

LIGHT IT

レベルの作り方

v1.0.0

はじめに	3
レベルを構成するファイル	3
レベル定義ファイル	4
文章定義ファイル	7
特殊な単位について	8
マップファイル	9
フラグ管理	11
イベント	13
レイヤーの種類	14
FLOOR フロアレイヤー	15
DIRECTION 方向レイヤー	17
DECOR 装飾レイヤー	17
ZONE ゾーンレイヤー	18
LIGHT ライトレイヤー	19
ENEMY 敵レイヤー	21
OBJECT オブジェクトレイヤー	36
ITEM アイテムレイヤー	45

レベルの作り方

はじめに

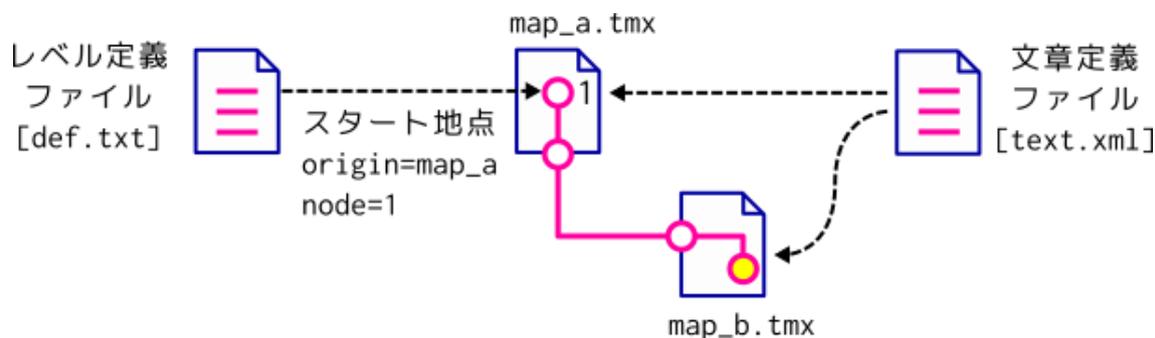
このドキュメントでは、レベルを構成するデータの仕様と、その設定方法を説明する。このドキュメントは、サンプルファイルを見ながら読むことを推奨している。サンプルファイルは公式ページ(<http://torazoit.com/lit/jp/level.html>)からダウンロードできる。

レベルを構成するファイル

1つのディレクトリに、以下のようなファイルを準備する必要がある。

```
レベルディレクトリ ─┬─ レベル定義ファイル [def.txt] (1つ・必須)
                   │─ 文章定義ファイル [text.xml] (1つ・必須)
                   │─ マップファイル [**.tmx] (1つ以上)
                   └─ 音楽ファイル (任意)
```

音楽ファイルを除くファイルの関係は下の図のようになっている。



レベル定義ファイルでは、プレイヤーのスタート地点、BGM など、レベル全般に関わる設定が定義される。

マップファイルは地形や敵の配置を設定する。レベルは1つ以上のマップからできていて、1つのマップは1つのtmxファイルで作られる。マップどうしは、**ノード**と呼ばれる点を経由して繋がっている。プレイヤーはそのノードを通じてマップ間を移動する。

文章定義ファイルは、ゲーム内で表示される文章をxml形式で準備する。レベルのタイトルも本ファイルで指定するため、用意することを推奨している。文章はキーと言語ごとに管理され、マップファイル内で**キー**を指定して、文章を引用する。

任意の**音楽ファイル**を同ディレクトリ内に格納することで、ゲーム内のBGMとして利用できる。

レベル定義ファイル

レベル定義ファイルには、レベル全般の設定が記述される。プレイヤーの初期位置や、BGM、タイトルなどである。

ファイル名

```
def.txt
```

ファイル名は def.txt、中身はプレーンなテキストファイルとする。

フォーマット

```
origin=スタート地点の tmx ファイル名  
node=スタート地点の ノード名  
bgm=BGM  
items=初期アイテムリスト（後述）
```

origin と **node** によりスタート地点を指定する。

bgm により全マップで流れる基本の BGM を指定する。ただし、tmx ファイルごとに個別にも指定可能で、その場合はそちらが優先される。

items では、プレイヤーが初期状態で所持しているアイテムを指定できる。アイテム指定の方法は、アイテムの章に示す。

BGM のフォーマット

指定方法(フォーマット)	
ゲーム標準の音楽から選択	BGM 識別子(後述)
同ディレクトリ内に置いた任意の音楽	@音楽ファイル名(拡張子含む)
無音	!

BGM 識別子リスト

ゲームで標準に用意されている音楽は、下の表のとおりである。これらは BGM の識別子であり、レベル定義ファイルの **bgm** や、tmx 内で指定する **bgm** プロパティで使う。

識別子	特徴
core	怪しい・静か
triplex	穏やか
stealthy	怪しい・静か
tribe	にぎやか・うるさい
popo	幻想的
myth	怪しい・静か
homeland	怪しい・うるさい

fate	うるさい
chant	低い声
steps	スタート画面
env_silent	無音(ホワイトノイズ)
env_sea	波音
env_river	川の濁流音

