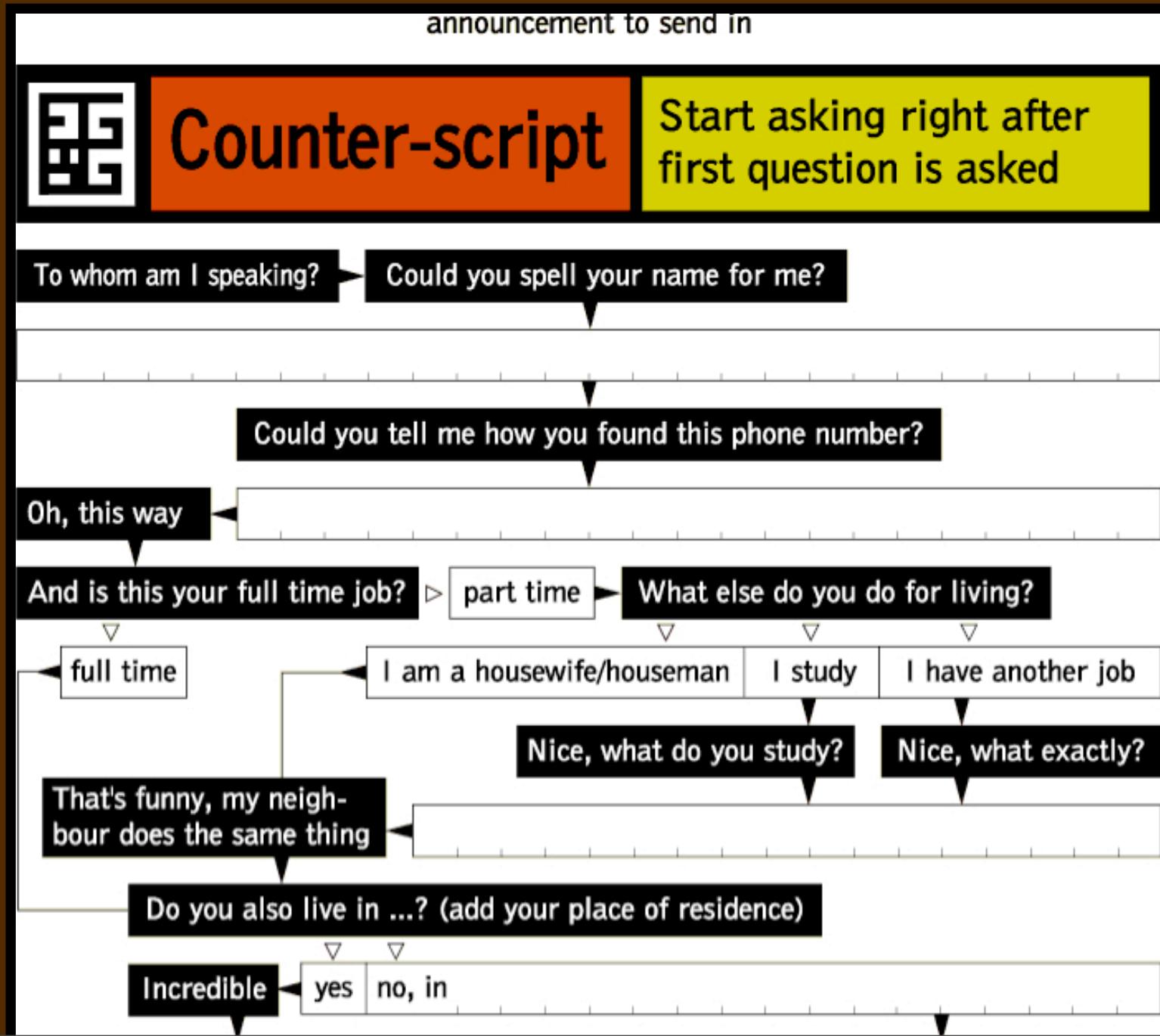
from Bill Barker's SCHWA

EGBG

Counter-script

Martijn Engelbregt

visual flow specifies the protocol of performance



HOW ARE YOU?

BAD

GOOD

VPRO GUIDE

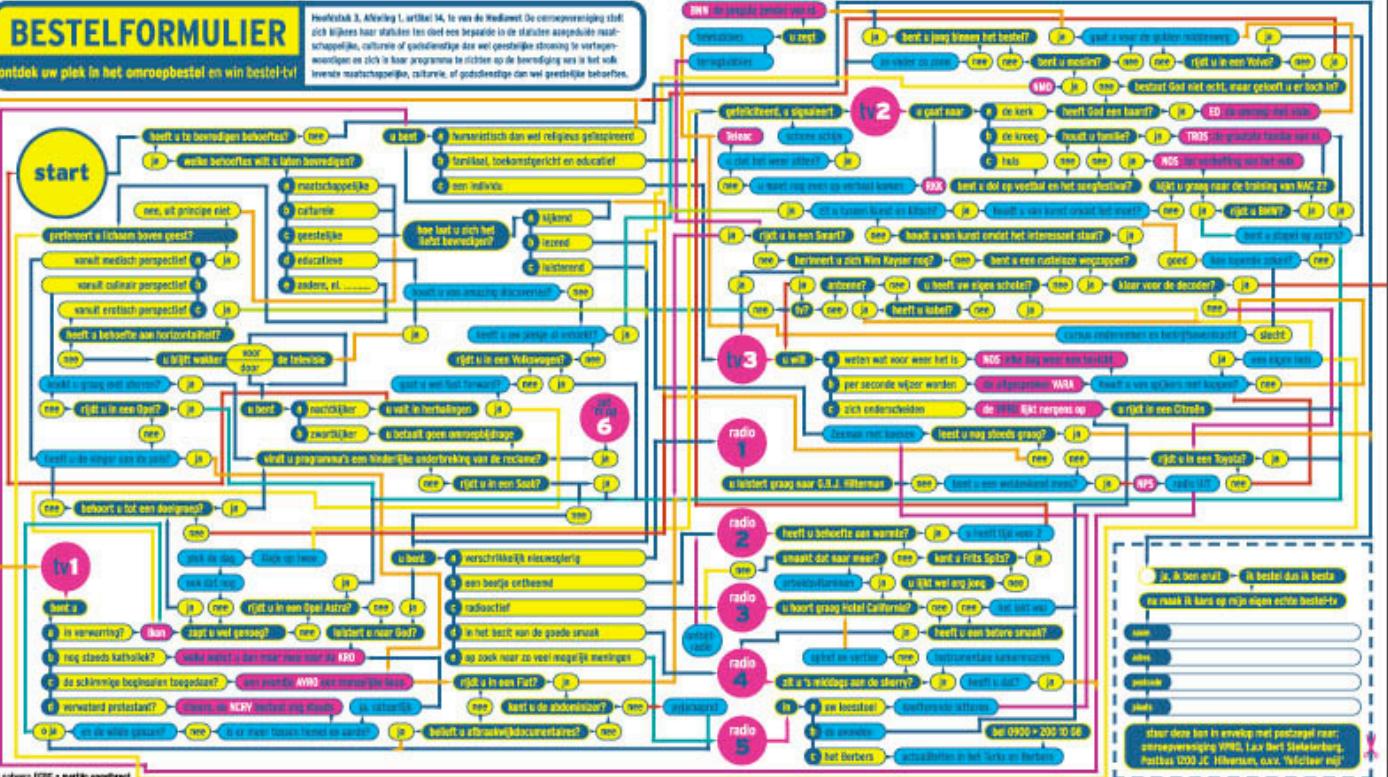
IMPROVE YOURSELF

8 FREE ISSUES FOR NEW SUBSCRIBERS

PHONE

+31-35-6712888

VPRO



TOE AAN EEN BURN OUT?

doe de test en maak kans op een uurtje ontspannende inspanning

Denk lang en diep na over elke vraag. De eerste antwoord is doorgaans niet het beste.
Volg zo'n lijstje met een blauwe pen.

start bij de rode ronde stip linksboven

wilt u ook alternatieve kalmtes u niet voor waar u eigen toe moet?

wilt u graag nachten lang aan de neus orangen zetten?

van welk van de volgende psychische klachten heeft u het liefst last?

- lasteloos gevoel

- prijsgevoeligheid

- snel geïrriteerd

- moeiteloos gevoel

- stemmingswisselingen

- rusteloos

- opgewoed

doet het u plezier te erkennen dat elke u kan schakelen?

