卒業研究報告書

題目　変分オートエンコーダによるビデオゲームの

レベル生成

指導教員　飯間　等　准教授

京都工芸繊維大学　工芸科学部　情報工学課程

学生番号　１９１２２０５９

氏名　吉貞　心

令和５年２月１３日提出

変分オートエンコーダによるビデオゲームのレベル生成

令和５年２月１３日　　19122059　吉貞　心

概　　要

ゲームに関する研究は,最高得点を獲得する手法については盛んに行われている一方,ゲームコンテンツを生成する手法については前者ほど盛んではない．しかし，近年のゲーム市場は拡大する一方で，ゲーム開発にかかる時間とコストは年々大きくなっている．本論文では，変分オートエンコーダ(VAE)を活用して，少数だけ用意されたゲームレベルを学習し，大量かつ多様なゲームレベルを生成する手法を提案する.学習用ゲームレベルをそのまま VAE に学習させると，情報量が多いためにゲームの仕様要件を満たさない画像が生成される恐れがある．そのため，学習用ゲームレベルを構成するオブジェクト毎に 0，1 の 2 次元 2 値データに変換することで，必要な情報量を削減する．また，生成したゲームレベルが実際にゲームとしてプレイできるよう，プレイ可能化器を導入することで,ゲームの仕様要件を満たすことを保証する．そのようにして生成したゲームレベルを，再度 VAE への学習用データとして採用することで,ゲームレベルが少数しか用意できない問題を解決する.提案手法を用いてゲームレベルを生成する実験を行った結果,より多様なゲームレベルを生成することに成功し,提案手法が有効であることが確認された.

目　次

1. はじめに・・・・・・・・・・・・・・・・・・・・・・・・・・・・1

2. 変分オートエンコーダ(VAE)・・・・・・・・・・・・・・・・・・・・・・・3

3. 提案手法・・・・・・・・・・・・・・・・・・・・・・・・・・・6

4. 実験・・・・・・・・・・・・・・・・・・・・・・・・・・・・12

4.1 学習法の設定・・・・・・・・・・・・・・・・・・・・・・・・12

4.2 プレイ可能化器・・・・・・・・・・・・・・・・・・・・・・・12

4.3 評価方法・・・・・・・・・・・・・・・・・・・・・・・・・・16

4.4 結果・・・・・・・・・・・・・・・・・・・・・・・・・・・・16

4.5 考察・・・・・・・・・・・・・・・・・・・・・・・・・・・・20

5. 結論・・・・・・・・・・・・・・・・・・・・・・・・・・・・22

謝辞・・・・・・・・・・・・・・・・・・・・・・・・・・・・・22

参考文献・・・・・・・・・・・・・・・・・・・・・・・・・・・23

1．はじめに

ゲームを対象とした人工知能に関する研究が活発に行われているが，その多くがゲームの最高得点を獲得することを目標にしていることが多い．そのような研究により，ゲームを素早くかつ効率的に攻略するための手法が数多く考案されてきた．一方で，ゲームのコンテンツを生成するような研究はそれほど盛んには行われていないのが現状である．

General Video Game-Artificial Intelligence(GVGAI)はゲーム用の強化学習環境で，ビデオゲーム記述言語(VGDL)によって記述された，OpenAI Gym[1]環境の一つである．GVGAI には，Space Invaders や Zelda など，多数の 2D ゲームのフレームワークが含まれているが，いずれのゲームについてもゲームレベルは 5 つしか設定されていない．GVGAI に限らず，一般に新たにゲームレベルを人の手で制作するのは時間もコストもかかる．そこで，既存の少数のゲームレベルから，機械学習を用いて多数のゲームレベルを生成することができれば，時間もコストも削減できることが期待できる．

Torrado らはこの課題に対して，敵対的生成ネットワーク(GAN)[1]を応用した条件付き畳み込み自己注意 GAN(CESAGAN)を用いることを提案した[2]．CESAGAN を用いると，生成したゲームレベルの多様性が通常の GAN を用いた場合よりも増加したと報告している．GAN と同様に，データを生成する他の深層学習モデルとして，オートエンコーダや変分オートエンコーダ[3]が知られている．

本論文では，VAE を用いて多様なゲームレベルを生成する方法を提案する．VAE の大きな特徴として，学習データの次元を削減し，正規分布に従って潜在空間を生成し，データを復元する点が挙げられる．これを活用すれば，より多様なゲームレベルを生成することができると考えられる．

しかし，GAN も VAE も大抵は画像に対して用いられることが多く，これをゲームコンテンツに応用するには解決すべき課題点がある．1 つ目の課題

[1] E.R. Musk らが率いる人工知能を研究する非営利団体である「OpenAI」が提供しているプラットフォームである．強化学習のためのいくつかの環境(ゲーム)が用意されている．

点は，上の GVGAI でも述べたように，ゲームコンテンツのデータ数が少ない点である．GAN や VAE の学習には大量の画像データが用いられることが一般的であるが，ゲームコンテンツではそれほど大量に用意することが難しい．2 つ目の課題点は，ゲームコンテンツには満たさなければならない要件が存在する点である．人間の顔の場合，多少片目がかすれてしまっていても人間の顔であると認識できる．一方ゲームコンテンツの場合では，要件に少しでも違反するとゲームが正しく動作することができない．例えばスーパーマリオブラザーズ(任天堂，1985 年)では，画面内にマリオが複数人いたり，土管が空中に浮かんでいたりするゲームレベルは仕様要件を満たしていないため，生成したゲームレベルは全く意味を持たないものとなる．

