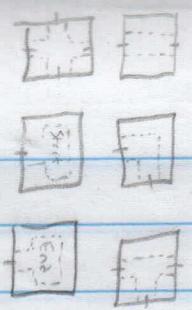


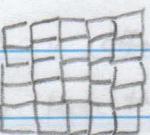
Maze Craze

Tile Examples

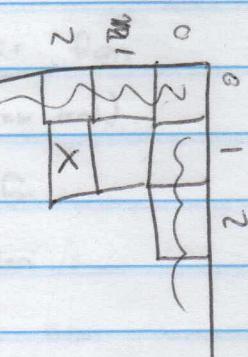


$$W = 1$$

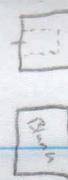
$$h = 2$$



Should fit!



* Go through check / add adjacent check value of sum, render specific tile based on that sum.



Wall



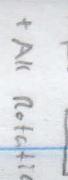
Floor



Door



Key



Wall



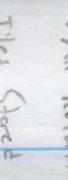
Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



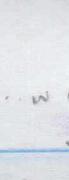
Floor



Door



Key



Wall



Floor



Door



Key



Wall



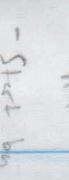
Floor



Door



Key



Wall



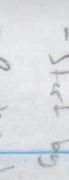
Floor



Door



Key



Wall



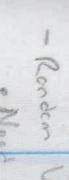
Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



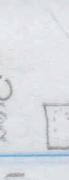
Floor



Door



Key



Wall



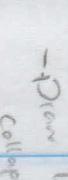
Floor



Door



Key



Wall



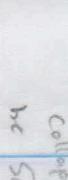
Floor



Door



Key



Wall



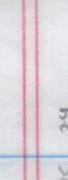
Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door



Key



Wall



Floor



Door

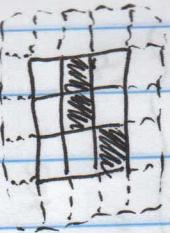


Key

- * All rotations of these tiles start in 0 array
- All start positions at first
- Start by generating start + end - Random wall + End?
- Need to find a way to move check for solution to move
- Draw line to end, then collapse until it's sellable
- * Define these values as enums $\frac{enum}{100}$ when checking to see which sum of $\frac{100}{100}$ is currently being used.
- * Numbers that each combination adds together will be unique. No other combination will result in this number.
- Need large different primes $\sqrt{100} \cdot \sqrt{100} \cdot \sqrt{100} \cdot \sqrt{100} \cdot \sqrt{100}$
- 1 10 100 1000 Doing up whatever

Flooding Algorithm for Maze ✓

Successfully marked collections as needed.



- * Start with a random **Perimeter Square** (will always be part of valid path)

- Add that cell to **Stack** (push)

- Pop top cell off stack until the stack is empty

- Then look at adjacent cells (N, S, E, W)

- If not a wall (path or object) or push onto stack [and not flooded]

- Set flag to **Flooded**

- Do until no more stacks

- Iterate through maze entries.

- If a path is found with no flooded flag, turn into a wall

To Do for Maze Craze

- 1.) Get game to sit devices screen
- 2.) Add controls to pan camera for testing
- 3.) Update Render to use sprites

