

Title	プロシージャルモデリングによる階層的3次元ダンジョンの制作支援
Author(s)	大川, 将広
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17085
Rights	
Description	Supervisor:宮田 一乗, 先端科学技術研究科, 修士(情報科学)

修士論文

プロシージャルモデリングによる階層的 3 次元ダンジョンの制作支援

大川 将広

主指導教員 宮田 一乗

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

令和 3 年 2 月

Abstract

In this paper, we propose a hierarchical three-dimensional dungeon production support system by procedural modelling. In addition to the parameter control, the proposed system is composited with global function to design the outer shape of the dungeon by sketching and local function to edit the model and support the user to directly modify it. With the help of these functions, the production efficiency of dungeon modeling by designers can be improved. We also proposed a novel index that indicates the difficulty of reaching the dungeon. The difficulty of reaching each dead point end from the start point of the dungeon is verified using the river order. This index enables practical usage such as placing objects such as treasure chests at points with higher river orders. Finally, we evaluated the usefulness of the proposed system. The participants were asked to make a dungeon using the proposed interface and a questionnaire survey was conducted. In addition, we made it possible to search the created dungeon in a VR environment and measured the arrival time costs from the start point to each dead point end to clarify the difficulty of reaching the introduced dungeon.

目次

第1章 はじめに	1
1. 1 研究背景	1
1. 2 研究の目的	1
1. 3 本論文の構成	1
第2章 関連研究	2
2. 1 ダンジョン生成	2
2. 2 迷路生成	4
2. 3 プロシージャルモデリング	7
第3章 提案手法	9
3. 1 システム設計	9
3. 2 ダンジョン生成手法	10
3. 3 編集機能	20
3. 4 到達しにくさの提案	23
第4章 実験と評価	25
第5章 おわりに	33
5. 1 まとめ	33
5. 2 課題	33
5. 3 展望	33

目次

図 1 : 機械学習を用いたダンジョンの自動生成 [3] : (a) 人間が作成したダンジョン, (b) 人間が作成したダンジョンを学習し自動で生成されたダンジョン.	2
図 2 : Roland らの手法により生成されたダンジョン [4]	3
図 3 : Roden らの手法により生成されたダンジョン [5]	3
図 4 : Bartosz らの手法により生成されたゲームレベル [6], (a) 迷路, (b) ダンジョン.	4
図 5 : 画像に基づく迷路の自動生成 [7]	5
図 6 : 有機的に見える迷路の自動生成 [8]	5
図 7 : デザイン中心の迷路生成 : (a) 生成された迷路, (b) 生成された迷路を使って制作されたゲーム [10].	6
図 8 : 都市のプロシージャルモデリング [14]	7
図 9 : 建物のプロシージャルモデリング [15]	8
図 10 : 水文学に基づく地形の生成 [16]	8
図 11 : フレームワーク	9
図 12 : スケッチからポリゴンを生成する手順 (a) スケッチ入力 (b) 整形されたスケッチ (c) ポリゴン	10
図 13 : 入力スケッチの整形 : (a) 入力スケッチ, (b) 整形されたスケッチ.	11
図 14 : ダンジョンの外郭形状の例	11
図 15 : 部屋の生成手順 : (a) 入力, (b) 母点を配置, (c) ボロノイ分割.	12
図 16 : 部屋数を変化させた例 : (a) 15 部屋, (b) 20 部屋, (c) 25 部屋.	12
図 17 : グラフ生成手順 : (a) 部屋のレイアウト, (b) 各部屋の中心 (青色の点) と各部屋の壁の midpoint の位置 (赤色の点) に点を配置, (c) 接続.	13
図 18 : 全域木生成手法 : (a) 図 16 の手順で生成したグラフ, (b) 解経路の生成 (緑がスタートで赤がゴール), (c) プリム法による全域木の生成, (d) 辺削除	14
図 19 : グラフ生成手順 : (a) 3 階層, (b) 階ごとにグラフ生成, (c) 辺を追加し重なっている部屋を繋げる.	15
図 20 : 壁の生成手順 : (a) 部屋のレイアウト, (b) 壁の作成 (青の領域), (c) 壁を上方向に押し出した結果.	16

図 21： 出入り口の生成：(a)押し出した壁と生成したグラフ，(b)交差点に Box 配置，(c)ブーリアン演算により穴をあける．	17
図 22： 階段配置の例：(a)2つの部屋，(b)部屋の垂直方向での重なりを求める(赤色の領域)，(c)重なっているセルの中心に階段を配置．	18
図 23： 階段の配置：(a)配置する階段の大きさ．壁の高さを h とすると， $x=h/4$ ， $y=h$ ， $z=h$ としている．(b)配置する階段の向きは，部屋の中心に向くようにしている．	18
図 24： オブジェクト配置の例	19
図 25： 特定の部屋を小さく編集した例：(a)編集前，(b)編集後．	20
図 26： 壁を移動させた例：(a)編集前，(b)編集後．	21
図 27： 壁を回転させた例：(a)編集前，(b)編集後．	21
図 28： 壁の編集によりグラフが変化した例：(a)編集前，(b)編集後．	22
図 29： 出入り口編集の例：(a)編集前，(b)編集後．	22
図 30： スタートとゴール位置の編集の例：(a)編集前，(b)編集後．緑がスタートで赤がゴール．赤の線が解経路で白の線が分岐経路．	23
図 31： スタートから行き止まりまでの到達しにくさの計算手順．緑の部屋がスタート地点．(a)割り当てられた次数(b)(a)で割り当てられた次数から計算された到達しにくさ	24
図 32： 到達しにくさの計算結果の例	24
図 33： システムの使いやすさについての実験結果	25
図 34： 被験者が制作したダンジョン	26
図 35： 実験で使用したダンジョン．緑の部屋がスタートで，色のついた部屋までの到達時間を測定した．数値は到達しにくさを表す．	27
図 36： 到達しにくさ，についての実験結果	28
図 37： 実験の様子：(a)システムの使いやすさについての評価実験，(b)到達しにくさについての評価実験．	28
図 38： 本システムにより制作したダンジョンの例．1 階層	29
図 39： 本システムにより制作したダンジョンの例．2 階層	30
図 40： 本システムにより制作したダンジョンの例．3 階層	31
図 41： 本システムにより制作したダンジョンの例．4 階層．	32

第1章 はじめに

本章では、まず本研究の背景について述べる．次に本研究の目的を説明し、最後に本論文の構成について記述する．

1.1 研究背景

近年のゲーム開発は、グラフィックスハードウェアが進歩し、プレイヤーの要求が高まったことによって、複雑で時間を要するようになってきている [1]．そのため、制作時間の削減を図り、プロシージャルによるコンテンツ制作の手法は、ますます重要になってきており、多くの研究が行われている．その中でも、本研究では、ダンジョンのプロシージャル生成に焦点を当てる．Roland ら [2] は、ダンジョンのプロシージャル生成についての広範な調査を行っている．調査結果から、Roland らは、ダンジョンのプロシージャル生成に欠けているのは、生成プロセスに対する、正確かつ豊富な制御であるとしている．また、デザイナーに提供する高レベルの制御機能の直感性とアクセシビリティを向上させる必要があるとしている．

1.2 研究の目的

本研究の目的は、ダンジョンを制作するゲームデザイナーの支援であり、プロシージャルモデリングにより、階層的三次元ダンジョンの制作支援システムを提案する．そのため、提案システムでは、プロシージャルモデリングと手作業による編集機能を組み合わせた手法をとっている．また、スケッチによるダンジョンの外郭形状の入力も可能である．

さらに、生成したダンジョン内のオブジェクトの配置に応用するための新たな指標として、ダンジョンのスタート地点から行き止まりまでの到達のしにくさを提案する．

1.3 本論文の構成

本論文は、全 5 章で構成する．第 2 章では関連研究を紹介し、本研究の意義を示す．第 3 章では、提案手法として、ダンジョン生成手法・編集機能・到達のしにくさについて説明する．第 4 章では、実験と評価として、システムの有用性の評価及び到達しにくさについての評価を行った．最後に第 5 章では、本研究を総括し、今後の課題を述べる．

第2章 関連研究

2.1 ダンジョン生成

関連研究として、ダンジョンの自動生成に関する研究がある。Evan ら [3]は、特定のレベルをプレイするのが楽しくなるものを作成するために、デザイナーが設計したタイルベースのダンジョンを強化学習により学習して、新しいダンジョンを自動で生成するシステムを提案した(図 1)。その結果、人間がデザインしたダンジョンに近づけることが示されている。

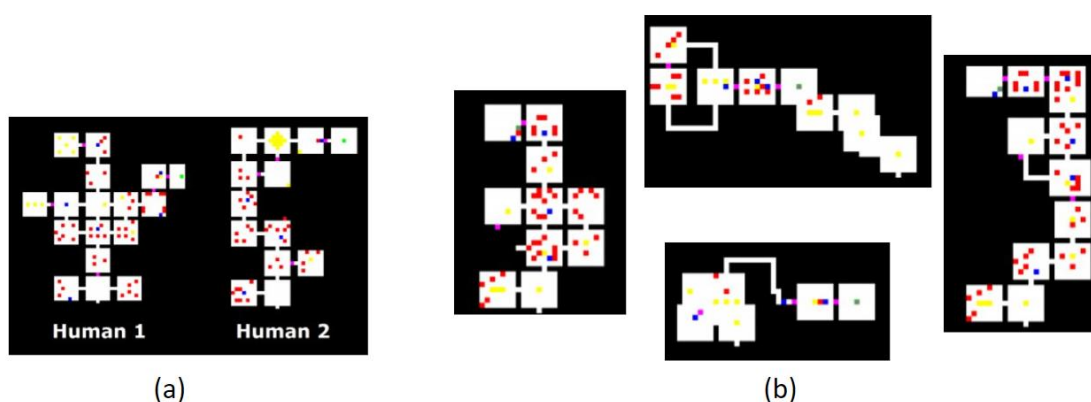


図 1：機械学習を用いたダンジョンの自動生成 [3]：(a)人間が作成したダンジョン，(b)人間が作成したダンジョンを学習し自動で生成されたダンジョン。

また、ダンジョンのプロシージャル生成に関する研究の中で、生成されるダンジョンのモデルが 3 次元であるものをいくつか取り上げる。Roland ら [4]は、ゲームプレイ関連の設計制約をアクショングラフによって表現することにより、ゲームデザイナーのダンジョンの作成及び制御を可能にした(図 2)。

Roden ら [5]は、ダンジョン生成のために、3 次元の無向グラフを作成し、作成したグラフの各頂点を実際のジオメトリに変換することで、3 次元のダンジョンを生成している(図 3)。

Bartosz ら [6]は、迷路やダンジョンなど多種多様なレベルを半自動生成する新しいアプローチを提案している(図 4)。ユーザは、グラフベースの抽象レベル構造を生成することで、対応するジオメトリが自動構築される。そして、実験結果により、提案されたアプローチが、複雑なレベルを効率的に作成し、レベルとゲームデザイナーの創造性の制限を最小限に抑えることを示している。