本論文ではこれらの課題について，VAE による学習によりゲームの仕様要件を満たすゲームレベルを生成する．また，Torrado らが提案した CESAGAN では生成したゲームレベルを再度学習用ゲームレベルとして採用する手法を活用することで，ゲームレベルが少数で学習が困難だという問題を解決している．そこで，提案手法においても，この手法を導入する．学習用ゲームレベル数が増えることで，より多様なゲームレベルの生成ができることも同時に期待することができる．

２．変分オートエンコーダ(VAE)

VAE はオートエンコーダと同様にニューラルネットワークの一種で，画像などの情報を圧縮して符号化したものを潜在変数とし，その潜在変数から元の情報を復元するモデルである．特に VAE は潜在変数を正規分布と仮定した生成モデルとなっている．

VAE は一般的に図 2.1(a)に示すように，入力層と隠れ層と出力層の 3 層が接続された構造を持つ．モデルは入力データXから正規分布の平均と分散を求めるエンコーダと，平均と分散を基にした正規分布からランダムにサンプリングしたデータを用いて元の入力データXを再構成するデコーダで構成される．VAE は出力が入力に等しくなるように学習を行うが，単にランダムなサンプリングを行うと，誤差逆伝播法を活用できなくなり，ネットワークの学習を行うことができない．

そこで，Reparametrization Trick という技法が用いられる．この技法は，デコーダに入力するベクトルzをランダムにサンプリングすることで求めるのではなく，分散σに非常に小さな乱数ϵをかけて平均μを加算することによって近似して求める技法であり，次式で表現される．

