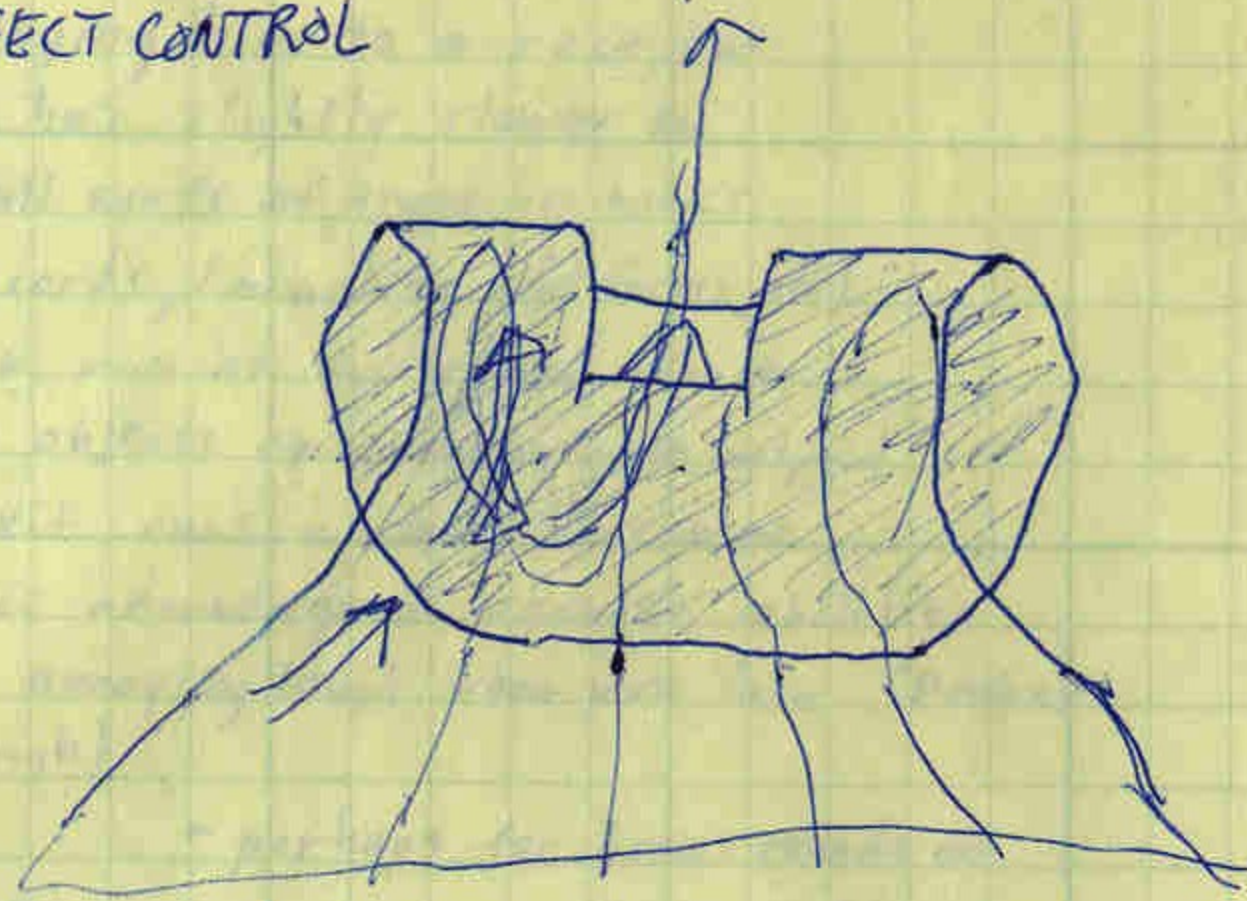
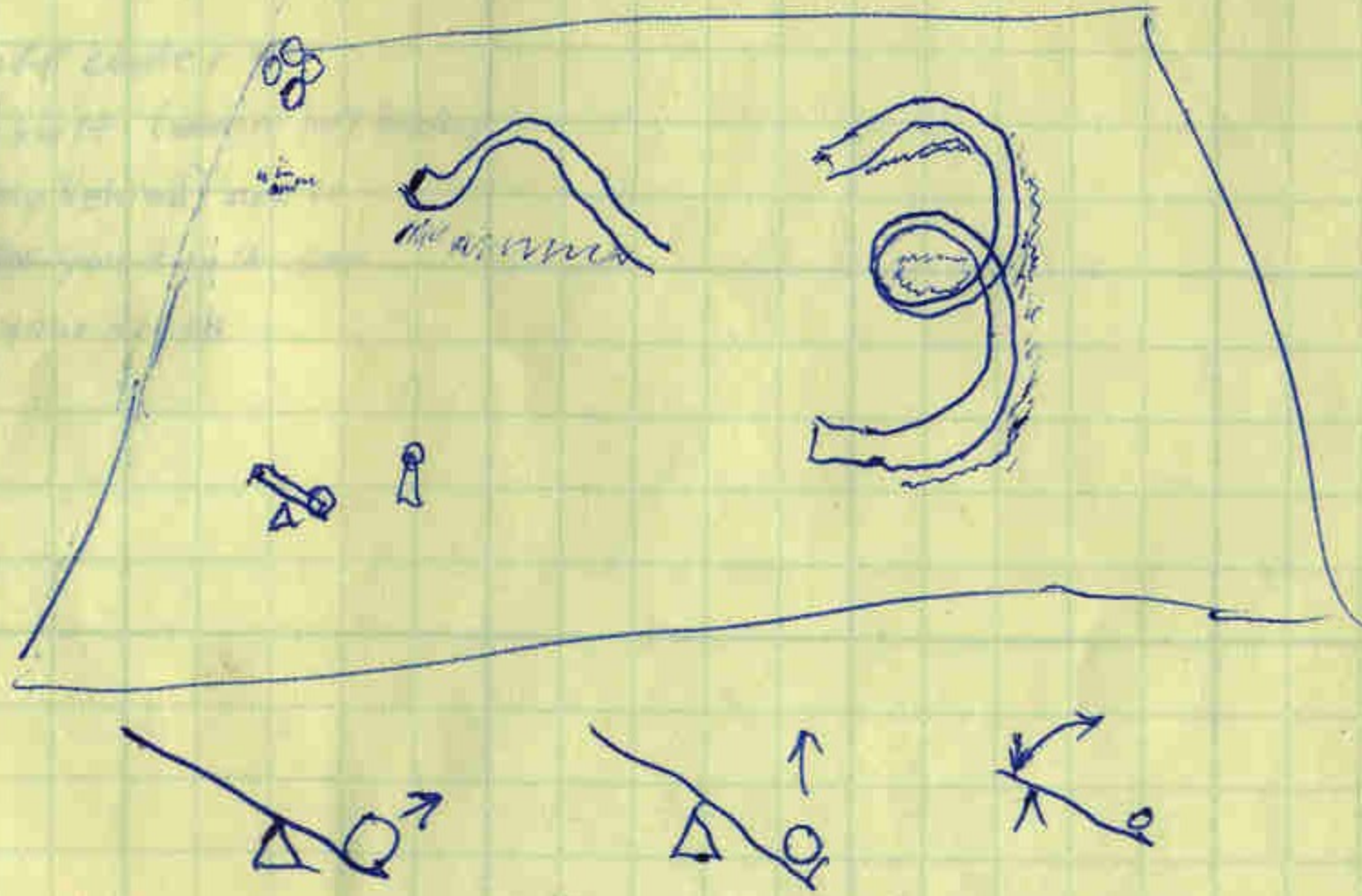


SONIC = SPEED
 BEAUTY
 SIMPLICITY
 PERFECT CONTROL



Concepts by Don Goddard
 Contacted by hnxg and drx

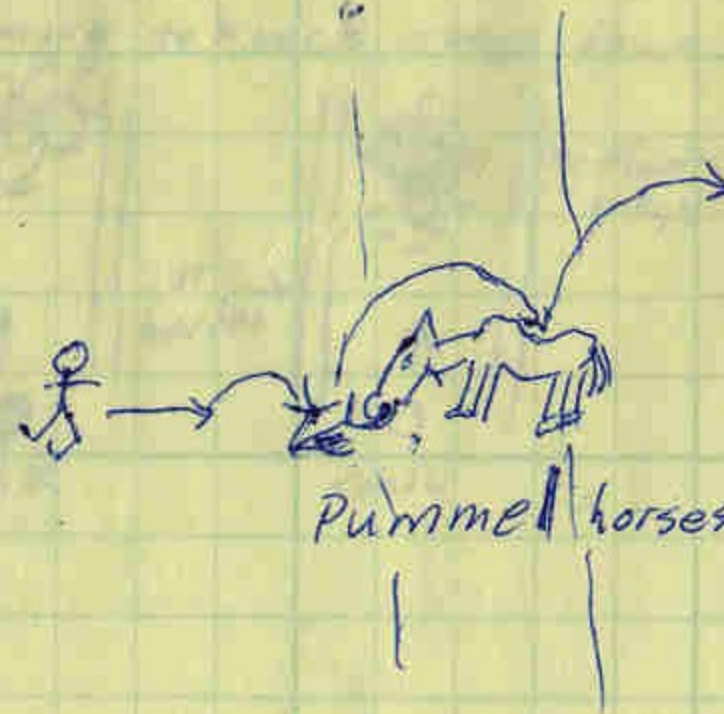


Shadow world

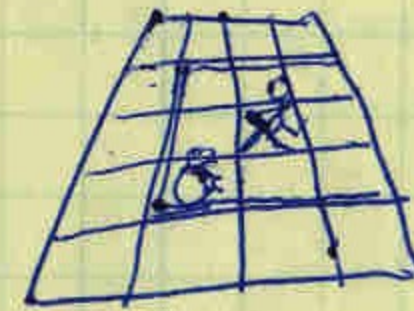


see thread spider swings in and out of shadows, if grabbed spider slings rings out in direction swinging

see eyes and outline, either vicious bad guy or innocent (bunny, squirrel) good guy



Pummel horses to get high or across a boundary



tic tac toe spider web

each step you make marks X each step spider makes is O. Three in a row either kills or lets you live or tiles fall away

-or their are the tic tac toe brothers who jump around and three O's or X's cause land to fall away