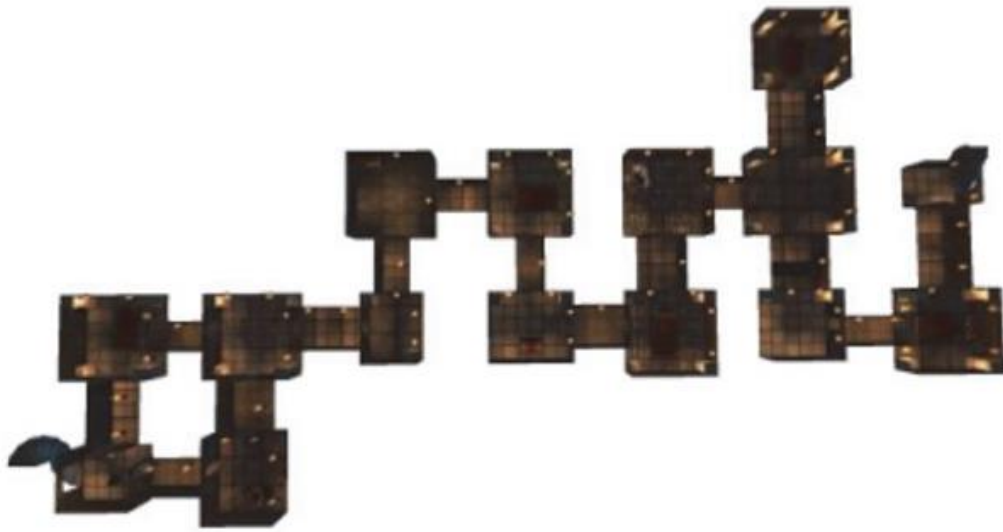


図 2 : Roland らの手法により生成されたダンジョン [4]

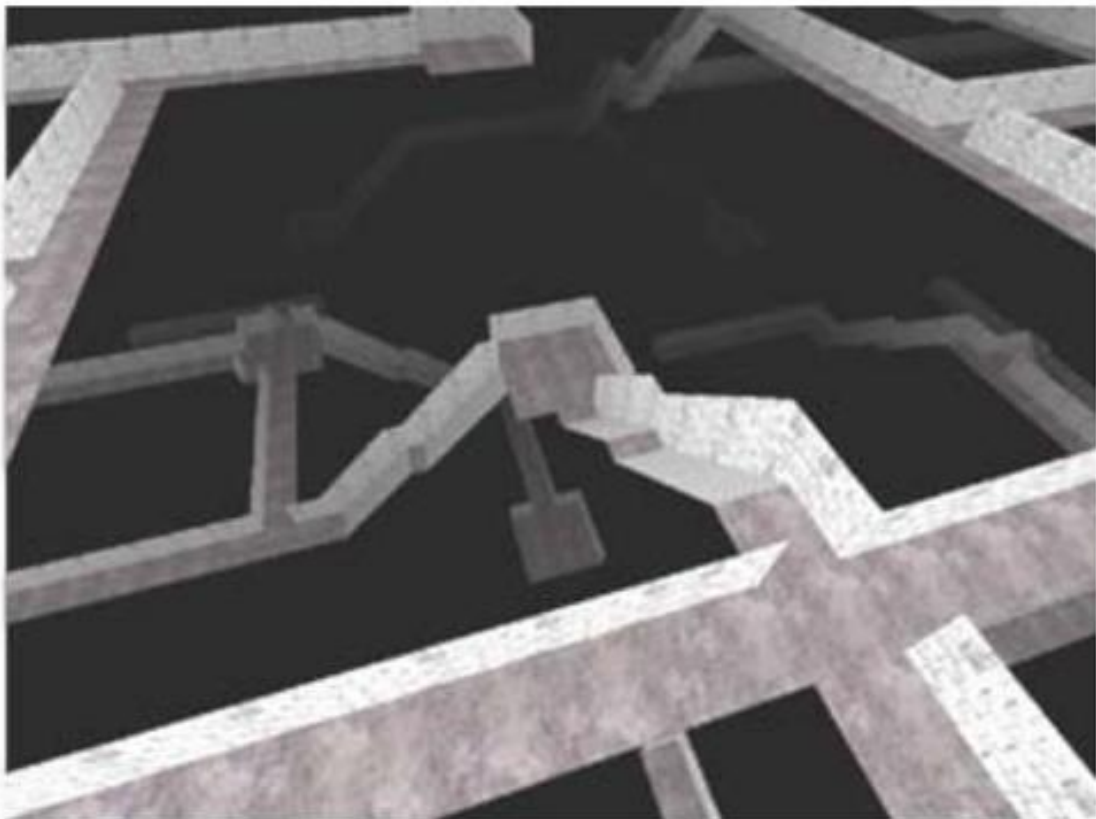


図 3 : Roden らの手法により生成されたダンジョン [5]



図 4 : Bartosz らの手法により生成されたゲームレベル [6], (a)迷路, (b)ダンジョン.

このように, ダンジョンの生成手法に関して, 多くの提案がされている. これに対し, 本論文で提案するシステムでは, 複数の階層を持ち, 階層ごとに異なる外郭形状を指定できる 3 次元のダンジョンをプロシージャルに生成することを目的とする.

2.2 迷路生成

ダンジョンの自動生成に関連する研究として, 迷路の研究がある. Jie ら [7] は, 画像に基づいて迷路を生成する手法を提案した(図 5). この手法では, ユーザがソリューションパスを入力すると, そのパスに準拠する迷路が構築される. また, 分割された領域ごとにスタイルパラメータを設定することで, 迷路の外観を決めることが可能である.

Hans ら [8]は, 有機的に見える迷路を自動生成するためのモデルを提案した. 生成される迷路は, 2D 多様体上の曲線として表される迷路であり, デザイナーはパスの複雑さと視覚的な美しさの両方を制御できると示されている(図 6).

井上ら [9]は, 重心ボロノイ分割の母点を最小木でつないだランダム迷路パターンを生成する手法を提案した. この手法により, モノクロ画像の明暗の会長をランダム迷路パターンの粗密によって表現することが可能である.

Paul ら [10]は, コンピュータゲームでの使用を想定し, デザイナーが希望のプロパティ(迷路のパスのねじれを表すプロパティなど)を入力して独自の迷路を作成できる, デザイン中心の迷路生成手法を提案した(図 7). この手法により, ゲームの仕組みに基づいた迷路の設計ができると示されている.

本研究では, ダンジョンを制作するデザイナーを支援するためのより直感的

な操作として、スケッチによる外郭形状の入力と編集機能を、実装している．従来手法では，本システムのように，直感的な操作で編集することはできない．

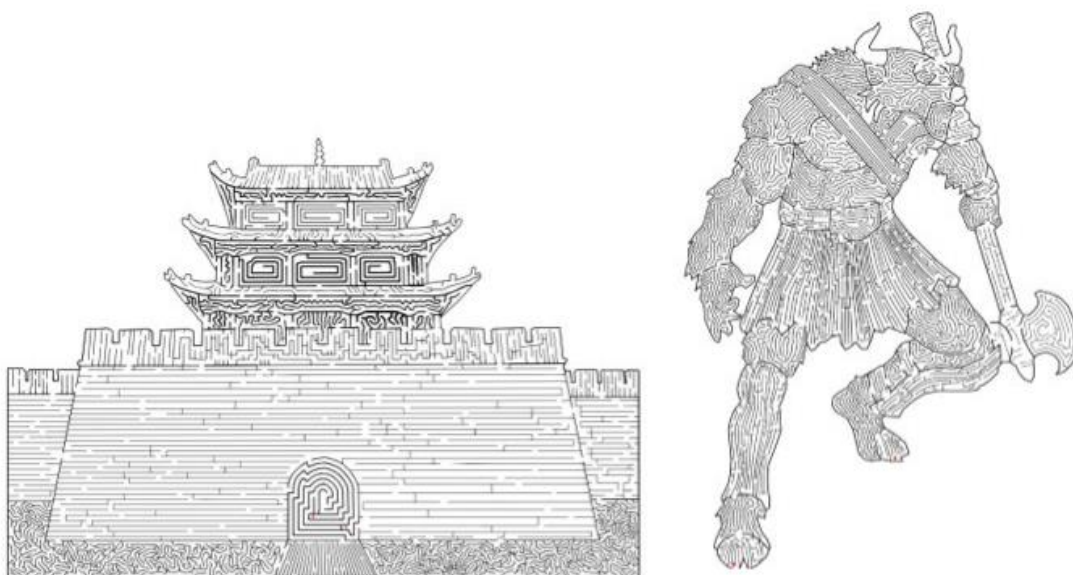


図 5：画像に基づく迷路の自動生成 [7]

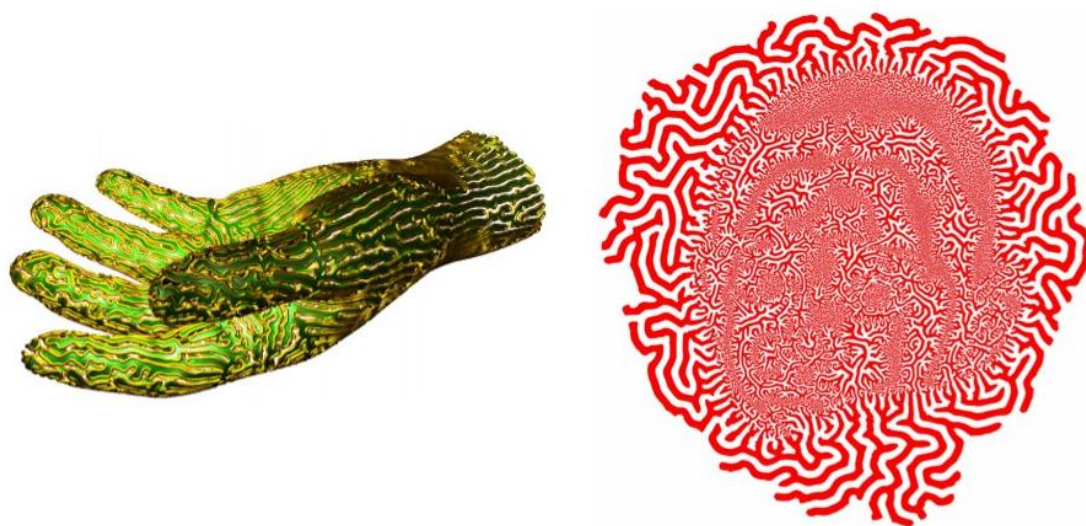


図 6：有機的に見える迷路の自動生成 [8]

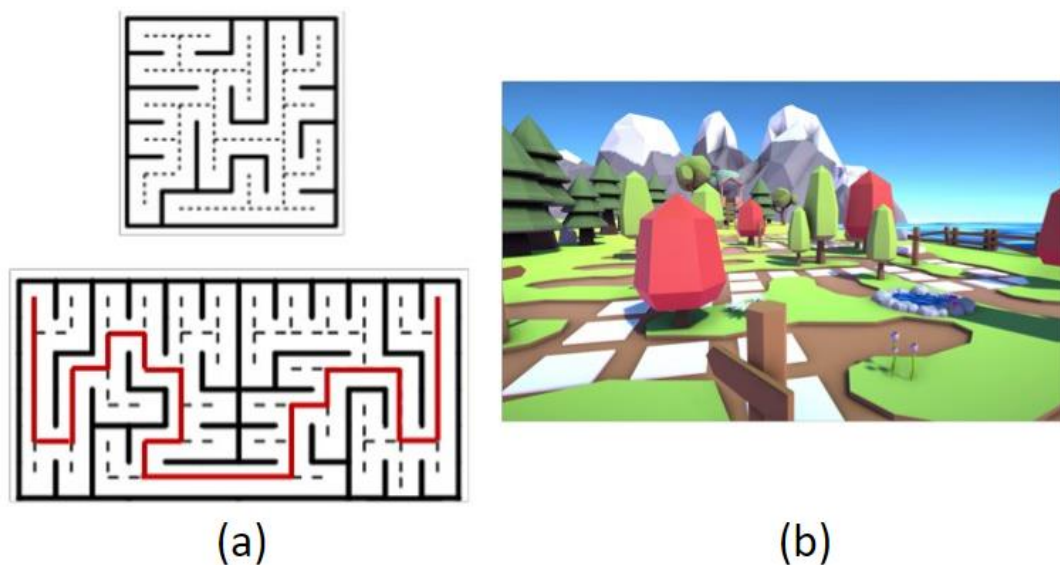


図 7: デザイン中心の迷路生成: (a) 生成された迷路, (b) 生成された迷路を使って制作されたゲーム [10].