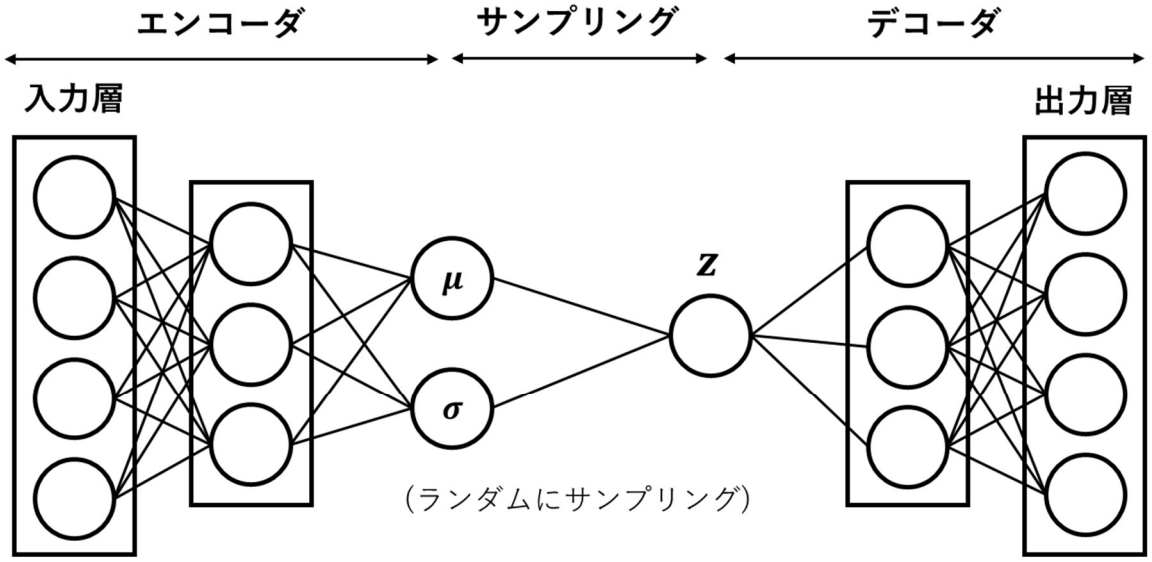
$$z = \mu + \epsilon\sigma \tag{1}$$

この技法により，VAE は図 2.2 に示すような構造となり，誤差逆伝播法を活用できるようになるため，ネットワークの学習を行うことができる．

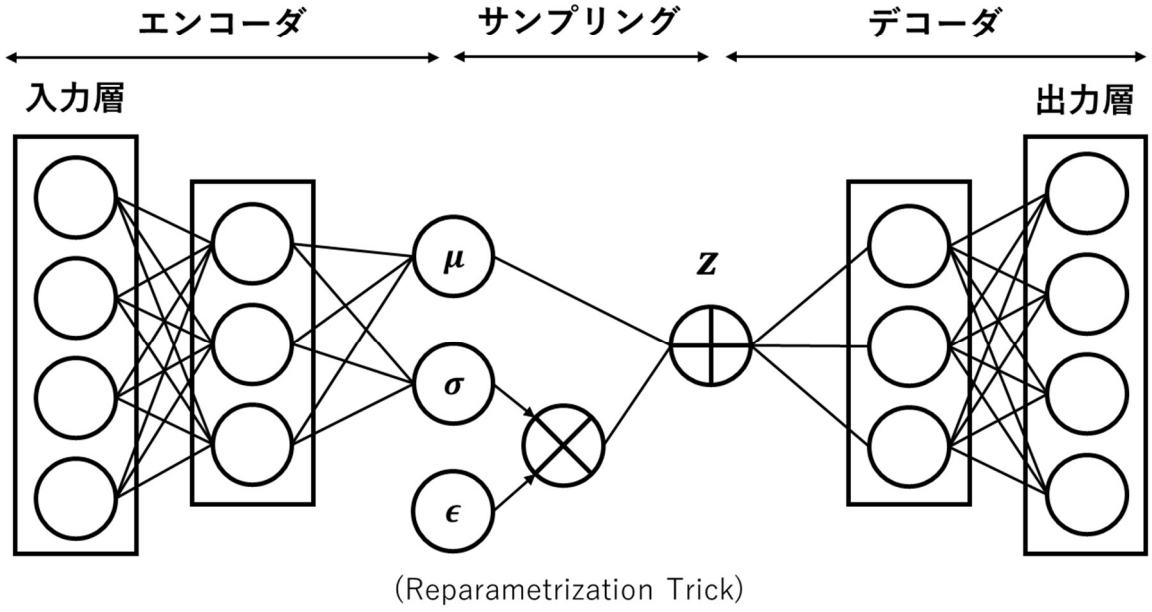
VAE の損失関数Lは変分下限と呼ばれ，潜在変数z，エンコーダ$q(z|X)$，デコーダ$p(X|z)$を用いて次式で表現される．

$$L = E_{q(z|X)}[\log p(X|z)] - D_{KL}[q(z|X)||p(z)] \tag{2}$$

第 1 項は再構築誤差で，$q(z|X)$に対する入力Xの対数尤度$\log p(X|z)$の期待値を最大化する．つまり，デコーダの出力が元の入力にどれだけ近いかを表す．第 2 項は正規化項で，D_{KL}はカルバックライブラー情報量である．$D_{KL} \geq 0$であり，$q(z|X)$と$p(X)$が近いほどD_{KL}は 0 に近づく．ここで，第 1 項は煩雑な積分計算が影響するため，計算量が膨大になりやすい問題がある．これを回避するために，デコーダの出力yと元の入力Xとの二乗誤差に置き換える．また，第 2 項は$p(z)$が標準正規分布であるとき，平均μと分散σを用いて近似値を適



(a) 一般的なモデル



(b) Reparametrization Trickを用いたモデル

図 2.1 変分オートエンコーダ

用することができる．これらを踏まえると，損失関数Lは次式のように書き換えることができる．ただし，Nはエンコーダの入力Xの次元数，Dは潜在変数の次元数である．

$$L = N\|y - X\|^2 - \frac{1}{2}\sum_{d=1}^{D}(1 + \log\sigma_d^2 - \mu_d^2 - \sigma_d^2) \tag{3}$$

この損失関数を用いることにより，計算量を抑えて学習を進めることが可能になる．

3．提案手法

VAE を用いて多様なゲームレベルを生成する手法を提案する．このためには，以下に示す 3 点に留意する必要がある．

1. 冗長な情報の削減
2. 品質の保証
3. 少数のゲームレベルから多様なゲームレベルの生成

これらの 3 点を，GVGAI に含まれる Zelda を例として説明する．Zelda のゲームレベルは図 3.1 に示すようなものとなる．このゲームは，剣を持ったプレイヤーが四方を壁に囲まれたダンジョンを探索し，ダンジョン内を動き回る敵(図では 3 体いる)に倒されないようにしながら鍵を入手してゴール地点である扉への到達を目指すゲームである．また，プレイヤーは剣を使って敵を倒すこともできる．このようなゲームレベル画像をそのまま学習に活用した場合，壁やプレイヤーの形状が全くそのまま再現される可能性は極めて低くなる．従って，ゲームレベルの冗長な情報を適切に削減する必要がある．また，プレイヤーや扉が壁に囲まれてゴールすることができないなど，ゲームの仕様要件を満たさないゲームレベルを生成する恐れもある．このような事態が生じないように，ゲームの品質を保証する必要がある．さらに，ゲームレベルは一般的に大量に用意することが難しい．そのため，VAE で学習する際の学習用ゲームレベル数を十分に確保することができない．従って，少数のゲームレベルから多様なゲームレベルを生成できるようにする必要がある．

以上の留意点を踏まえ，提案手法のフレームワークを図 3.2 に示す．第 1 の点に関しては，壁やプレイヤーなどの各オブジェクトの位置を，オブジェクト毎に 0，1 の 2 次元データで表したものを VAE への入力Xとする．例えば，GVGAI に含まれるゲームのゲームレベルはテキストデータで管理されており，ゲームレベルはn種類のオブジェクトがi行j列の長方形状に配置されている． Zelda には 8 種類のオブジェクトが 9 行 13 列に配置されており，表 3.1 に示すオブジェクトとシンボルとなるテキストの対応をもとにオブジェクトを配置すると，図 3.1 のようなゲームレベルが生成される．この