wilt u het drukke wanneer mogelijk a voorhoede vragen naar de

- geestelijker

- hoofdhoofd

- voorhoede

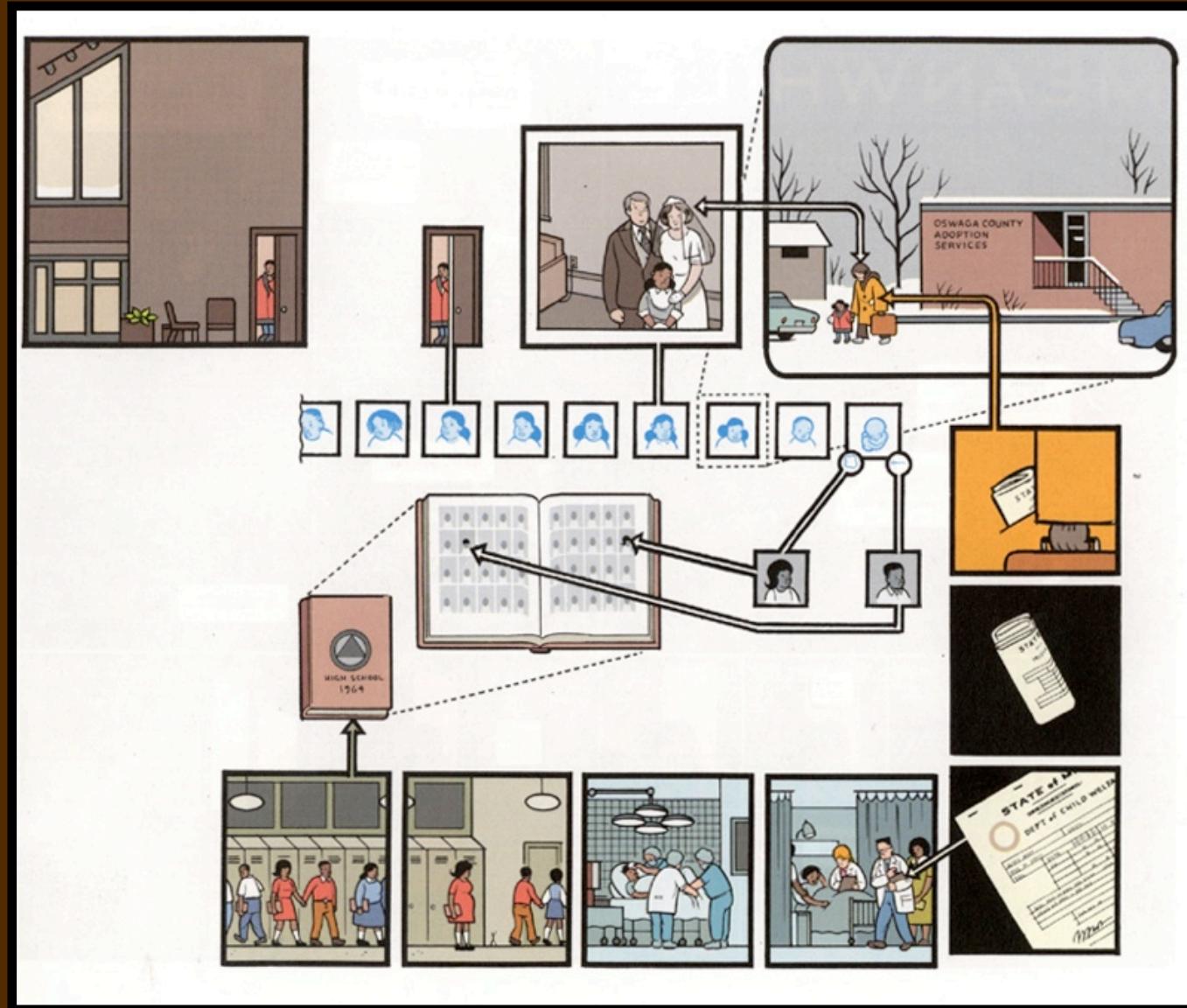
</

document mapping to trace branching paths



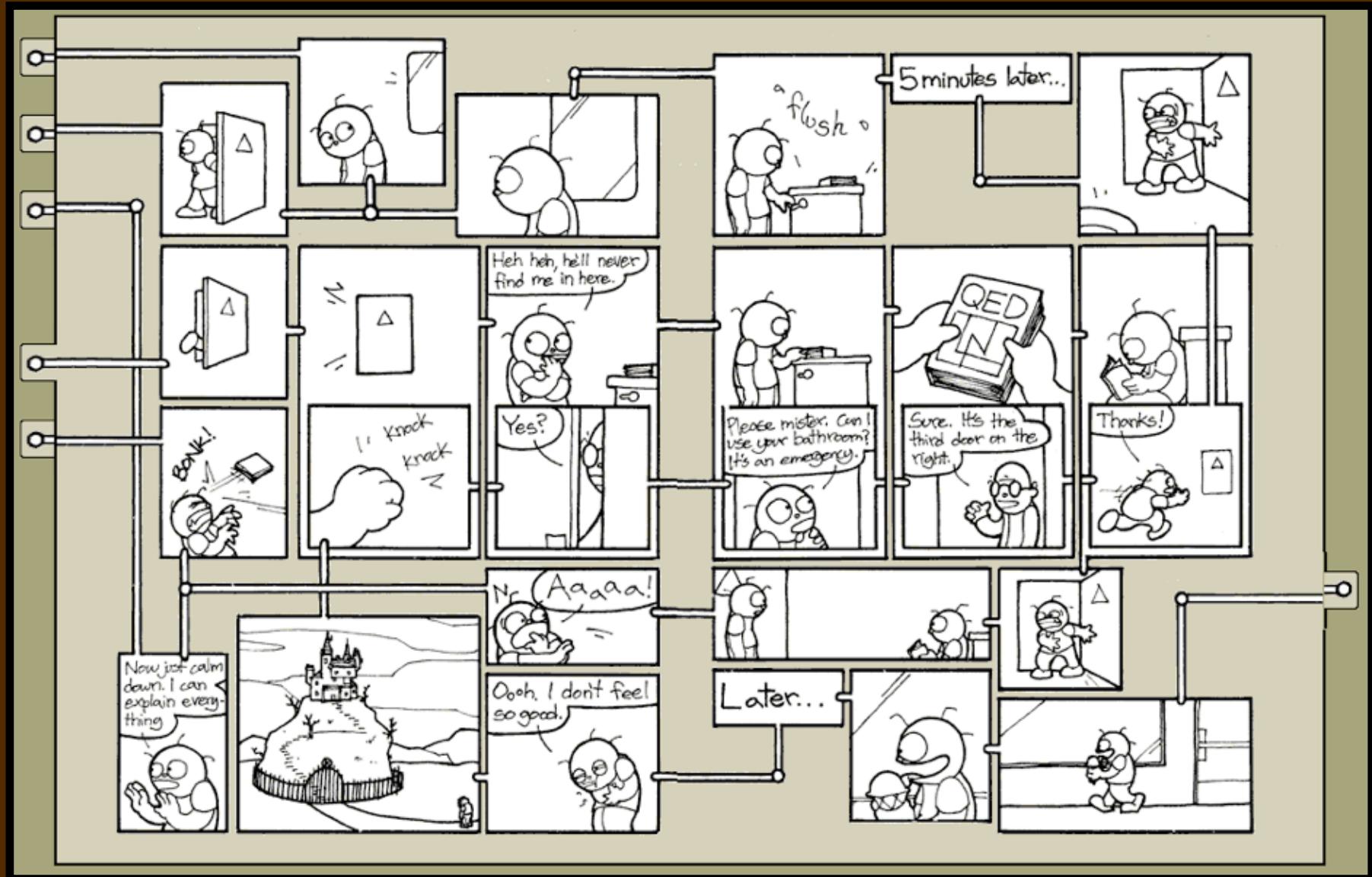
House of Leaves: mapping the labyrinth (2008)

comic links complex backstories with paths



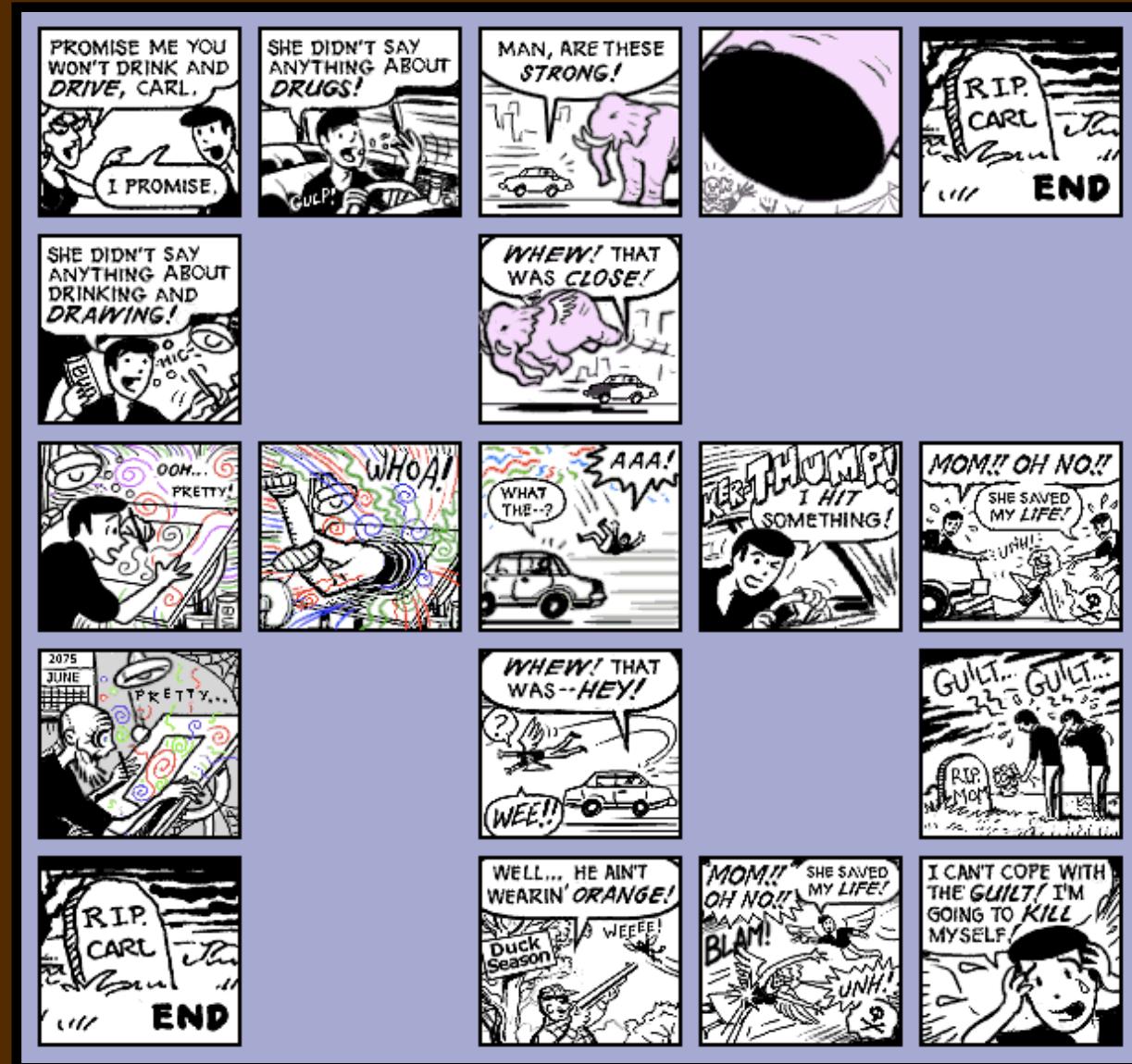
Chris Ware's *Jimmy Corrigan* [detail]

branching comic connects choices with paths



Jason Shiga's Meanwhile

branching comic embeds choices in a grid



Scott McCloud's *Carl* [detail]

