

条件付き変分オートエンコーダを用いたビデオゲームのレベル生成

京都工芸繊維大学 吉貞 心(6Q-02)・飯間 等

背景

- 近年のゲーム開発は大規模化 & 複雑化が進んでいる
 - 機械による開発支援が必須化している
- 近年生成AIによるコンテンツ生成が流行している
 - ゲームコンテンツにも応用することで、開発への直接的な支援に繋がると期待されている

先行研究

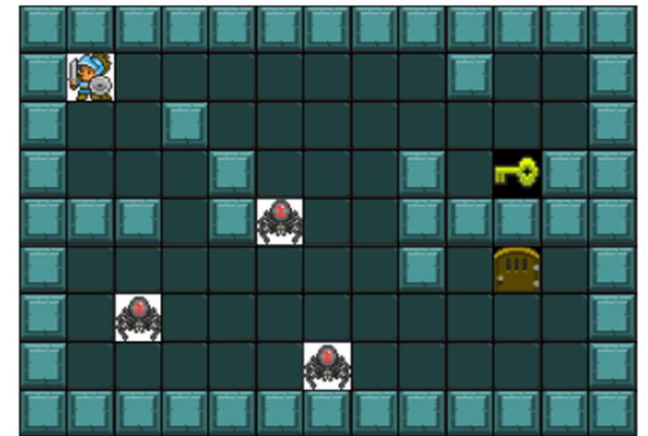
- 敵対的生成ネットワーク(GAN)による生成[1]
 - 学習・生成したゲームレベルが本物かどうか検知&学習する
 - 生成したゲームレベルは開発者の意図通りのものとは限らない
- Mario GPTによる生成[2]
 - 開発者が自然言語によって要求を指定できる言語処理ベースの手法
 - 使用できる言葉(little, manyなど)が限られ, 生成されるゲームレベルも限定的

[1] 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.

[2] S. Sudhakaran, M. Gonzalez-Duque, C. Glanois, M. Freiburger, E. Najarro, and S. Risi, "MarioGPT: Open-Ended Text2Level Generation through Large Language Models," Proceedings of the 37th Annual Conference on Neural Information Processing Systems, 2023.

目的

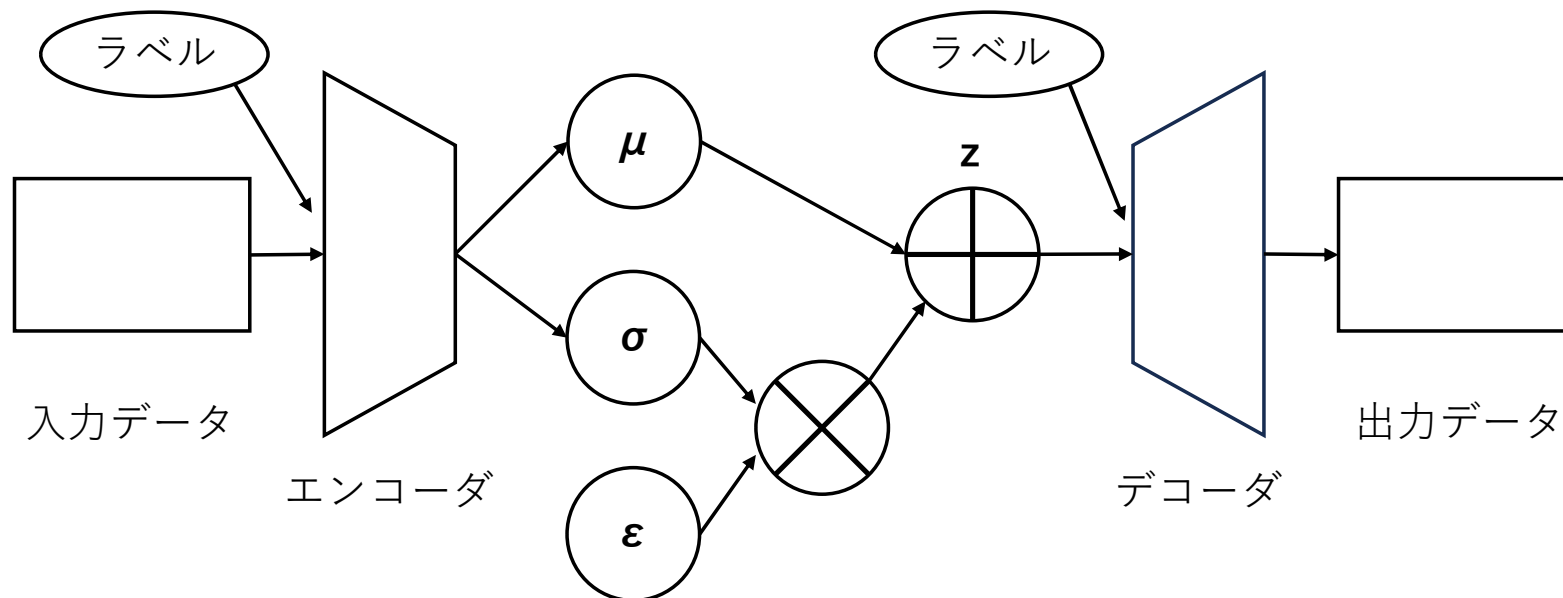
- 少数のゲームレベルから開発者の要求通りのゲームレベルを生成する
 - ゲーム開発初期段階では学習に使えるゲームレベルは少数しか準備できない
 - 開発初期のうちから要求を満たすゲームレベルを制作できれば、コンテンツを生成する点だけでなくデバッグ対象の早期発見にも繋がる
- 開発時間とコストの面で改善が期待できるのではないか