Diff angles

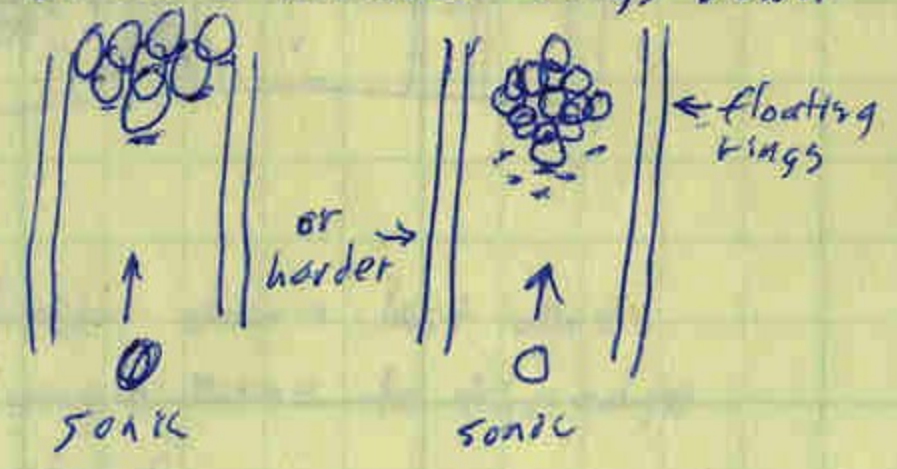
- Jealous rabbit character - rabbit races
- rabbit is jealous of sonic's speed and keeps challenging him to a race, but rabbit knows he's slightly slower so he sets up all sorts of traps in sonic's path (trip cords, launching platforms, etc) so sonic must run at full speed while jumping over objects or jumping onto helpful objects. Rabbit runs a clear course or one that takes advantage of a rabbit's abilities. Rabbit gives annoying laugh when you lose. (Donkey or Alan type laugh)



Screen off center to show rabbit (if he's beating you, or way behind) and so that ~~you~~ you can't see all of your track

- perhaps for Bonus rounds on each or certain worlds
- if rabbit gets far enough ahead, he'll get cocky and sit against a tree watching you fumble and falter (Forces you to get used to terrain or traps)

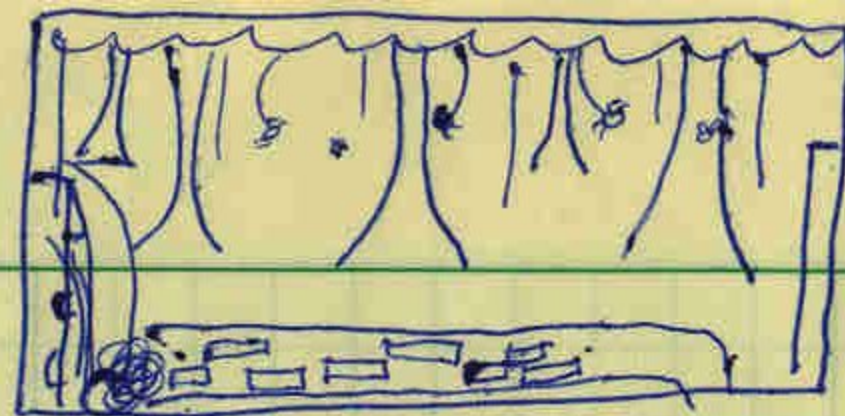
- Mark's idea - sonic bowling
- sonic's on a bowling alley and spins to bowl down pins or giant rings. works like a bowling game. perhaps you could make it harder by arranging pins/rings in different 3D patterns so you have to jump or move to knock rings down



- possible enemies - high above or in the air so ledge and water will affect if you get near
- enemies or some animal who is built starting out in some wild place and looks like you to know you know you

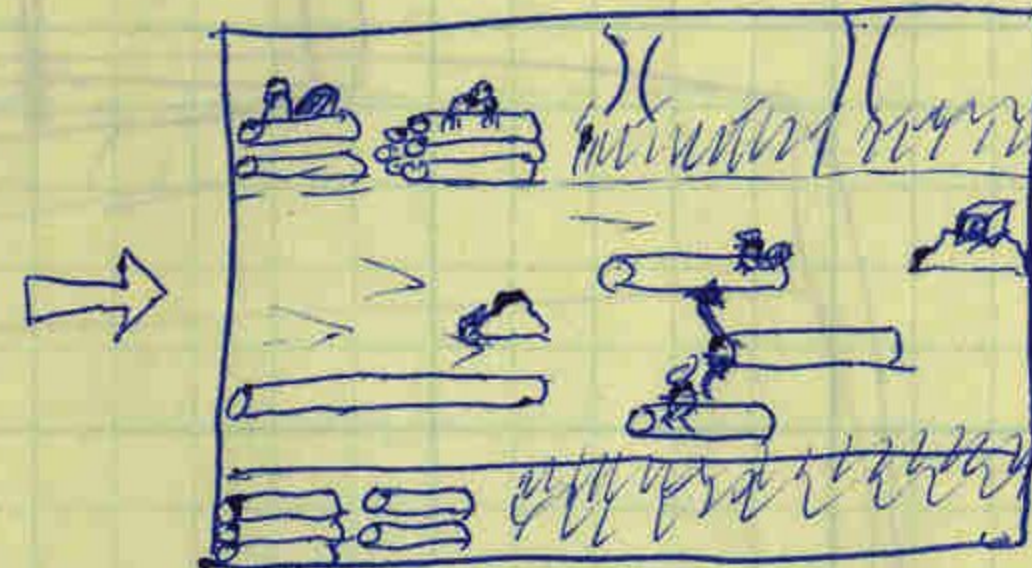
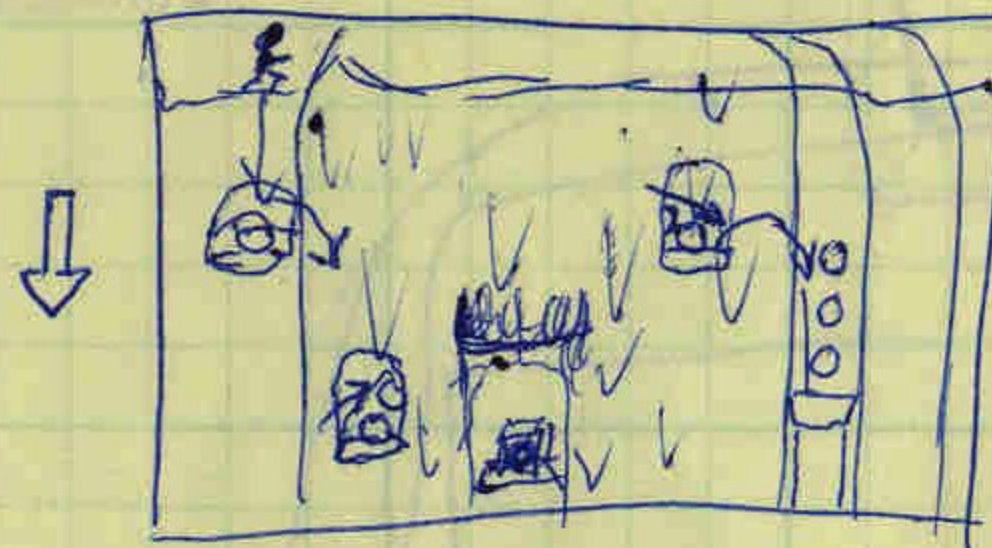
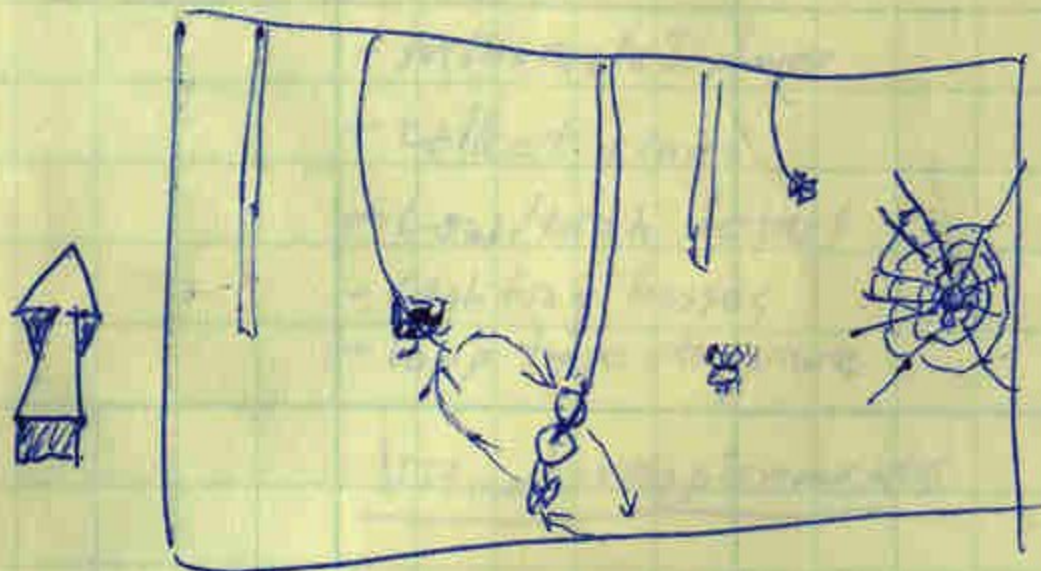
Jungle World

Act 1 - Jungle Zone



Act 2 - Waterfall zone

Act 3 - Logjammer zone



- start by jumping from cliff onto vine (can't see any ground)
- you swing vine by pushing in ANY direction
- hold down a button and up and down moves you up and down vine
- must swing from vine to vine til you get to a very long vine that takes you to next zone

- enemies - spiders create their own 'vines' and start swinging after you.
- flies or mosquitos try to leech onto you causing you to slide down vines or lose rings til you swat them (spindash)
- ~~the~~ spider webs catch you if you fall or swing into them, must wrestle free and defeat spider, then you can jump on spider web and will always spring you back onto a vine
- falling in center of screen kills you, falling near side will make you fall in web or onto the side of giant tree

- ~~fall~~ climbs down last vine from Jungle zone to the edge of a cliff
- mostly sonic is dodging the rushing water fall so he won't be dashed into the sharp rocks at the bottom
- he slowly jumps from ledge to ledge and in and out of holes behind waterfall
- possible enemies - birds whose nests are on ledges and mother will attack if you get near
- squirrels or some animal who is busy storing food in caves will throw food (nuts) at you to knock you from ledge