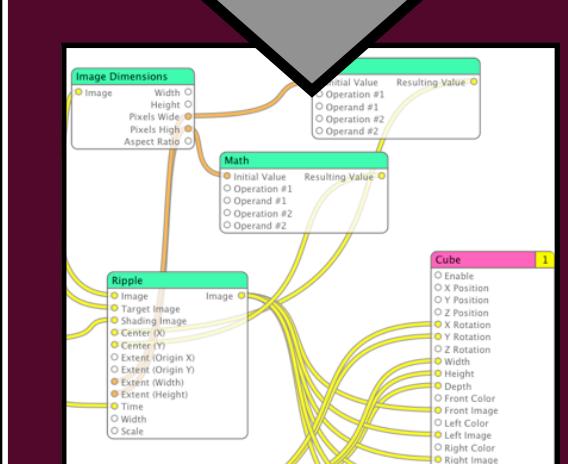
a variety of ways to signify
sequence,
relation, &
procedure

**what techniques do
flow art &**

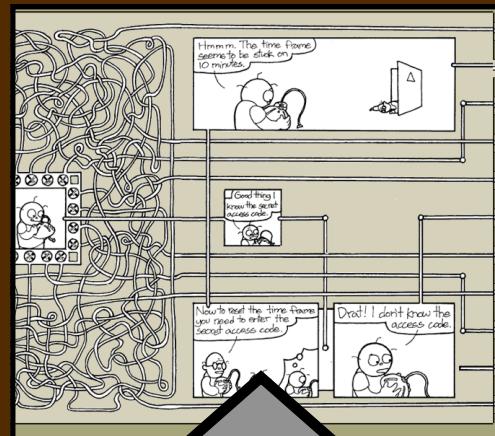
**flow programming
have in common?**

VISUAL PROGRAMMING

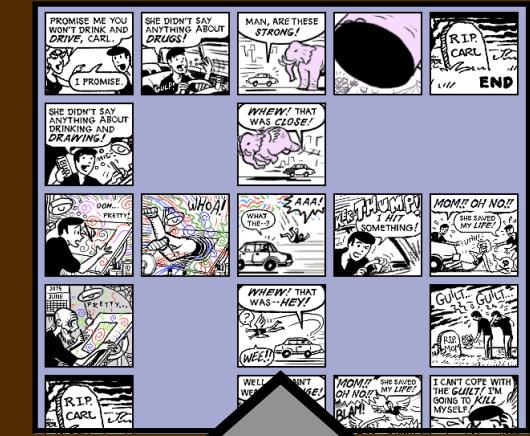
FLOW ART



PATCH



PATCH



GRID

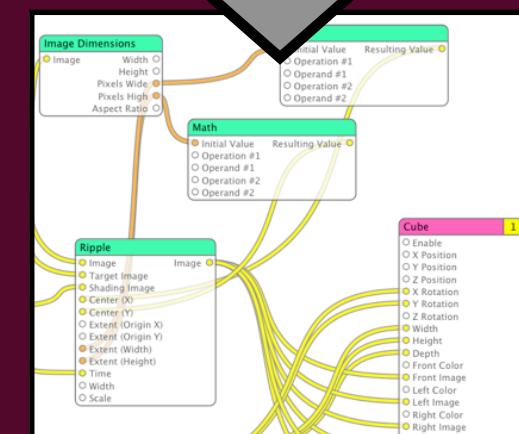
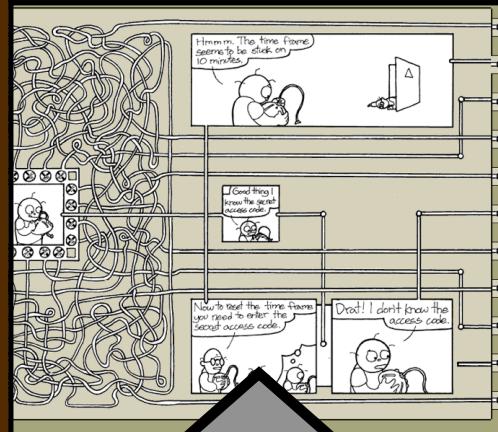