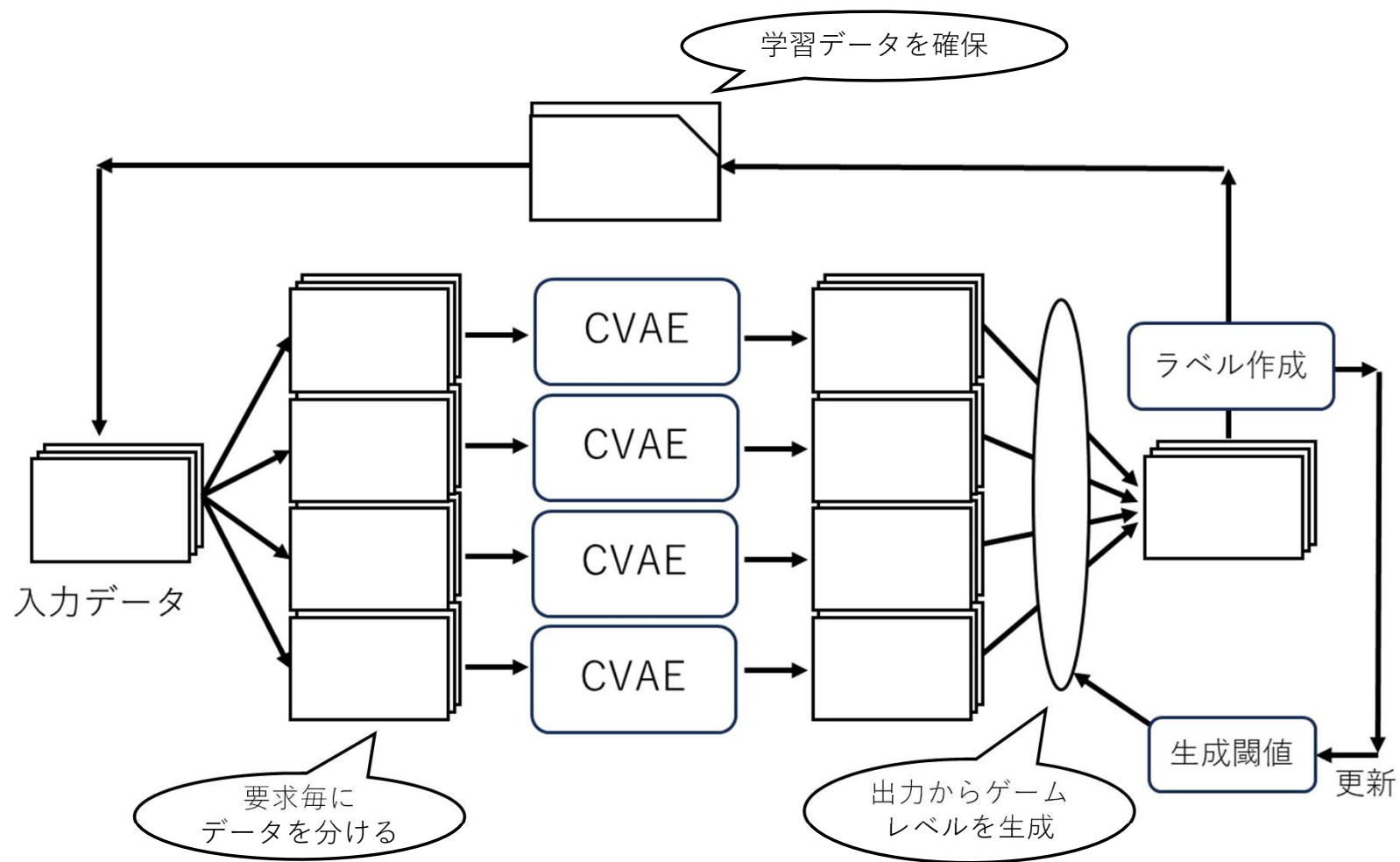
ゲームレベルの例 4

提案手法

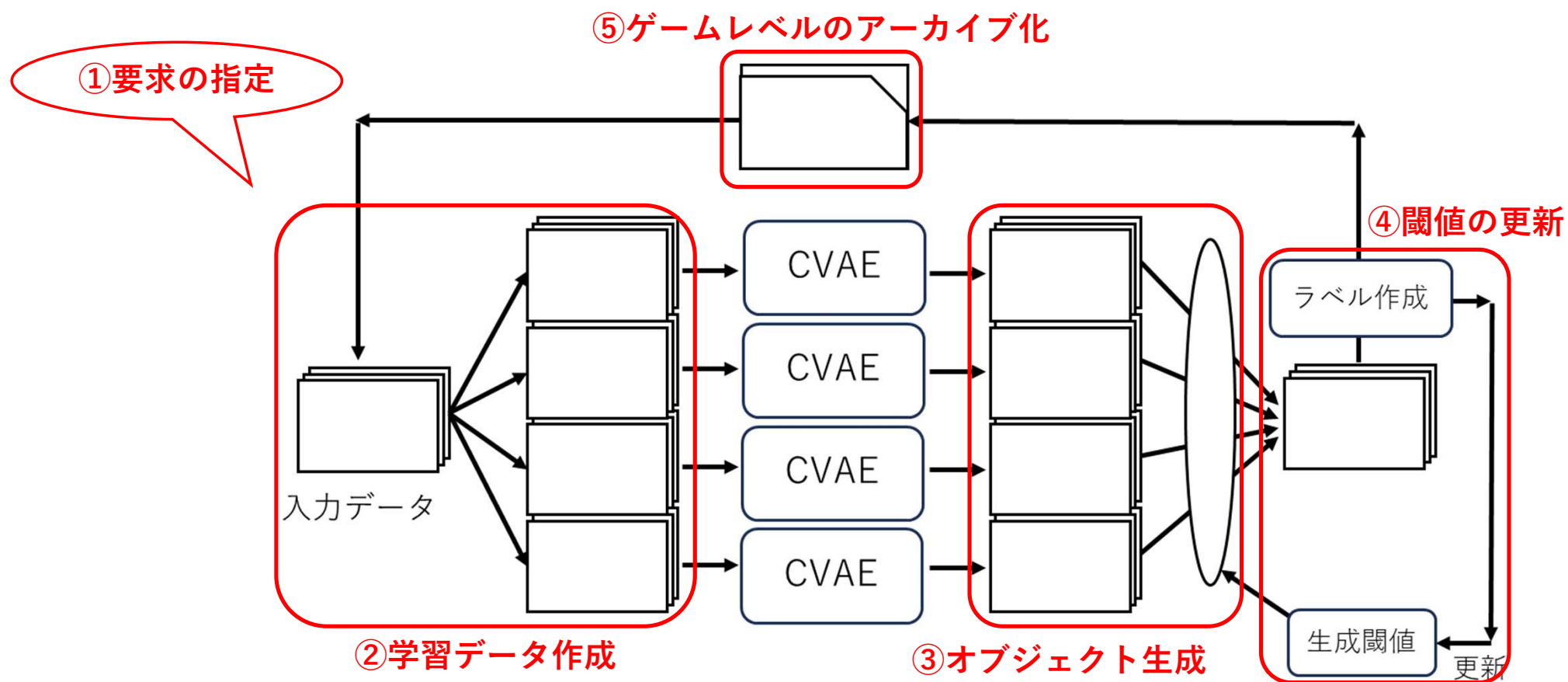
- 条件付き変分オートエンコーダ(Conditional Variational Autoencoder : CVAE)を使用
 - 生成モデルの1種