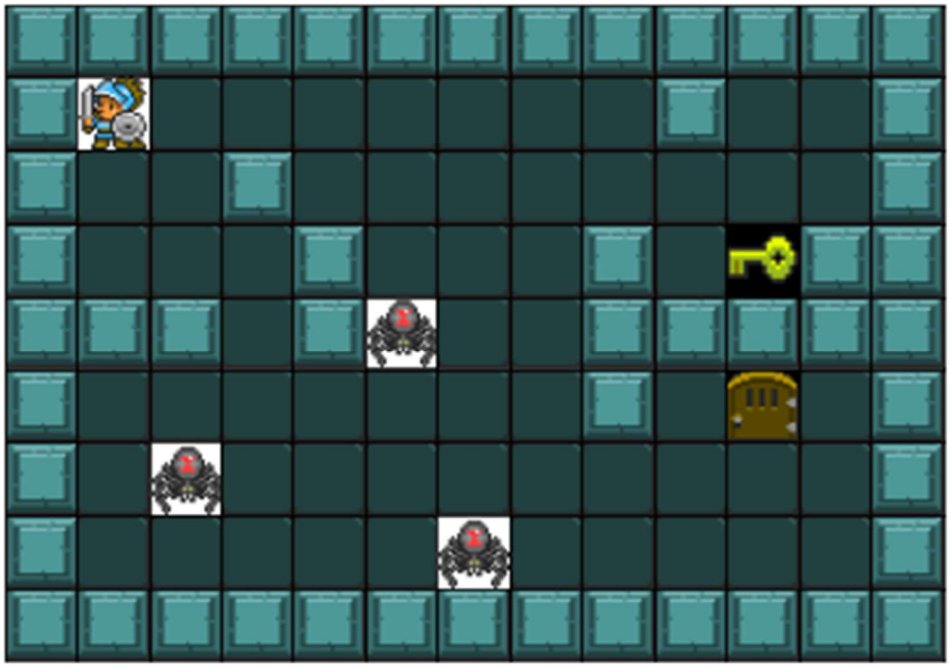


図 3.1 Zelda のゲームレベルの例

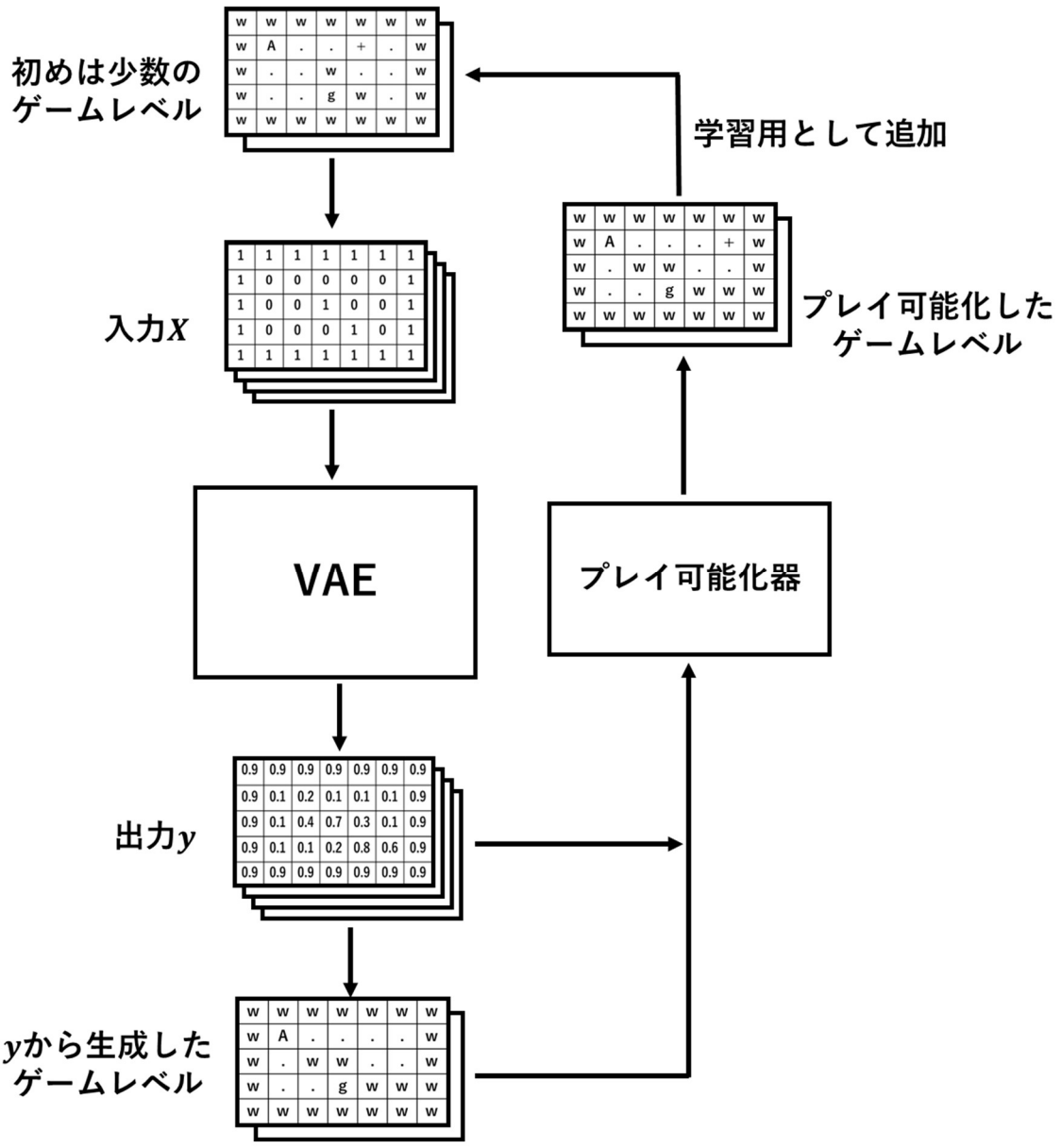


図 3.2 提案手法のフレームワーク

表 3.1 オブジェクトとシンボルの対応

オブジェクト名	オブジェクト	シンボル
壁		w
床		.
鍵		+
扉		g
敵1		1
敵2		2
敵3		3
プレイヤー		A

テキストデータに対して，例えば壁についてのデータは，'w'の位置を 1 で表し，それ以外の位置を 0 で表したi行j列の 2 次元データとなる．図 3.2 の入力Xでは，壁のオブジェクト'w'についての 2 次元データの例を示している．このような変換を各オブジェクトに対して行うと，1 つのゲームレベルのテキストデータをn種類のi行j列の 2 次元 2 値データに変換することができ，元の冗長な情報を含むデータよりも情報量を削減することが可能になる．Zelda の場合は 1 つのゲームレベルが 8 種類の 9 行 13 列の 2 次元 2 値データとなり，これが VAE への入力Xとなる．

第 2 の点に関しては，プレイ可能化器を導入する．VAE の出力は，図 3.2 に示すようにn種類のオブジェクトそれぞれに対してi行j列の 0〜1 の実数値となる．ここからゲームレベルを生成するには，各行各列毎にすべてのオブジェクトの出力値を比較し，最も数値の大きいオブジェクトをゲームレベルに出現させる．例えば図 3.2 の場合は，壁オブジェクトの出力の 1 行 1 列目の数値 0.9 が他のオブジェクトの同じ位置の数値より大きくなっていたことから，生成したゲームレベルの 1 行 1 列目は'w'となる．他の位置についても同様に，その位置での出力値が最大のオブジェクトを出現させることで，ゲームレベルを生成することができる．しかし図 3.2 の生成されたゲームレベルをよく見ると，鍵のオブジェクトが生成できていないことがわかる．このように，VAE の出力のみではプレイ可能なゲームレベルが生成できない事態が生じる可能性が高い．こうしたゲームレベルをプレイ可能にするために，プレイ可能化器を活用する．プレイ可能化器は，VAE からの出力yと，yから上に述べた方法で生成したゲームレベルを入力とし，プレイ可能化したゲームレベルを出力する．これにより，生成されたゲームレベルはゲームの仕様要件を満たすことが保証される．図 3.2 では，鍵のオブジェクトが生成されていることが確認でき，プレイ可能なゲームレベルとなっている．