文章定義ファイル

レベルのタイトルや、ゲーム内のドキュメント、ポップアップメッセージの文章を xml 形式で記述する。

ファイル名

同フォルダに'text.xml'という名前 で保存する。

フォーマット

text.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<root>

  <text key="@title">

    <en>Title</en>

    <ja>タイトル</ja>

  </text>

  <text key="key1">

    <en>Hello.</en>

    <ja>こんにちは。</ja>

  </text>

</root>
```

例えば、上のファイルでは、識別子 @title と key2 (緑の網掛け) と、対になる文章 (黄色の網掛け) が、言語ごとに記述されている。@title は規定の識別子で、レベルのタイトルとして扱われる。言語は、英語が 'en'、日本語が 'ja' として指定する。

特殊な単位について

フレーム

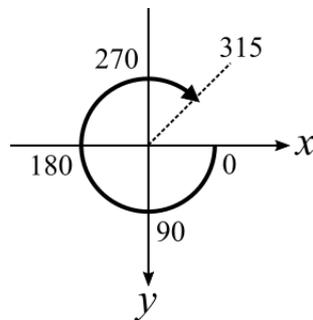
時間の設定には、**フレーム数**という単位を使う項目が多くある。

このゲームでは、1秒あたり画面を60フレーム更新するように調整される。従って、**1フレームは約1/60秒**に対応する。

角度

ゲーム内の座標系は右方向をx軸の正、下方向をy軸の正とする直交形座標である。そのため、右を0度とし、時計周りに角度が大きくなる座標系となる。

角度はすべて、ラジアンではなく、度(0~359)で設定する。



長さとしてのセル

セルという単語が長さとして使われたとき、1セルはタイル一辺の長さとする。

マップファイル

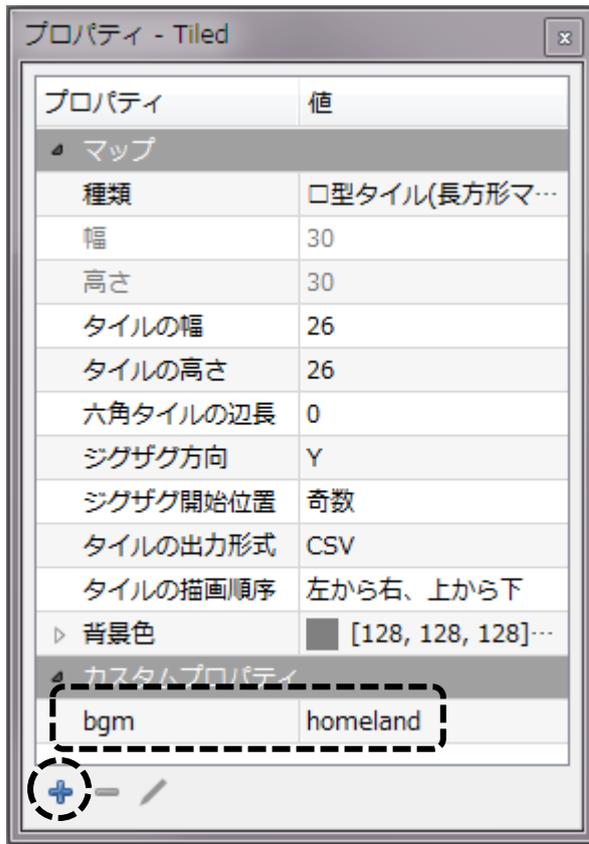
マップ作成ソフトウェア [Tiled](#) を利用してマップを作る。このゲームは、ソフトウェアの標準フォーマットである `tmx` 形式のファイルを読み込む。

カスタムプロパティ

名前	フォーマット	説明	
bgm	BGM	BGM(音楽)を <code>tmx</code> ファイルごとに指定できる。指定しなかった場合は、レベル定義ファイルで指定した音楽が使われる。	任意

カスタムプロパティの設定方法

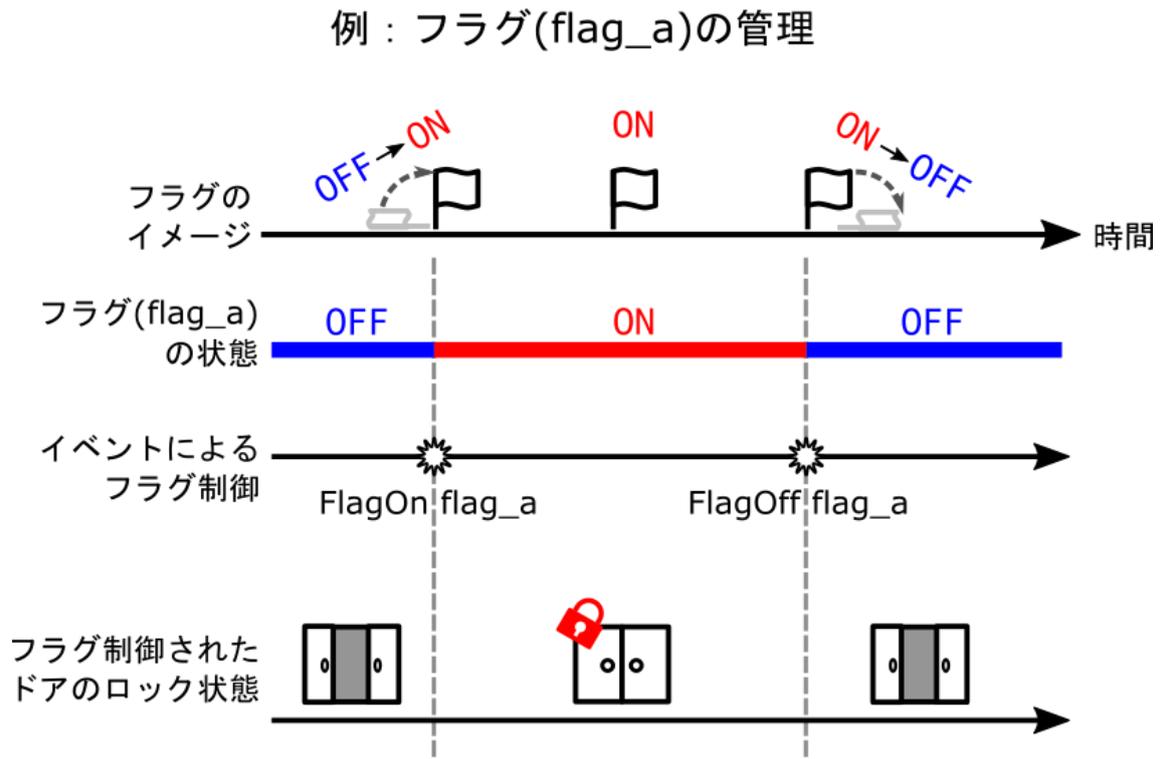
カスタムプロパティの設定を、**bgm** を例にとって示す。この操作は、様々な設定で何度も使う。



1. マップ->マップのプロパティ
でプロパティを表示
2. プロパティウィンドウ左下の '+' を押
す
3. ポップアップウィンドウの '名前'
に 'bgm' と入力
4. 任意の値を入力

フラグ管理

フラグは、様々なイベントの条件となるものである。例えば、「ある場所を通過すると、フラグAをONにする。フラグAがONの時、あるドアをロック状態にする。」というような使い方がされる。下の画像は、そのようなフラグ管理の概要である。



フラグをON・OFFする方法

- **イベントジェネレータ**により FlagOn・FlagOff イベントを生成する。
- 特定の敵 (BlockerA, BlockerB, Yama) の 'die-event' プロパティに FlagOn・FlagOff イベントを指定する。

フラグの使いどころ

- **ドア** の 'lock' プロパティ。このプロパティにフラグを指定すると、そのフラグが ON の間、ドアがロックされて開かなくなる。
- **イベントジェネレータ** の条件として。指定したフラグが ON の場合に、イベントを発生させることができる。
- **オブジェクト** である **メッセージ** の条件として。指定したフラグが ON の場合に、メッセージを表示させることができる。

イベント

イベントを使うことで、フラグ管理、ドアの開閉、画面のフラッシュ、メッセージの表示ができます。

各種イベントは、EventGenerator を使って発生させます。

イベントのフォーマット

フォーマット	説明
FlagOn □フラグ識別子	指定した識別子のフラグをオンにする
FlagOff □フラグ識別子	指定した識別子のフラグをオフにする
OpenDoor □名前	指定した名前のドアを開ける
CloseDoor □名前	指定した名前のドアを閉める
GenEffect □Flash□ フレーム数□赤□緑□青	画面を指定フレーム数の間フラッシュさせる。色は赤緑青で指定し、それぞれ 0~255 の値で指定する。
Message □フレーム数□ 文章の識別子	インスタントメッセージを表示する。インスタントメッセージは、画面下に白色で数秒間表示されるメッセージのことです。

レイヤーの種類

マップファイルは複数のレイヤーで構成される。レイヤーごとに、設定する項目が異なる。以下に各レイヤーの概要を説明する。

レイヤー名称	概説
system	スタート地点やゴールを配置する。
item	プレイヤーが拾えるアイテムを配置する。