提案手法



提案手法



提案手法 ①要求の指定

- 今回想定する要求は4項目

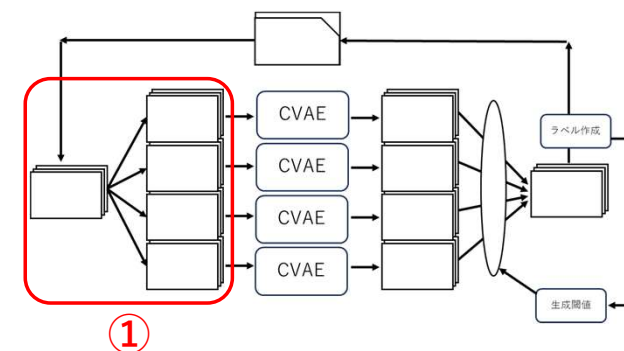
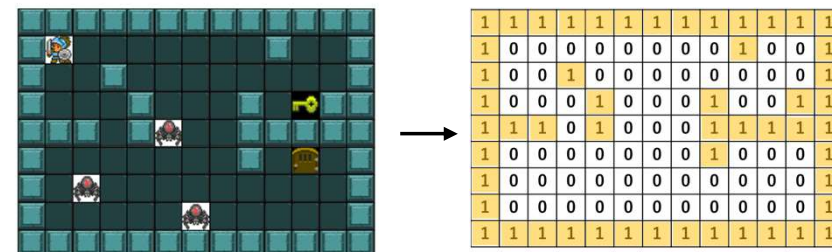
要求項目	関連するオブジェクト
壁の数	壁
敵の数	敵
アバターと鍵との距離	アバター, 鍵, 床
アバターと扉との距離	アバター, 扉, 床



- 要求は0～1の実数値で指定
 - 0に近いほど生成する数/距離は小さく, 1に近いほど生成する数/距離は大きくなる
 - 柔軟な要求を実現している

提案手法 ①学習データ作成

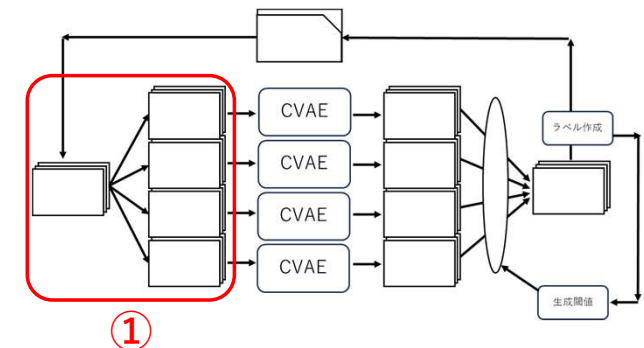
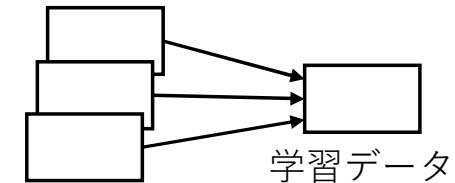
- ゲームレベルは情報量が多い
 - そのまま学習させるとオブジェクトの形状など正しく復元できない恐れがある
- オブジェクト毎に0,1データを生成
 - 冗長な情報を削減



提案手法 ①学習データ作成

- 開発者の要求に合わせた学習データを作成
 - 例：壁の数を増やしたい
 - 学習データに壁のオブジェクト情報をもつ0,1データを作成
- 学習は要求項目毎に実行
 - 学習データとラベルの情報が1対1対応
 - 要求別の学習データ間に相関は無いため学習器を分けている

要求項目	関連するオブジェクト
壁の数	壁
敵の数	敵
アバターと鍵との距離	アバター, 鍵, 床
アバターと扉との距離	アバター, 扉, 床

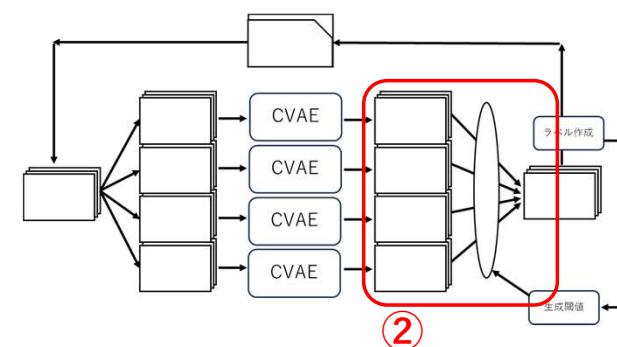


提案手法 ②オブジェクト生成

- 出力の実数値データを0,1データに変換
 - ゲームレベルにおいて実数値は意味をもたない
 - オブジェクトが存在するかどうかを示す0,1データに変換する
- 閾値を定めて実数値データとの比較でオブジェクトを生成