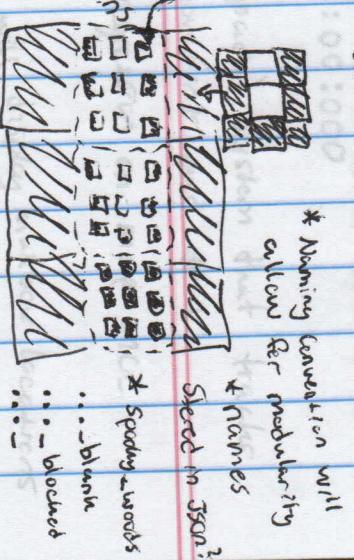
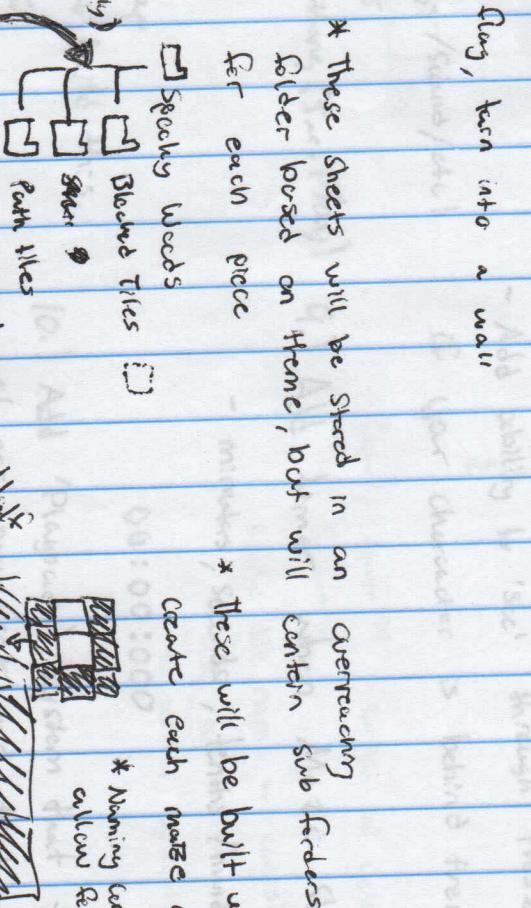
- Create tilesheets

→ **Front Page**

- Update renderer to use 3×3 per cell

- Update to use selected tilesheets

- Assign / Develop a "weight" system to allow for river tiles (could just add more basic ones to sheet) → **Open** :



To Do for Maze Craze (Cont.)

4.) Add "Special" Object maze generation

- Set objects into the maze

- will be replaced by sprite / collisions
- acts as "walls" before render!

* These will not iterate through the maze to place them. They will randomly select a cell, check it, and then move on to another random cell.

5.) Player Spawning / Movement

- Selecting Spawn Location
- Player Movement
- Player Collision with maze

6.) Add UI to allow for maze creation through menus.

- Title menu
- Options menu (potentially screen size settings / sound, etc)
- Play menu with maze setup settings

o True random option to not use settings (Start, Play)

7.) Implement Map Features

- 'ini' to display aerial view of map
 - will most likely use maze object to build this.
 - will reveal as progress is made

= Also will help displaying mini map

Real Name:



Copy & Maps

← Generate + View → mirror view with your own player

View → mirror view with your own player

8.) Add decoration generation to 'wall' files.

- Randomly fill in "walls" with various decorative sprites
- Rocks, trees, etc.

- Add ability to 'see' through these if your character is behind them.

9.) Add timer when Maze Starts

- minutes, seconds, tenths / hundredths

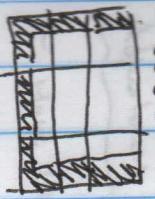
00 : 00 : 00

10.) Add 'playback' system that tracks players movement and then

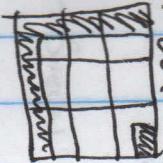
- displays progression on map once finished

- Also will display mailbox locations so you can see how close some were.

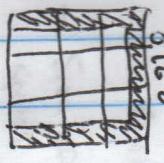
6001



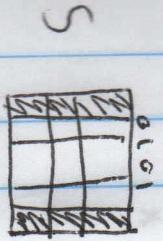
2



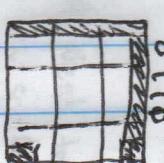
3



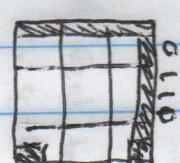
4



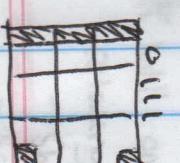
5



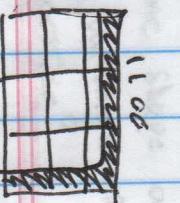
6



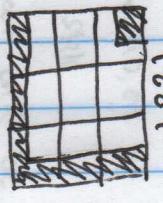
7



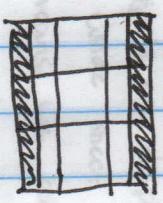
8



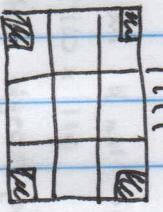
9



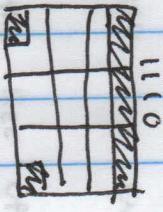
10



11



12



* Insert spawn

Spawn X

Y

Origin

of cell

(top right)