object	ドアやブロックなど、多種のオブジェクトを配置する。
enemy	敵を配置する。
light	固定型ライトを配置する。
zone	敵の縄張りなどのゾーンを配置する。
decor	藻やツタなどの装飾を配置する。
direction	floor のタイルごとの方向を決定する。
floor	床や壁などを配置する。

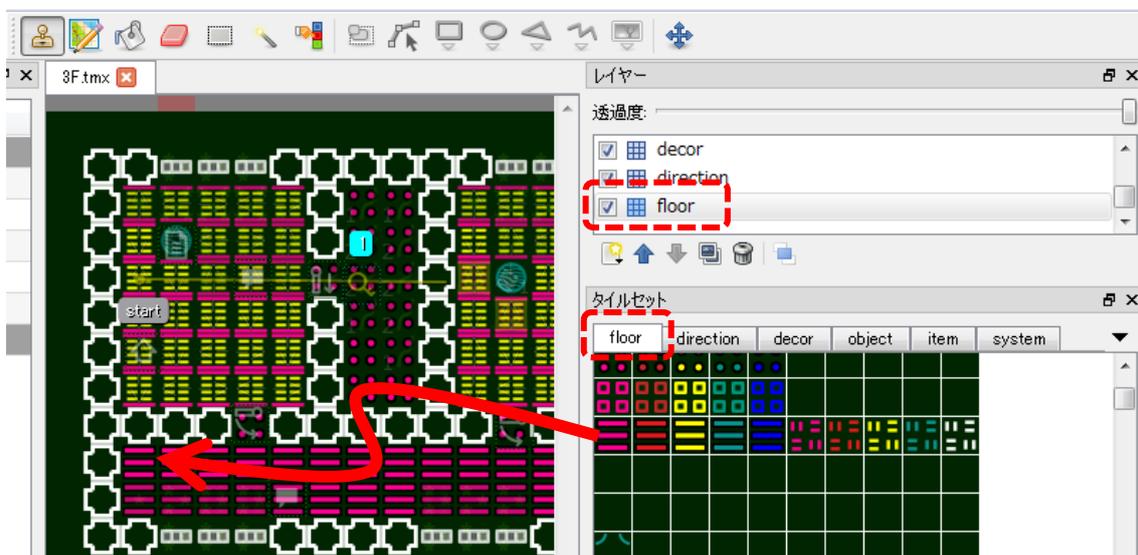
これより、下のレイヤーから順に説明する。

floor フロアレイヤー

フロアレイヤーは、壁や床、椅子や机などの障害物を配置するレイヤーであり、レベルの構造を決定する必須のレイヤーである。

設定方法

タイルセットの **floor** からタイルを選び、**floor** レイヤーにタイルを置く。



留意点

このレイヤーではタイルの方向は指定できない。例えば、トイレのタイルは標準で東向きだが、これを南向きに置きたい場合は、**direction** レイヤーで南向きを指定する。

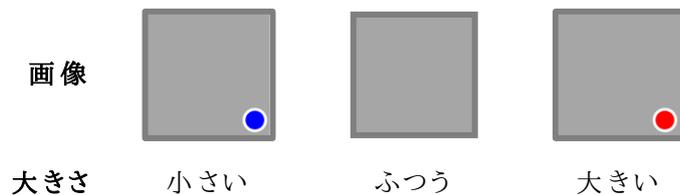
タイル特性

タイルには見た目の違いの他にも、足跡の大きさ、光の透過性、通行の可不可という違いがある。その見分け方を以下に示す。

足音の大きさ

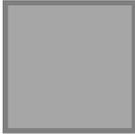
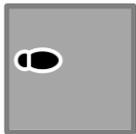
タイルを通過した際に発生する足音の大きさには 3 段階ある。

それぞれのタイルの右下に、色をついたマークが示されている。これが足音の大きさを示す。



光の透過性と通行の可不可

各タイルの左上のマークは、光の透過性と通行の可不可を示している。

画像				
光の透過性	×	○	○	×
通行の可不可	×	○	×	○
例	壁	床	低い障害物	高い茂み

direction 方向レイヤー

方向レイヤーは、フロアタイルレイヤーで置かれたタイルの方向を 8 段階で指定する。



方向は、東向きを 0 度、時計回りとする。従って、東向きを指定した場合は、タイルの向きに変化はない。南向きを指定した場合、時計回りに 90 度回転することになる。

設定方法

タイルセットの **direction** からタイルを選び、**direction** レイヤーにタイルを置く。

decor 装飾レイヤー

装飾レイヤーは、フロアタイルの上に、藻やツタなど、グラフィックに装飾を加えるためのレイヤーである。指定してもゲーム性に変化はなく、グラフィックの変化のみとなる。

設定方法

タイルセットの **decor** からタイルを選び、**decor** レイヤーにタイルを置く。

zone ゾーンレイヤー

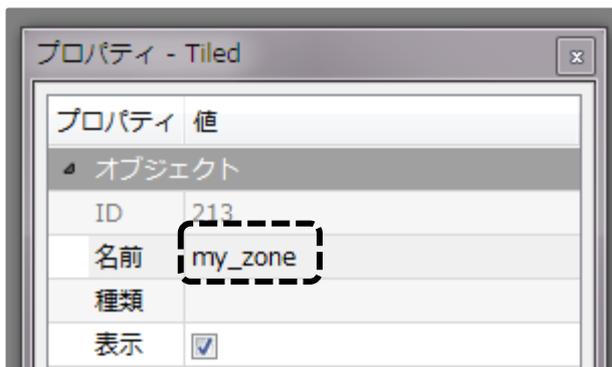
敵の縄張り(後述の **territory** プロパティにて指定)のためにゾーンを設定できる。

設定方法

1. **zone** レイヤーをクリックして選択し、操作できる状態にする。



2. “ポリゴンを追加”を選択
3. タイルの中心点を含むようにしてポリゴン(多角形)を作る。
4. ‘プロパティ’の‘名前’に**識別子**を指定する。



light ライトレイヤー

固定型のライトを設置する。

設定方法

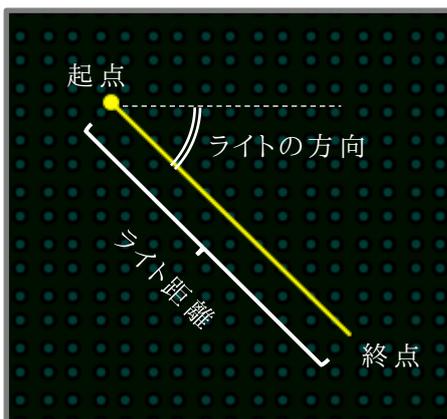
1. **light** レイヤーをクリックして選択し、操作できる状態にする。

2. “ポリラインを追加”を選択

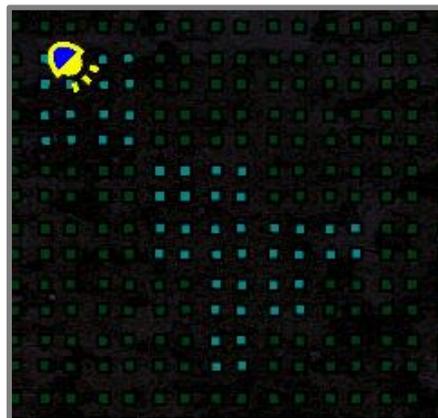


3. 2点を指定する。起点がライトを設置する点となり、終点で方向と光の届く距離が決まる。

4. プロパティを設定



→



標準プロパティ

項目	フォーマット	説明	
名前	文字列	ライトの識別子。オブジェクトの Switch の connect プロパティに指定する。	任意

カスタムプロパティ

名前	フォーマット	規定値	説明	
pow	True False	True	電源の初期状態	任意
switch	True False	True	切り替え可能かどうか	任意
deg	実数 [度]	30	ライトの視野角	任意
rotatable	空白		回転可能かどうか	任意
rot	実数 [度/秒/60]		自動回転の角速度	任意
btry	*バッテリー	inf	(*)フォーマットは後述	

バッテリーのフォーマット

種類	フォーマット
使いきり 	btry_残量[秒]
充電可能 	re_容量[秒]_残量[秒]
無限 	inf

enemy 敵レイヤー

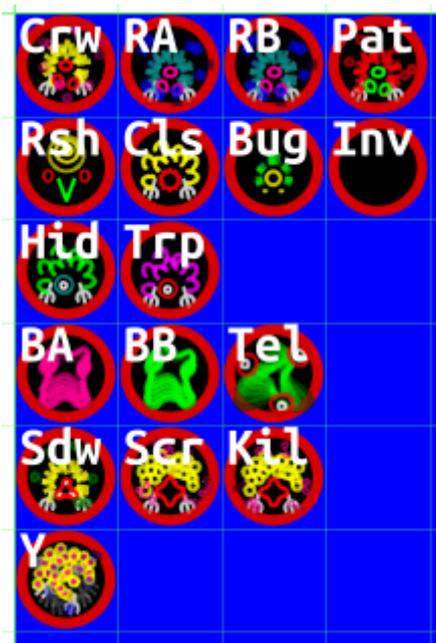
敵レイヤーには敵を設置する。

設置方法

タイルセット **object** からタイルを選び、“タイルを追加”するモードに切り替える。敵の描かれたタイルセットから配置したい敵を選択し、**object** レイヤーにタイルを置く。(Ctrl キーを押しながら配置すると、セルにスナップされ綺麗に配置できる。)



タイルを追加