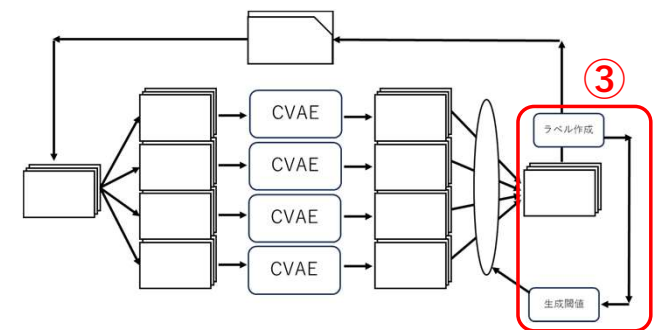
例
閾値0.45

0.6	0.7	0.6	0.8	0.9	1	1	1	1	1
0.5	0.4	0.2	0.3	0.7	1	0	0	0	1
0.8	0.6	0.7	0.9	0.9	1	1	1	1	1



提案手法 ③閾値の更新

- 閾値を動的に更新することで要求を満たすゲームレベル生成を実現している
 - 閾値を小さくする→オブジェクトの個数は増加
 - 閾値を大きくする→オブジェクトの個数は減少
- 閾値は要求それぞれで固有に設ける

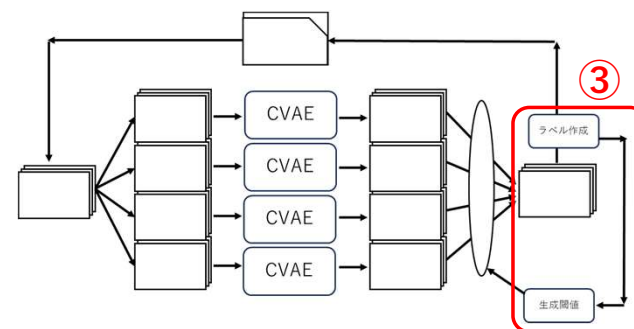


提案手法 ③閾値の更新

- 更新手法1：1エポック毎に1つずつ更新
 - 各要求は相互に影響し合っている
 - 学習結果により全要求の閾値を同時更新しても改善できないことがある

$$\begin{aligned}Thres &= Thres + sign \cdot update \\ update &= \gamma \cdot update\end{aligned}$$

Thres : 閾値
sign : 1 or -1
update : 閾値更新幅
 γ : 更新率(=0.9)

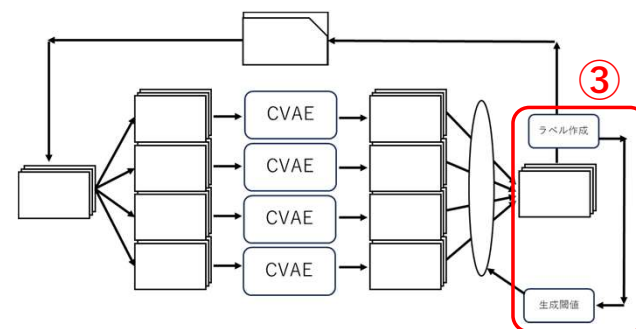


提案手法 ③閾値の更新

• 更新手法2：差分進化法による更新

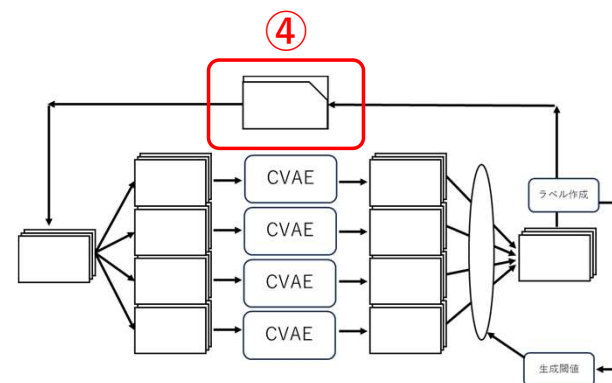
1. パラメータセット $x = [x^1, \dots, x^n]$ を用意し、 x^k の k を決める (初期は $k = 0$)
2. パラメータ x^{now} でゲームレベルを生成
3. x^{now} での生成結果が x^k での結果を上回れば、 $x^k \leftarrow x^{now}$
4. $req \leftarrow$ ラベル1が最も多く出現した要求項目番号
5. req に対応する要求について、 $k \leftarrow \operatorname{argmax}_k |x^{now}[req] - x^k[req]|$
6. $m^k \leftarrow x^a + F * (x^b - x^c)$ ($F: \text{const}$)
7. $\text{if } Prob = R : y_j^k \leftarrow m_j^k, \text{ else: } y_j^k \leftarrow x_j^k (\text{if } j = req : y_{req}^k = x^{now}[req])$