- sonic gets to the bottom of waterfall and finds a ~~great~~ large beaver colony sending logs down the river, sonic must get down river by riding logs and avoiding beavers
- logs move in and out of screen as they move ~~right~~ downstream
- ~~water~~ beneath water are rings or bonus items, water is shallow
- sonic must avoid being crushed by logs
- enemies - beavers ~~can~~ can swim in river to logs and spin the logs to knock sonic off or they can get on top and swat sonic with their tail. If sonic is ~~in~~ in water then beavers can't bite sonic and they will swim faster
- swappy or treacherous lands will keep sonic from jumping off river until the right opening appears at the end of river

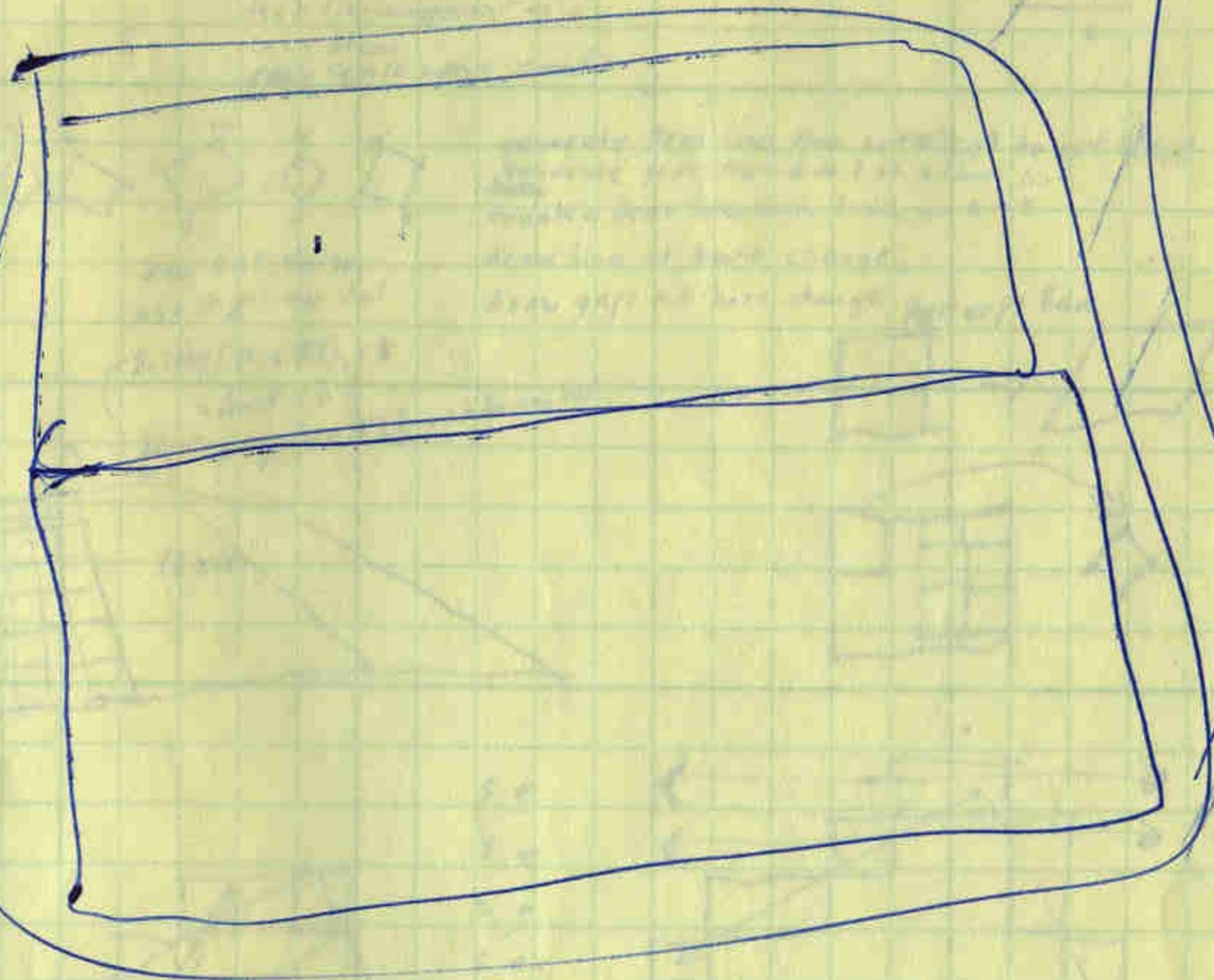
Concepts by Don Goddard
Contacted by hxg and dx

Basic requirements

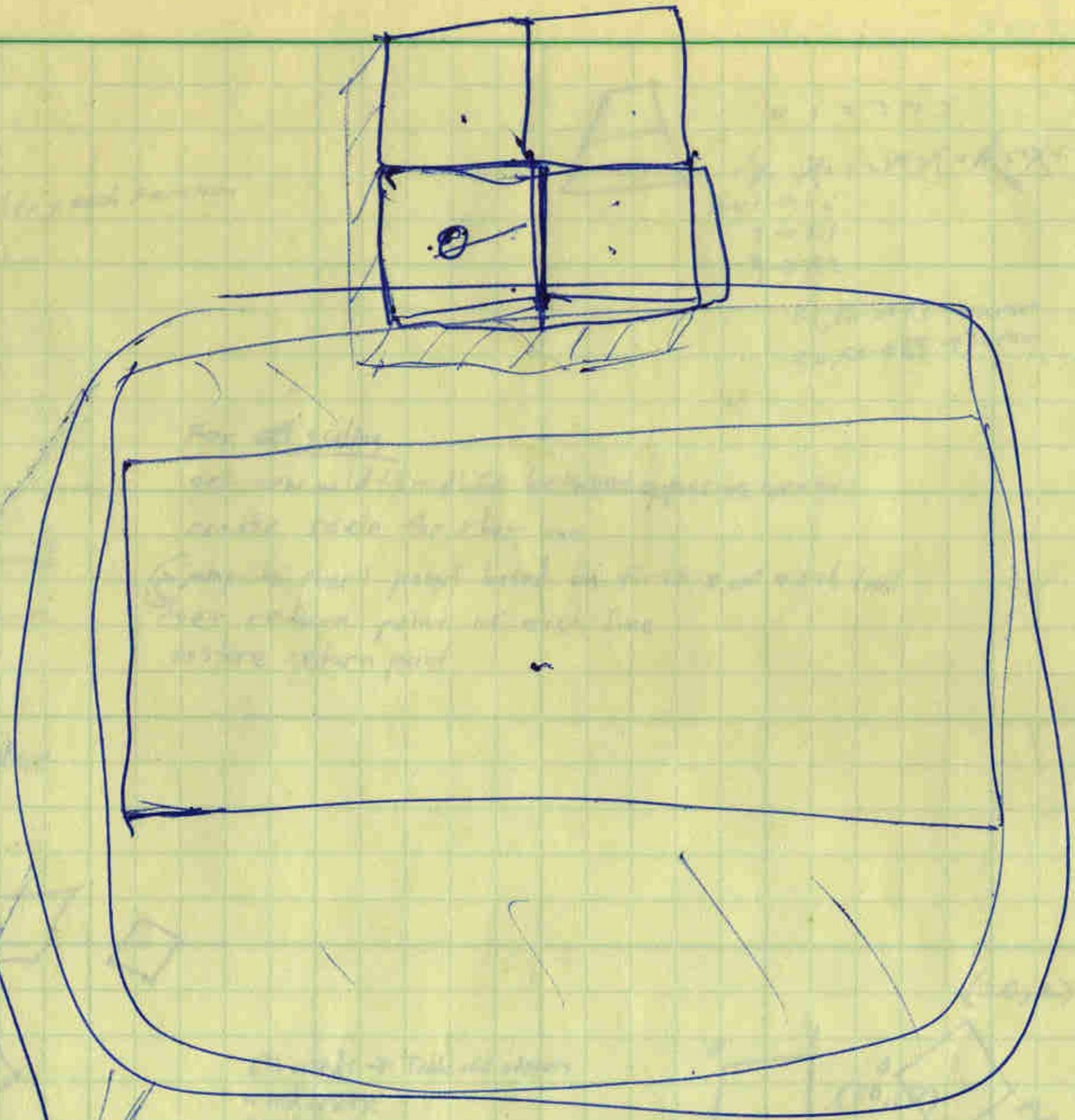
- Sonic 3-D
- 8-bit
- 75% screen
- minimum 1 player
- collect rings
- low/high detail
- fighting bosses
- loop type structure
- triggered regulated/moving 3-D
- breakaway/fall away 3-D
- one direction camera