ただしプレイ可能化器は，例えば Zelda であればプレイヤーは 1 人しか存在しない，プレイヤーの初期位置から鍵の位置や扉の位置までは必ず到達できる，といったゲームの仕様要件に関わる内容を補正するための機能である．そのため，四方を壁で囲めていない，壁しか存在しないなど，プレイ可能化することが困難である場合は，そのゲームレベルを破棄する．なお，プレイ可能

化手法はゲームに合わせて適切に設定する.

第 3 の点に関しては，プレイ可能化器を用いてプレイ可能となったゲームレベルを，学習用ゲームレベルとして追加し，VAE を初期化した上でそれらの学習用ゲームレベルを用いて再度 VAE で学習し，プレイ可能化器でプレイ可能化したゲームレベルを再度学習用ゲームレベルに追加する，という一連の流れを繰り返す．初期の学習用ゲームレベルの個数をN個，生成用ゲームレベルの個数をM個とした場合，1 回の VAE での学習で生成できるゲームレベルはM個となり，プレイ可能化器を通して破棄されるゲームレベルを無視すれば，学習用ゲームレベル数はk回の学習後に$N + kM$個となる．このように，学習用ゲームレベルの数が増加することでより多様なゲームレベルの生成が見込まれる．

4．実験

本章では，第 3 章で提案した手法を用いてゲームレベルを生成する実験を行った結果を示し，提案手法の性能を評価する．

4.1 学習法の設定

学習用ゲームレベルのデータセットには， GVGAI に付属している Zelda の$N=5$ つのゲームレベルを使用する．これらを図 4.1 に示す．また，提案手法で用いる VAE のパラメータは表 4.1 の通りである．第 3 章で述べたように，VAE の学習結果として得られるゲームレベルも学習用として用いるため，データセットの数は最初の 5 つから VAE での学習毎に増加する．ところが, データセットの個数が膨大になると学習に時間がかかり過ぎるので, 本実験では学習用ゲームレベルが 5000 個を超えて以降は学習用ゲームベレルの中から 5000 個をランダムに選択し，それらのみを VAE の学習に用いることとする．VAE の学習後にゲームレベルを生成するが，このときに用いる生成用ゲームレベルにも図 4.1 の$M=5$ つのゲームレベルを用い，学習用ゲームレベルが 5000 個を超えて以降は選択した 5000 個の学習用ゲームレベルからランダムに選択した 5 つを生成用ゲームレベルとする．この学習を学習用ゲームレベルが 15000 個になるまで繰り返し行い，この間に生成したすべてのゲームレベルを評価の対象とする．