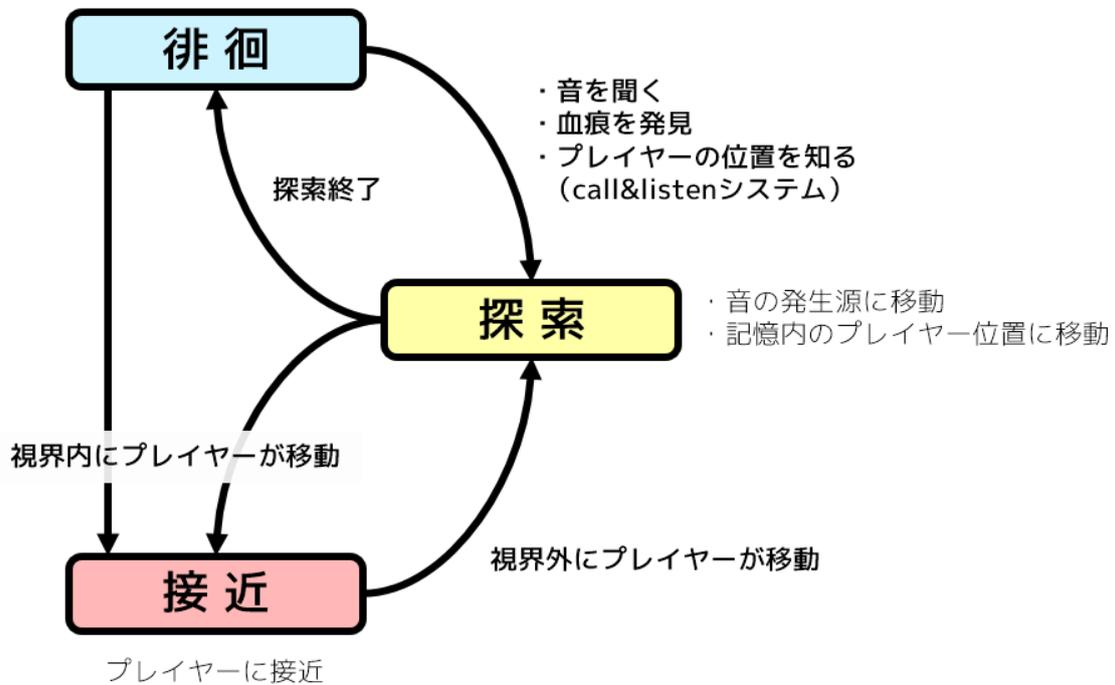
敵タイルセット

敵の行動モード

敵は3つの行動モードを持つ。巡回、探索、接近の3つである。敵はこれらを切り替えながら行動している。

移動速度を行動モードごとに設定するため、敵プロパティの設定方法を示す前に、行動モードの説明をしておく。

下図のような条件で、行動モードが遷移させながら行動している。



カスタムプロパティ

名前	フォーマット	規定値	説明	
dir	実数 [度]	0	初期角度	任意
wear	文字列	-	敵が所持しているアイテムを指定	任意
pos	pin 識別子 ;区切り・複数	-	初期位置をランダムに指定できる。 指定したピンからランダムに選ばれる。	任意
pins	pin 識別子 ;区切り・複数	-	巡回点のリストなどをピンの識別子で指定する。複数指定可能。	敵種類による
territory	zone 識別子 ;区切り・複数	-	縄張りをゾーンの識別子で指定する。	敵種類による
step	実数 (セル数)	敵種類 に依存	敵1歩の歩幅。単位はセル数。	任意
wnd-step	〃	〃	巡回時の歩幅	任意
app-step	〃	〃	プレイヤー接近時の歩幅	任意
src-step	〃	〃	探索時の歩幅	任意
itval	分布形 (後述)	敵種類 に依存	一歩と次の一歩の間隔 (Interval)。定数ではなく、分布形で指定する。	任意
wnd-itval	〃	〃	巡回時の itval	任意
app-itval	〃	〃	プレイヤー接近時の itval	任意
src-itval	〃	〃	探索時の itval	任意
freeze	Freeze(後述)	-	一定時間光を当てて停止	任意
call	チャンネル ;区切り・複数	-	“call&listen システムについて”の項を参照	任意
listen	チャンネルリス ;区切り・複数	-	“call&listen システムについて”の項を参照	任意

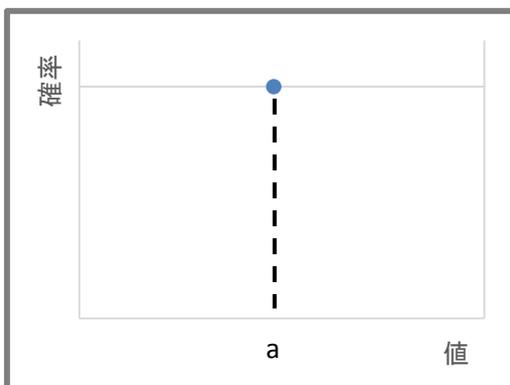
分布形

種類	概要	フォーマット
Constant	定数	Constant $_a$
Uniform	一様分布	Uniform $_a_b$
NormDist	正規分布	NormDist $_mu_sigma_min_max$
Log	ログ関数	Log $_a_b_min$

Constant (定数)

フォーマット: Constant $_a$

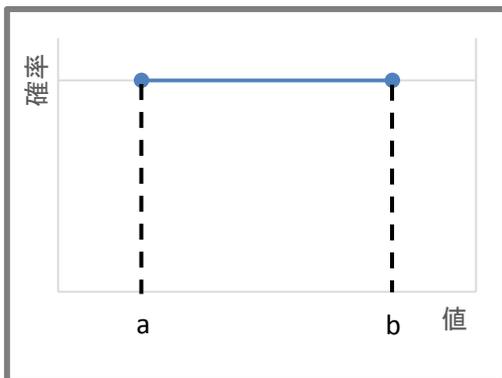
常に a を返す。



Uniform(一様分布)

フォーマット: `Uniform` a b

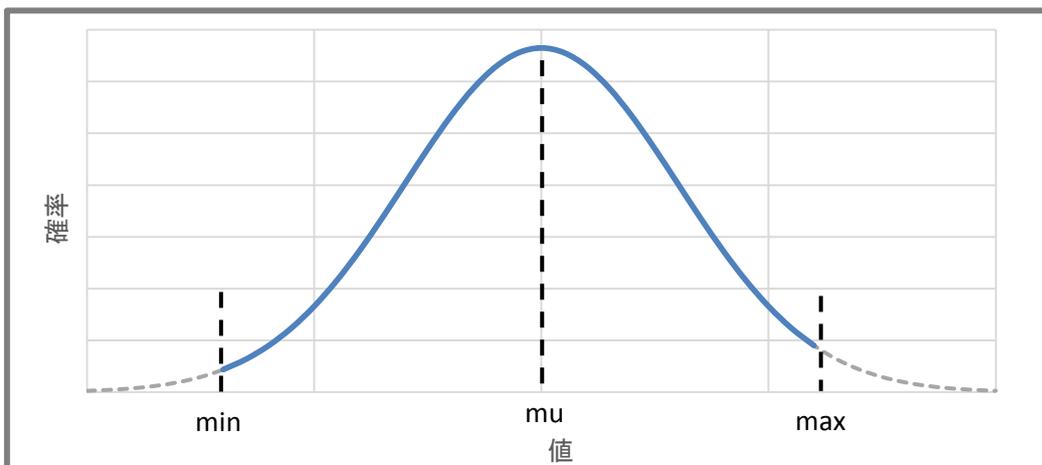
a から b までの値を一様に返す。



NormDist(正規分布)

フォーマット: `NormDist` μ σ min max

μ : 平均値 σ : 標準偏差(ばらつき) min : 最小値 max : 最大値



Log (ログ関数)

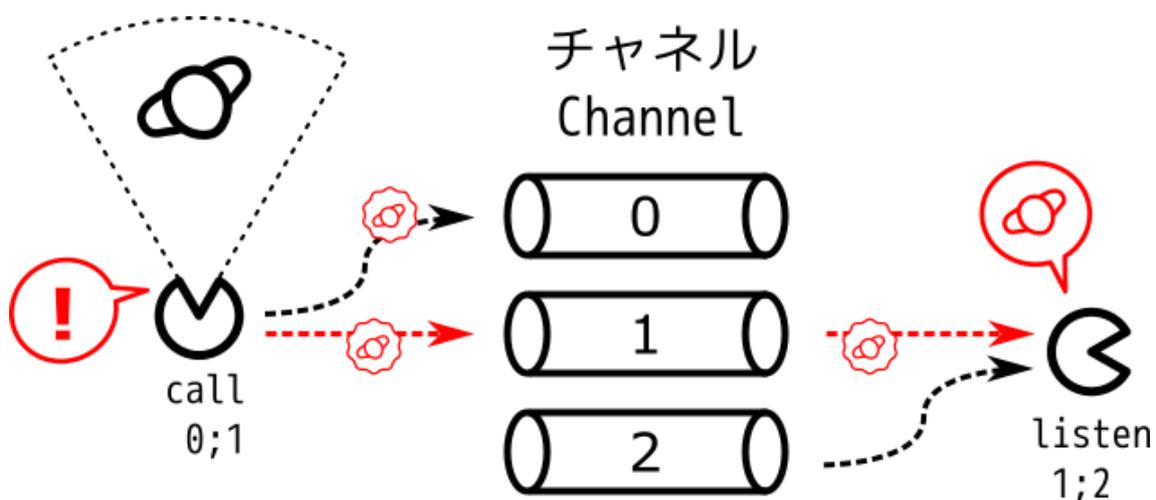
Log \sqsubset $a \sqsubset b \sqsubset min$

$$F(x) = a * \log(1 + bx)$$

$$x(0 \leq x \leq 1)$$

call&listen システムについて

ある敵が別の離れた場所にいる敵に、プレイヤーの位置を知らせることができる。これを call&listen システムという。概要を図で示す。



call を設定された敵は、プレイヤーを発見した時に、指定されたチャンネルにプレイヤーの位置を送信する。**listen** で該当のチャンネルを受信する設定がされた敵は、それを受信し、プレイヤーを探しにくる。(敵の種類によっては探しに行かない。動かない敵や攻撃しない敵など。)

敵一覧

	名前	特性	関連 プロパティ
	Crawler	<ul style="list-style-type: none"> ・もっとも基本的な敵。 ・止まったり、近づいたりする。 	
	Closer	<ul style="list-style-type: none"> ・非被照明時、常に接近。 	
	Invis	<ul style="list-style-type: none"> ・透明な敵。 	
	RounderA	<ul style="list-style-type: none"> ・pins のポイントを巡回。 	pins
	RounderB	<ul style="list-style-type: none"> ・A はランダム、B は順番に巡回。 	
	Patroller	<ul style="list-style-type: none"> ・territory の zone 内を巡回。 	territory
	Shadow	<ul style="list-style-type: none"> ・プレイヤーが視界外に出ると、高速で移動。 	
	Screamer	<ul style="list-style-type: none"> ・視界内のライトを一定時間だけ利用不能にする。 	