Visual requirements

Concepts by Don Goddard
Contacted by hxc and dix

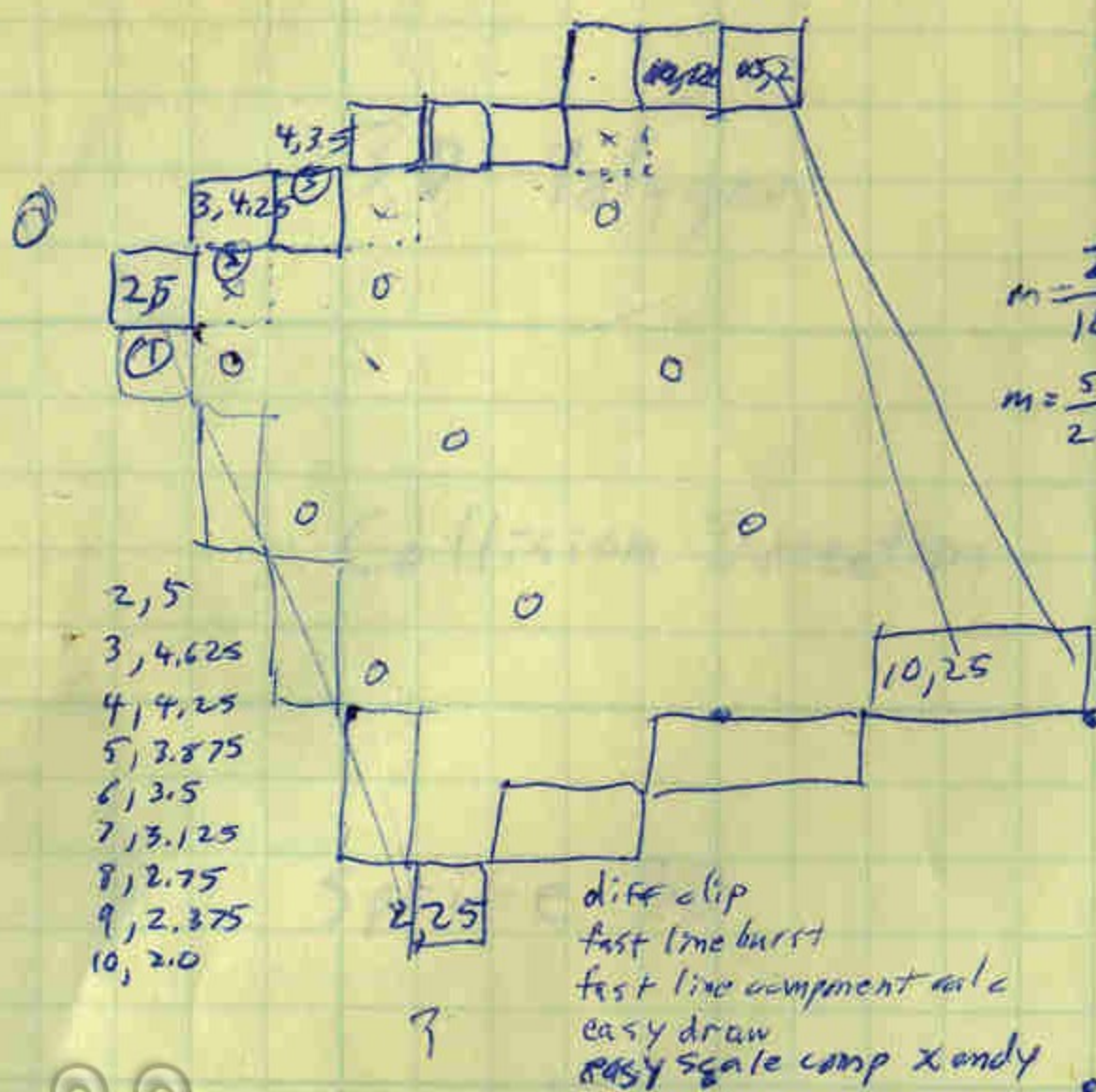
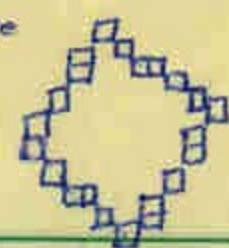


- rotational sprite



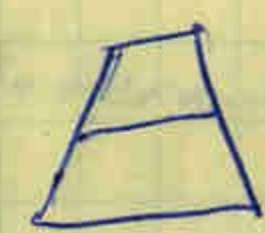
Low/HIGH DETAIL

X = Fill order = 27 (4x3)
 O = Gap = 9



$$m = \frac{2-5}{10-2} = \frac{-3}{8} = -.375 \text{ add to y each iteration}$$

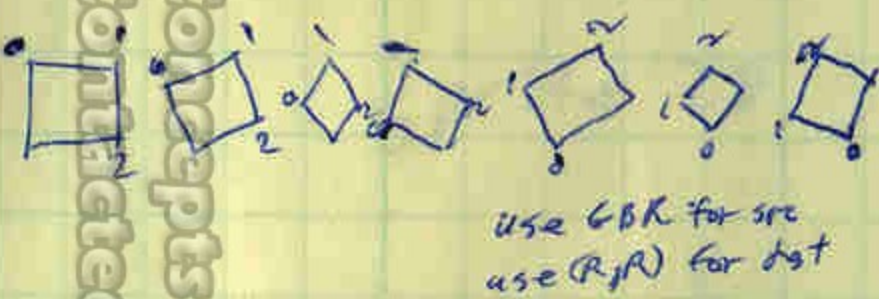
$$m = \frac{5-3}{2-10} = \frac{2}{-8} = -.25$$



0 1 2 3 4 5
 0x0 1x1 2x2 3x3 4x4 5x5
 start → R0
 3 → R1
 4 → R2
 R1, R0 → R3 → screen
 R2, R0 → R3 → screen

For scaling
 get new width - diff between opposite corners
 create scale for that line
 jump to start point based on first x of each line
 set return point of each line
 restore return point

Concepts by Don Stoddard
 Contacted by hng and ax

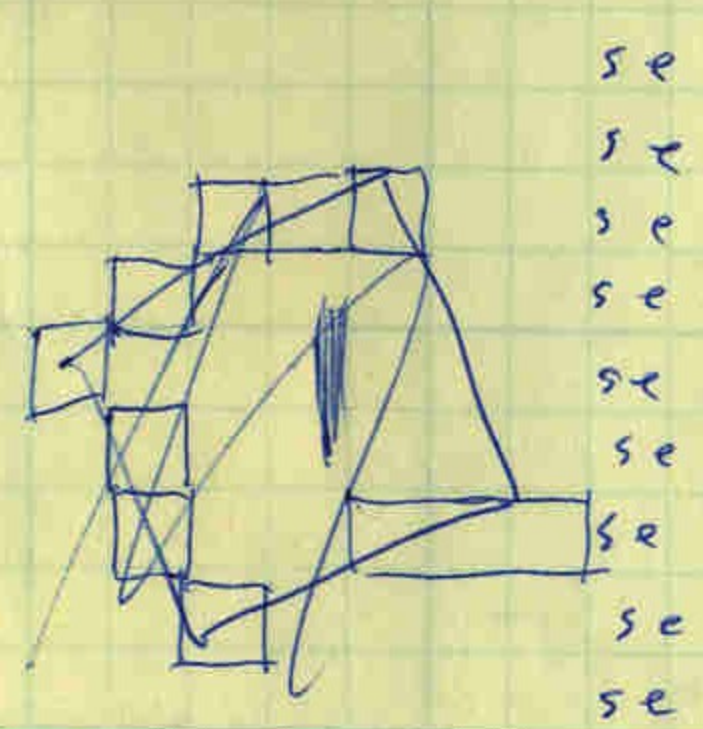
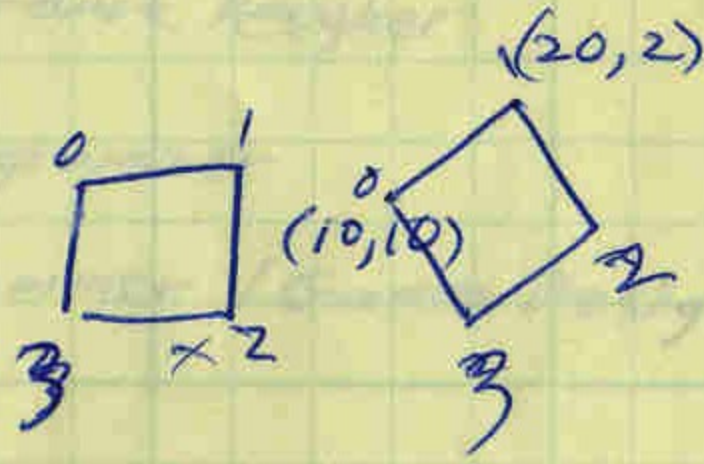


use GBR for src
 use R1/R for dst
 Build(#,GBR),r0
 = dest,r0
 r1,r2 = r0,r2 r2 = frame buf

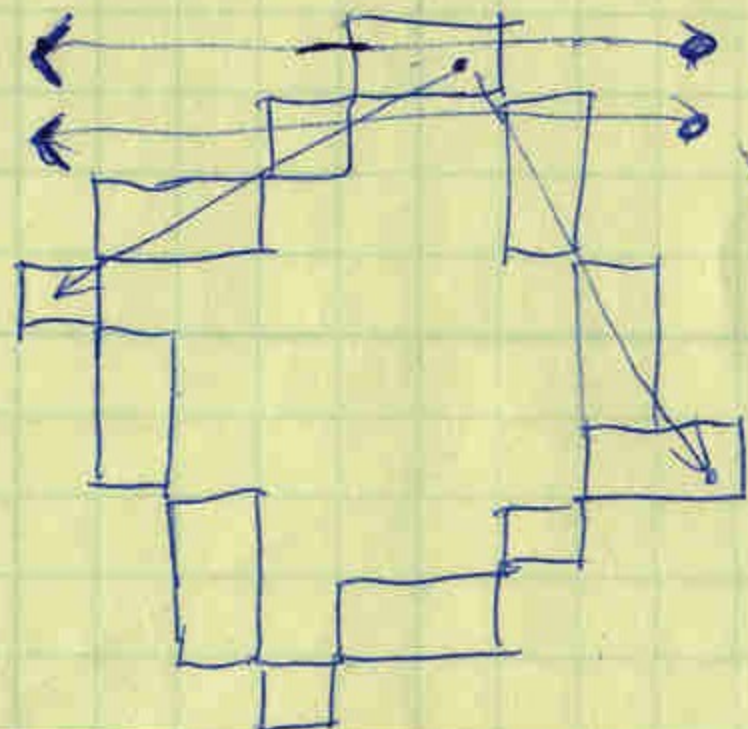
generate Bres line from rot0(x,y) to rot1(x,y)
 generate gaps from 0 to 1 as other line
 track a Bres line from 1→2 or 0→3
 draw line at vert change,
 draw gaps at horz change



64 words → Table off offsets
 word next x
 word next y



s e
 s e
 s e
 s e
 s e
 s e
 s e
 s e
 s e
 s e



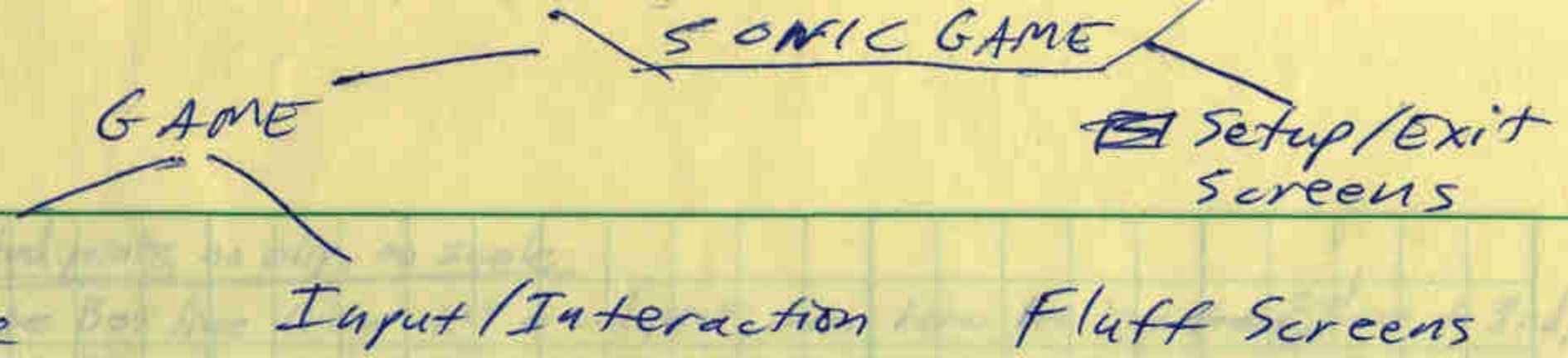
get max width
 build uv line
 so from top to opp corners
 when bend use last x's to draw line
 first x is start, 2nd is stop (or s)