4.2 プレイ可能化器

プレイ可能化器は，第 3 章で述べたように VAE の出力yと，yから再構成したゲームレベルの 2 つを入力とする．Zelda に合わせたプレイ可能化器のプレイ可能化項目と手法を示す．まずプレイ可能化項目を以下に列挙する．

1. プレイヤーは一人だけ存在する
2. 鍵は一つだけ存在する
3. 扉は一つだけ存在する

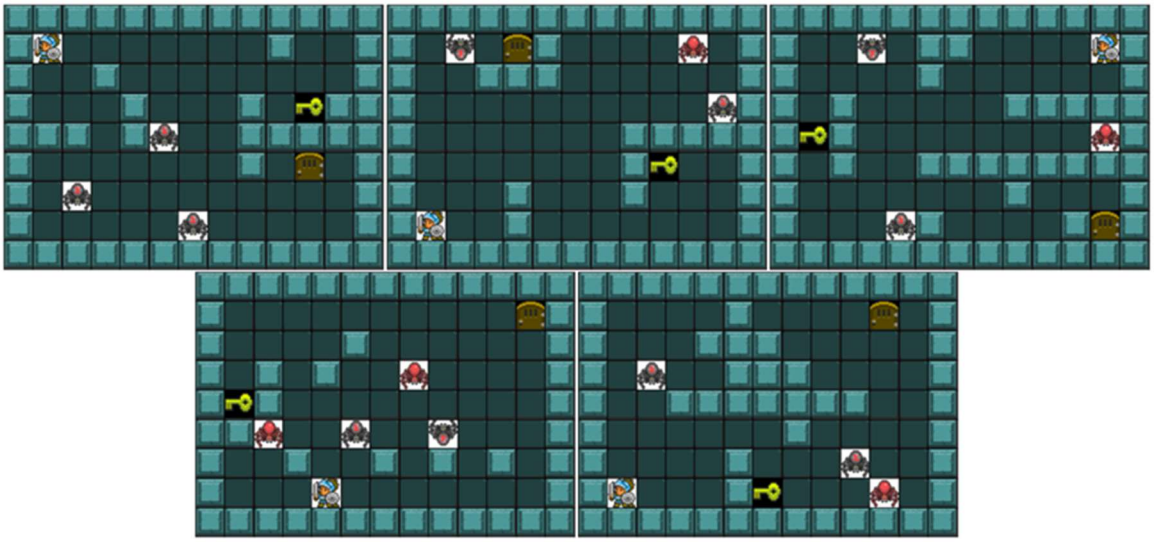


図 4.1 Zelda の 5 つのゲームレベル

表 4.1 VAE の設定

入力層の次元数	8×9×13
層の種類(すべて共通)	全結合層
エンコーダの隠れ層の数	2
エンコーダの1つ目の隠れ層の次元数	64
エンコーダの2つ目の隠れ層の次元数	32
エンコーダの隠れ層の活性化関数	ReLU関数
潜在変数zの次元数	2
デコーダの隠れ層の数	2
デコーダの1つ目の隠れ層の次元数	32
デコーダの2つ目の隠れ層の次元数	64
デコーダの隠れ層の活性化関数	ReLU関数
バッチサイズ	32
出力層の活性化関数	sigmoid関数
エポック数	50

4. 敵の数は床の数に対して 60%未満となるようにする
5. ゲームレベルは最も外側のマスが必ず壁でなければならない
6. プレイヤーは鍵の位置まで到達可能である
7. プレイヤーは扉の位置まで到達可能である

次に，これらの項目に対するプレイ可能化手法を示す．第 1〜第 3 項目については，入力したゲームレベルにプレイヤー，鍵，扉がそれぞれ一つずつだけ含まれているかを調べる．もしそれぞれ一つも存在していなければ VAE の出力yを用いてプレイヤー，鍵，扉の最も出現しそうな位置，すなわち該当オブジェクトのi行j列の 2 次元出力値の中で最大出力値を持つ位置に，またはその位置の四方の壁を除く上下左右の隣接する位置に一つオブジェクトを追加する．逆に二つ以上存在している場合は，その二つ以上ある位置のうち一つをランダムに選択し，その位置以外を床に変更する．第 4 項目については，入力したゲームレベルのすべての敵の数と床の数を数え，敵の数が床の数に対して 60%以上となっている場合は，そのゲームレベルを破棄する．逆に，3 種類の敵それぞれに対して，その敵が存在しない場合，$\frac{1}{4}$の確率でプレイヤー，鍵，扉の位置を除くランダムな位置にその敵を追加する操作を行う．敵のオブジェクトは 3 種類存在するため，最大 3 体の敵が追加されることになる．第 5 項目については，入力したゲームレベルの最も外側のマスを調べ，一つでも壁でないマスが存在すれば，そのゲームレベルを破棄する．第 6 項目および第 7 項目については，入力したゲームレベルについて A*アルゴリズムを用いて，プレイヤーの位置から鍵の位置，およびプレイヤーの位置から扉の位置まで到達可能か調べ，到達不可能であれば，そのゲームレベルを破棄する．