X

Y

Z

W

V

U

Sprite file name - 'cubism'

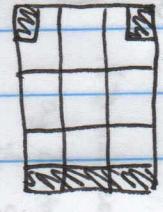
- use to build string

to render specific sprite

* Annoying Way

- Specific Render Instruction for Each Case 1-9, super unreadable, but works

13



* Helper Scripts?

- One for each just to move code elsewhere for readability.

loop i to 3

loop j to 3

case

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

X * "Better Way" → Less Randomization

- More tile 48 x 48 rather than 6x6 and build these up

Adjustment to Object generation:

- Have object arrays created via a sprite sheet (4×9)



- * Black pixels will be part of the object
- * White pixels will be empty space
- * (0,0) is middle of sprite

$\rightarrow [0, 1], [0, 0], [0, -1], [1, -1]$

To find center, divide length/height by 2, then round up. Or ~~floor it is starting at zero index~~

- Special objects could be 11x11 or bigger + account for size

- * would still have (0,0) at center/middle and would have custom sprite still

- Size of sprite sheet will be $10 \times$ the size of the sprites

- * If the objects are displayed in an 11×11 square + size of 110

then the whole sheet will be 110×110

- This means we can store 100 objects per sheet, and not need to iterate too much.

* Need to find out if we are able to iterate over a sprite manually and check color values, etc.

- * Will only store one shape and use matrix rotating to get variations of the shape

- Should be possible?

- Possible!

- Look up stack overflow

- "How do you return a two-dimensional array."

- Just store as 2D using of 0's and 1's (1's being filled)

[0 0 1 0 0
0 0 1 0 0
0 1 0 0 0
0 1 1 0 0]

(My world will have millions of objects, so this is not a good idea)

Inside double loop of 10:
divide by 10 to get size of individual (I)
increment by 10

divide by 2 and floor to get center
 $I/2$
centerObjArray = [I]
Iterate through $i = 0, \frac{I}{2} < i < I; i++$

Iterate through $j = 0, j < I; j++$
if pixel == (255, 255, 255)
objArray[i][j] = new pair(I, j, ObjArray)

* This object array will need to be part into another holding cell objects. This only holds the current object @points

Main Array • append (ObjArray)

To Do for Maze Craze (Cont.)

11.) User generated maze objects

- * See previous page for new system

- * Have sprite editor where users can create their own sprites to use in generation.

12.) Multiplayer * - WAY AT END

- * Split Screen for Sure

- * Maybe online

(Ghosts will be visible)

- leaderboards

- * same user points

- Racer mode

- Tag/Infection Mode

(Hide one seen)

13.) "Normal" Maze generation

- Just a standard maze

- * Would have multiplayer?

14.) Daily Maze with leaderboards:

- Can only be done once?

- Repeat attempts for high score (or random placement).

- Rewards currency on global generation. (High Score, Global Leaderboard).

14.) Power Ups / Abilities

- * Selectable Abilities in the mazes

ex. (They would all have some cooldown)

O. 1. Standing still extends 'sight'

O. 2. Coins tea drop and mark paths

"these will be here when I get back... right?"

Δ 3. Some kind of 'Hot/Cold'

meter when near a mailbox

- * too good? Maybe for only

so many?

Δ 4. Teleport to a random path tile?

O. 5. Displays 'Zone' boundaries

- * pretty good rather than checking map

Δ 6.1 Clone Jitson

- leave or clone at a spot, teleport back to it.

- * time limit on clone until it forces you to teleport back. (This is how we balance this).

16.) Sprites that move in wind

- * wave over map! depending on intensity

- sprite will "bend" more

- * Need more research though.

O = exploring
 Δ = Speed
 ↗ = Perfect scenario
 ↘ = In or perfect we know where we want them (know who) down there which gets faster

↑ = Definitely explore sooner (CPU under).