$R = 1 - (\text{要求} j \text{ のラベル 1 個数割合})$



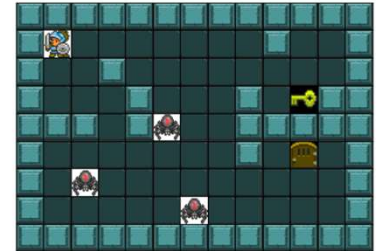
提案手法 ④ゲームレベルのアーカイブ化

- 生成したゲームレベルをアーカイブ化
 - 生成したゲームレベルも学習データに加える
 - 初期にゲームレベルが少数しかない問題を解消している
 - 機械学習に必要な学習データ数を確保



実験

- ゲームとしてGVGAIのZeldaを用いた
- 初期ゲームレベル数5, エポック数1200として, 生成実験を行った
- 学習が進んだときの要求を満たしたゲームレベル生成数の検証と, 閾値の更新手法1,2に対して生成結果の比較を行う



結果

- 更新手法1：1エポック毎に1つずつ更新

	要求値：要求達成率(%)		
要求項目	(a) 要求1	(b) 要求2	(c) 要求3
壁の数	0.5 : 58.40	0.6 : 58.62	0.2 : 60.00
敵の数	0.3 : 69.20	0.3 : 69.23	0.6 : 55.33
アバターと鍵との距離	0.3 : 60.35	0.6 : 16.67	0.3 : 61.38
アバターと扉との距離	0.4 : 39.80	0.4 : 43.06	0.8 : 3.58
生成総数	2656	2970	5386
要求達成総数：達成率(%)	278 : 10.89	100 : 3.37	43 : 0.80