90 H
 90 270
 H
 q
 p
 v
 X
 H
 q
 p
 v
 X
 H
 q
 p
 v
 X

$$\frac{10}{20} \quad \frac{10}{2}$$

$$x=10 \quad y=10$$

$$m = \frac{20-10}{2-10} = \frac{10}{-8} = -1.25$$



3D Polygon

Rendering Phase
clear

Input/Interaction
Genesis Input

Fluff Screens

Collision Detection

Sorting
Sprites/3D Polygons

Sprite 3D

Overlays
Genesis

Sprite 2D

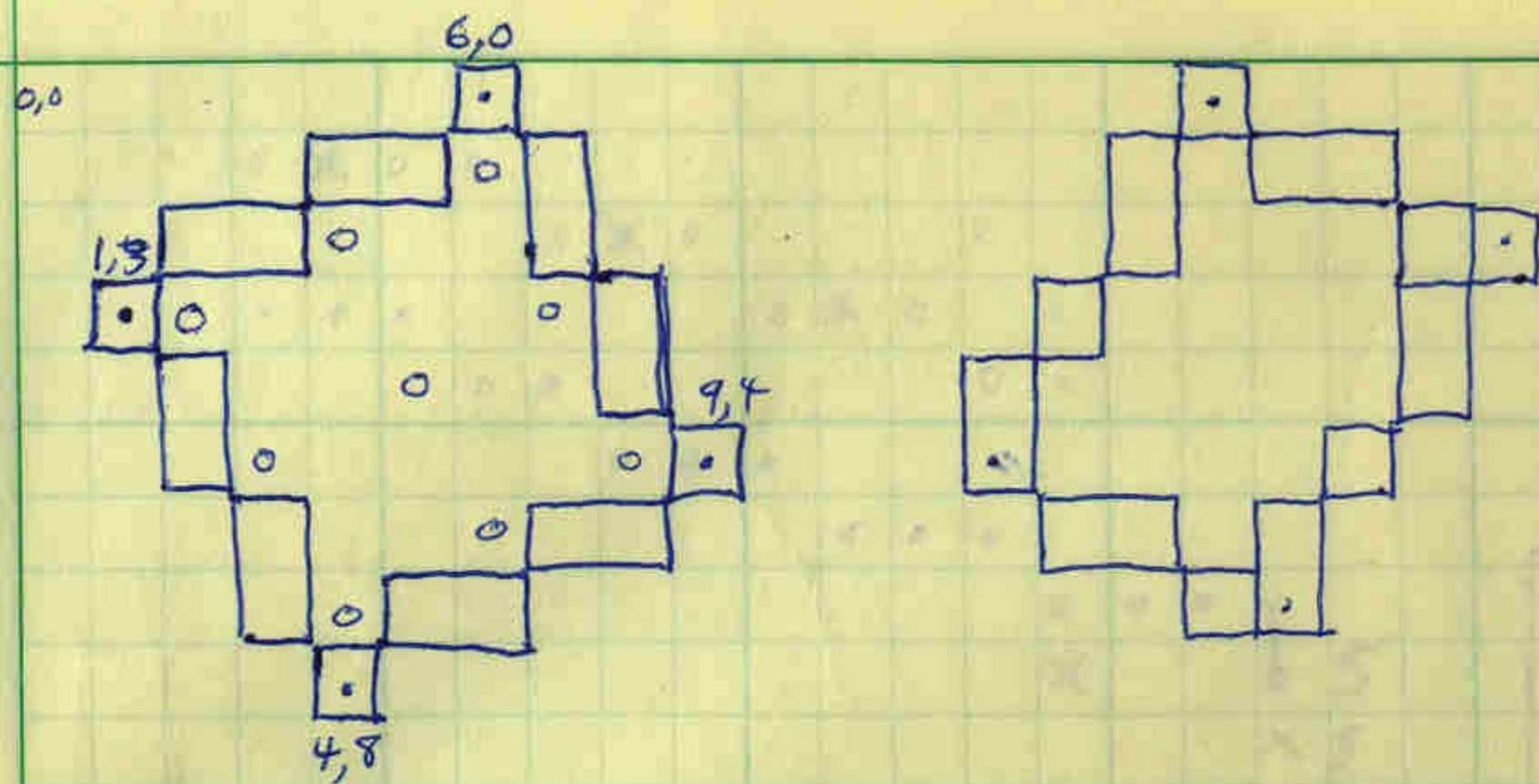
Input

- Chris - Director/Game Designer
- Ofer - Technical Programmer
- Me - Lead Programmer / Game Design
- Mark } Tools Programmers
- Robert }
- Adrian - Reference
- Mike Wallis - Producer

Concepts by Don Goddard
 Contacted by hxxg and drx

< 45°

> 45°



- line burst algorithm
- clipping
- flat X,Y rotation about Z-axis
- scalable

Step 1 - rotate points of orig image using base as origin and scale, later can use alternate basepoint for rotation about an outside object

Step 2 - using scaled, rotated points, check for clipping. If no clip use faster no clipping rotations.



Rotated points, no clip, no scale

Create Bes line from 1st to 2nd point, then draw Bes line from 2nd point to 3rd point

@create_burst_line

```

mov.l #pt2y, r1
mov.l pt1y, r3
mov.l #pt2x, r2
mov.l pt1x, r4
sub.l r3, r1
sub.l r4, r2
mov.l r2, @divbot
mov.l r1, @divtop

```

$$m = \frac{pt.2y - pt.1y}{pt.2x - pt.1x} = \text{amt. to add to } y \text{ each iteration}$$

x always adds 1
loop til x = pt1x

```

mov.l source-scale-x, r5 ;(src index)
mov.l divtop, r7
mov.l r5, r6
mov.l #xxx, r12
mov.l #C500, r14
mov.l #lineburst, r13

```

@next_pixel

```

mov.l #add320, y
mov.l #-1, r7
swap.w r6, r0
and.b #FFF, r0
or.l r14, r0
mov.w r0, #3 @-r13
add.l r7, r1

```

```
add #5, r6
```

```

cmp.l r0, r4
bf @next_pixel
add #5, r6

```

```

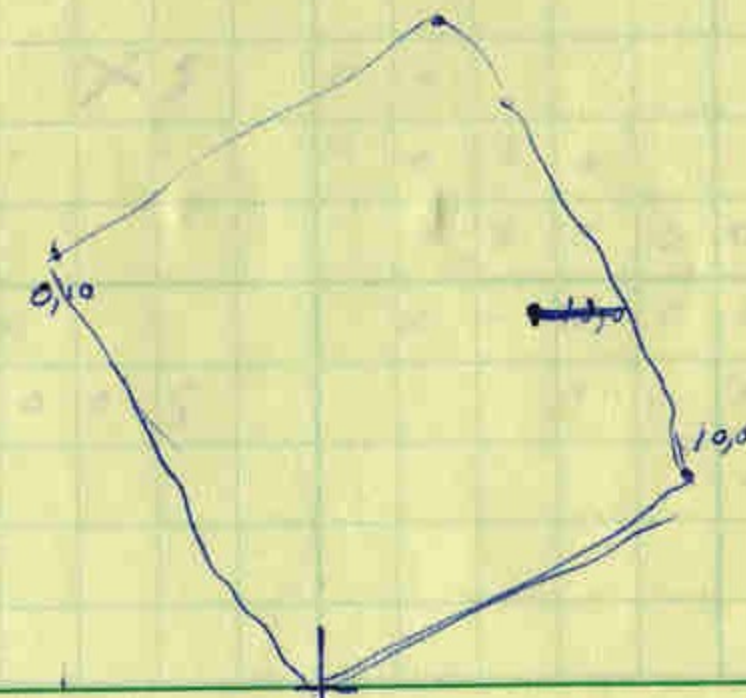
mov.w r0, r4
add #1, r4
add.v r7, r8
bf @same-y
mov.w r11, @-r13
sub.l #2, r13
@same-y

```

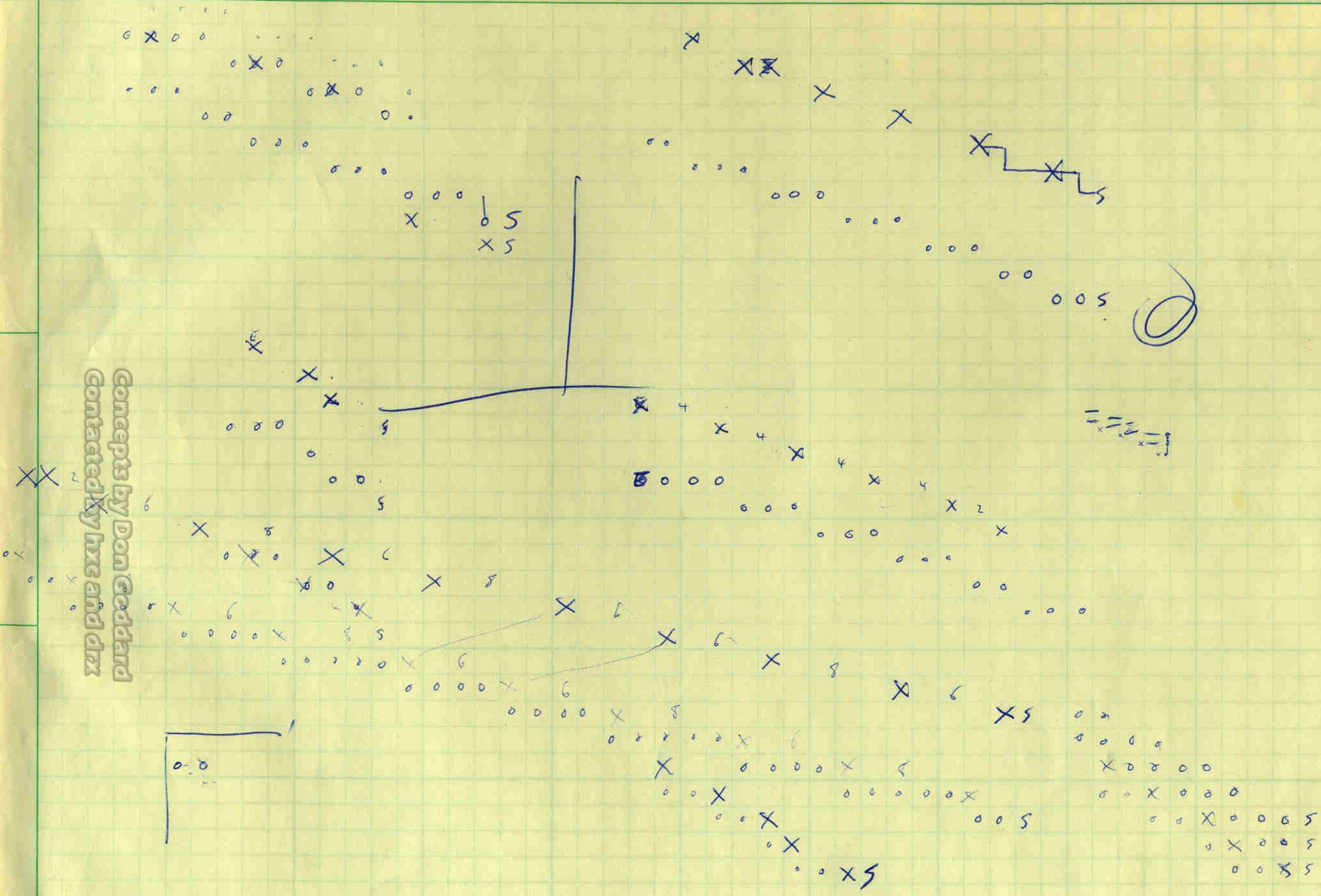
```

cmp.l r6, r4
bf ls @next_pixel
add r5, r6