	Killer	<ul style="list-style-type: none"> ・一定時間だけプレイヤーの足を止める。 ・pins のポイントをランダムで巡回。 	pins
	Rusher	<ul style="list-style-type: none"> ・直線的に接近。 	
	Bug	<ul style="list-style-type: none"> ・短い視界の距離 	
	BlockerA	<ul style="list-style-type: none"> ・通路をブロックする。 ・一度光が当たってから指定時間後に消滅。 	die die-event
	BlockerB	<ul style="list-style-type: none"> ・通路をブロックする。 ・指定時間光を当て続けると消滅。 	die die-event
	Telepos	<ul style="list-style-type: none"> ・標準で call が設定されている。チャンネルは'0'。 	

	Hider	<ul style="list-style-type: none"> ・逃げる。 ・攻撃しない。 	<ul style="list-style-type: none"> territory stay,freeze
	Tripper	<ul style="list-style-type: none"> ・territory の zone 内を巡回。 ・攻撃しない。 	<ul style="list-style-type: none"> territory stay,freeze
	Yama	<ul style="list-style-type: none"> ・ライト不能とプレイヤー足止めの能力 ・territory の zone 内を徘徊。 	<ul style="list-style-type: none"> territory die die-event

敵の種類(特徴・プロパティ)

敵の種類ごとの特徴と、設定が必要、または推奨されるプロパティを説明する。

Crawler



最も基本的な敵。徘徊し、プレイヤーを見つけると接近する。

カスタムプロパティ

名前	フォーマット	説明	
territory	zone の識別子 ;区切り・複数可	指定されたゾーンから外れた場合、ゾーン内に戻る。	任意

Closer

プレイヤーを見つけると、光を当てられていない間は常に接近する。



Invis



性質は Crawler と同じだが、透明なグラフィックを持つ。

RounderA / RounderB



pins で指定された地点を徘徊する。

カスタムプロパティ

名前	フォーマット	説明	
pins	pin の識別子 ;区切り・複数可	指定されたピンの地点を巡回する。 RounderA はランダムに巡回し、 RounderB は指定された順番で巡回する。	必須

Patroller



territory で指定された zone 内を巡回する。具体的には、「zone 内の点をランダムに選択し移動」を繰り返す。

カスタムプロパティ

名前	フォーマット	説明	
territory	zone の識別子 ;区切り・複数可	指定された zone 内を巡回する。	必須

Shadow



基本的な性質は Crawler と同じだが、プレイヤーが視界外に出ると、高速で移動する。

Screamer



光を当てられると叫び、敵の視界内にあるライトを一定時間だけ利用不能にする。

Killer



プレイヤーが Killer の視界内に入ると叫び、一定時間だけプレイヤーの足を止める。

Bug



短い視界の距離を持つ。

Rusher



視野が狭く、プレイヤーを発見すると直線的に接近する。プレイヤーが視界外にいる場合は、初期位置に戻る。

敵の初期状態での方向は、ランダムに決定されるので、**dir** プロパティの設定が推奨される。

BlockerA / BlockerB

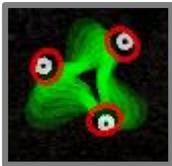


通り道を塞ぐ。光を当てて指定時間経過後に消える。時間は **die** プロパティで指定する。

BlockerA は、一度光が当たってから指定時間後に消えるのに対し、BlockerB は、指定時間だけ光を当て続けると消える。

名前	フォーマット	説明	
die	秒	消滅のための指定時間。	必須
die-event	イベント ;区切り・複数	消滅時に発生させるイベント。	

Telepos



動かず、攻撃もしないが、視野が広く、プレイヤーを発見すると敵にプレイヤーの位置を知らせる。デフォルトで **call** プロパティにチャンネル 0 が設定されている。

名前	フォーマット	規定値	説明	
call	チャンネル(整数) ;区切り・複数可	0	プレイヤーの位置を流すチャンネルを指定する。listen に対応するチャンネルが設定された敵は、その位置に駆けつけることになる。	推奨

Hider