また、迷路の研究として、迷路の難易度についての研究がある。水澤ら [11] は、迷路問題の難易度指標として、障害物密度を提案した。この研究では、二次元正方格子状のグラフからなる迷路において、障害物密度に応じて、ランダムに障害物を配置することで、迷路を作成している。そして、生成される迷路の難易度のピークは、障害物密度が約 41% のときであると示している。

また、横田ら [12] は、迷路問題の難易度指標として、ゴールまでの T 字分岐の個数、および、迷路コストを提案している。実験結果から、どちらも同程度の難易度パラメータであることが示されている。

Daryl ら [13] は、仮想迷路における人間のパフォーマンスにおいて、照明やランドマーク、オーディオキューなどが及ぼす影響についての調査を行った。

本研究では、ダンジョンの探索時におけるスタートから行き止まりまでの、到達しにくさを表す指標を提案する。この指標を導入することによって、到達しにくい行き止まりに宝箱などのオブジェクトを配置する、というような実用的な使い方が可能になる。

2.3 プロシージャルモデリング

プロシージャルモデリングの研究としては、地形や都市、建物、など多くの研究が行われている。Parish ら [14]は、都市のモデリングのために、L-system に基づくプロシージャルアプローチを提案している(図 8)。

Muller ら [15]は、プロシージャルモデリングのために、CGA 形状と呼ばれる新しい形状文法を提案した(図 9)。CGA 形状により、屋根の設計を含む建物のプロシージャルモデリングが可能となった。

Genevaux ら [16]は、水文学に触発された概念を使用して、地形の迅速かつ直感的なモデリングを可能にするフレームワークを提案した(図 10)。

Ruben ら [17]の研究では、プロシージャル生成と手作業による編集をシームレスに統合して、仮想世界をモデリングするシステムを提案している。プロシージャルと手作業による編集機能を組み合わせることによって、作業を効率化しつつ繊細な作業が可能になると示唆している。したがって本研究では、プロシージャルモデリングと手作業による編集を組み合わせた手法をとっている。

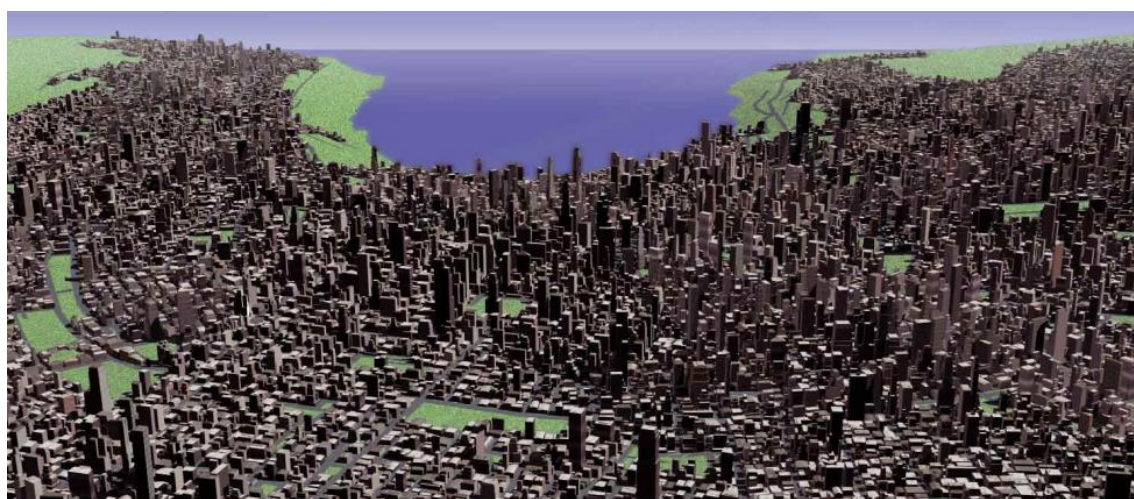


図 8 : 都市のプロシージャルモデリング [14]



図 9：建物のプロシージャルモデリング [15]



図 10：水文学に基づく地形の生成 [16]

第3章 提案手法

本論文では，デザイナーにとって使いやすいダンジョン生成支援システムを提案する．そのために，システムには，スケッチによるダンジョンの外郭形状の入力と，編集機能を実装している．また，ダンジョン生成のオブジェクトの配置に応用するための新しい指標として，到達しにくさを提案する．本章では，提案手法として，システム設計・ダンジョン生成手法・編集機能・到達しにくさについて，解説する．

3.1 システム設計



図 11：フレームワーク

提案システムのフレームワークを図 11 に示す．入力は，ダンジョンの外郭形状であり，スケッチにより行う．複数階層のダンジョンを制作する場合は，階ごとに外郭形状を入力することになる．このとき，図 11 のように重ね機能があるため，解のスケッチ通しの位置確認が可能である．その後，ダンジョンのプロシージャル生成や編集機能により，出力のダンジョンが完成する．このように編集機能があることで，プロシージャル生成したダンジョンをそのまま使うのではなく，デザイナーに改変の余地を与えている．これにより，デザイナーの意図を反映することが可能になると考えられる．

これらのシステムは，プロシージャルモデリングが可能な DCC ツールである Houdini に実装した．

3.2 ダンジョン生成手法

まず、コンピュータゲームにおけるダンジョンについて説明する．デジタル大辞典では、「ロールプレイングゲームなどの舞台となる，迷路に似た構造を持つ空間」とされている．単に，地下迷宮あるいは迷宮，と表現されることも多い．一般的に，ダンジョン生成は，部屋の生成と通路の生成に分けることができるが，本論文で提案するシステムで制作できるダンジョンは，部屋のみであり，より迷路に近い構造になっている．

外郭形状の入力と整形 本システムでは，ダンジョンの外郭形状をスケッチ入力する．次のステップである部屋の生成のために，図 12 のように外郭形状をポリゴンに変換する．ポリゴンに変換するために，まず，入力のスケッチを整形している．例えば，図 13(a) のスケッチのような複数の曲線がある場合，近い距離にある点を結合することで，1つの曲線に変換している．1つの曲線にすることで，曲線の長さを計算し，その曲線を均一の長さのセグメントでサンプリングし直している（図 13(b)）．スケッチ入力は描いた速度などによって，点が疎らになるため，このようにリサンプリングし均一にすることで，外郭形状を整えている．

その後，サンプリングし直した点に基づいて一つのポリゴンを作成している．このような手順で，ダンジョンの外郭形状を生成するが，階層数が複数の場合は，階層ごとにスケッチを入力することになる．図 14 に，スケッチ入力された外郭形状の例を示す．

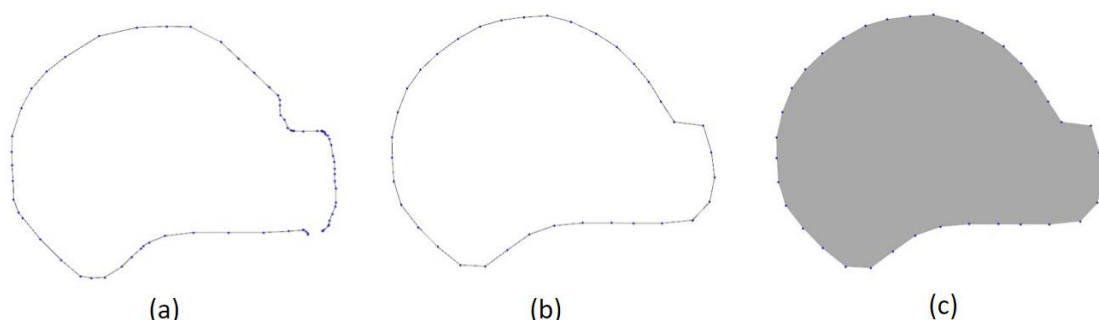


図 12：スケッチからポリゴンを生成する手順 (a) スケッチ入力 (b) 整形されたスケッチ (c) ポリゴン

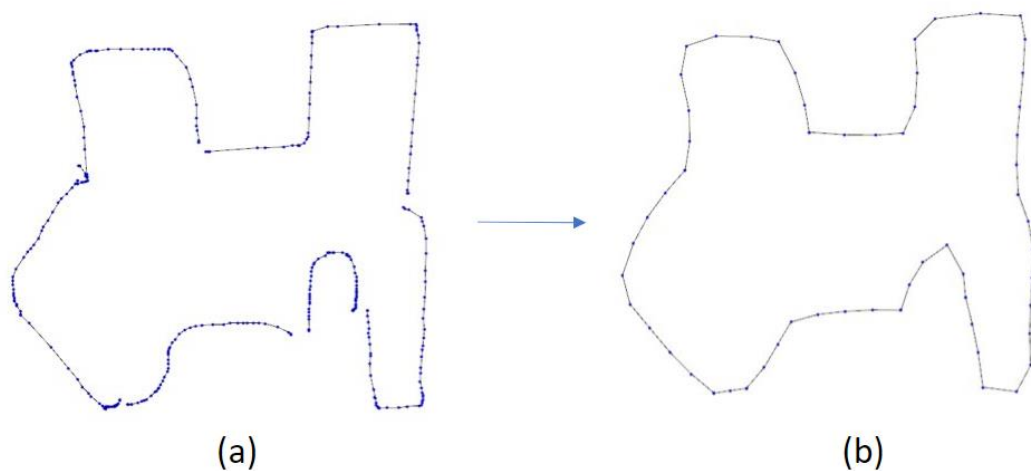


図 13：入力スケッチの整形：(a)入力スケッチ，(b)整形されたスケッチ.



図 14：ダンジョンの外郭形状の例

部屋生成 次に、多角形をボロノイ分割することで、部屋の生成を行う．部屋は、図 15 のような手順で生成する．ボロノイ分割を使う理由としては、母点数によって簡単に部屋数の制御が可能となるためである(図 16)．母点の配置では、領域内に母点を均一に配置することで、極端に小さい部屋が生成されないようにしている．そのために、まず適当に母点を配置した後、通常のボロノイ分割から (1) (2) を繰り返している．