結果

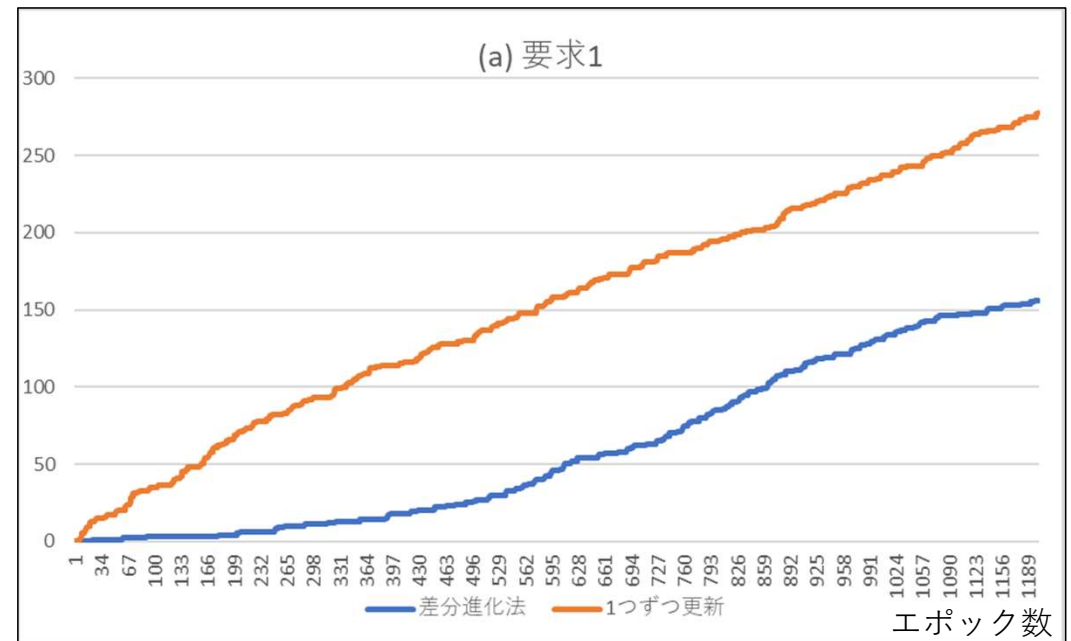
- 更新手法2：差分進化法

	要求値：要求達成率(%)		
要求項目	(a) 要求1	(b) 要求2	(c) 要求3
壁の数	0.5 : 31.98	0.6 : 9.84	0.2 : 45.17
敵の数	0.3 : 37.80	0.3 : 39.93	0.6 : 39.49
アバターと鍵との距離	0.3 : 61.54	0.6 : 20.55	0.3 : 62.33
アバターと扉との距離	0.4 : 45.82	0.4 : 47.45	0.8 : 4.45
生成総数	4553	5479	6974
要求達成総数：達成率(%)	156 : 6.22	18 : 0.33	37 : 0.53

結果

- 要求1の更新手法比較
 - 差分進化法は後半の傾きが大きい
 - 学習が進むにつれて閾値が最適化されていると考えられる
 - 1つずつ更新する手法は一定割合で増加

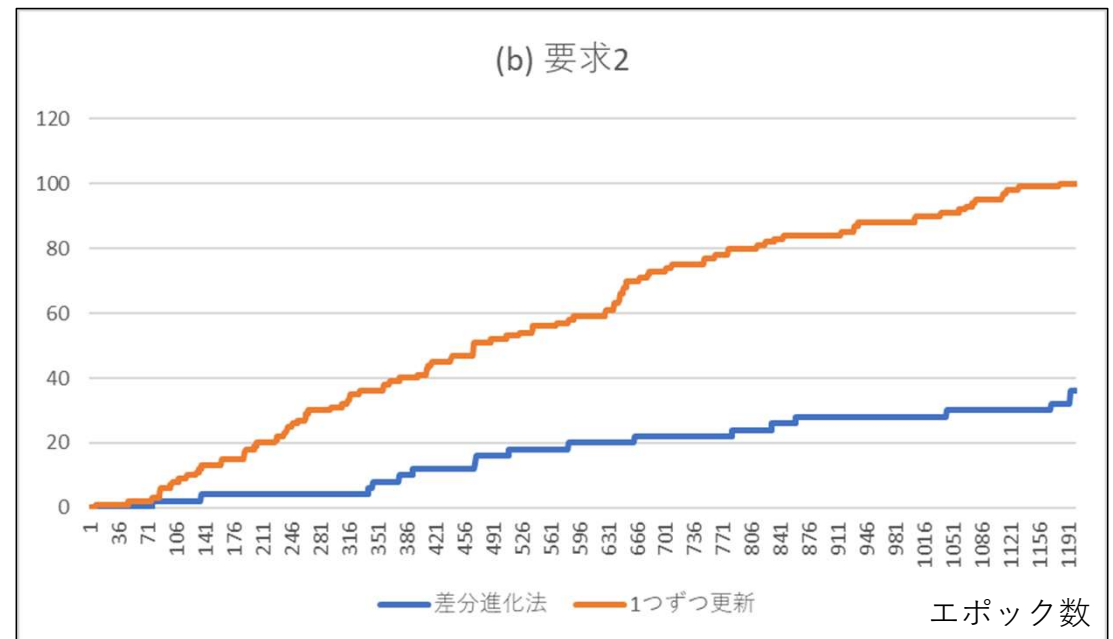
要求を満たした生成数



結果

- 要求2の更新手法比較
 - 1つずつ更新する手法で傾きが急増している箇所あり
 - このあたりで閾値がほぼ一定になっていた
 - 差分進化法では思うように生成数が増えなかった