```



Concepts by Don Geddard
 Contacted by hxg and dxg

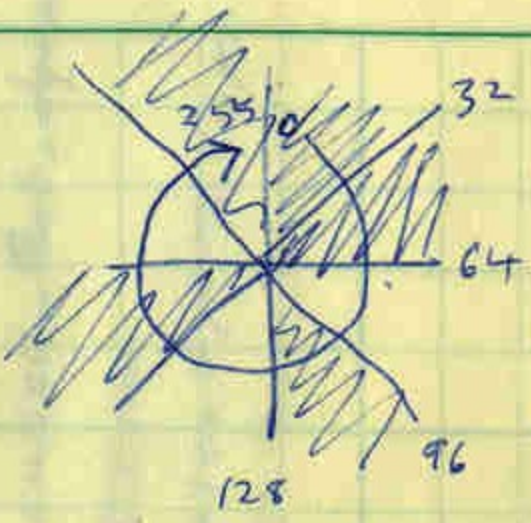


Sega Interactive
 (909) 860 3009
 Mike Terlecki

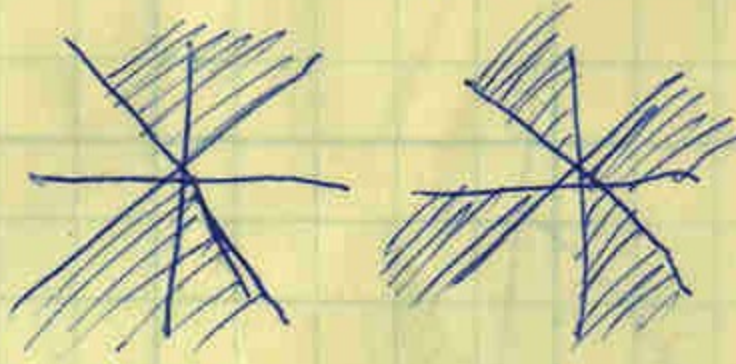
(909) 612 1219

Debug code
 Scaled sprites 2 insts/pixel → 3 cycle
 Rot sprites 3 insts/pixel -
 Yersp sprites 4 insts/6-1=5

0 1 2 3 3 4 4 5 5-0 5-1 5-1
 5 4 4 . . .

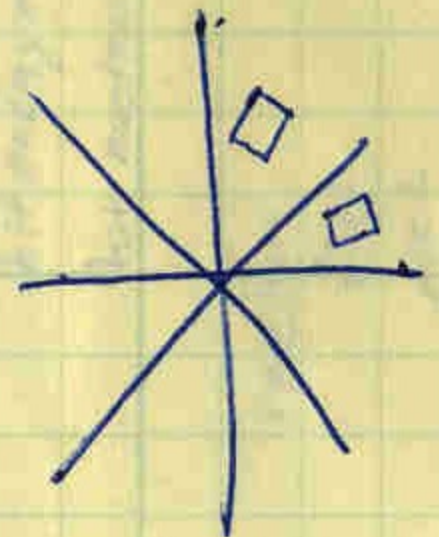


every other 32 angles (45°) must switch
 drawing routines, bit 6 (32) on uses alternate routine



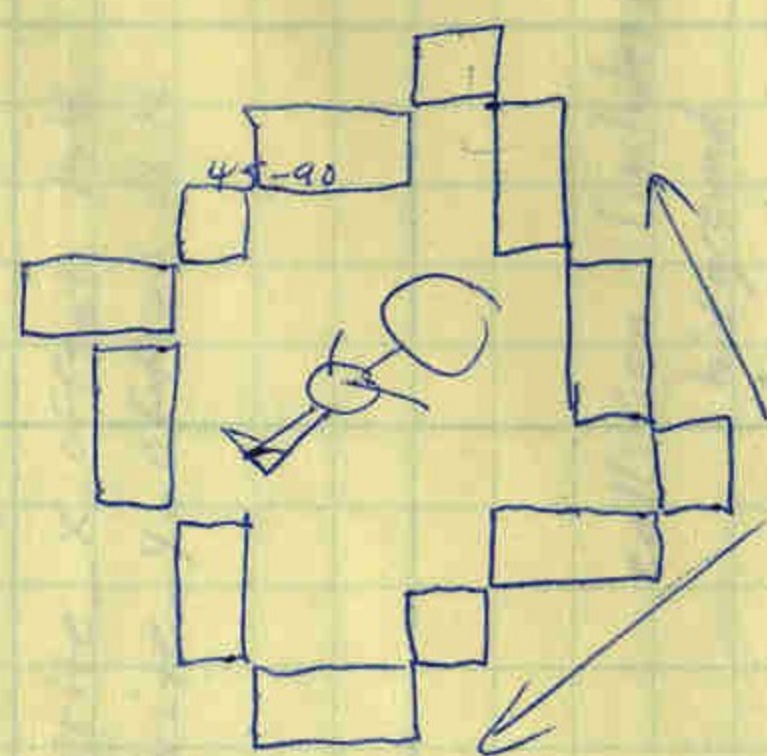
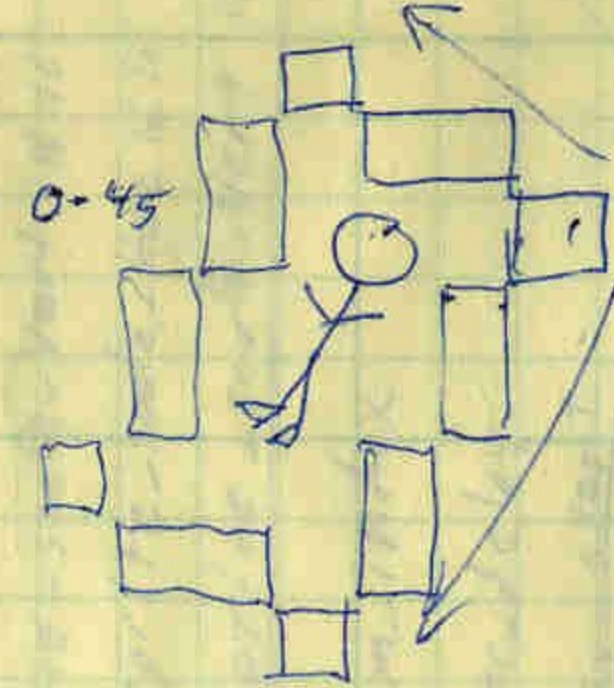
81813
 992-0888