(1) ボロノイ分割後、各領域の中心を求める．

(2) 母点を中心に移動して、もう一度ボロノイ分割する．

これを繰り返すことによって、均一に母点を配置している．本システムでは、20 回程度で母点の位置が収束したため、繰り返し回数を 20 としている．

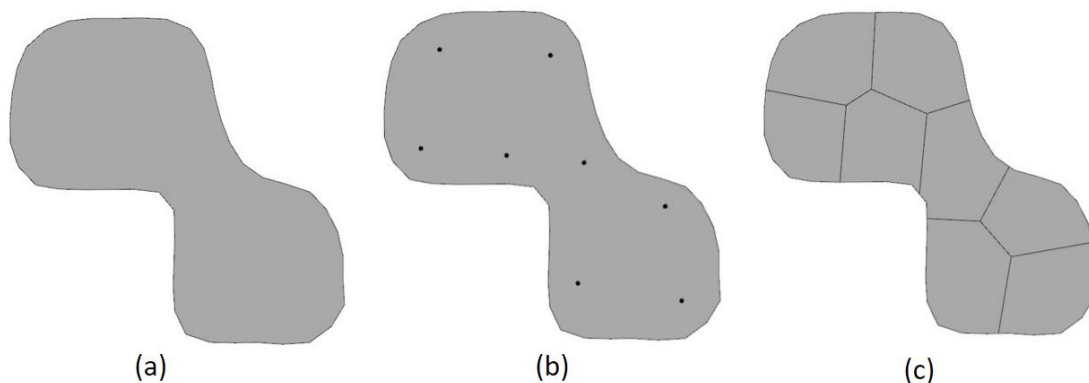


図 15：部屋の生成手順：(a)入力，(b)母点を配置，(c)ボロノイ分割．

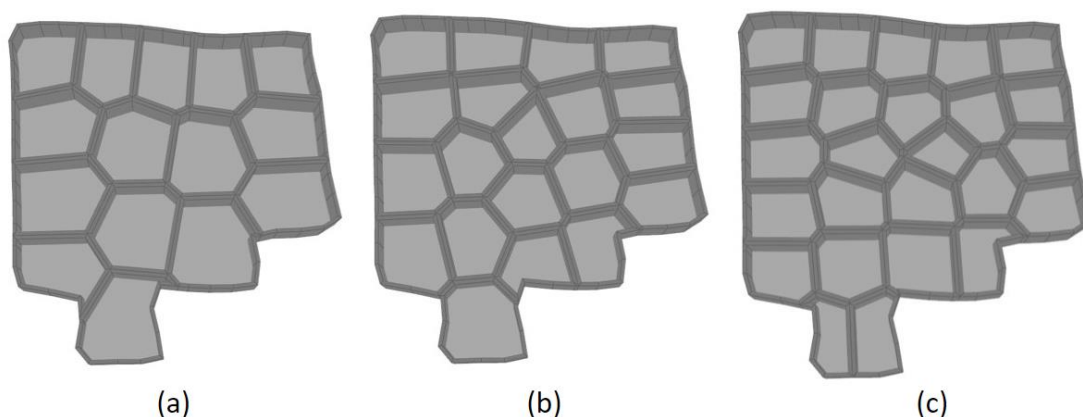


図 16：部屋数を変化させた例：(a)15 部屋，(b)20 部屋，(c)25 部屋．

グラフ生成 部屋の生成後，生成した部屋に対応するグラフを，図 17 のような手順で生成する．まず，図 17(b)のように部屋の中心と部屋の出入り口・階段(階層数が複数の場合)の配置場所となり得る位置に頂点を配置する．次に，配置した頂点を辺で接続する(図 17(c))．

生成したグラフから，図 18 のような手順で，部屋の中心に配置された頂点をつなぐ全域木を生成する．解経路を生成してから，いくつかの辺を削除することで，全域木となる．解経路の生成は，最短経路アルゴリズムであるダイクストラ法 [18]を用いて生成している．このとき，距離ではなく辺にランダムな重みを割り当てて，経路を生成している(図 18(b))．解経路の生成後，プリム法 [19]により，全域木を生成している(図 18(c))．その後，壁に配置されている頂点(図 17, 18 における赤色の頂点)が葉頂点(頂点に接続している辺の数が 1)の場合，その頂点と接続している辺を削除する(図 18(d))．

階層数が複数の場合は，図 19 のように階ごとにグラフを生成してから，階層間を繋げている．図 19 のような 3 階層の場合，A と B, C と E, D と F の部屋が

重なっているため，それらの部屋を繋げる辺を追加している．生成したグラフから，同じように図 17 の手順で全域木が生成される．

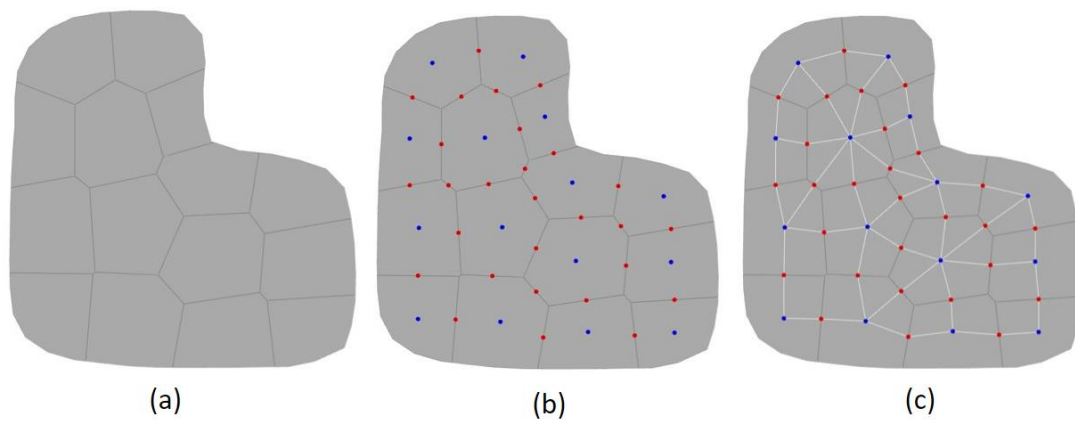
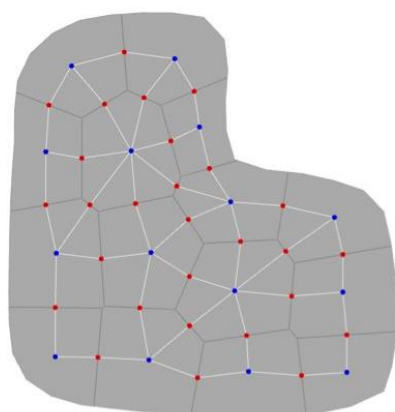
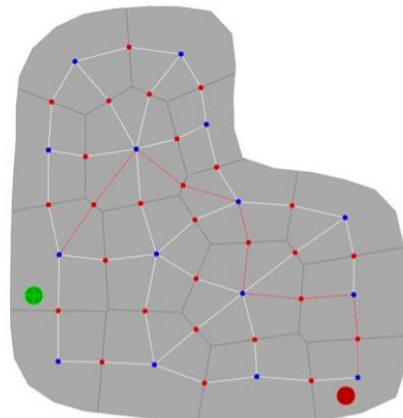


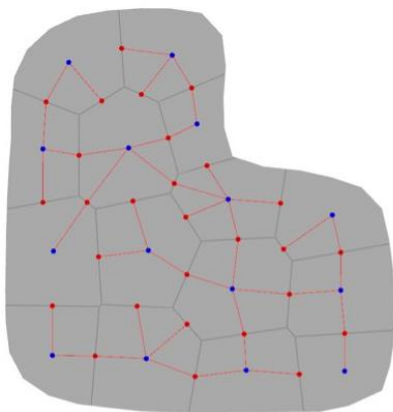
図 17：グラフ生成手順：(a) 部屋のレイアウト，(b) 各部屋の中心(青色の点)と各部屋の壁の中点の位置(赤色の点)に点を配置，(c) 接続．



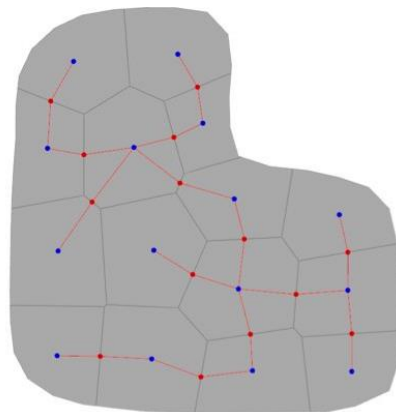
(a)



(b)



(c)



(d)

図 18：全域木生成手法：(a)図 16 の手順で生成したグラフ，(b)解経路の生成（緑がスタートで赤がゴール），(c)プリム法による全域木の生成，(d)辺削除

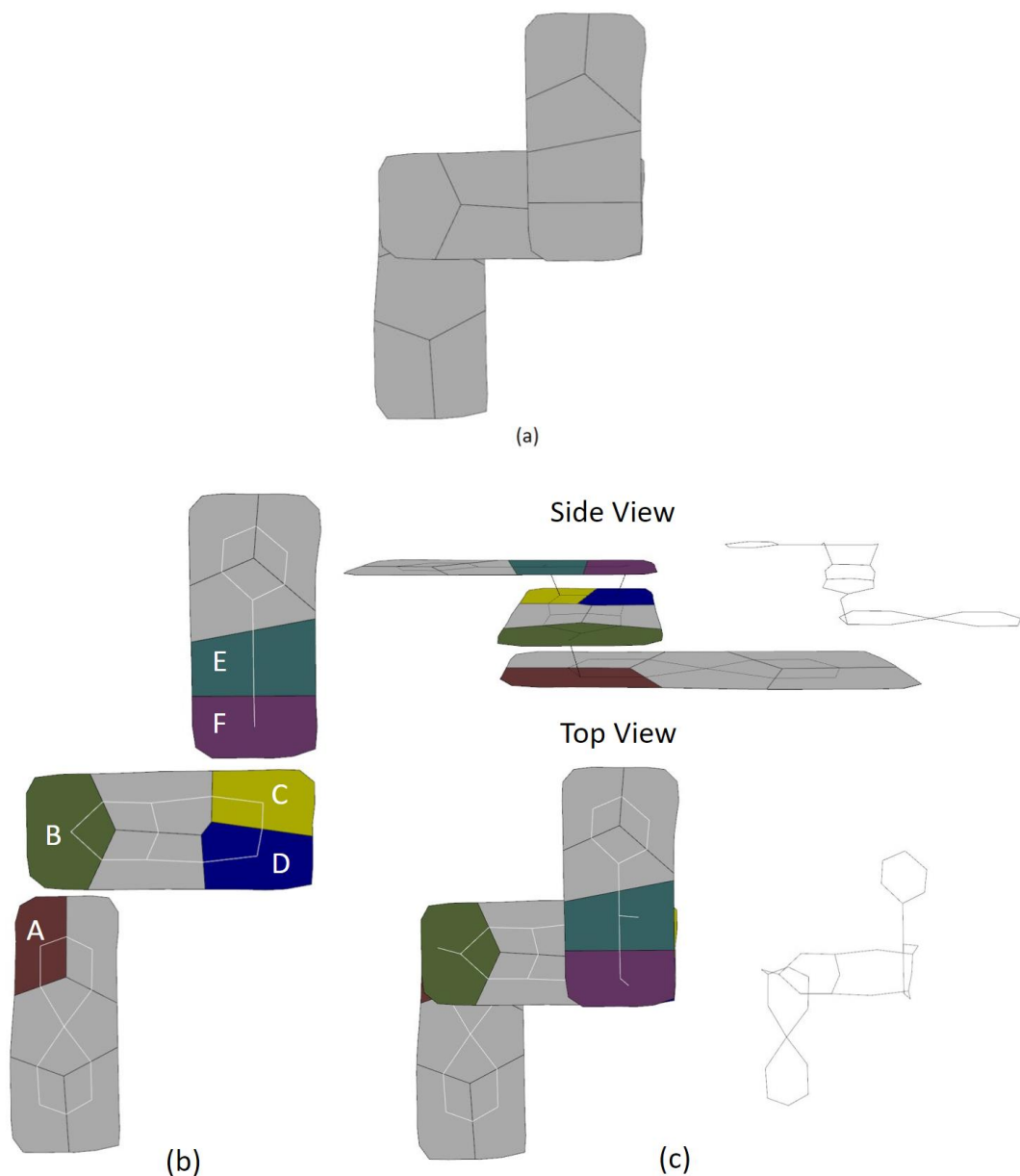


図 19：グラフ生成手順：(a)3 階層，(b)階ごとにグラフ生成，(c)辺を追加し重なっている部屋を繋げる．

壁の作成 次に，部屋とグラフから，プロシージャルモデリングによって，対応する 3D モデルを生成する．まず，生成した部屋のレイアウトから壁の押し出しを行う．壁に厚さを持たせるために，各部屋の内側方向に新たな面を作成する．そして，パラメータにより壁の高さを決定し，作成した面を上方向に押し出すことで壁を生成する (図 20)．押し出した壁と生成したグラフの交差点に，Box を配置し，ブーリアン演算により，出入り口の部分に穴をあける (図 21)．配置する Box の大きさは，Box の横幅を x ，高さを y ，奥行を z ，壁の高さを h ，壁の厚

さを w とすると, $x=h \times 0.4$, $y=h \times 0.7$, $z=w \times 2$ としている. 横幅 x より高さ y を少し大きめに設定することで, ダンジョン探索時に人が出入りできる大きさにしている. また, 奥行を $z=w$ とすると, ブーリアン演算時の計算誤差により壁に穴が開けられないことがあるため, 奥行は壁の厚さより大きめに設定している. 壁の高さ h と厚さ w は, デザイナーがパラメータにより指定できるようにしている.

