{'north' : false},

if south != true,
[0] → [2]
north → south
[1] → [3]
west → east

Tile

north
east
south
west

exits[] = [true, F, F, F]

true/f & white arrow

Dining Room - Final indoor tile

white_arrow[] = [T, F, F, F]

nine_oclock
ten
eleven
item
number-of-cards-left = 7

devcard - assign manual id

[dev

discard_pile
cards_pile.[ ] ← Draw a card; random fn move to discard pile.
random fn (1 - number of cards_left)

nine.oclock={ 1: "You try ....", 2: battle(), 3: item() ..., 9: }
ten = { 1: ..., ... }

DevCard
nine_fn
:
item_fn

item action → draw another card. item

DevCard

Battle

In- Game class.

In DevCard class, a field called no_of_zombies, usually is zero.

Battle

item.action()

[2, 3, true]

↑

Item

cowerd() in Game class

-1 DevCard but no card action.

Cower

tile_fn

Tiles

hours_left in Game e.g. 8-1

Time

look for the next Tile exit.

E.g. new Tile has East

then the old Tile exit is West

No Exit Room (Bathroom + storage)

+1 Health Tile w/ ~~Tile~~ health property?

Garden / Kitchen.

Patio initially ~~rotated~~ so that white ~~arrows~~ match

white_arrow[] matches w-a dining room

Patio

Tile Placement

2D Array ~~8x8~~; 16 x 16

Starting from center

Tiles

exits[ ] = [T, F, T, F]

Exit to South

Coordinate y+1

Exit.