To Do for Maze Craze (cont.)

17.) Have some tile blocks (set of walk tiles) able to turn into another factor

(ex. Have a group that work be water tiles rather than grass for some variation)

* Wizard needs to redo how randomized Spiders are determined

- Would need to place each annuated tile in a folder and somehow find how many tiles are in that folder?

18.) Ability to Enter a Seed for Maze Generation / Save Favorite mazes (including Daily mazes).

- a.) Unlocked cosmetics through points
 - earned by completing mazes
 - + bonus for Daily placement on Daily maze
 - Tools (rainbow, etc.) - Titles (Prefix/Suffix)

- Clothing

- Aura (Rain, Fog, etc.)

- Texture packus?

- Shaders (Solid gold, Rainbow, etc.)

- Displays

Aura None

In maze. for main player.

(except for 'enemy' players in fog / thickets)

- Optimal middle way of from 'the'

Special Objects

20.) Lookout towers ^ that zoom the map / view out and lets you 'see' through decorative spiders (similar to wings the towers at tcm's).

- Special Object
- Could be a power up?

21.) Randomly found collectables while playing

- Placed on a random path or wall similar to a mailbox

22.) Lighting System

- Maybe have a 'dark' version of mazes and then have 'vision cones' with lights / flashlights.

23.) Similar to previous one, have a horor 'char' made with a roaming monster. Some game, but he places you back all the Start if caught.

~ Displays

Aura None

In maze. for main player.

(except for 'enemy' players in fog / thickets)

- Optimal middle way of from 'the'

All optional (1)

- could just have one and that's it.

Maze Create Scene Theme Ideas

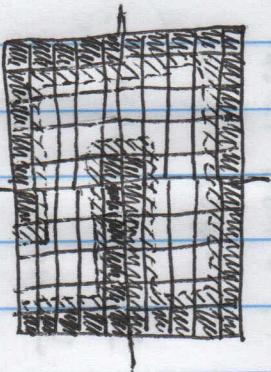
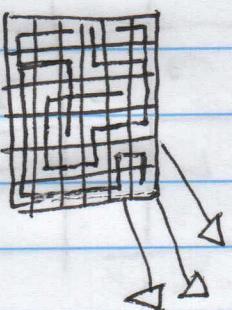
- Haunted Forest
 - Dead Trees
 - Fog
 - Spooky Stuff
 - Normal Forest
 - Evergreens
 - Dirt Paths
 - Small Ponds
 - Office Space
 - Carpet Floors
 - Cubical Walls
 - Water-cooler Fountain
 - Army with other office stuff
 - Beach
 - Path will be sand
 - "walls" will be water
 - Opportunity for animals.
 - Cosmic
 - Similar to rainbow road
 - In Space
 - Desert
 - Snowy Mountain / Forest
 - City / Village
 - Paths would be ponds
 - Lots of "special" objects for buildings otherwise just sidewalks or walls
- Inside Veins?
 - All paths would be veins surrounded by tunnels.
 - Some abilities would be OP
 - Field with very tall grass
 - Corn Field Obviously!!

Normal Maze Algorithm

- Typically go through a stack and create paths, we construct walls

* Already can render
the inner 3x3

5	4	7	6	35	36
6	3	14	9	34	33
7	2	18	10	11	32
8	16	17	12	31	
9	11	15	14	13	30
10	12	26	21	28	22



- Could do this and then just render without "wall" cells
- OR
 - Each cell is 5x5 with walls surrounding those areas that need them
 - * Path would always be 3 wide?
 - Technically 2 with one being a "full" path
- Cell will be determined based on 'neighbor' cells
 - Array of neighbors will be stored for each cell, one added to to during the generation algorithm.
- Start and End will be based on longest path
 - will need an algorithm for this.