GRID

FLOW
ART

VISUAL
PROGRAMMING

PATCH

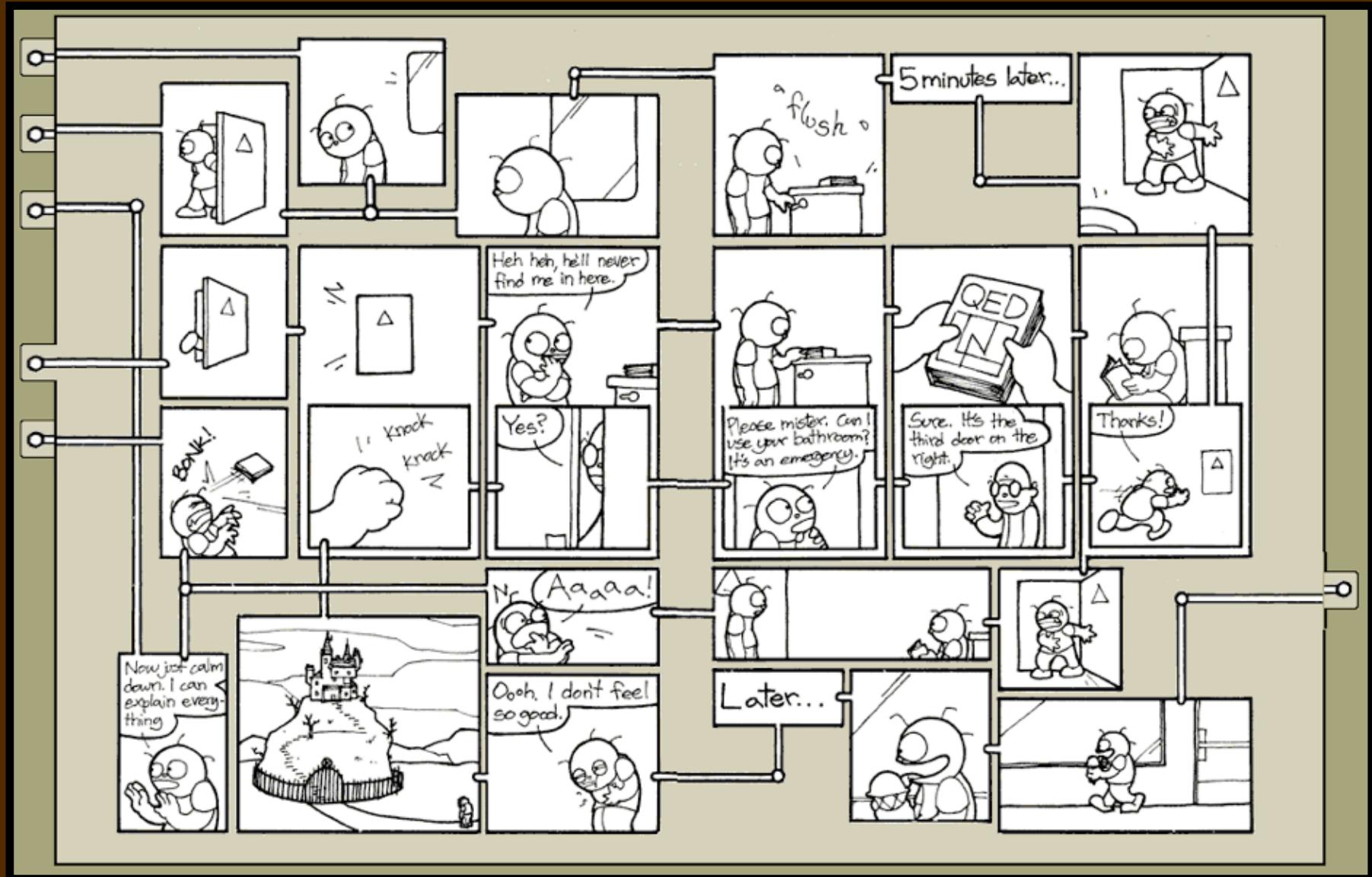


PATCH

GRID

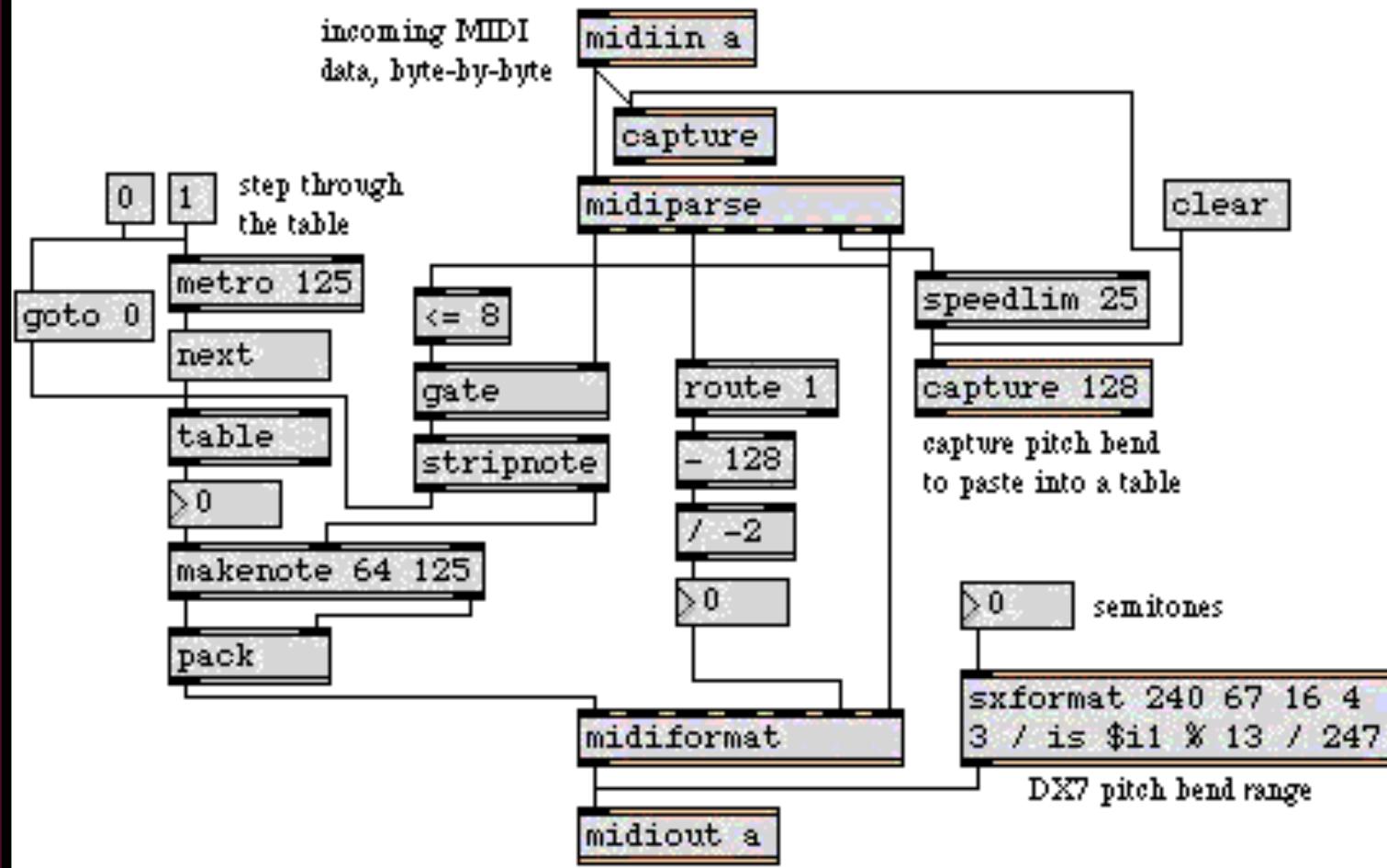
GRID

branching comic connects choices with paths

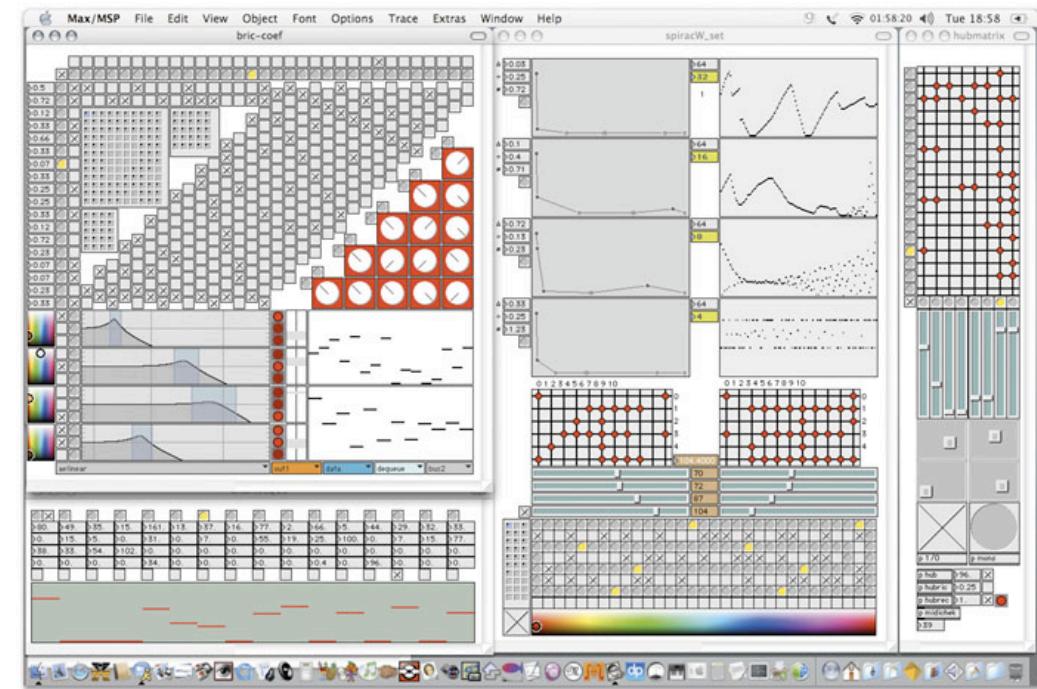
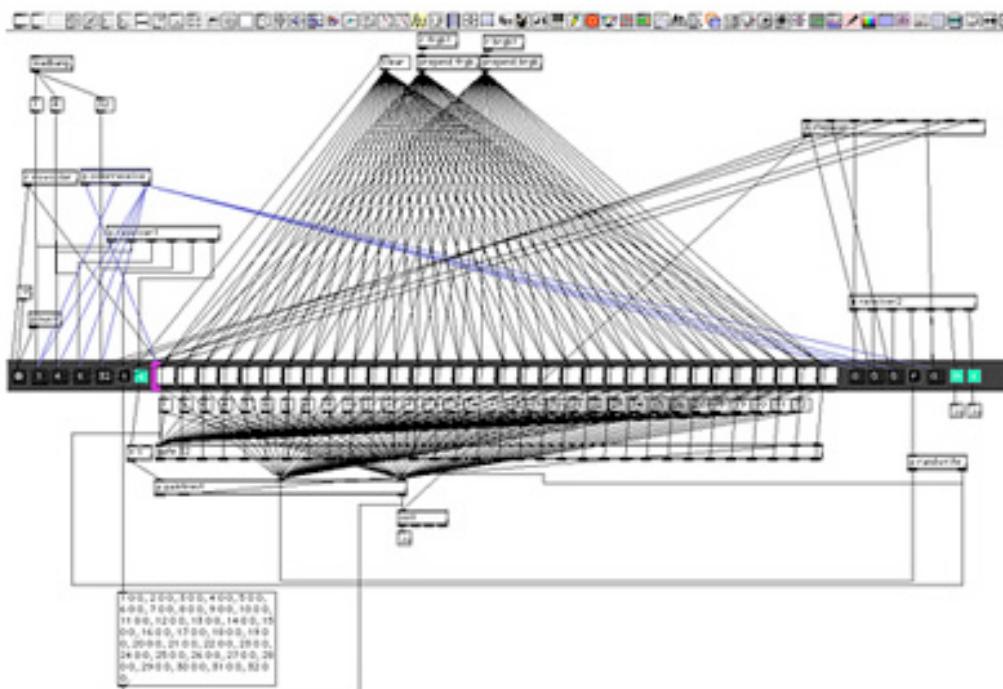
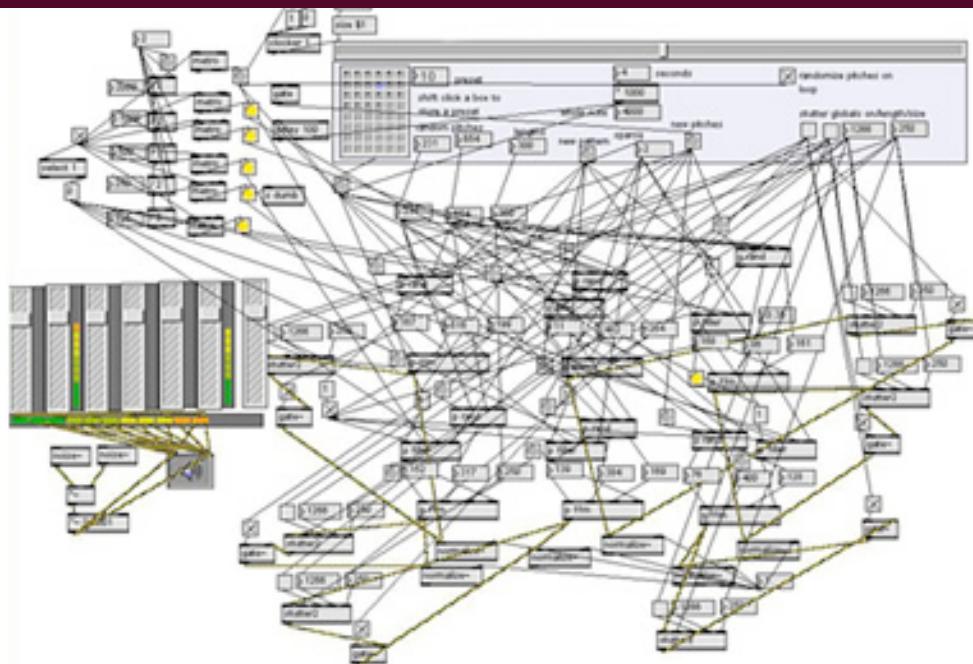
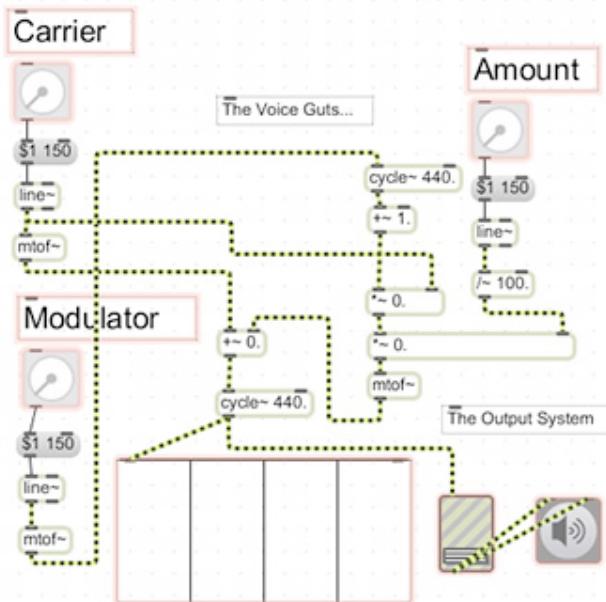