プレイヤーが近づくと、ライトを当てると逃げる敵。逃げる範囲は **territory** で指定する。攻撃をしてこないが、ライトを当てても止まらない。

名前	フォーマット	規定値	説明	
territory	zone 識別子 ;区切り・複数可	-	逃げる範囲の zone を指定する。	必須
stay	分布形	Constant┘60		

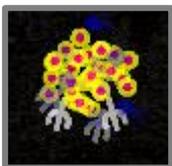
Tripper



常に移動している。移動範囲は **territory** で指定する。攻撃をしてこないが、ライトを当てても止まらない。

名前	フォーマット	規定値	説明	
territory	zone 識別子 ;区切り・複数可	-	移動範囲の zone を指定する。	必須
stay	分布形	Constant┘300		

Yama



詳細不明。

名前	フォーマット	規定値	説明	
territory	ゾーン識別子 ;区切り・複数	-	移動範囲を zone で指定。	必須
die	秒	600	死亡までの時間。	任意
die-event	イベント ;区切り・複数	-	死亡時に発生するイベント。	任意

object オブジェクトレイヤー

オブジェクトレイヤーでは、ドア、カーテン、スイッチなどを設置する。

設定方法

タイルセット **object** からタイルを選び、“タイルを追加”するモードに切り替え、**object** レイヤーにタイルを置いていく。(Ctrl キーを押しながら配置すると、セルにスナップされ綺麗に配置できる。)



タイルを追加

オブジェクトの種類



ドア



開き戸 引き戸

それぞれに4種類の方角の違うタイルが用意されている。方角の違いはグラフィックのみに影響し、ゲーム内における役割に違いは生じない。

プロパティ

名前	フォーマット	規定値	説明
name	文字列		OpenDoor/CloseDoor イベントで指定するのドアの識別子。 (‘イベントジェネレータ’オブジェクトを参照。)
			任意

カスタムプロパティ

名前	フォーマット	規定値	説明
open	空白でよい	-	設定すると、開いた状態が初期状態となる。
key	鍵の識別子 ;区切り・複数可	-	指定した鍵でロックする。
kickable	整数		設定すると、蹴り開けるドアになる。プレイヤーが蹴ると開く回数を指定する。
			任意

breakable	整数		設定すると、壊れるドアになる。敵に叩かれると壊れる回数を指定する。	任意
dummy	空白でよい		設定すると、開かないドアとなる。ただし、 breakable を設定した場合、プレイヤーは開けられないが、敵は壊せるドアとなる。	任意
lock	フラグ識別子		指定されたフラグがオンになっていると、ドアが開かなくなる。 (‘イベントジェネレータ’オブジェクトを参照。)	



カーテンレール

カーテンレールの種類



布付き
閉状態



布付き
開状態



布無し

それぞれに 4 種類の方角の違うタイルが用意されている。方角の違いはグラフィックのみに影響し、ゲーム内における役割に違いは生じない。

カスタムフォーマット

名前	フォーマット	規定値	説明
removable	True/False	True	布の付け替え可不可 任意



電源 (コンセント)

充電式懐中電灯 (アイテム) を充電するための電源。



給水場

ペットボトル (アイテム) を給水するための給水場。



ライトのスイッチ

light レイヤーに配置された固定ライトの電源を、遠くから ON/OFF するためのスイッチ。

カスタムプロパティ

名前	フォーマット	規定値	説明	
connect	ライトの識別子 ;区切り	-	ここで指定したライトの電源が ON/OFF される。	必須



水たまり

永続的な水たまり。この上を敵が通過すると、敵は一定時間だけ足跡を残す。



移動型ブロック

プレイヤーが移動させることのできるブロック。プロパティで移動にかかるフレーム数を指定できる。

名前	フォーマット	規定値	説明	
count	フレーム数	60	1マス動かすのに必要なフレーム数。	任意
breakable	整数	-	設定すると、壊れるブロックになる。敵に叩かれると壊れる回数を指定する。	任意



壊れるブロック

プレイヤーが破壊できるブロック。プロパティで破壊するのに必要な叩く回数を指定できる。

カスタムプロパティ

名前	フォーマット	規定値	説明
count	整数	10	プレイヤーが壊すのに必要な叩く回数。



ドキュメント

ゲームプレイを一時停止し、記述された文章を読む画面に移行する。

カスタムプロパティ

名前	フォーマット	規定値	説明	
key	文章の識別子	-	同フォルダの'文章定義ファイル'に記述した文章の識別子を指定する。	必須



イベントジェネレータ

ドアの自動的な開閉、音を鳴らす、フラグの管理などのイベントを発生させる。プロパティには、発生する条件とイベントの内容を設定する。

カスタムプロパティ

	名前	フォーマット	規定値	説明	
発生イベント	@while	Event ;区切り・複数		条件に合った毎フレーム発生させるイベント。	