0 5 5
 0 0 X 5 5
 0 X 0 0
 0 0 X 0
 0 0 0 0
 0 0

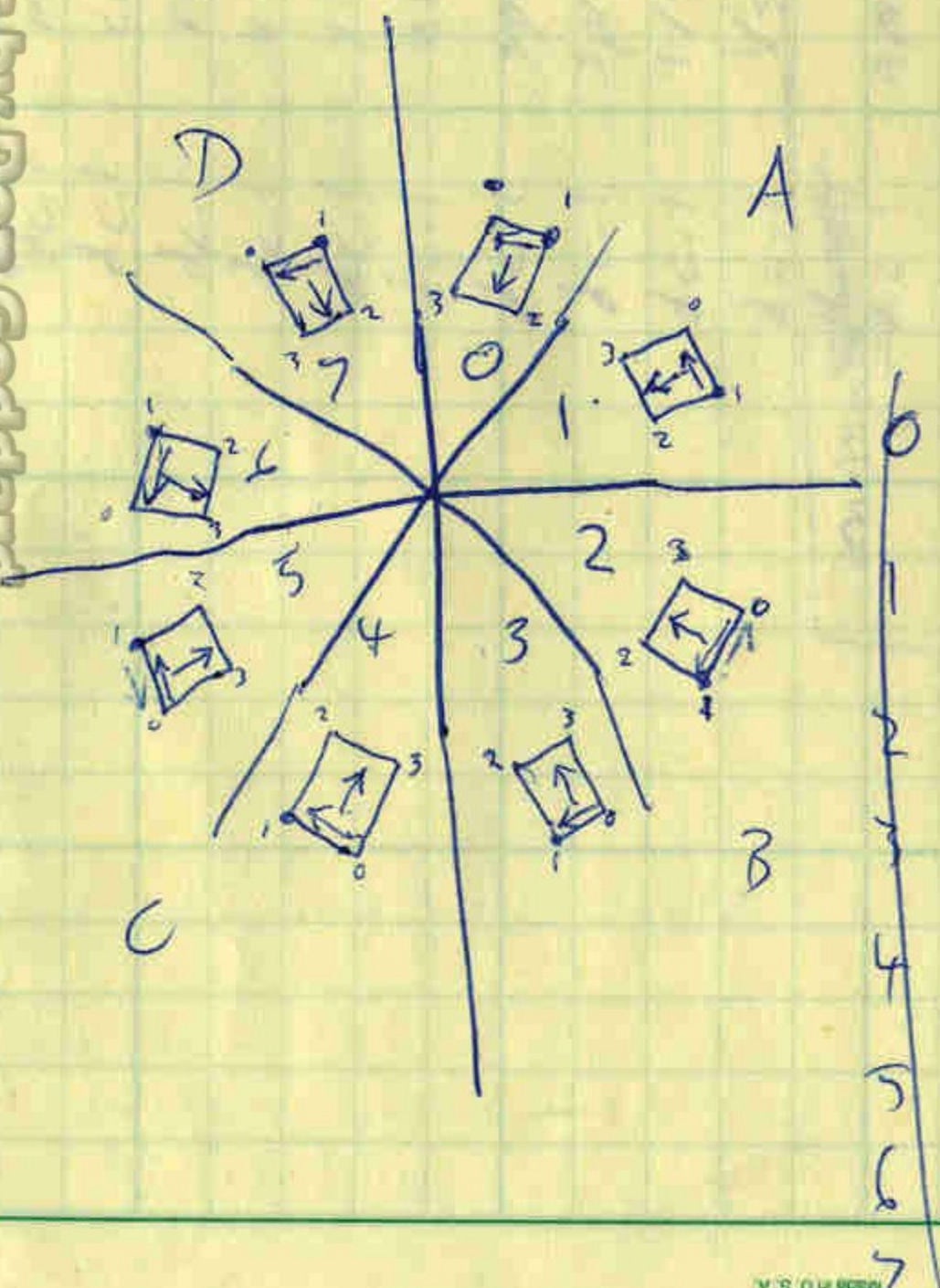


Calcs

Draws



concepts by Don Goddard
 contacted by hxs and dx



build draw
 0 sub r2,r3 add #-2,r11
 1 dest=pt+2x,y
 2
 3 add r2,r3 add #-2,r11
 dest=pt+x,y

build draw
 4 use sub r2,r3 add #-2,r11
 5 dest=pt+x,y
 6
 7 add r2,r3 add #-2,r11
 dest=pt+2x,y

clip accurate

do 1,2,6,5 quadrants

clip minimally all quads

quadrant	top clip	right	bottom	left
A { 0	0y	1x	2y	3x
1	0	1	2	3
B { 2	3y	0x	1y	2x
3	3	0	1	2
C { 4	2y	3x	0y	1x
5	2	3	0	1
D { 6	1y	2x	3y	0x
7	1	2	3	0

rot amount >> 6 = 0-3 (A-D)

larry_russell_cat_dynamix@
~~dynamix.com~~
 mailgate.dynamix.com

ROM

Sprites

RAM

contacted by larry russell
 prepared by larry russell

Sprite name	
dc.1	anim_struct
dc.1	movement_struct
dc.1	collision_struct
dc.1	render_struct
dc.1	x16.16
dc.1	y16.16
dc.1	z16.16
dc.1	sprite_scale_Horz 16.16
dc.1	sprite_scale_Vert 16.16
dc.1	1/sprite_scale_Horz 16.16
dc.1	1/sprite_scale_Vert 16.16
anim_struct	anim_index
dc.1	frame_delay
dc.1	delay_start
dc.1	frame loop back
dc.1	sprite_x_offset 16.16
dc.1	sprite_y_offset 16.16

sound effect info
 sprite triggers

movement (AI)

- track 1 (follow objects pattern)
- track 2 (circle and dive)
- track 3 (~~go str~~ beeline player)
- random
- spastic
- circular
- orbital
- avoid
- waiting
- jumping
- pulsing / throbbing

collision (includes pointer to reaction)

- hit ground
- hit 3D object
- hit player
- hit player missiles
- hit enemy
- hit enemymissles
- hit neutral object

renderer

- scaled_sprite
- rotated ~~sprite~~ scaled_sprite
- polygon_style 1
- polygon_style 2
- polygon_style 3
- mode 7_sprite
- transparent_sprite
- derez_sprite
- shadow_sprite

reaction

- die 1
- die 2
- break in two
- break into many
- bounce off
- stop direction moving
- switch AI

500 SHEETS, FILLER, 5 SQUARE
 50 SHEETS EYE-EASE® 5 SQUARE
 100 SHEETS EYE-EASE® 5 SQUARE
 200 SHEETS EYE-EASE® 5 SQUARE
 100 RECYCLED WHITE 5 SQUARE
 200 RECYCLED WHITE 5 SQUARE
 Made in U.S.A.



alpha, beta, final, when more detailed ready

Neutral non-moving objects

Animating/neutral non-moving

Animating and moving neutral objects

Tree/rock	Waterfall/ tree blowing tree	Birds in background, leaves, dust
RAM { X	X	X
RAM { Y	Y	Y
RAM { Z	Z	Z
ROM { X-off	X-off	X-off
ROM { Y-off	Y-off	Y-off
RAM OR ROM { sprite-scale-h	scale-h	scale
RAM OR ROM { sprite-scale-v	scale-v	"
RAM OR ROM { 1/scale-h	1/scale-h	"
RAM OR ROM { 1/scale-v	1/scale-v	"
RAM OR ROM { no-move	no-move	move-pattern 1
RAM OR ROM { no-collision	no-collision	no-collision
RAM OR ROM { no-anim	anim-loop	anim-loop
RAM OR ROM { scaled-sprite	scaled-sprite	rotated-scaled-sprite

Animating, non moving, collidable

Animating, moving, collidable

unintelligent animating moving collidable

Rings/boxes/store

enemies/bosses/sonic/Norvaga

missiles/rings/objects

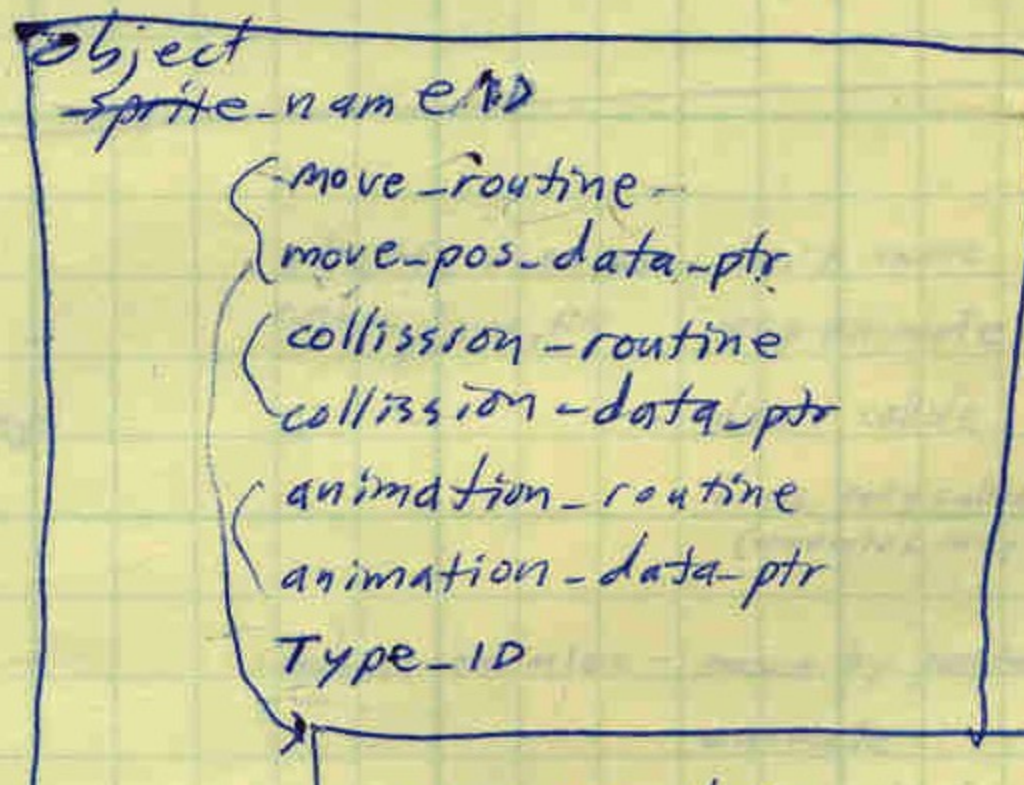
no-move
ring/box/store
~~bullet~~-collision (reaction inside collision routine)
anim-loop
scaled-sprite

moving/waiting pattern
collision-list-routine (reaction included)
anim-sequence
scaled-sprite

moving-loop-pattern
collision-list
anim-sequence
rotated-scaled-sprite

wait for trigger
move in

position-movement-struct-~~ptr~~
scale-struct-~~ptr~~
move-routine-~~ptr~~
collision-data-struct-~~ptr~~
collision-routine-~~ptr~~
animation-~~ptr~~
scaled-sprite-routine-~~ptr~~



process-sprites
jsr move-routine
mov move-pos-data, r0
jsr collision
mov
jsr animation

animation_data

normal animation {
scale-struct-~~ptr~~ } scale-struct-~~ptr~~
render-routine } render-routine
anim_sprite_list-~~ptr~~ } anim_sprite
anim-sequence-~~flip-flags~~ } x-offset
current-anim-frame } y-offset
anim-fps

move-pos-data

X	X	X	X
Y	Y	Y	Y
Z	Z	Z	Z
X2	X'	radiusX	ptr
Y2	Y'	radiusZ	chase_obj_ptr
Z2	Z'	speed	

collision-data

bounding box {
Xmin } X
Ymin } Y
Zmin } Z
Xmax } radius
Ymax } sphere
Zmax } plane (wall)

anim_sprite_list

dw # of frames
dw goto frame # when done
dl sprite_addr + flip-flags
...

scale-struct

scale-h
scale-v
1/scale-h
1/scale-v

obj-34 is a spinning ring

Object_list

Obj_#
1
2
3
:
:
294

Obj_34

* rts, nop
→ ring-34-pos-data-ptr
* rts, nop
→ ring-34-collision-data-ptr
* looped-anim-routine
→ ring-34-anim-data-ptr

Ring ID

ring-34-pos-data

X 16.16
Y 16.16
Z 16.16

ring-34-collision-data

X
Y
Z
X' or W
Y' H
Z' D

ring-34-anim-data

→ standard-ring-scale-ptr
* scaled-sprite-routine
→ spinning-ring-frames-ptr
No-flip-sequence (0)
2
FPS 10 (6)

Good Points

Extremely Flexible
Space saving-reusability, pointers to ROM
Fast movement and animation process
Good SH2 access

Bad Points

Awkward collision detection and sorting, slow?