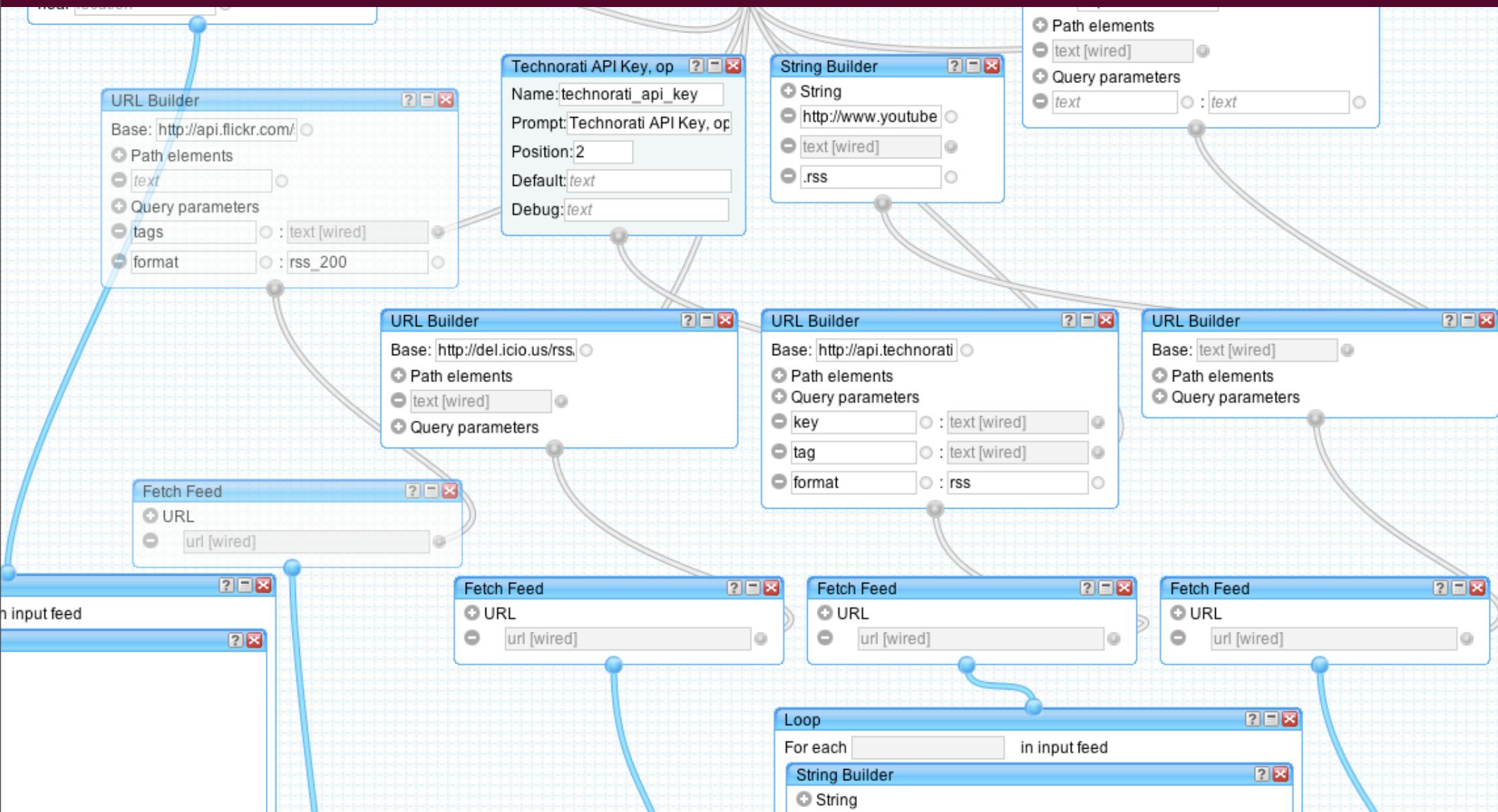
Jason Shiga's Meanwhile

Max/MSP

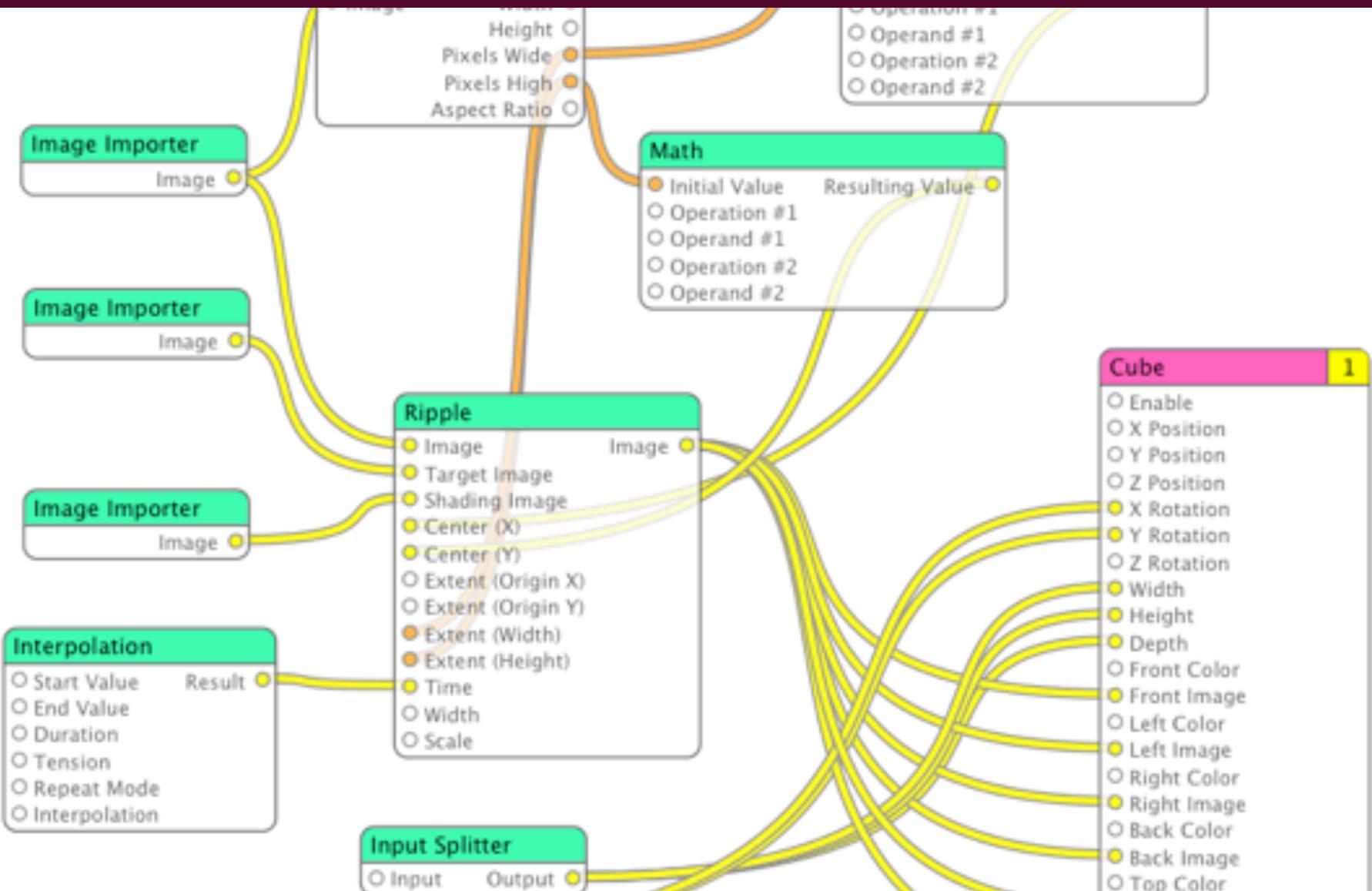


Miller Puckette / Cycling74's Max / MSP



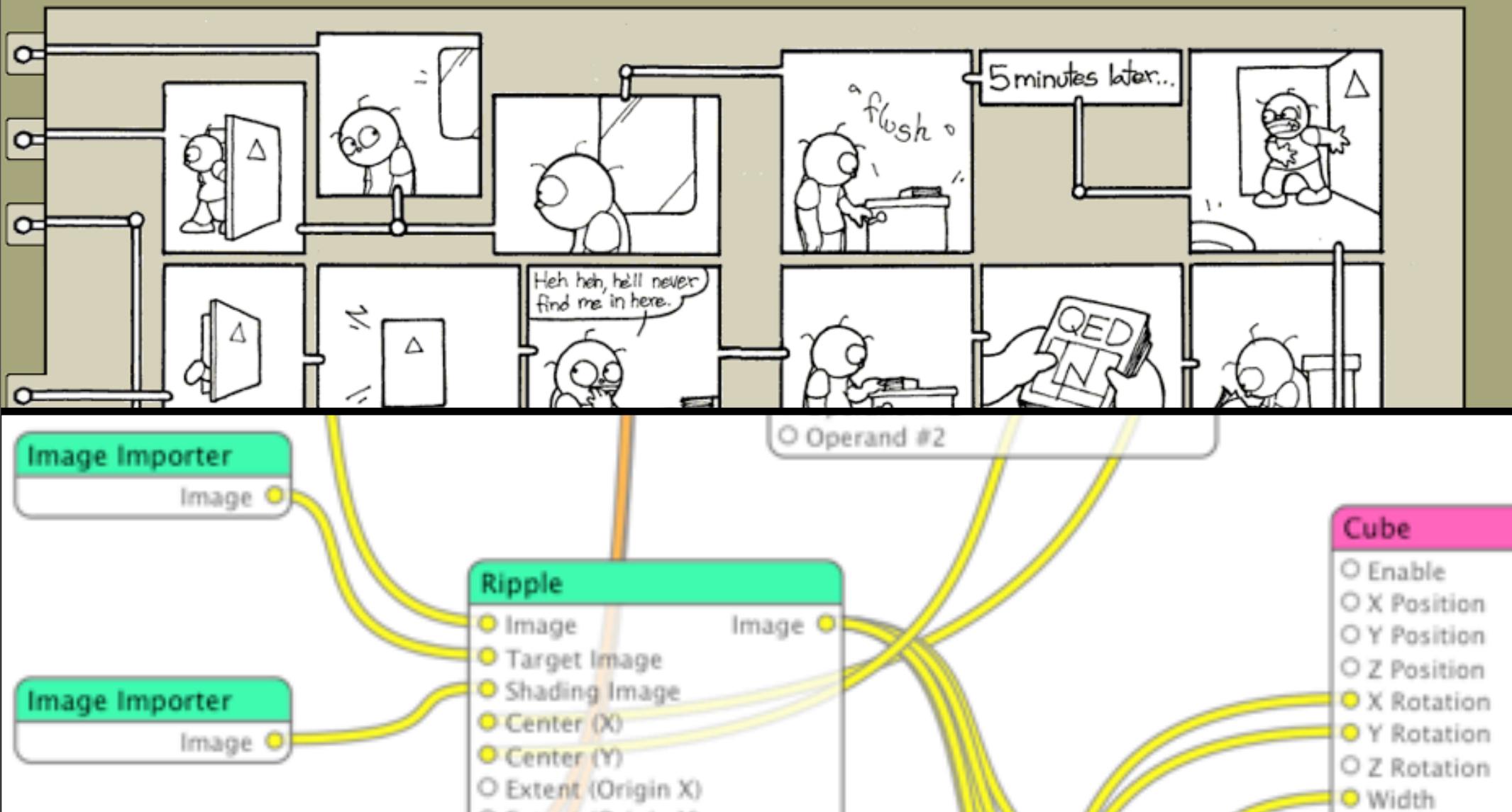


Yahoo! Pipes



Quartz Composer (OS X)