	@once	同上		条件に合った場合、一度だけ発生するイベント。	
	@in	同上		条件に合ったタイミングに発生するイベント。	
	@out	同上		条件から外れたタイミングに発生するイベント。	
条件	distance	実数		距離の条件。1単位は1マスに対応する。	推奨
	lighted	空白		ライトで照らされている場合。	任意
	sight	空白		視界内にある場合。	任意
	time	秒数		この秒数が経過した場合。 (経過秒数は tmx ごと)	任意
	flag	フラグ識別子 ;区切り・複数		指定のフラグ全てがオンの場合。	任意

条件の各項目は、記述しない場合は常にオン(真)の扱いとなる。



メッセージ

画面に緑色のポップアップメッセージを表示する。チュートリアルなどの、プレイヤーに直接伝えたい内容を表示するのに利用される。

プロパティには表示する条件を指定する。全ての条件を満たしている最中にメッセージが表示される。

カスタムプロパティ

	名前	フォーマット	規定値	説明	
	key	文章の識別子	-	同フォルダの'文章定義ファイル'に記述した文章の識別子を指定する。	必須
条件	distance	実数		距離の条件。1単位は1マスに対応する。	推奨
	lighted	空白		ライトで照らされている場合。	任意
	sight	空白		視界内にある場合。	任意
	time	秒数		この秒数が経過した後。 (秒数は tmx ファイルごと)	任意
	flag	フラグ識別子 ;区切り・複数		指定のフラグ全てがオンの場合。	任意

item アイテムレイヤー

アイテムレイヤーに鍵やカメラなどを置くことで、プレイヤーがアイテムを拾わせることができるようになる。

設定方法

タイルセット **item** からタイルを選び、“タイルを追加”するモードに切り替え、**item** レイヤーにタイルを置いていく。(Ctrl キーを押しながら配置すると、セルにスナップされ綺麗に配置できる。)



タイルを追加

プロパティ (‘名前’欄に記述)

	プロパティ	説明	
	鍵	鍵番号	整数(0~4)
	懐中電灯	距離 □ 角度 □ (*)バッテリー	距離はセルの一辺の長さを1とする。 (*)バッテリーはライトの章を参照。
	布	無し	
	カメラ	無し	
	GPS	無し	
	笛	無し	

	鈴	無し	
	タイマー	無し	
	ボトル	回数	整数 (0~3) 省略した場合は 3
	生体スキャナー	無し	

カスタムプロパティ

全アイテム共通のプロパティがある。

名前	フォーマット	説明	
pos	pin 識別子 ;区切り・複数	初期位置をランダムに指定できる。指定した ピン からランダムに選ばれる。	任意
setup	(空白)	この項目を追加すると、プレイヤーがアイテムを設置したのと同じ状態を、初期状態で実現できる。例えば、鈴を object レイヤーに置いておけば、はじめから敵が通過した際に音が鳴るようになる。 注意 : 以下のアイテムにのみ有効。 懐中電灯、カメラ、GPS、鈴、タイマー、生体スキャナー	任意

レベル定義ファイル内のアイテム記述方法

記述方法

	鍵	key 〱 鍵番号
	懐中電灯	light 〱 距離 〱 角度 〱 バッテリー
	布	cloth
	カメラ	cam
	GPS	gps
	笛	whistle
	鈴	bell
	タイマー	timer
	ボトル	bottle
	生体スキャナー	scanner

種類



ノード

レベルのスタート地点、レベル間を移動するための点を指定する。

プロパティ

項目	フォーマット	説明	
名前	文字列	ノードの識別子。スタート地点や、移動先に指定するためには設定が必要だが、そうでなければ不要。	任意

カスタムプロパティ

名前	フォーマット	説明	
to	移動先 tmx 名;ノード名	指定すると、このノードから指定されたノードにプレイヤーが移動する。同 tmx ファイル内の移動も有効。	任意
type	NodeType	画面遷移の種類。次の項を参照。デフォルトは'Flat'。	任意

NodeType

識別子	Flat	Fall	Stair
説明	地続き	落下	階段
エフェクト	歩行音	落下音 画面のズームイン	階段の音



ゴール

ゴールに到達するとレベルクリアとなる。ゴールは複数設定できる。



チェックポイント

チェックポイントを通過すると、その地点からやり直すことができる。



ピン

敵の巡回地点 (**pins** プロパティ) や、敵・アイテムのランダムな初期位置 (**start** プロパティ) に指定するための基準点となる。

項目	フォーマット	説明	
名前	文字列	ピンの識別子。敵の 'pins' プロパティや、敵・アイテムの 'start' プロパティで利用される。	必須

自作レベルで遊ぶ

レベルデータの配置

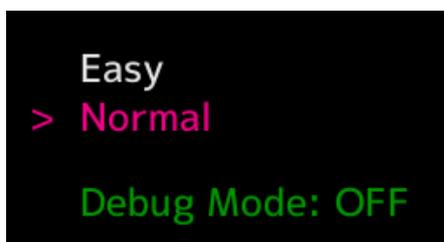
1つのレベルは1つのフォルダに格納される。そのフォルダをゲームのローカルディレクトリ以下にある、下記のパスに配置する。その際、フォルダ名がレベルの識別子となる。この識別子は半角英数字の組み合わせとする。