Fluff anims - may move
leaves, dust, waterfall animates
deaths, explosions
no collision
no collision into

Trees, rocks, scenery - don't move
may animate
don't collide into things
may be collided into

Rings - normally don't move
may animate
don't collide into things
sonic may collide into

Moving Rings - same as above but
move in pattern

Runaway Rings - bounce away
- animate
- collide into 3-D world
- sonic may collide into

static enemies - don't move
spikes, lava, fire may animate
don't collide
sonic may collide into
(enemies may?)

mobile enemies - move by pattern, or track sonic
animate
may collide into world
sonic may collide into

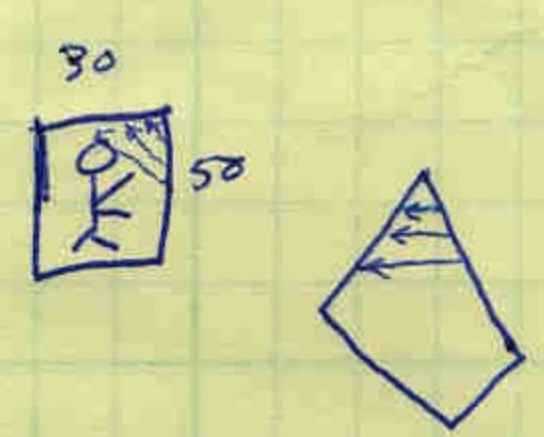
bosses - complex movement by pattern, track, damage, time
animates
collides into world and sonic
sonic collides into

13-782
42-381
42-382
42-383
42-384
42-385
42-386
42-387
42-388
42-389
42-390
42-391
42-392
42-393
42-394
42-395
42-396
42-397
42-398
42-399
42-400
500 SHEETS, FILLER, 5 SQUARE
50 SHEETS EYE-EASE, 5 SQUARE
100 SHEETS EYE-EASE, 5 SQUARE
200 SHEETS EYE-EASE, 5 SQUARE
100 RECYCLED WHITE, 5 SQUARE
200 RECYCLED WHITE, 5 SQUARE
Made in U.S.A.



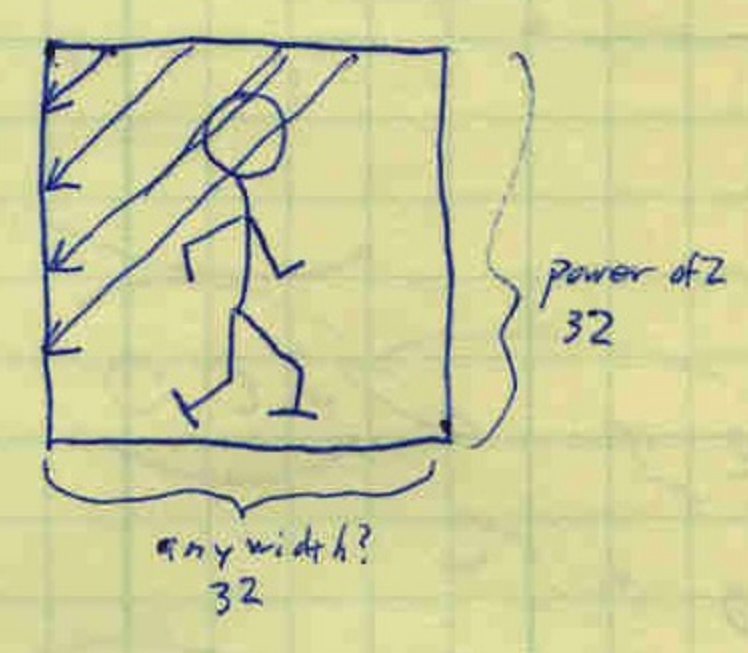
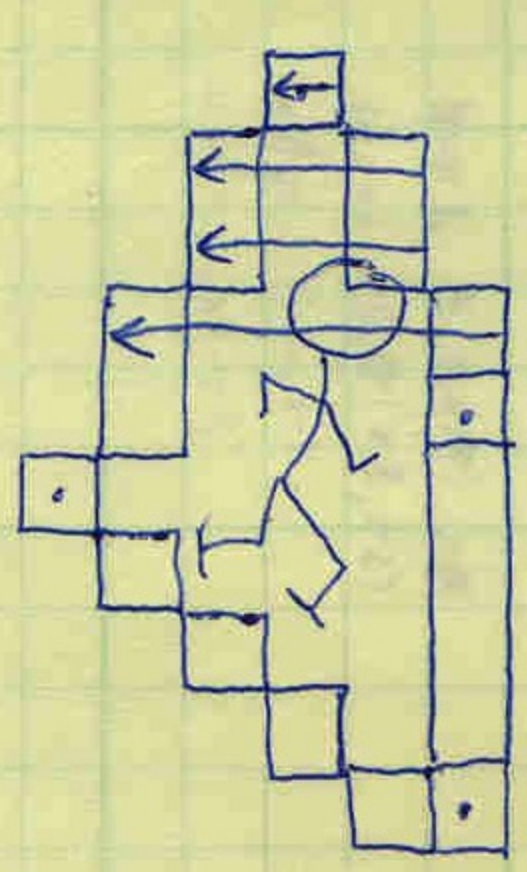
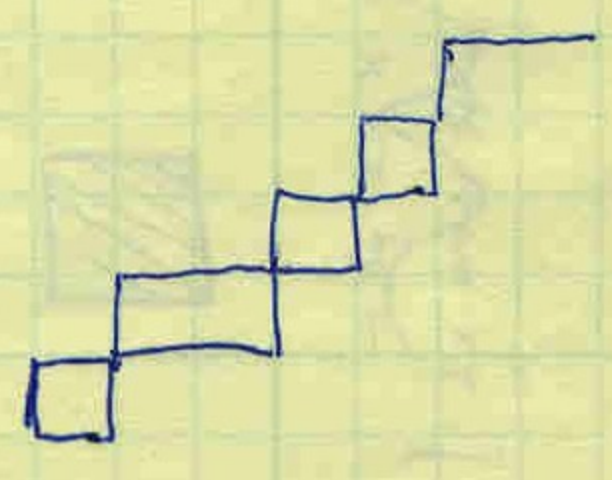
Concepts by Don Goddard
 Contacted by hxc and drx

5/24/12
 more work to be done
 about a game direction
 about a game direction



```

r3 = temp
r1 is source addr texture
r2  Dest screen
r0  index int
r4  index frac
r5  step step int
r6  step frac
addc r6 r6, r4
addc r5, r0
mov.l @r1, r3
mov.l r3, @-r2
  
```



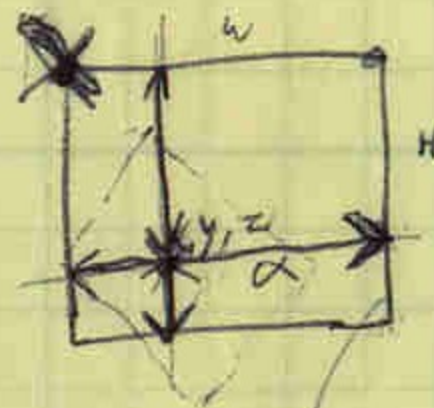
13-782
 42-381
 42-382
 42-383
 42-384
 42-385
 42-386
 42-387
 42-388
 42-389
 42-390
 42-391
 42-392
 42-393
 42-394
 42-395
 42-396
 42-397
 42-398
 42-399
 42-400
 42-401
 42-402
 42-403
 42-404
 42-405
 42-406
 42-407
 42-408
 42-409
 42-410
 42-411
 42-412
 42-413
 42-414
 42-415
 42-416
 42-417
 42-418
 42-419
 42-420
 42-421
 42-422
 42-423
 42-424
 42-425
 42-426
 42-427
 42-428
 42-429
 42-430
 42-431
 42-432
 42-433
 42-434
 42-435
 42-436
 42-437
 42-438
 42-439
 42-440
 42-441
 42-442
 42-443
 42-444
 42-445
 42-446
 42-447
 42-448
 42-449
 42-450
 42-451
 42-452
 42-453
 42-454
 42-455
 42-456
 42-457
 42-458
 42-459
 42-460
 42-461
 42-462
 42-463
 42-464
 42-465
 42-466
 42-467
 42-468
 42-469
 42-470
 42-471
 42-472
 42-473
 42-474
 42-475
 42-476
 42-477
 42-478
 42-479
 42-480
 42-481
 42-482
 42-483
 42-484
 42-485
 42-486
 42-487
 42-488
 42-489
 42-490
 42-491
 42-492
 42-493
 42-494
 42-495
 42-496
 42-497
 42-498
 42-499
 42-500



Made in U.S.A.

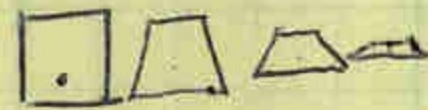
static
 move_circle on x,y,z radius speed
 point to point direction speed
 point to point curve

RANdom
 Fixed



$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} =$$

$$\begin{pmatrix} \alpha \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha a \\ \alpha d \\ \alpha g \end{pmatrix} \propto \begin{pmatrix} a \\ d \\ g \end{pmatrix}$$



GBR H=src
 r2=dist

mov.b @(#,GBR), r0
 mov.b r0, @(r3, r2)
 addc. frac, r4
 addc. int, r3

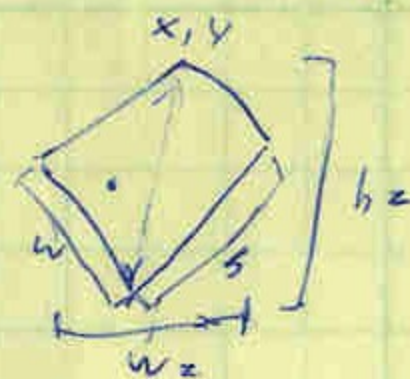


get x,y,z
 determine x,y,z for all four pts

get all four x,y pts
 add same z to all
 rotate x,y,z about axis x,y,z

~~get z~~
 get z
 rotate x,y
 determine pts from x,y
 rotate each x,y,z by z

translate
 rot x
 rot y
 rot z
 translate
 $x = x/2$
 $y = y/2$ for all pts



Concepts by Don Goddard
 Contacted by hxg and dx