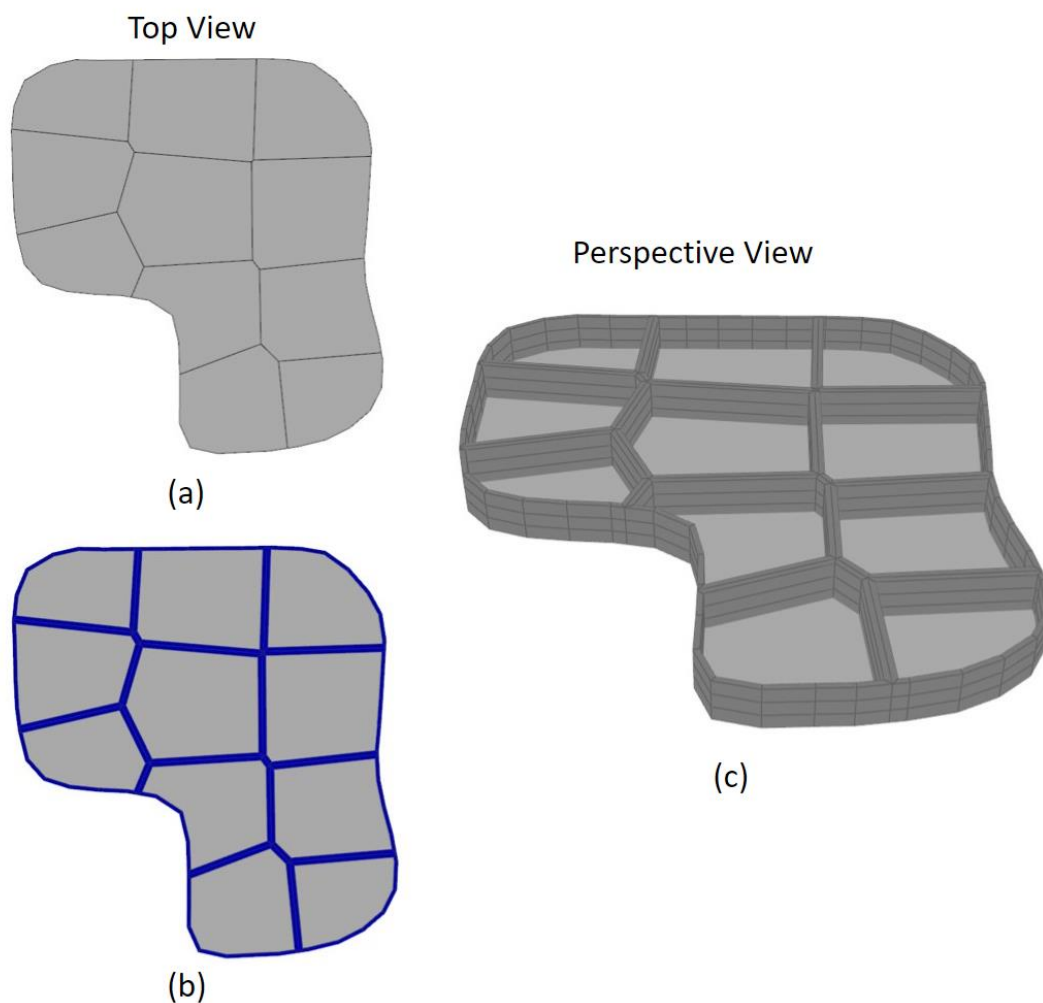


図 20 : 壁の生成手順 : (a) 部屋のレイアウト, (b) 壁の作成 (青の領域), (c) 壁を上方向に押し出した結果.

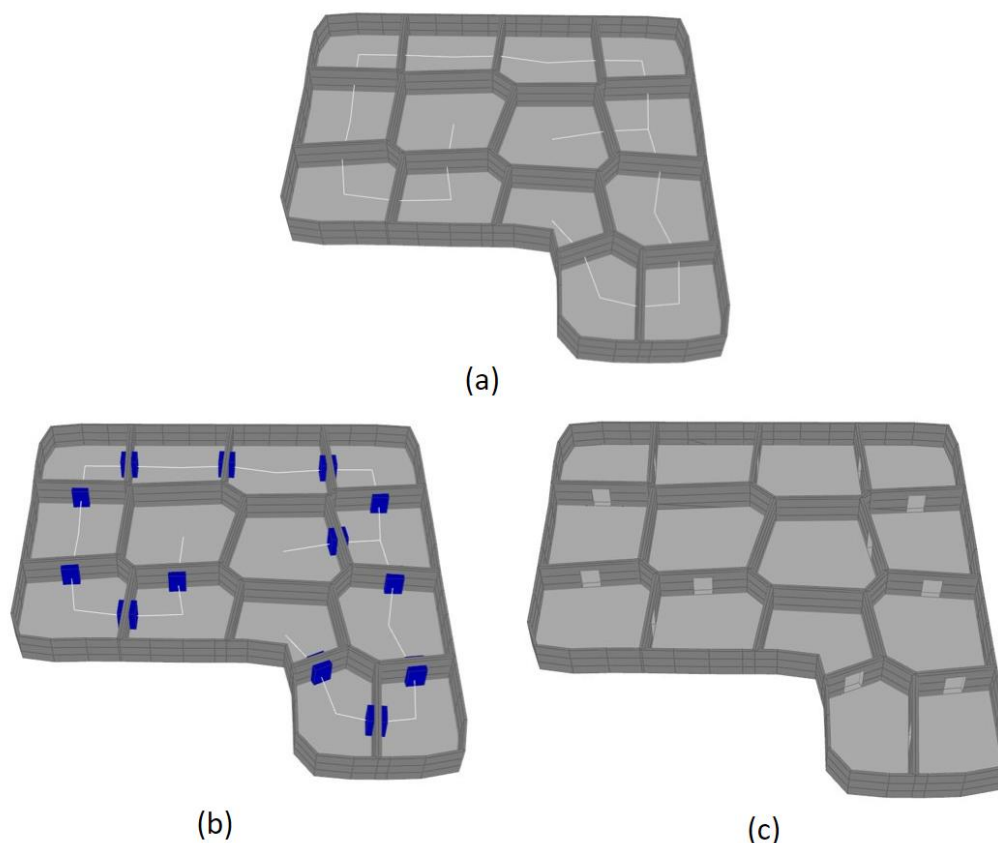


図 21：出入り口の生成：(a)押し出した壁と生成したグラフ，(b)交差点に Box 配置，(c)ブーリアン演算により穴をあける．

階段の配置 階層数が複数の場合は，階層間を階段で繋げている．階段を配置するときは，階ごとに部屋のレイアウトが異なるため，配置場所を考慮する必要がある．例えば，図 22(a)のような2つの部屋を繋げる階段を配置する場合，2つの部屋の垂直方向での重なりを求め，重なっているセルの中心の位置に階段を配置する．配置する階段の大きさは，重なっているセルの面積を a ，壁の高さを h ，階段の幅を x ，階段の高さを y ，階段の奥行を z とすると， $x=(a \times 0.2)/z$ ， $y=h$ ， $z=h$ としている．天井は，階段の幅 x と奥行 z の大きさで穴を開けている．このようにすることで，天井に開ける穴の大きさが，重なっているセルの $1/5$ の大きさとなり，穴がセルより大きくなることはない．また，階段の向きは，部屋を中心に置くように配置している(図 23)．

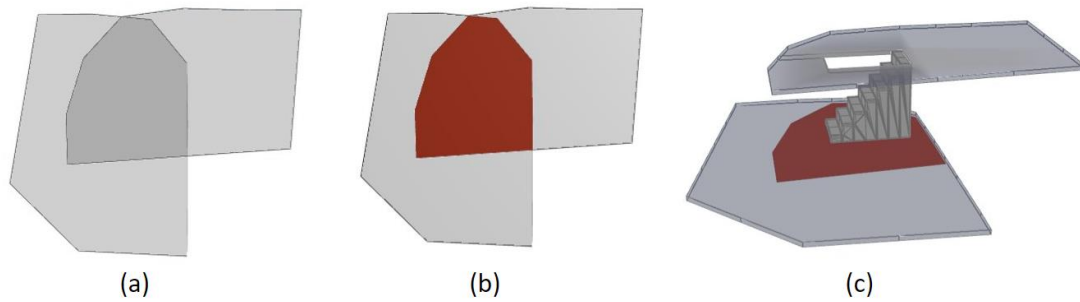


図 22：階段配置の例：(a)2つの部屋，(b)部屋の垂直方向での重なりを求める（赤色の領域），(c)重なっているセルの中心に階段を配置．

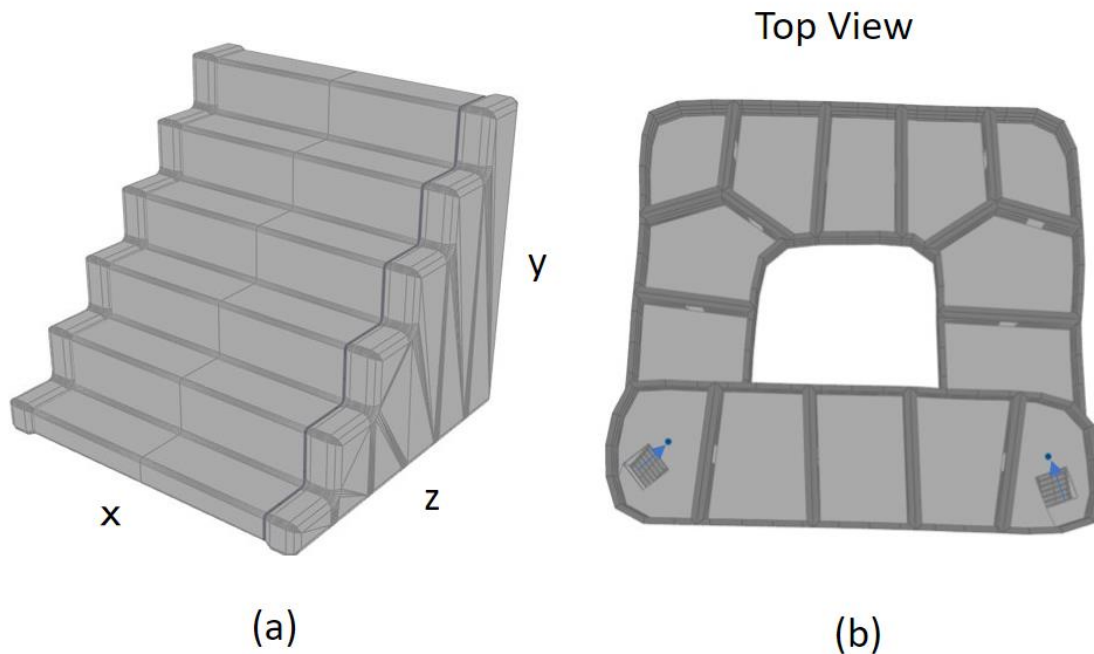


図 23：階段の配置：(a)配置する階段の大きさ．壁の高さを h ，重なっているセルの面積を a とすると， $x=(a \times 0.2)/z$ ， $y=h$ ， $z=h$ としている．(b)配置する階段の向きは，部屋の中心に向くようにしている．