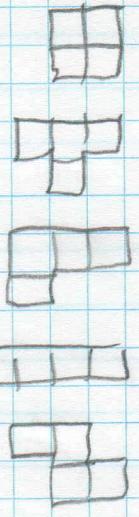
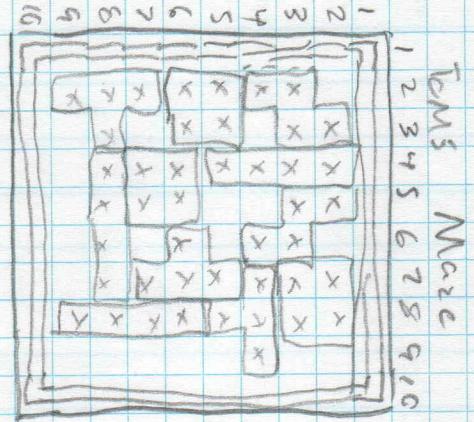
1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
l	m	n	o	p	q	r	s	t	u
5	6	7	8	9	10	1	2	3	4
4	5	6	7	8	9	10	1	2	3

- Grid
Array[10][10] of Cells

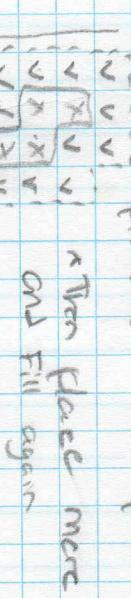


- Cell Object
 - Visited (Boolean)
 - Adjacent Nodes (Array of Cells)
 - Associated Sprite - Prefab with Collision Info.
 - (Discoverer at end)

1. Fill outside with path
2. 2D Bin Filling with Tetrominoes



- * These are 'Blocked areas'
- * Filter assume paths are already 'around'
- * These
- * Outline placement with path?



- Validation code
- be After all placement
- * Refinement to single wide path

- * How to determine single wide path?

* Check each nearby floor, then check for path with next edge.

* Counter clever count up to three is bad

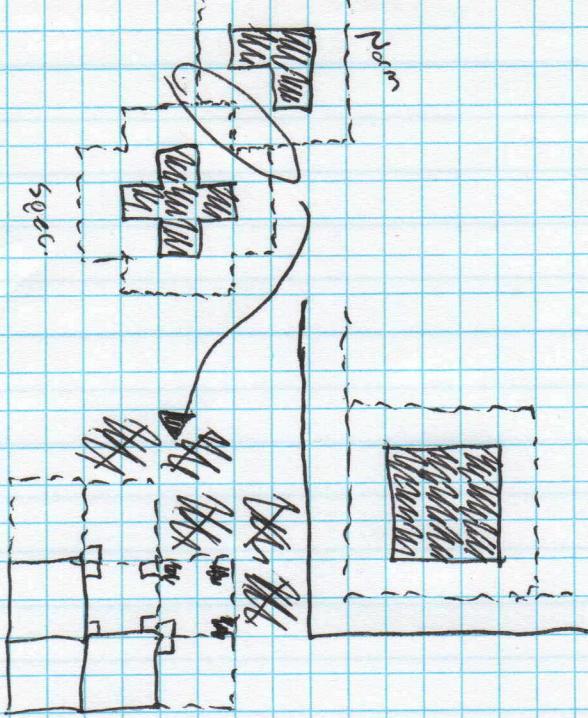
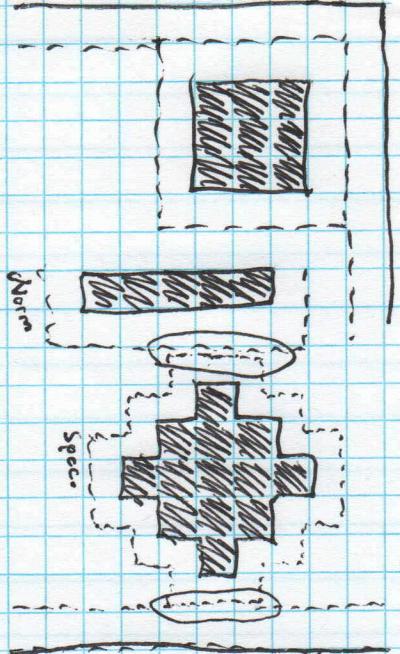
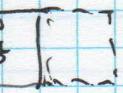
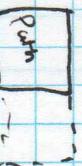
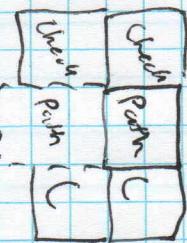
- Eliminate third count as blocked