Jason Shiga's Meanwhile



Quartz Composer (OS X)

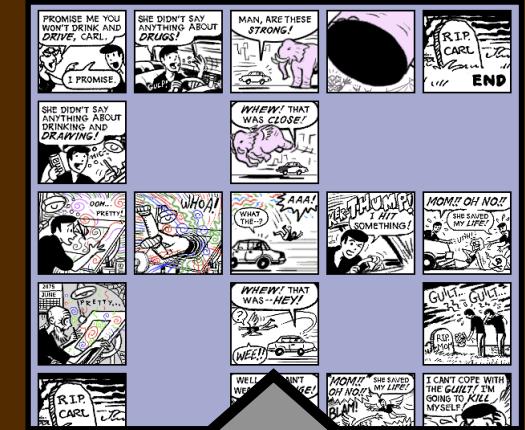
**FLOW
ART**

**VISUAL
PROGRAMMING**

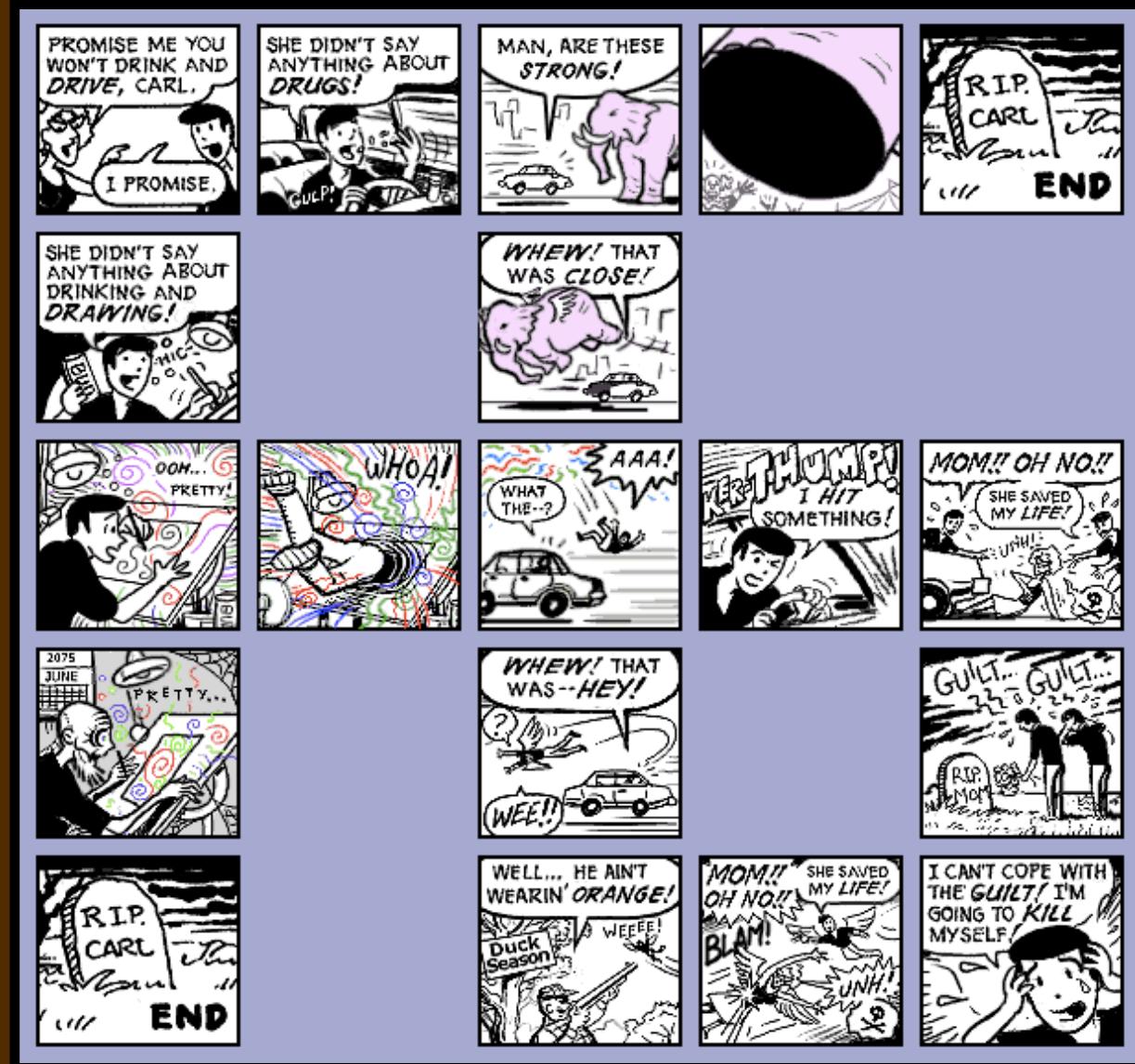
PATCH

PATCH

GRID



GRID



Scott McCloud's *Carl* [detail]