オブジェクトの配置 図 24 は，ダンジョン内にオブジェクトを配置した例である．図 24 では，行き止まりに宝箱，分岐点に石を配置し，ランダムな場所に樽を配置している．行き止まりに宝箱を配置する場合は，到達しにくさを計算(3.4 節で説明)することで，到達しにくい行き止まりにのみに宝箱を配置する，ということも可能である．また，ランダムに配置した樽は，パラメータにより配置する樽の数を制御できる．

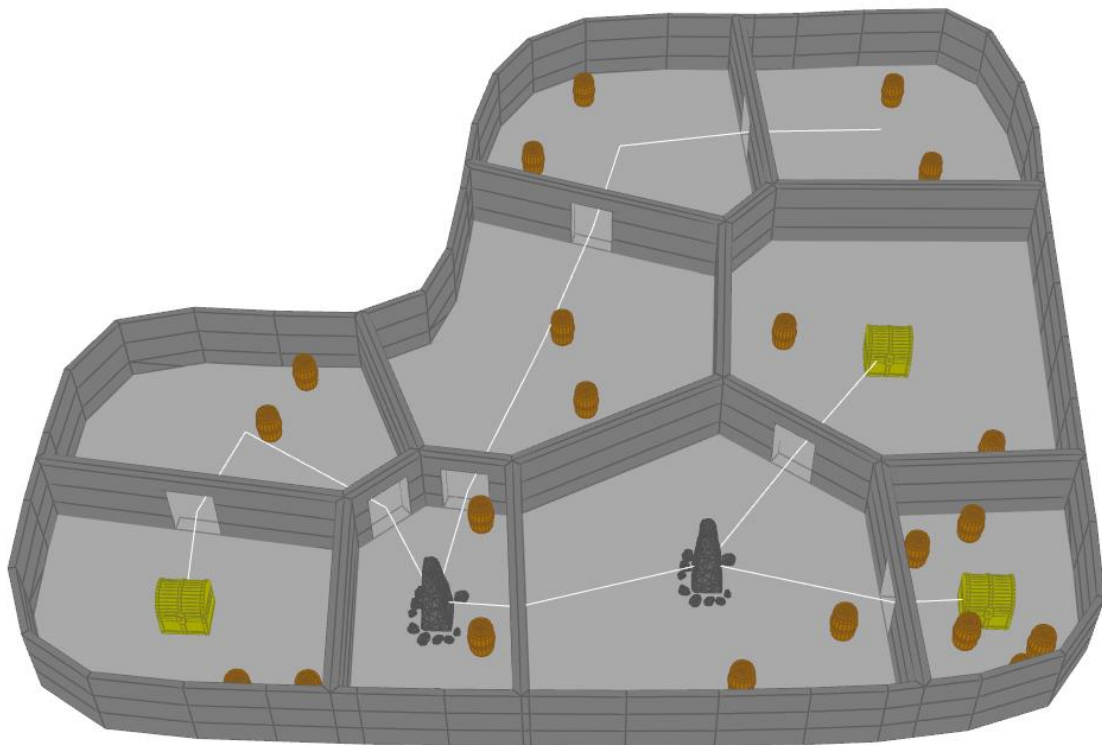


図 24 : オブジェクト配置の例

3.3 編集機能

本システムでは、デザイナーのための編集機能を提供する。編集機能として、部屋の編集・壁の編集・出入り口の編集・スタート、ゴール位置の編集の4つを提供する。これらの編集機能は、マウスを用いた編集することになる。

部屋 本システムでは、部屋の生成をボロノイ分割によって行っている。そのため、ボロノイ分割時の入力である母点の位置を操作することによって、部屋の大きさや位置の調整が可能になる。図25は、母点の位置を上方向に動かすことで、部屋を小さく編集をしている例である。ダンジョンでは、部屋の大きさがある程度ばらばらであることが好まれるため、このような編集機能を提供している。

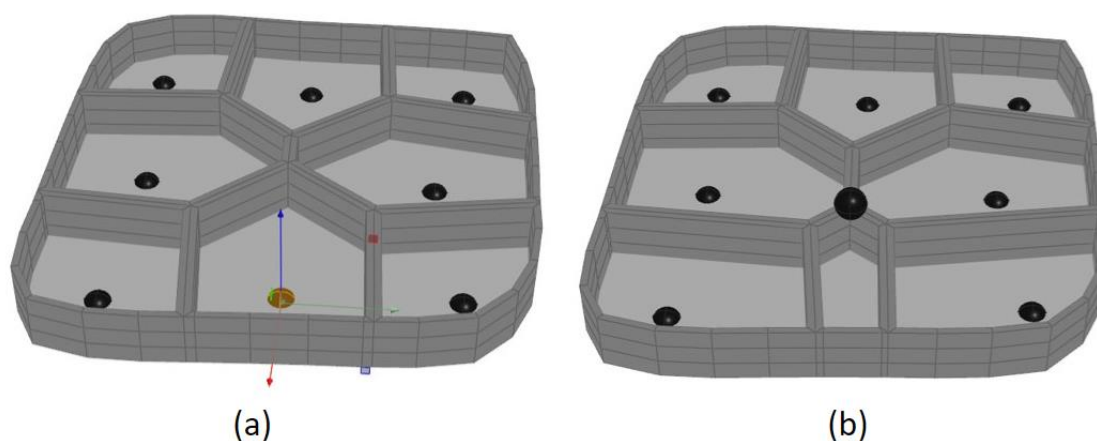


図 25：特定の部屋を小さく編集した例：(a)編集前，(b)編集後。

壁 上記の部屋の編集では、大まかな調整しかできないため、細かい調整として、壁の編集を提供する。壁の編集では、壁の移動・回転の2つの操作が可能である。図26，27が壁の編集をしている例である。壁の編集は、部屋生成後にエッジ(部屋の壁となる部分)抽出を行い、抽出したエッジに対して処理をすることによって実現している。また、図28に、壁の編集によって、対応するグラフが変化している例を示す。

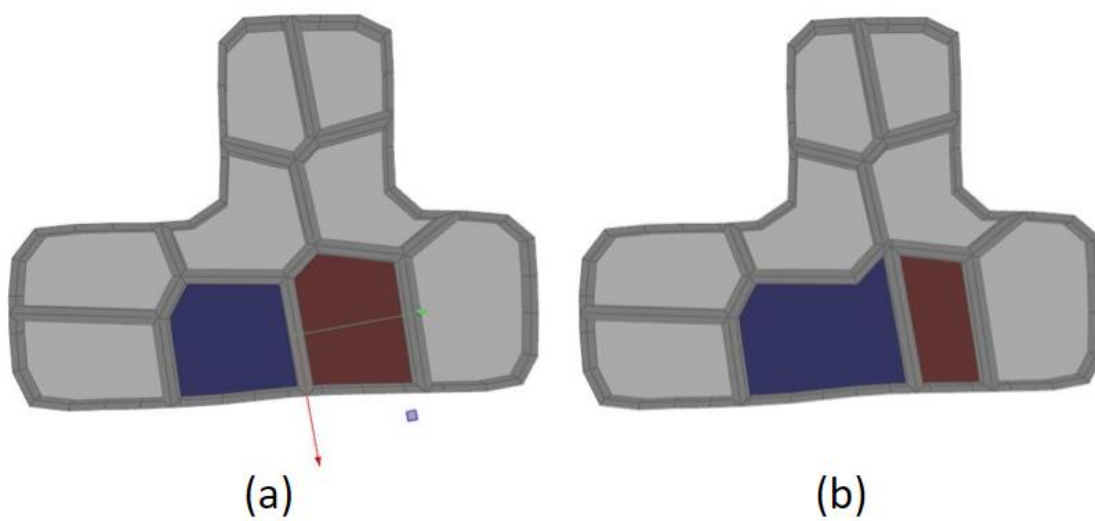


図 26：壁を移動させた例：(a) 編集前，(b) 編集後．

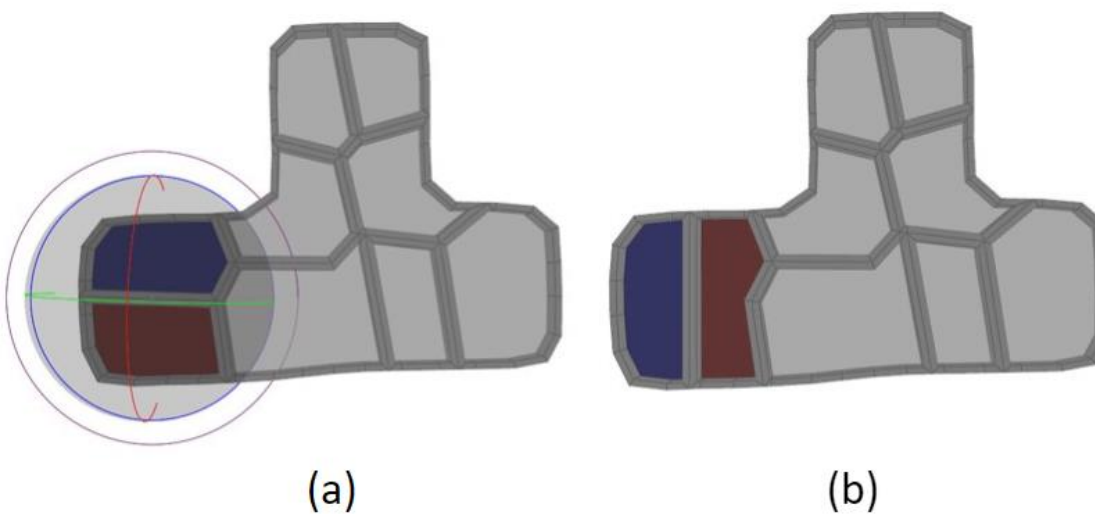


図 27：壁を回転させた例：(a) 編集前，(b) 編集後．

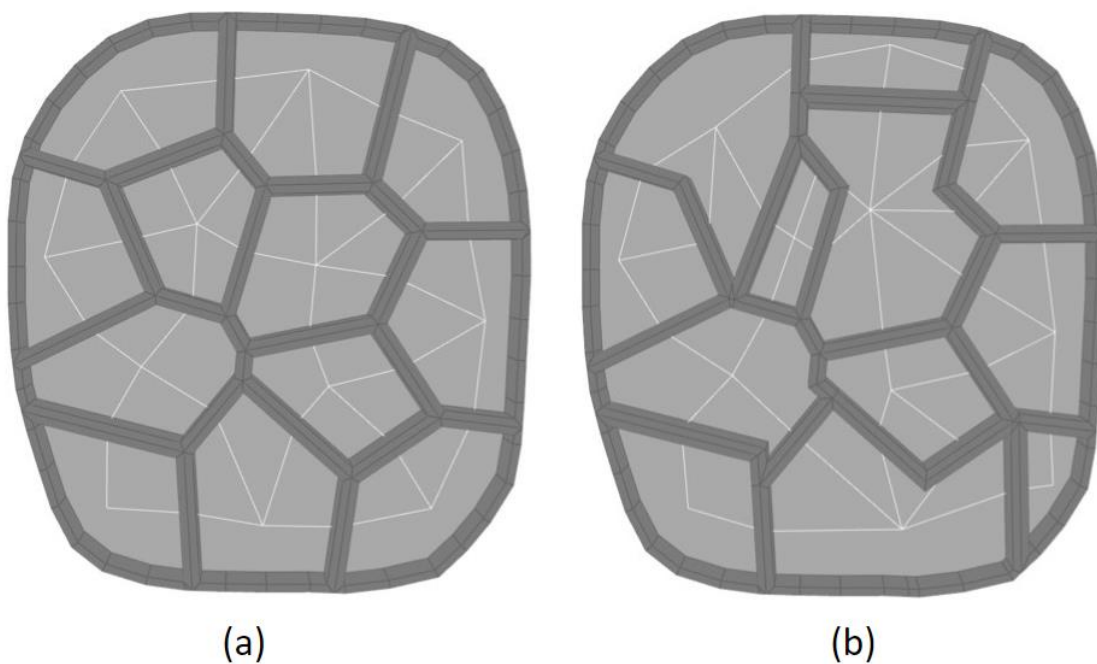


図 28：壁の編集によりグラフが変化した例：(a)編集前，(b)編集後．

出入口 ダンジョン内に配置される出入口位置の編集は，図 29 のように可能である．この編集操作は，壁に穴をあけるときに Box と壁とのブーリアン演算によって壁に穴をあけているが，その Box をマウスで指定し，位置をマニュアルで動かすことで実現している．