Count as path

Hot Spot

Path

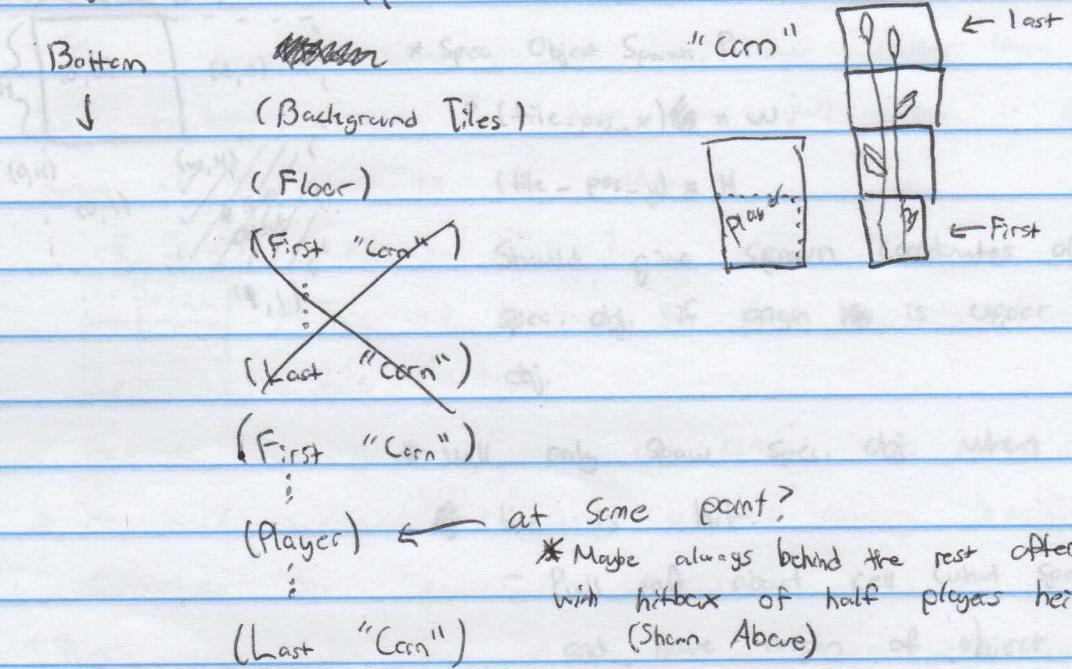
* Path needs adjacent special points
N,S,E,W



* If surrounded by paths, one or one of these is 'special', replace with "fun" tile.

Maze Craze

- ## • Layers for Tilemaps



* ~~May~~ Most likely need Z-sorting

and objects and the corn tiles

- How many layers to file map?

• Could just make 80+ layers

(nah its way more than 80) ~~BAD IDEA~~

- May need to see where the

file is in respect to open paths

(if no open paths, then don't worry)

About 50%). (Just do "corn" seat)

...
...
...
...

Digitized by srujanika@gmail.com

Bottom Corn > Special Object (temp. Sothe. Alaska)

{ Not Above Path \Rightarrow At least 4 layer

Top Corn

Player / Entities / Objects

At least I might

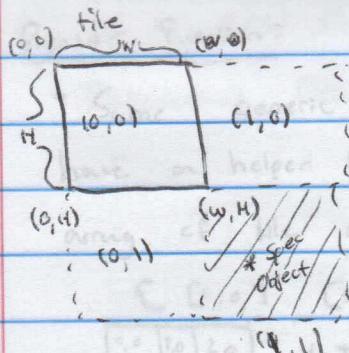
Battam (1980)

100 (100)

When Above Path At least 4 layers

• Could have been at Sprite empty to allow
for randomization of the Specie Object that it describes.

Maze Game



* Spec Object Spawns

$$= (\text{tile_pos_x}) * w$$

$$(\text{tile_pos_y}) * H$$

Should give spawn coordinates of

spec. obj. if origin is upper left of

obj.