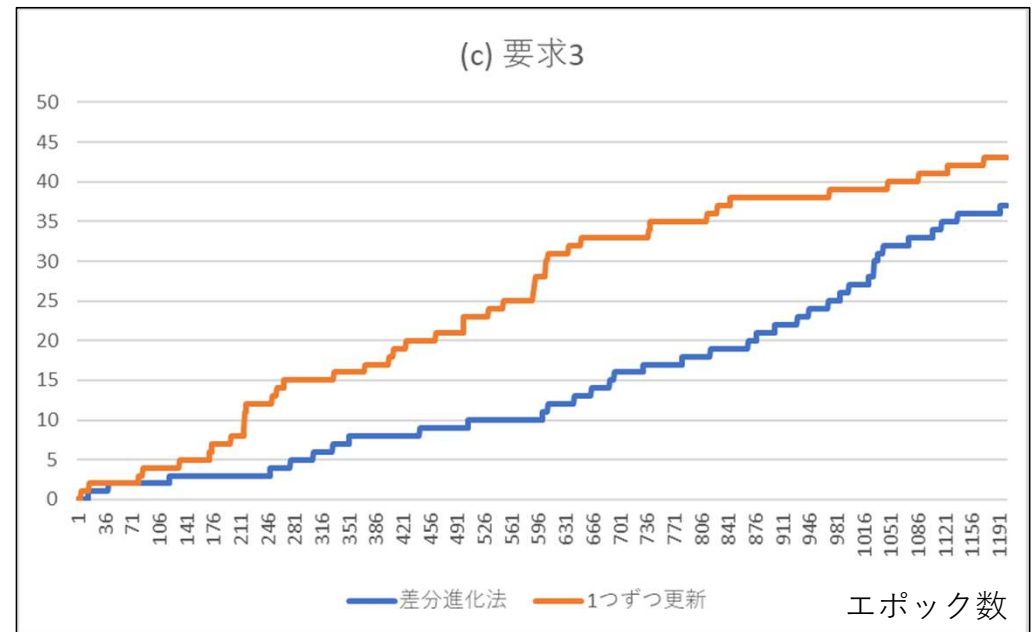
要求を満たした生成数



結果

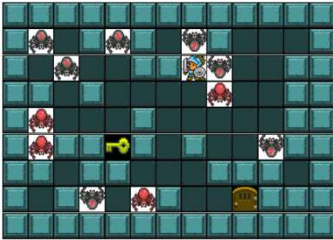
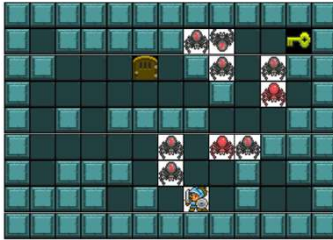
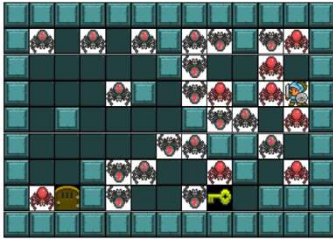
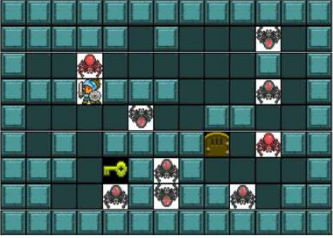
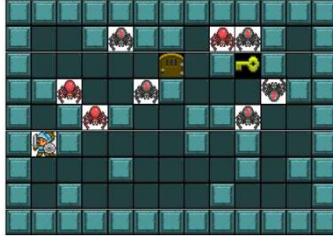
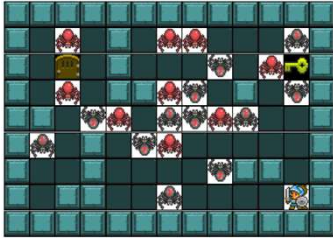
- 要求3の更新手法比較
 - 差分進化法は後半の傾きが大きい
 - 1つずつ更新する手法でも傾きが急増する箇所あり

要求を満たした生成数



結果

- 実際に生成できたゲームレベル

	要求 (壁の数,敵の数,アバターと鍵との距離,アバターと扉との距離)		
	(a) 要求1 (0.5, 0.3, 0.3, 0.4)	(b) 要求2 (0.6, 0.3, 0.6, 0.4)	(c) 要求3 (0.2, 0.6, 0.3, 0.8)
1つずつ更新する手法			
差分進化法			

考察

- 学習が進むにつれて要求に近いゲームレベルの生成数が増加している
 - 提案手法が有効であるとともに、閾値の最適化により開発者の要求を満たすゲームレベルが生成できることを意味する
- 一部の要求にはあまり応えられていない
 - 特に大きな値で要求した時の要求達成率が低い
 - 初期の学習データによる影響が大きいと考えられる
 - 要求1が初期のゲームレベル5つに比較的近いものであることから、要求1の達成率が高くなっている

まとめ

- CVAEによるゲームレベル生成手法について提案した
 - 少数のゲームレベルから開発者の要求を満たすゲームレベルを生成することに成功した
- より多様な要求への対応や他のゲームプロトタイプへ応用できるか
 - 昨今は3Dゲームが主流
 - 様々なゲームへの適用方法を検討したい