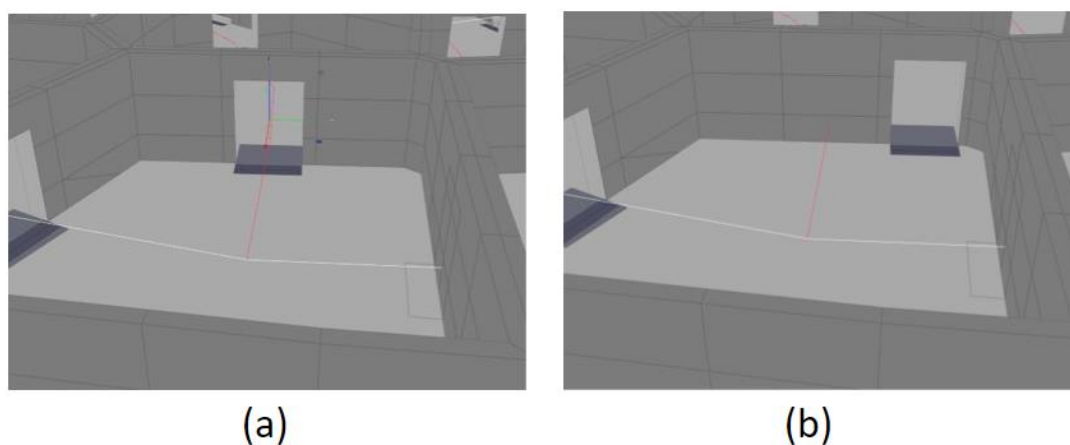


図 29：出入口編集の例：(a)編集前，(b)編集後．

スタートとゴール ダンジョンを設計するとき、ダンジョンのスタートとゴールの位置も重要である。そこで、スタートとゴールの位置を編集できるようにし、スタートからゴールまでの解を見ながら、位置を決定することができる。図 30 がスタートとゴールの位置を編集している例である。

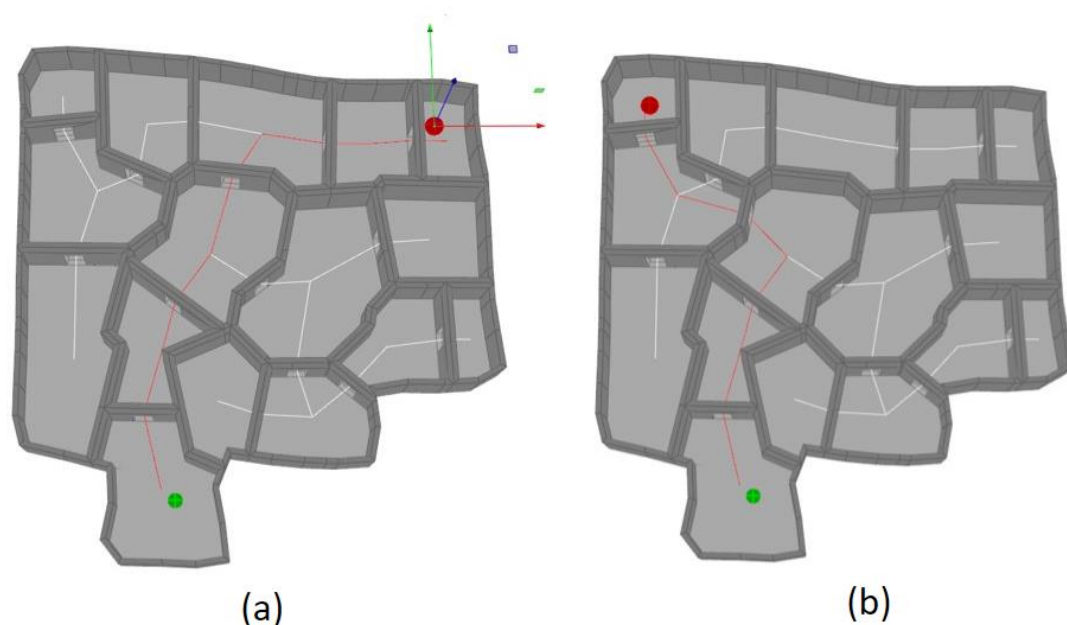


図 30：スタートとゴール位置の編集の例：(a) 編集前，(b) 編集後．緑がスタートで赤がゴール．赤の線が解経路で白の線が分岐経路．

3.4 到達しにくさの提案

生成したダンジョンのスタートから行き止まりまでの到達のしにくさを、河川の順位付けの手法である Shreve [20]の手法により計算した。計算は、図 31 のような手順で行った。まず、全ての行き止まりに 1 次を割り当てる。次に分岐点の次数を計算する。分岐点の次数の計算は、例えば、1 次の行き止まり 3 つに接続している場合、その分岐点には 3 次が割り当てられ、3 次の分岐点と 1 次の行き止まりに接続している分岐点の場合には、4 次が割り当てられる(図 31(a))。そして、スタートから行き止まりまでの次数を加算していき、行き止まりまでの到達のしにくさが計算される(図 31(b))。このように行き止まりまでの到達のしにくさを計算することで、ゴールの設定や宝箱などのアイテムの配置などに生かすことができる。具体的には、到達のしにくい行き止まりには、良いアイテムを配置しておく、などが可能になる。図 32 に到達しにくさの計算結果の例を示す。

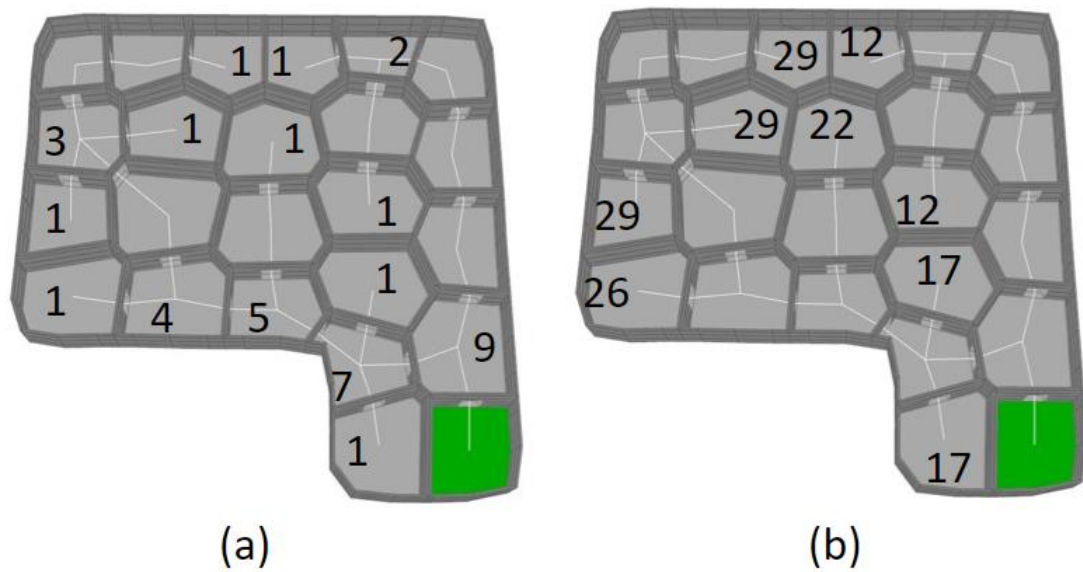


図 31：スタートから行き止まりまでの到達しにくさの計算手順．緑の部屋がスタート地点．(a)割り当てられた回数(b) (a)で割り当てられた回数から計算された到達しにくさ

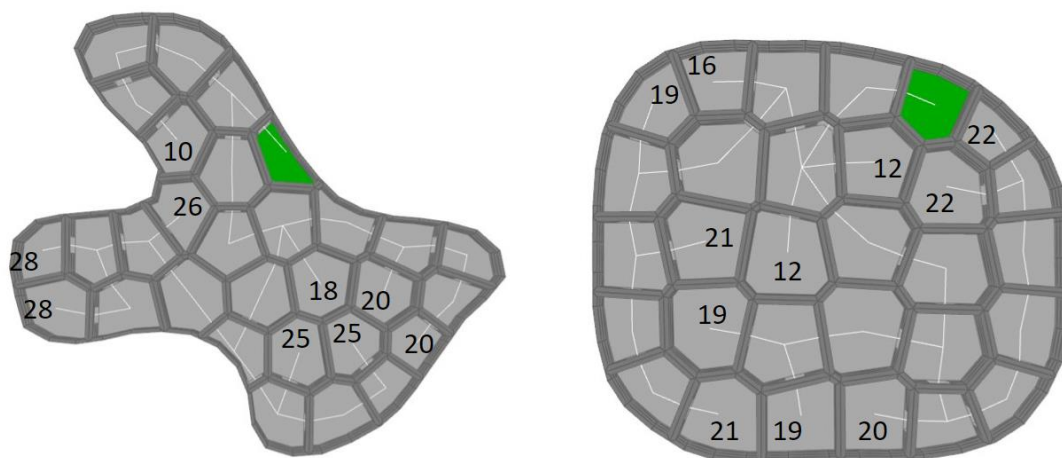


図 32：到達しにくさの計算結果の例

第4章 実験と評価

実験は、システムの使いやすさについての評価実験及び到達しにくさについての評価実験の2つを行った。

まず、システムの使いやすさについての評価実験として、大学院生 9 名に対して行った。実験の様子を図 37(a)に示す。被験者には実際にシステムを使ってダンジョンを制作してもらった。その後、システムの使いやすさについてのアンケート調査を行った。アンケートの結果を図 33 に示す。アンケート結果から、9 名中 2 名が使いにくいと答えていることがわかる。理由として、本システムは Houdini 上で動作するものであるが、インターフェースなどがまだ分かりやすく実装できず、どのような機能が使えるか分かりにくい、ということがあったためと考えられる。そのため、ユーザインターフェースに関しては、改善する必要があるということが分かった。図 34 に実際に被験者が制作したダンジョンを示す。