PAL



Carnage
Heart

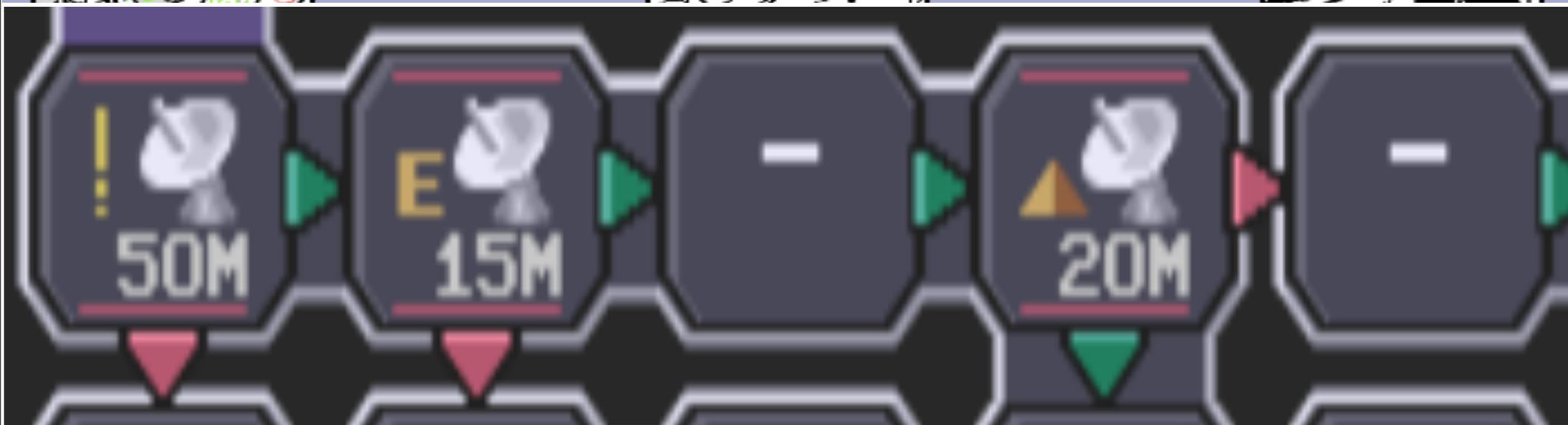
ARTDINK

PlayStation

Carnage Heart

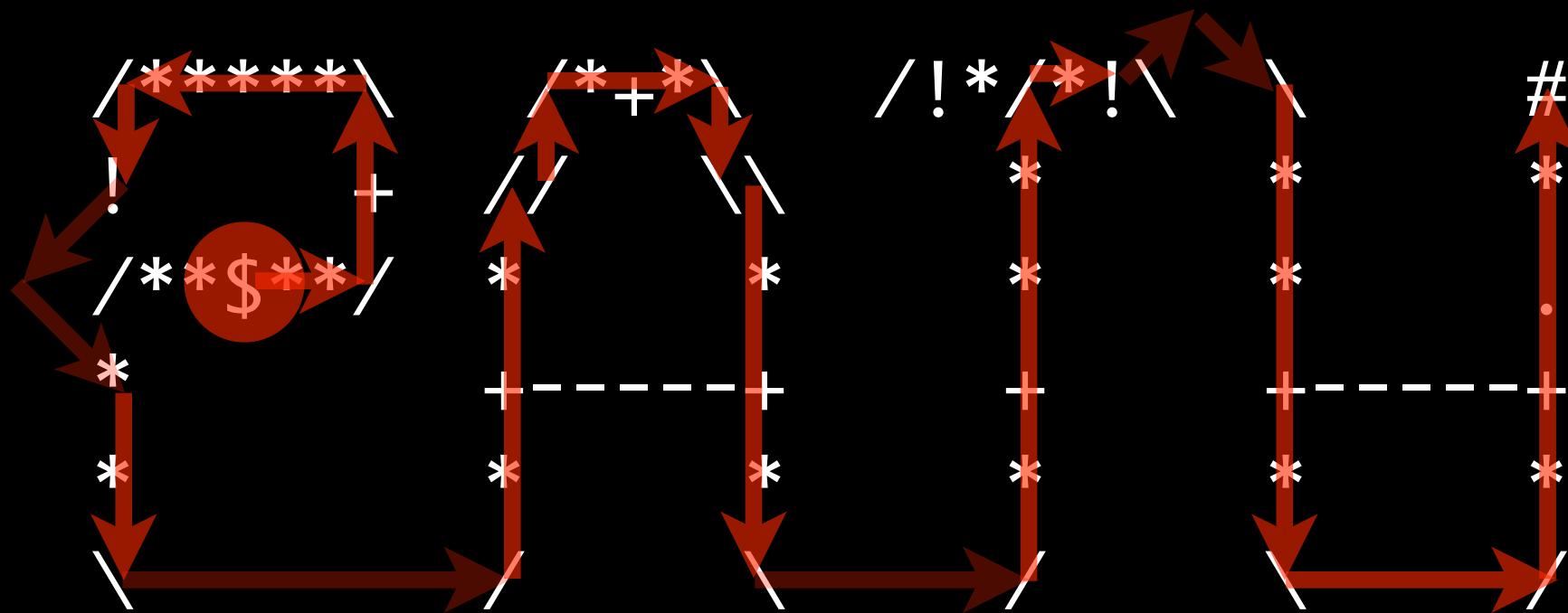


Scott McCloud's *Carl* [detail]



Carnage Heart: robot program [detail]

... “cartesian programming”



PATH esoteric / weird programming language

“ the PATH interpreter follows a path of instruction symbols which make up a program. The program can go up, down, left and right, and can also overlap itself. Also, the interpreter simply skips over any characters that aren't valid instruction symbols. ”

- \$ Start the program here, heading right.
- # End the program.
- / Turn 90 degrees without crossing line.
- \ Turn 90 degrees without crossing line
- + Increment the current memory cell.
- Decrement the current memory cell.
- ! Jump over the next symbol.
- .
- . Output a character from current memory.
- ,
- , Input a character from current memory.
- }
- } Move to the next memory cell.
- {
- { Move to the previous memory cell.
- ^
- ^ If current memory is not 0, turn up.
- <
- < If current memory is not 0, turn left.
- >
- > If current memory is not 0, turn right.
- v
- v If current memory is not 0, turn down.

PATCH

GRID

objects

directions

sequences

axes

relationships

spatial relations

conclusion

SOURCE

IMPERATIVE PROGRAMMING

```
#include <control.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "cold.h"
#include "joy.h"

char * patc="\n\nPress any key to continue

char move(char now){
    char k;
    if(Up[now]==0) Up[now]=now; //if key UP
    0 page
    while(k=joyinput()){
        //  if (now==0 && k==JOY_UP) return now
        if(k==JOY_A &&A[now]) return A[now]
        if(k==JOY_B &&B[now]) return B[now]
        if(k==JOY_SEL &&Select[now]) return
        if(k==JOY_STA &&Start[now]) return
        if(k==JOY_UP &&Up[now]) return Up[now]
        if(k==JOY_DN &&Down[now]) return Down[now]
        if(k==JOY_LF &&Left[now]) return Left[now]
        if(k==JOY_RT &&Right[now]) return Right[now]
    }
}

void clsO{
    waitblank();
    clrscr();
}
```

NATURAL

NATURAL LANGUAGE PROGRAMMING

The castle exterior is scenery in the drawbridge. The printed name of the castle exterior is "castle". Understand "tower" or "tower" or "drawbridge" or "bridge" as the castle exterior. The description is "The drawbridge looks longer than it actually is; the towers are so high that the tops are lost in cloud, and looking east or west, you cannot see the furthest extent of the walls. An optical illusion: it is smaller inside.

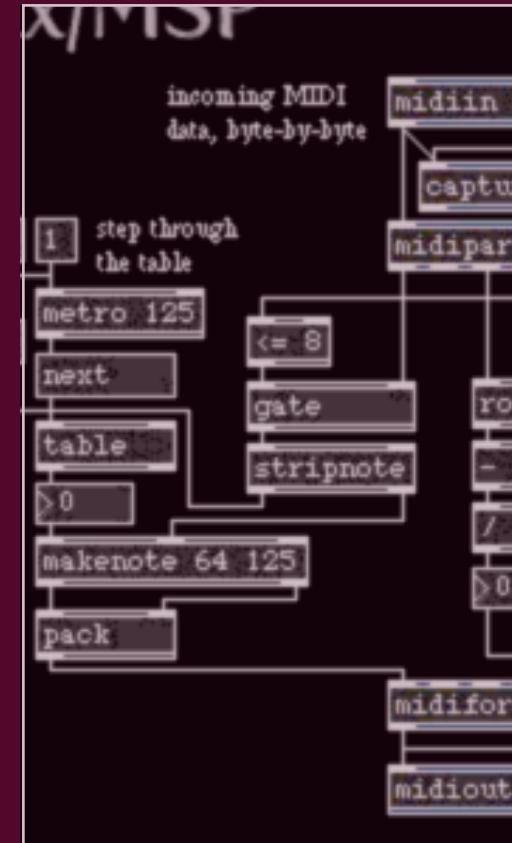
Probably."

The iron-barred gate is a door. "An iron-barred gate leads [gate direction]." It is north of the Drawbridge and south of the Entrance Hall. It is closed and openable. Before entering the castle, try entering the gate instead. Before going inside in the Drawbridge, try going north instead. Understand "door" as the gate.