上記に加えて，壁の出現パターンを増やすために，壁の総数が四方に位置する壁を含んで 55 個未満の場合に，ランダムな位置を 10 箇所選び，その位置のオブジェクトが床であれば壁に変更する操作を導入する．すなわち，最大で 10 箇所の床が壁に変更される．ただし，床を壁に変更しないゲームレベルも多様性に貢献するので，この操作は$\frac{1}{2}$の確率で実行する．ここでの 55 という数字は，GVGAI 付属の 5 個のゲームレベルにおける壁の平均個数が約 54 であったことから設定している．

4.3 評価方法

提案手法の性能を評価するために，生成したゲームレベルの重複率を算出し，Torrado らの CESAGAN[2]で生成したゲームレベルの重複率と比較する．重複率pを「生成したゲームレベルのうち，既に生成したゲームレベルと重複したものの割合」と定義する．生成したゲームレベルの総数をN_g，重複したゲームレベルの種類数をN_dとすると，重複率pは次式のように表現される．

$$p = \frac{N_d}{N_g} \tag{4}$$

重複率が小さいほど多様なゲームレベルを生成したことを表す．

また，提案手法を用いて生成したゲームレベルの各オブジェクトについて，そのオブジェクトが各位置に出現した頻度をヒートマップに表して可視化することにより，生成したゲームレベルの多様さが増しているかどうかを確認する．

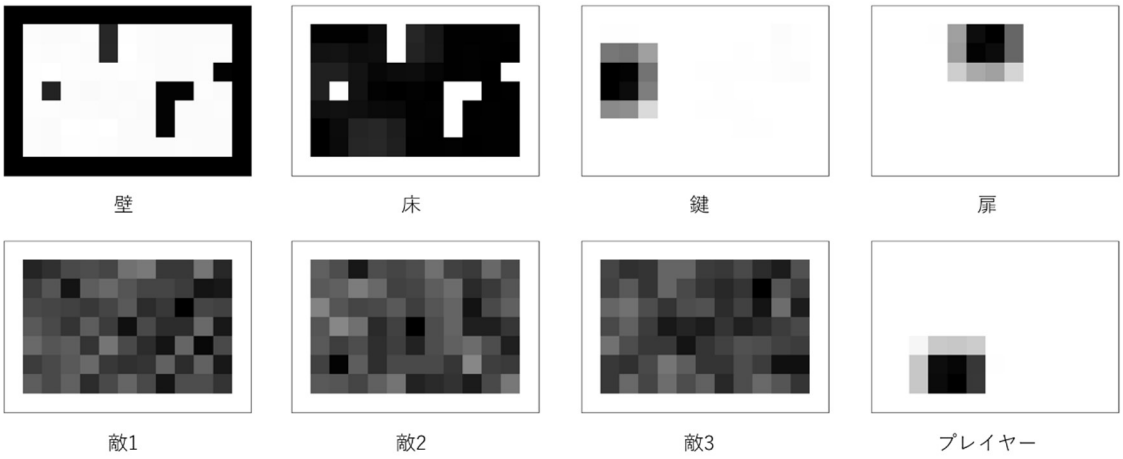
4.4 結果

提案手法を用いて生成したゲームレベルと，Torrado らの CESAGAN を用いて生成したゲームレベルの重複率を表 4.2 に示す．提案手法の重複率の方が小さいことがわかる．

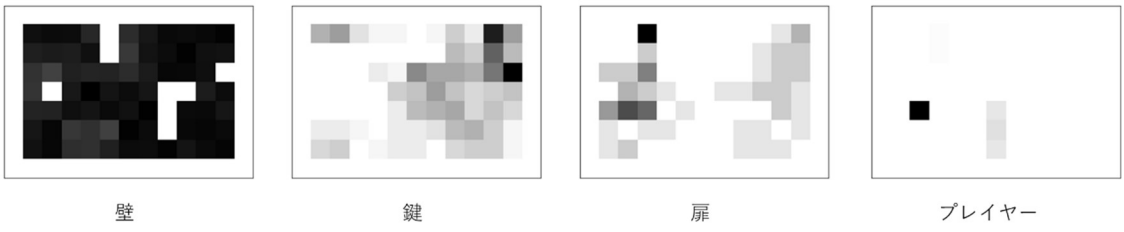
また，各オブジェクトのヒートマップを図 4.2(a)に示す．このヒートマップは，生成したゲームレベルの各オブジェクトについて 9 行 13 列の各位置での出現数を表したもので，黒い方が多く出現したことを示している．図より，壁，鍵，扉，プレイヤーの出現位置に偏りが生じていることがわかる．また，図 4.2(a)の壁，鍵，扉，プレイヤーのヒートマップに対して，出現頻度が大きい位置を除き，残りの位置の出現数に応じて濃淡をつけ直したヒートマップを図 4.2(b)に示す．壁，鍵，扉については出現頻度が大きい位置以外でもある程度万遍なく出現していることがわかる．一方プレイヤーのヒートマップでは，出現場所の偏りが依然存在すると考えられる．

表 4.2 重複率の比較

学習モデル	重複率(%)
CESAGAN	57
提案手法	14



(a) オブジェクト毎のヒートマップ



(b) 値の大きい位置を除いたヒートマップ(一部オブジェクト)

