Variables

Variable problems can be solved by proof or by value substitution

Value substitution is possible, proof may not be

Given that value substitution is possible, it is the route we will choose

For value substitution to work, we must assign values to our variables

We can do this globally or locally

For local assignments, we might simply have a block that assigns a value to a variable in a radius

This poses a problem however, because variables may have multiple values assigned to them in overlapping areas

Values may also be assigned a single value initially, but as a value travels to a new definition the value may become ambiguous as the definition begins to transmit

This problem could be circumvented through constrained level design, but it leaves a large hole in the engine

Local assignment need not work through radius

variable takes value from closest (if 2 closest no value)

variable takes value through wiring (if 2 wired values on value)

Local assignment

Lasers don’t fire if in range of 2 emitters

For global assignment we need a single authority for a variable's value

That authority might be (ranked from best to worst):

**Shrines (beams)**

The player activates a particular shrine by grabbing it, thereby selecting a single source for variable value

Players won’t understand why they have to do this

**Shrines (input)**

The player can directly set the value of a variable at any shrine

The value is then set at all shrines and emitters

Would require a conversion function from infix to reverse Polish

**An inventory interaction (tablet and stylus)**

An inventory system might be frustrating (for me and the player)

If tokens are found to use with tablet, the player might not know if he has the correct ones

If a stylus is used, things might be easier

Pull out the stylus and type a value

No algebra used, just values

The contents of the stylus remain

**Pedestal and tokens**

Only the tokens present at the pedestal may be used

New and complicated system

Fixed and moveable tiles

**Summons**

Summon the spirit of Xar, Zed, Yah into a beam and they will broadcast that beam

Summon in front of the player and push?

Summon in front of the player and control?

Physical or ethereal (idols or familiars)?

Place the idol on the ground or hold it in front of the player?

Familiar absorbs laser or simply stands in it?

Press x or hit the button to summon the idol in front of the player

**A companion**

x, y, z, w fairies, p, q, r, s sprites

How do they get their values?

**An item**

Wear the x ring and absorb a laser

Must be directional or multiple values might be accepted

Depends on the player interacting with lasers

**A spell cast**

Tracing out a shape in the ground?

**A single emitter on the map (the shrine of x)**

The player blasts a beam into a globally emissive shrine

It would be frustrating to return to a central shrine for every variable reassignment

**The most recent definition by the player**

Different definitions could happen simultaneously

**A tile**

The player puts the summon on an x tile and

Anything that receives a beam must do so from a specific direction to avoid multiple inputs

A beam solution is the easiest from a programming standpoint (no new interface or parsing) and is more thematically coherent

What is the nature of the puzzles?

Use a, b, c to reduce ambiguity with times sign and teach that variables exist independently of graphs

Solving for x is fun

3x-5=x/5

15x-25=x

14x=25

x=25/14

2, 7, 5,

(3x-5)/x=⅕

3-5/x=⅕

3-⅕=5/x

14/5=5/x

x(14/5)=5

x=25/14

Place the above equation into a rigid function built from connected emitters and functions

The player must now define x such that the equation emits true (or false)

Pedestal and tablet both work, tablet is more straightforward

Tablet could have a built in solving helper (red for bad format, gold for good format, gold allows symmetrical permutation)

Solution should be local

If I am trying to get past a gate in the room, the ability to get past should exist in the room

Spells (right-click)

Shield - You can now pass through beams unharmed

Flash - Jump across distance until object encountered

Redirect - Beam an value in a new direction

Pull - Move object toward player at distance

Ride - Instead of dying to beams, you now can jump into a beam and travel to its destination (antiparallel beams don’t work)

Fire - Produce an arbitrary beam

Tether - grab a block at distance that moves with the player

Animate - animate a block at distance

Spirit walk - Leave the arithmagician’s body and move what can be moved

Something involving symmetry/rotation

Companions

Map and teleports

A map overlay that allows the player to teleport would be nice

Gloves - Allow the player to move blocks of arbitrary size

Map - Allows the player to see his progress

Cloak - Allows the player to teleport to shrines

Hat - Allows the player to define variables

Wand - Allows the player to fire a laser

Ring -

Shield -

Boots -

Lantern - Illuminates darkness

Lens - Reveals hidden

Wiring layer (powered effect (switches and doors, unpowered lasers))

Attachment layer (object connectors)

Save camera position to backtracking

set initial position if no camera

How to prevent player from moving into lasers

if performing movement ends up putting player in laser, no movement

the player may move into a laser's previous path, but be protected by a moving block

naive solution

perform movement, fire lasers, see if player hit after simulated move, deny movement

better solution

destroy arithmagician, implement backtracking?

find solution that doesnt involve simulated movement?

create a new character that doesnt cause contact problems?

Global definitions for variables a-z

Stone overlays for both messages and variable definitions?

Fire constants and variables at the start of turn

Implement single pass recursive firing of lasers

String to value type and operands

Operands and value type to string

Value type and operands render

dict of strings for messages, no string array

Fix not being able to drag through connections

-switch to square 2/3 grid and check for movement across borders

Add file selector dialog for saves

Add release selection

Allow panning/selecting with other tools

Implement tool specific cursors/graphics

Add default display blocks for builder tool, show at low alpha when build active?

Separate binding, bracing, roof layers

Flowerbeds, waterways, grass, stone tile, bushes

Multi-cell blocks

Make a function obscuring tileset

Complicated multi parameter functions through opaque roofs

Enter the function to see the meaning of its inputs

Fully opaque connection roofs that hide the operators make black box functions!