After opening the gate:
say "You shouldn't be able to

VISUAL

PATCH AND FLOW CONTROL PROGRAMMING



```
#include <conio.h>
#include <string.h>

#define FULLSCREEN 512
char buf[FULLSCREEN+1];

#include "col.h"
#include "joy.h"

char * patc="";

char move(char k);
char k;
if(Up[now]==0) page++;
while(k!=j)
// if (n==0) {
if(k==1) if(k==2) if(k==3) if(k==4) if(k==5) if(k==6) if(k==7) if(k==8) if(k==9) if(k==10) if(k==11) if(k==12) if(k==13) if(k==14) if(k==15) if(k==16) if(k==17) if(k==18) if(k==19) if(k==20) if(k==21) if(k==22) if(k==23) if(k==24) if(k==25) if(k==26) if(k==27) if(k==28) if(k==29) if(k==30) if(k==31) if(k==32) if(k==33) if(k==34) if(k==35) if(k==36) if(k==37) if(k==38) if(k==39) if(k==40) if(k==41) if(k==42) if(k==43) if(k==44) if(k==45) if(k==46) if(k==47) if(k==48) if(k==49) if(k==50) if(k==51) if(k==52) if(k==53) if(k==54) if(k==55) if(k==56) if(k==57) if(k==58) if(k==59) if(k==60) if(k==61) if(k==62) if(k==63) if(k==64) if(k==65) if(k==66) if(k==67) if(k==68) if(k==69) if(k==70) if(k==71) if(k==72) if(k==73) if(k==74) if(k==75) if(k==76) if(k==77) if(k==78) if(k==79) if(k==80) if(k==81) if(k==82) if(k==83) if(k==84) if(k==85) if(k==86) if(k==87) if(k==88) if(k==89) if(k==90) if(k==91) if(k==92) if(k==93) if(k==94) if(k==95) if(k==96) if(k==97) if(k==98) if(k==99) if(k==100) if(k==101) if(k==102) if(k==103) if(k==104) if(k==105) if(k==106) if(k==107) if(k==108) if(k==109) if(k==110) if(k==111) if(k==112) if(k==113) if(k==114) if(k==115) if(k==116) if(k==117) if(k==118) if(k==119) if(k==120) if(k==121) if(k==122) if(k==123) if(k==124) if(k==125) if(k==126) if(k==127) if(k==128) if(k==129) if(k==130) if(k==131) if(k==132) if(k==133) if(k==134) if(k==135) if(k==136) if(k==137) if(k==138) if(k==139) if(k==140) if(k==141) if(k==142) if(k==143) if(k==144) if(k==145) if(k==146) if(k==147) if(k==148) if(k==149) if(k==150) if(k==151) if(k==152) if(k==153) if(k==154) if(k==155) if(k==156) if(k==157) if(k==158) if(k==159) if(k==160) if(k==161) if(k==162) if(k==163) if(k==164) if(k==165) if(k==166) if(k==167) if(k==168) if(k==169) if(k==170) if(k==171) if(k==172) if(k==173) if(k==174) if(k==175) if(k==176) if(k==177) if(k==178) if(k==179) if(k==180) if(k==181) if(k==182) if(k==183) if(k==184) if(k==185) if(k==186) if(k==187) if(k==188) if(k==189) if(k==190) if(k==191) if(k==192) if(k==193) if(k==194) if(k==195) if(k==196) if(k==197) if(k==198) if(k==199) if(k==200) if(k==201) if(k==202) if(k==203) if(k==204) if(k==205) if(k==206) if(k==207) if(k==208) if(k==209) if(k==210) if(k==211) if(k==212) if(k==213) if(k==214) if(k==215) if(k==216) if(k==217) if(k==218) if(k==219) if(k==220) if(k==221) if(k==222) if(k==223) if(k==224) if(k==225) if(k==226) if(k==227) if(k==228) if(k==229) if(k==230) if(k==231) if(k==232) if(k==233) if(k==234) if(k==235) if(k==236) if(k==237) if(k==238) if(k==239) if(k==240) if(k==241) if(k==242) if(k==243) if(k==244) if(k==245) if(k==246) if(k==247) if(k==248) if(k==249) if(k==250) if(k==251) if(k==252) if(k==253) if(k==254) if(k==255) if(k==256) if(k==257) if(k==258) if(k==259) if(k==260) if(k==261) if(k==262) if(k==263) if(k==264) if(k==265) if(k==266) if(k==267) if(k==268) if(k==269) if(k==270) if(k==271) if(k==272) if(k==273) if(k==274) if(k==275) if(k==276) if(k==277) if(k==278) if(k==279) if(k==280) if(k==281) if(k==282) if(k==283) if(k==284) if(k==285) if(k==286) if(k==287) if(k==288) if(k==289) if(k==290) if(k==291) if(k==292) if(k==293) if(k==294) if(k==295) if(k==296) if(k==297) if(k==298) if(k==299) if(k==300) if(k==301) if(k==302) if(k==303) if(k==304) if(k==305) if(k==306) if(k==307) if(k==308) if(k==309) if(k==310) if(k==311) if(k==312) if(k==313) if(k==314) if(k==315) if(k==316) if(k==317) if(k==318) if(k==319) if(k==320) if(k==321) if(k==322) if(k==323) if(k==324) if(k==325) if(k==326) if(k==327) if(k==328) if(k==329) if(k==330) if(k==331) if(k==332) if(k==333) if(k==334) if(k==335) if(k==336) if(k==337) if(k==338) if(k==339) if(k==340) if(k==341) if(k==342) if(k==343) if(k==344) if(k==345) if(k==346) if(k==347) if(k==348) if(k==349) if(k==350) if(k==351) if(k==352) if(k==353) if(k==354) if(k==355) if(k==356) if(k==357) if(k==358) if(k==359) if(k==360) if(k==361) if(k==362) if(k==363) if(k==364) if(k==365) if(k==366) if(k==367) if(k==368) if(k==369) if(k==370) if(k==371) if(k==372) if(k==373) if(k==374) if(k==375) if(k==376) if(k==377) if(k==378) if(k==379) if(k==380) if(k==381) if(k==382) if(k==383) if(k==384) if(k==385) if(k==386) if(k==387) if(k==388) if(k==389) if(k==390) if(k==391) if(k==392) if(k==393) if(k==394) if(k==395) if(k==396) if(k==397) if(k==398) if(k==399) if(k==400) if(k==401) if(k==402) if(k==403) if(k==404) if(k==405) if(k==406) if(k==407) if(k==408) if(k==409) if(k==410) if(k==411) if(k==412) if(k==413) if(k==414) if(k==415) if(k==416) if(k==417) if(k==418) if(k==419) if(k==420) if(k==421) if(k==422) if(k==423) if(k==424) if(k==425) if(k==426) if(k==427) if(k==428) if(k==429) if(k==430) if(k==431) if(k==432) if(k==433) if(k==434) if(k==435) if(k==436) if(k==437) if(k==438) if(k==439) if(k==440) if(k==441) if(k==442) if(k==443) if(k==444) if(k==445) if(k==446) if(k==447) if(k==448) if(k==449) if(k==450) if(k==451) if(k==452) if(k==453) if(k==454) if(k==455) if(k==456) if(k==457) if(k==458) if(k==459) if(k==460) if(k==461) if(k==462) if(k==463) if(k==464) if(k==465) if(k==466) if(k==467) if(k==468) if(k==469) if(k==470) if(k==471) if(k==472) if(k==473) if(k==474) if(k==475) if(k==476) if(k==477) if(k==478) if(k==479) if(k==480) if(k==481) if(k==482) if(k==483) if(k==484) if(k==485) if(k==486) if(k==487) if(k==488) if(k==489) if(k==490) if(k==491) if(k==492) if(k==493) if(k==494) if(k==495) if(k==496) if(k==497) if(k==498) if(k==499) if(k==500) if(k==501) if(k==502) if(k==503) if(k==504) if(k==505) if(k==506) if(k==507) if(k==508) if(k==509) if(k==510) if(k==511) if(k==512)
```

Max/MSP

incoming MIDI
data, byte-by-byte



SOURCE

sequence

literacy

write

FLOW

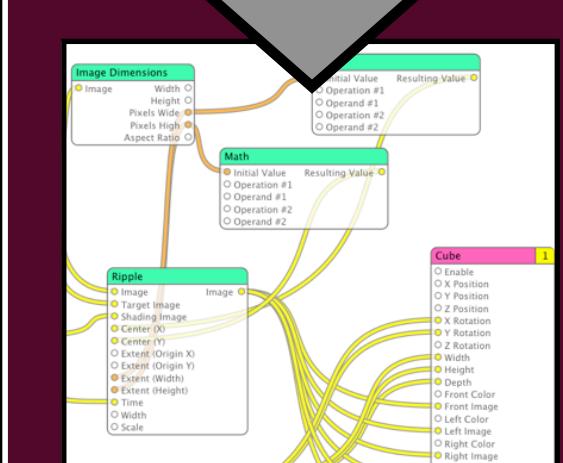
space

visual literacy

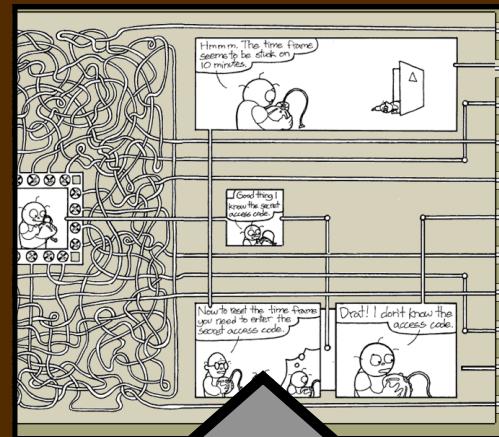
map / graph

VISUAL PROGRAMMING

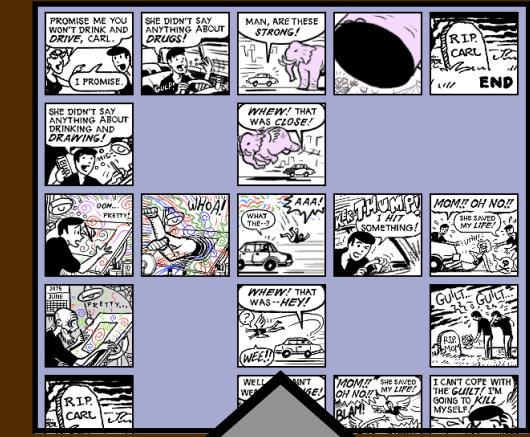
FLOW ART



PATCH



PATCH



GRID



GRID

PATCH

GRID

objects

directions

sequences

axes

relationships

spatial relations

there is no one type of
code to criticize

every logic of specification
is its own
paradigm for critique

Jeremy Douglass

Postdoctoral Researcher
Software Studies Initiative
University of California San Diego