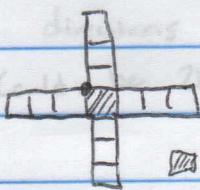
* Then a single hit * will only spawn spec. obj. when "main spec"

* May need special file is hit.

- Pull info about cell (what spec obj. it is)

and have origin of object base on

where the main tile is.



- main tile
- object origin

"Create player controller"

Resource Needs

- Maze Resource (Input into maze object)

- Tile Set

- width and height

- Sprite Sheet

- Create main -> Sprite Mask

- Arrays

- Special Object List

- Add win condition -> Special Object (temp Sprite Mask)

- Add win Screen

- Sprite Mask player back to main screen

- Create Standard for each cell

- mask the

- Create resource -> origin on upper left of main tile for each type of

tile so we can * Actual Scene vs. tile map cells (maybe overlapping?)

* Or design the * Could leave areas of Sprite empty to allow for randomization of the Special Object floor if desired.

Maze Craze

Render Function:

- Same generic rendering (3×3 loop) but rather than have a helper function for each case, just pass in an array of tile coordinates to render

$\{ [0,0], [0,1], \dots [2,2] \}$

0,0	1,0	2,0
0,1	1,1	2,1
0,2	1,2	2,2

* This will determine the tiles we pull from in tile sheet.

- Then a single helper function can render each passed in tile.

* May need special tiles array to help rendering specific

Scenarios. See gamemaster project!

- This array will be set based on sum of the cardinal directions (that 16 case switch statement.)

* Could be 2D array? May be more complicated than necessary.

TODO:

To Catch Up

Game Loop

• Create player character ✓

- Sprite, animation, collision, movement, camera ✓

• Add player character to available cell (Path) ✓

- Test movement and collisions ✓

• Fix/Adjust camera zoom on player ✓

• Create main menu to generate maze from other scene

- Eventually add ability to set maze settings → *mazeState*, *mazeSettings*, *globalFov*

• Add win condition to game

• Add win screen that will take player back to main screen

• Add overlatable function "add_fence()" that can be called by render section.

• Create standard for sprite tile layout

- Create resource that also describes boundaries for each type of tile so we can add randomization back in (and maybe weighting?)

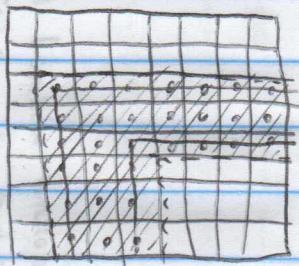
• Or design the files in layers to not need something like this?

Current objectives
func, etc

mazeState
mazeSettings → *globalFov*
★ *playerSettings* " "
playerStats " "

Maze Craze

- Generate Border Tiles



• = Path

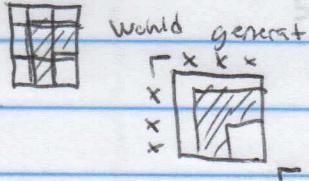
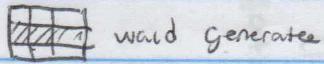


Potential cases



- * Generate during the tile rendering cases

Ex,



- The Special corners would be handled by these turn tiles

- Overloadable function so that it doesn't always need called. Initially empty function, but some filesets will have fences or fence behaving objects and can program the behavior they need.

Maze Craze

Resources

• Maze

- Size - will contain an x and y size

- Maze Array → array of maze cell objects

- ↳ • Maze Cell (custom resource)

- Various flags and data regarding current cell

- Spec Obj., Available, wall bool, etc.

- ~~Tile Map~~

- Maze Resource (custom resource)

- ↳ • Tile Map

- Contains tile set textures

- " " " collisions, how many, etc.

- ■■■ JSON of tile limits

- Indices to allow for random choice of type of tile

- ■■■ Weights

- Spec Obj. Resource (custom resource)

- PNG of special objects

- JSON of Spec. Obj data (Attributes)

- Weights (Array for each Obj.)

- Spec. Obj size limit (to know how to parse) (width + height of objects).

- (OPTIONAL) (Array for each Obj.)

- ~~Sprite~~ Sprites for each special object

- (to allow for custom sprite objects in maze)