Expanding central circle, no safety circle

Circles, central safety circle, ever expanding and more difficult region, battery, energy productions, shields

Selector

Selection is a grid of booleans

All booleans are set to false by default

SelectTypeNone

SelectTypeAdd

SelectTypeRemove

this.\_selectType

Add to selection begins with a left shift down and a left mouse click

Remove from collection begins with a left shift down and a left mouse click

Releasing shift changes the select type to none

Left mouse up with SelectTypeAdd adds to selection

Right mouse up with SelectTypeRemove removes from selection

Selection begins with a left-shift + left mouse click

Releasing shift ends the selection without adding cells

Cut out inner and outer perimeter at source

Copy exactly relative to control

Cut out outer at destination

Paste exactly relative to control

Orthos the arithmagician

Numerian blocks

Temple of Bool

Numerian blocks

empty (empty tablet, no ports)

tablet (tablet, no ports)

integer

rational

TF

arithmetic?

Boolean?

conduit (empty tablet, linear ports)? glass

reflector (empty tablet, 1 in port, 1 out port at 90 degree angle) mirror

distributor (empty tablet, one in port, multiple out ports)

combinator (empty tablet, multiple in ports, single out port)

mixer (empty tablet, 2 in ports, 2 out ports)

arithmetic

add (2 in ports, 1 out port)

subtract (2 in ports, 1 out port)

multiply (2 in ports, 1 out port)

divide (2 in ports, 1 out port)

negate (1 in port, 1 out port)?

reciprocal (1 in port, 1 out port)?

Numerian blocks

value overlays

integer: -99 ... 99 = 199

numerator: -99 ... 99 = 199

denominator: 1 ... 99 = 99

fraction: width 1, 2, 3 = 3

variable: a ... z = 26

true

false

addition

subtraction

multiplication

division

negation

and

or

not

TypeGate

immobile

no inputs or outputs

integer

rational

boolean

letter

1 image

TypeDefinition

1 input

letter

4 images

TypeValue

no inputs, 1-4 outputs

integer

rational

boolean

letter

2^4 - 1 = 15 images

TypeUnary

1 input, 1-3 outputs

empty (reflector, distributor)

negation

not

4 \* (3 + 3 + 1) = 28 images

TypeBinary

2 inputs, 1-2 outputs

addition

subtraction

multiplication

division

and

or

4 \* 3 \* 3 = 36 images

79 total (awesome)

Block

Immobile

Value tablet

No inputs

No outputs

Destroyed by a beam of its value

Literal

Integer

rational

Boolean

Variable

Alphabetic

No inputs, up to 4 outputs

Definition

Single in port

No out ports

Alphabetic tablet

Defines a variable in a radius

Unary operator

Binary operator

Numerian blocks

Literals

Non-beam

Numeric

Boolean

Beam

Numeric

Boolean

256x256

Each cell has an implicit 'id' of its position (x bits y bits)

for types that have extra data, have a dictionary with a

Face blocks

string

Function

value

'',

'+', '-', '\*', '/', '=',

'and', 'or', 'not'

'1', ..., '999'

'1/2', ..., '998/999'

'true', 'false'

'a', ..., 'z'

Function

no 3 inputs

no 4 inputs

no 2 1 dot inputs

no 2 2 dot inputs

no 1 2 dot input

no more than 2 inputs

either 1 dot input or 1 dot 2 dot inputs

Cell

type (powerups)

connections

operation

hidden op

value (string)

io

Dialog

Wall

Block

Glass

Function

Strength

Function

Floor

type (teleports simply look for corresponding tiles)

connections

Block

Roof

Objects have ids? Ids look up whatever extra resources that id needs?

tiling style

none

vertical

horizontal

continuous

arbitrary connections

Don't worry about type, only empty/non-empty

Any physical object can be bound to any other physical object

If objects are the same type, use tileset

If not, use binding

'substrate' tilesets - floor, objects?

object types/structs

edit toolbar

functions

definitions

teleports

messages

player

Entity

2d array

cell hierarchy

How to make functions

functions are offering tables

type function

function nothing (repeater), constant (number), add, subtract, multiply, divide, and, or, equals

hidden bit

value1 numerator, T/F,

value2 denominator

connections normal, handled with binding straps

List of function objects

io

ioEast none = 00, input1 = 01, input2 = 10, out = 11

ioSouth none = 00, input1 = 01, input2 = 10, out = 11

ioWest none = 00, input1 = 01, input2 = 10, out = 11

ioNorth none = 00, input1 = 01, input2 = 10, out = 11

00\_00\_00\_00 - IO could be stored in a byte

The arithmagician gets the ability to animate functions in a radius

The definitions send signals to variable functions in a radius

How to make definitions

type = definition

tools

Creator

Connector

Inspector

Selector

Regardless of tool selection, space pans

download templates, build tilesets on startup

Finalize room transitions

World editor

Camera regions

Single world

Single world makes editor harder and transitions easier

From template, create object connections

0b\_1111\_1111

Design block template scheme

Contiguous types joined by self

Distinct types joined by clasps

Create cell generation system

Create text boxes

Create shader applicator

Room editor or world editor?

World editor

Pros

No need for room transitions

Cool musical/graphical transitions

Percentage of floor < 50 a single tileset: Ambience

else music/shaders as a ratio of tiles over 50% / total possible tiles over 50%

Cross world interactions and puzzles

Camera regions?

Where a camera region is defined the camera will be bounded

Cons

Fast transit camera movement might be strange

Camera follows faithfully, transition is immediate

Possible performance hit

Limit rendering to on-screen floor/cells

Design levels to only allow for local effects or trivial global effects

Zoom and pan in world editor

Initial

save file

flattened uint8array for floor types

flattened uint8array for object types

flattened uint8array for floor connections

flattened uint8array for object connections

object containing powers and strength

undo stored locally for session

don't put objects near edge and I don't need to do bounds checking

Create grid with zoom and pan