```
{ゲームのローカルディレクトリ}/user/{レベル識別子}/マップなどの各種データ
```

ゲーム内でのレベル選択

ゲームを起動し、タイトルから“ユーザーレベル”を選ぶ。user フォルダにあるレベルのリストが表示されるので、遊びたいレベルを選ぶ。

次に、難易度とデバッグモード ON/OFF の状態が表示される。難易度を選ぶとゲームが始まる。その際、デバッグモードを選択して ON にしておくこと、様々なテスト用の操作ができるようになる。また、デバッグモードでレベルをクリアしてもスコアは記録されない。



デバッグモードでの操作

デバッグモードを ON にしてレベルを開始すると、左上にデバッグ情報が表示される。それぞれの項目は、ON/OFF の状態を持っており、切り替え可能である。デバッグボタンを押しながら、上下ボタンで選択し、左右ボタンで状態を切り替えることができる。または、ファンクションキーでも切り替えることができる。

“F8: Virtual Check Point”は特別で、いつでもゲーム状態を保存できる。その状態からゲームを再開するには、ポーズメニューからリトライを選ばばよい。

```
* Holding Debug button ...
* Up | Down - Move the cursor
* Left | Right - Switch on & off
> F1: Show Help on
  F2: Show Info off
  F3: Check Visibility off
  F4: Through Wall off
  F5: Show Enemy off
  F6: Infinity Energy off
  F7: Light All off
  F8: Virtual Check Point
```

それぞれの項目を下の表で説明する。

項目	説明
Show Help	この情報を表示する。
Show Info	付加情報を表示する。 例えば、フラグ状態など。
Check Visibility	プレイヤーの視界を可視化する。
Through Wall	壁を通行可能にする。
Show Enemy	敵を常に表示する。
Infinity Energy	プレイヤーのライトの電力量を減らさない。
Light All	全てを照らし、可視化する。