図 4.2 生成したゲームレベルにおける各オブジェクトの出現数

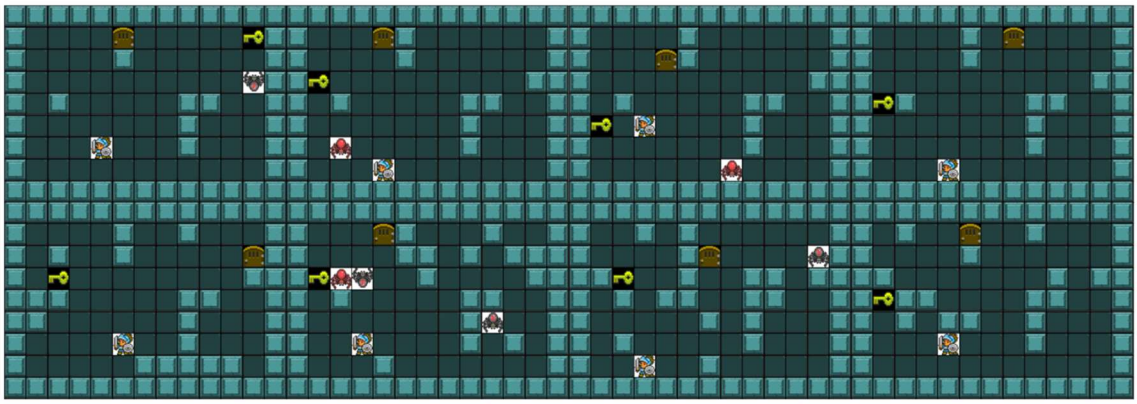


図 4.3 提案手法で生成されたゲームレベルの例(8 種類)

さらに，実際に生成したゲームレベルの例を図 4.3 に示す．いずれのゲームレベルについても，図 4.2(a)の壁についてのヒートマップで黒く現れている形状が確認できる．また，図 4.3 の下段のゲームレベルでは，壁の個数が増えたことで上段のゲームレベルよりも複雑さが増していることがわかる．このように，オブジェクトの配置が変化したり複雑さが増加したりしたゲームレベルが多数確認できる．

4.5 考察

提案手法の重複率は，CESAGAN より 43%小さくすることができている．この点で，提案手法ではより多様なゲームレベルを生成することができていると言える．この成果はプレイ可能化器による影響が強いと考えられる．Torrado らは CESAGAN から出力されたゲームレベルについて重複率を調査しており，提案手法では VAE からの出力を用いて生成したゲームレベルに，さらにプレイ可能化器で補正をかけたゲームレベルに対して重複率を調査している．特に床を壁に変更する場合には，位置をランダムに 10 箇所選んで変更していたことが，重複率が大きく減少したことの原因であると推察される．

また，ヒートマップについては，図 4.3 に示すゲームレベルには図 4.2 のヒートマップ特徴がよく現れている．特に図 4.2(a)の特徴はどのゲームレベルにも多く見られる特徴となっており，VAE による復元が適切に行えていることが窺える．一方で，敵についてはどの位置においてもほぼ同数で生成されており，プレイ可能化器による貢献が特に大きくなっている点であると考えられる．また，ゲームレベルの多様性については図 4.2(b)のヒートマップに示す通り，最も多く出現した位置以外においてもある程度ランダムに出現していることがわかることから，生成するゲームレベルを多様化できていると考えられる．特に鍵と扉については，ゲームレベルのほぼ全体に出現した例があることから，ゲームレベルの多様化ができていると推測できる．一方で，プレイヤーの出現位置は，出現頻度が高い部分を除いても出現位置が

偏った結果となっている．これは，GVGAI 付属の 5 個のゲームレベルのプレイヤーの位置が左の方に偏っていたためであると考えられる．

５．結論

本論文では，VAE を用いて少数のゲームレベルから多様なゲームレベルを生成する手法について提案した．提案手法では，VAE への入力であるゲームレベルをオブジェクト毎に分けて学習し，プレイ可能化器を通じて生成したゲームレベルを学習用データとして活用することで，学習用ゲームレベルが少数である問題を解消している．実験の結果，提案手法により多様なゲームレベルを生成することに成功した．一方で，この成功はプレイ可能化器による貢献が大きすぎるようにも考えられる．

今後の課題として，Zelda 以外のゲームに対しても，プレイ可能化器を適宜設定することで同様にゲームレベルを生成することができるかどうかを検証することが挙げられる．そして，プレイ可能化器を介すること無くプレイ可能であるようなゲームレベルを生成できるような改良方法を開発する必要がある．

謝辞

本実験を行うにあたり，研究課題の設定や研究に対する姿勢，本報告書の作成に至るまで，すべての面で丁寧なご指導を頂きました，本学情報工学・人間科学系飯間等准教授に厚く御礼申し上げます．本報告書執筆にあたり貴重な助言を多数頂きました，情報知能システム研究室の皆様，学生生活を通じて著者の支えとなった家族や有人に深く感謝致します．

参考文献

[1]I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D.Warde-Farley, S. Ozair, A. Courville, Y. Bengio, “Generative Adversarial Networks, “ Advances in Neural Information Processing Systems, 2014.

[2]R.R. Torrado, A. Khalifa, M.C. Green, N. Justesen, S. Risi, J. Togelius, “Bootstrapping Conditional GANs for Video Game Level Generation, ” Proceedings of 2020 IEEE Conference on Games, pp.41-48, Osaka, Japan, August, 2020.

[3]D.P. Kingma, M. Welling, “Auto-Encoding Variational Bayes, ” Processings of International Conference on Learning Representations, 2014.