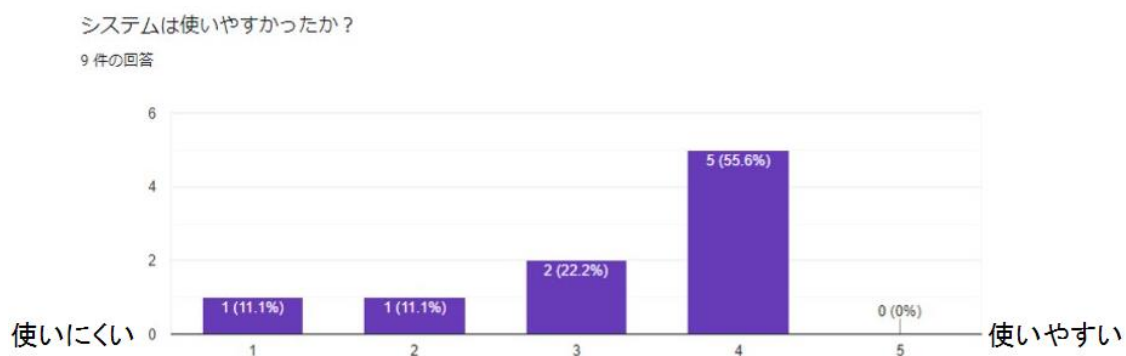


図 33：システムの使いやすさについての実験結果

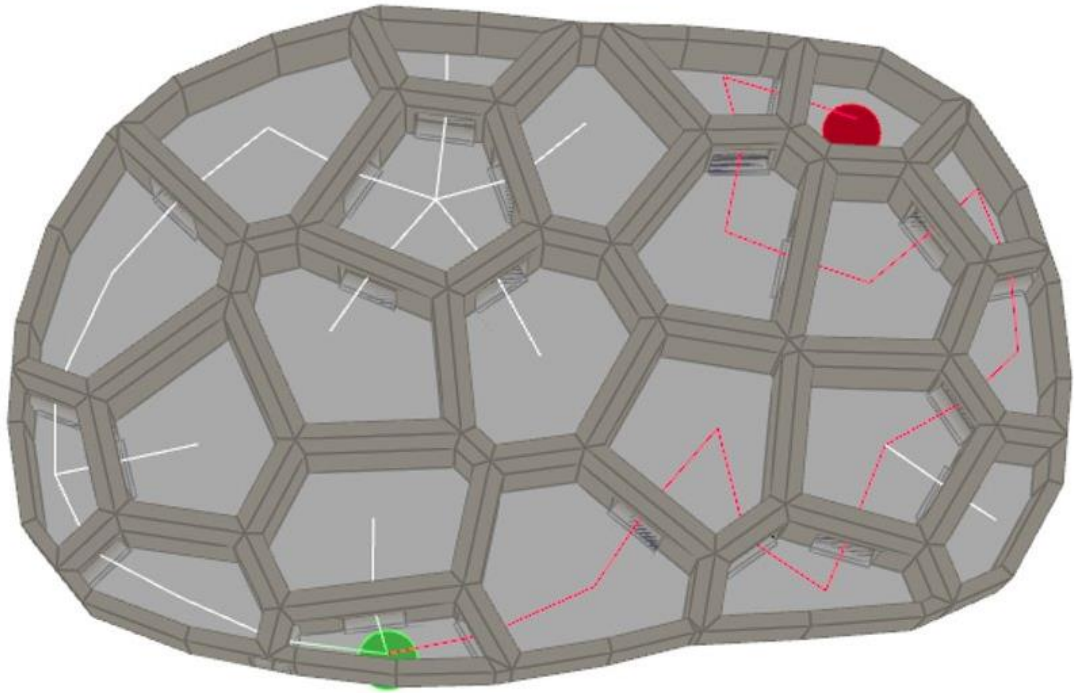


図 34：被験者が制作したダンジョン

また、到達しにくさを表す指標についての評価実験を、大学院生 6 名に対して行った。実験の様子を図 37(b)に示す。評価実験のために、HTC Vive, SteamVR, Unreal Engine を用いて VR 環境を構築し、被験者に図 35 のダンジョンを探索してもらった。実験では、全ての部屋に到達することを「探索完了」と定義し、被験者には探索完了するまで続けてもらった。そして、スタートから行き止まりの部屋までの到達時間を測定した実験結果を図 36 に示す。各地点の平均の到達時間を見ると、 $A \rightarrow B$, $C \rightarrow D$, E の順で平均の到達時間が大きくなっていることが分かる。実験結果より、到達しにくさの有効性を確認することができた。

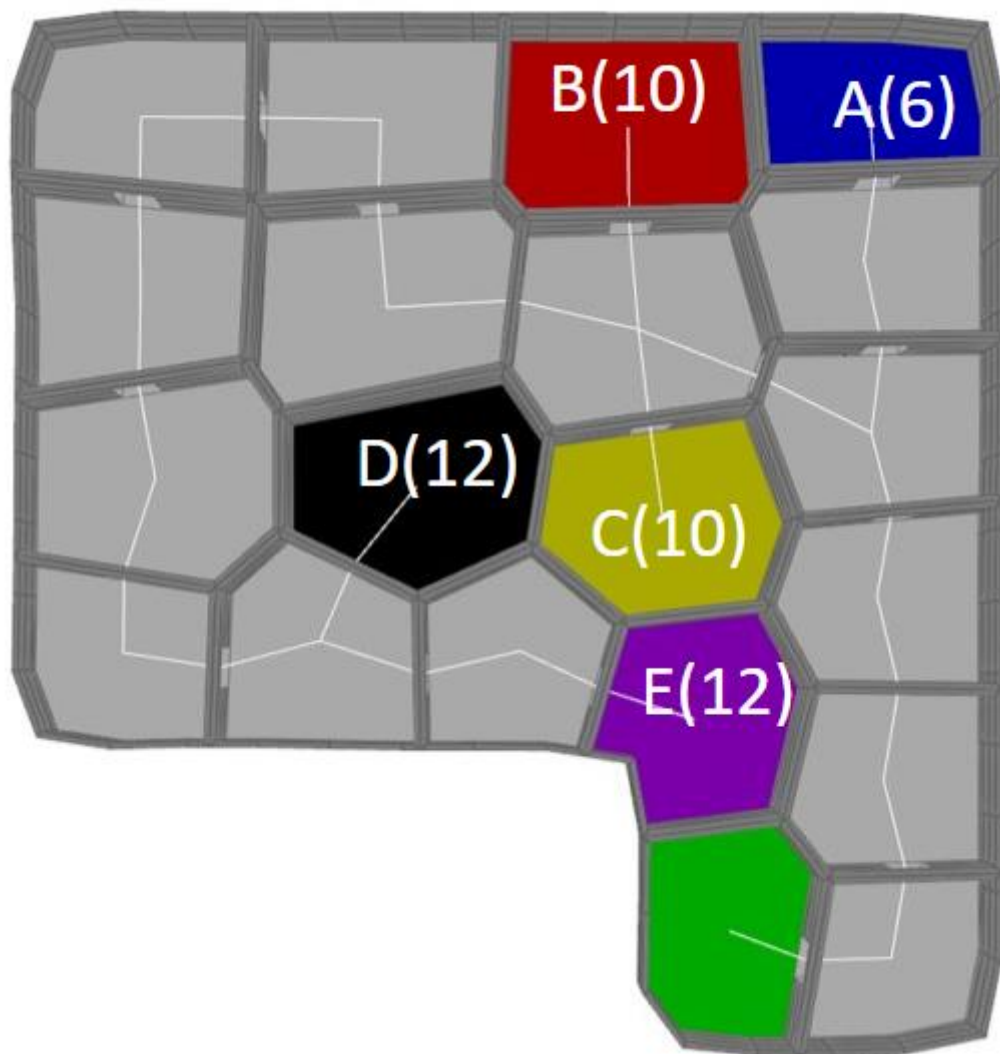


図 35：実験で使ったダンジョン．緑の部屋がスタートで，色のついた部屋までの到達時間を測定した．数値は到達しにくさを表す．

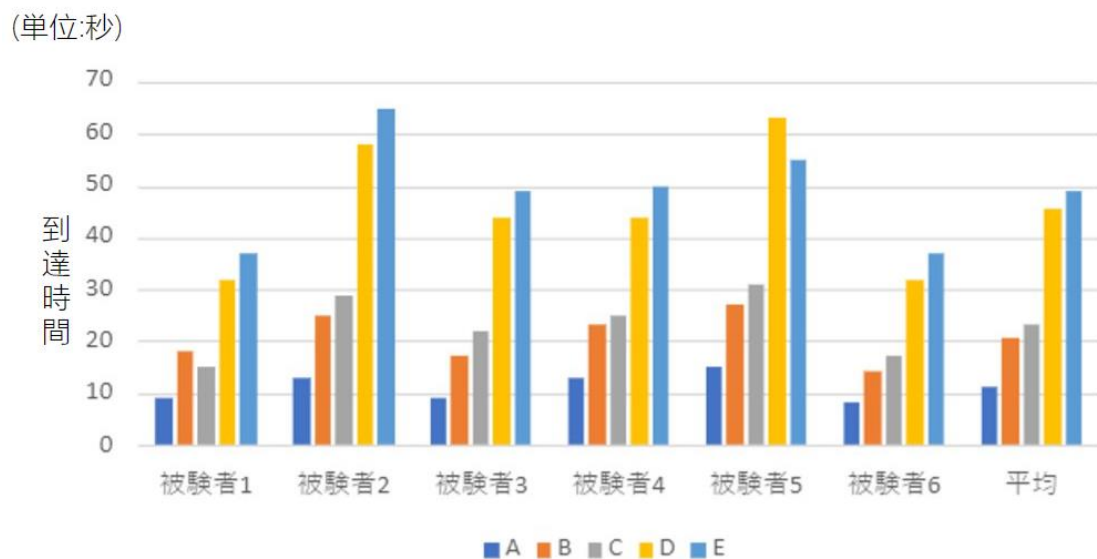


図 36：到達しにくさ，についての実験結果

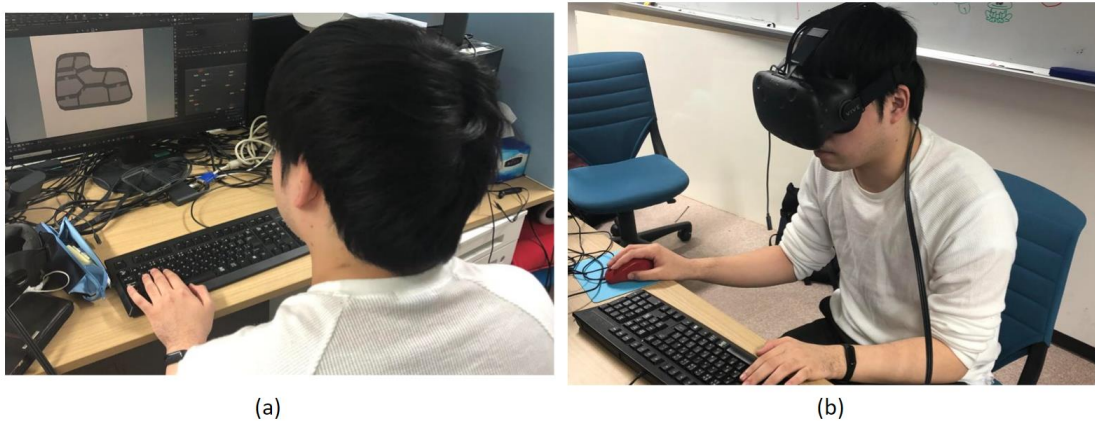


図 37：実験の様子：(a)システムの使いやすさについての評価実験，(b)到達しにくさについての評価実験．

最後に，提案システムにより制作したダンジョンの例を図 38，39，40，41 に示す．このように，複数の階層を持つダンジョンの制作が可能である．



図 38 : 本システムにより制作したダンジョンの例. 1 階層



図 39 : 本システムにより制作したダンジョンの例. 2階層

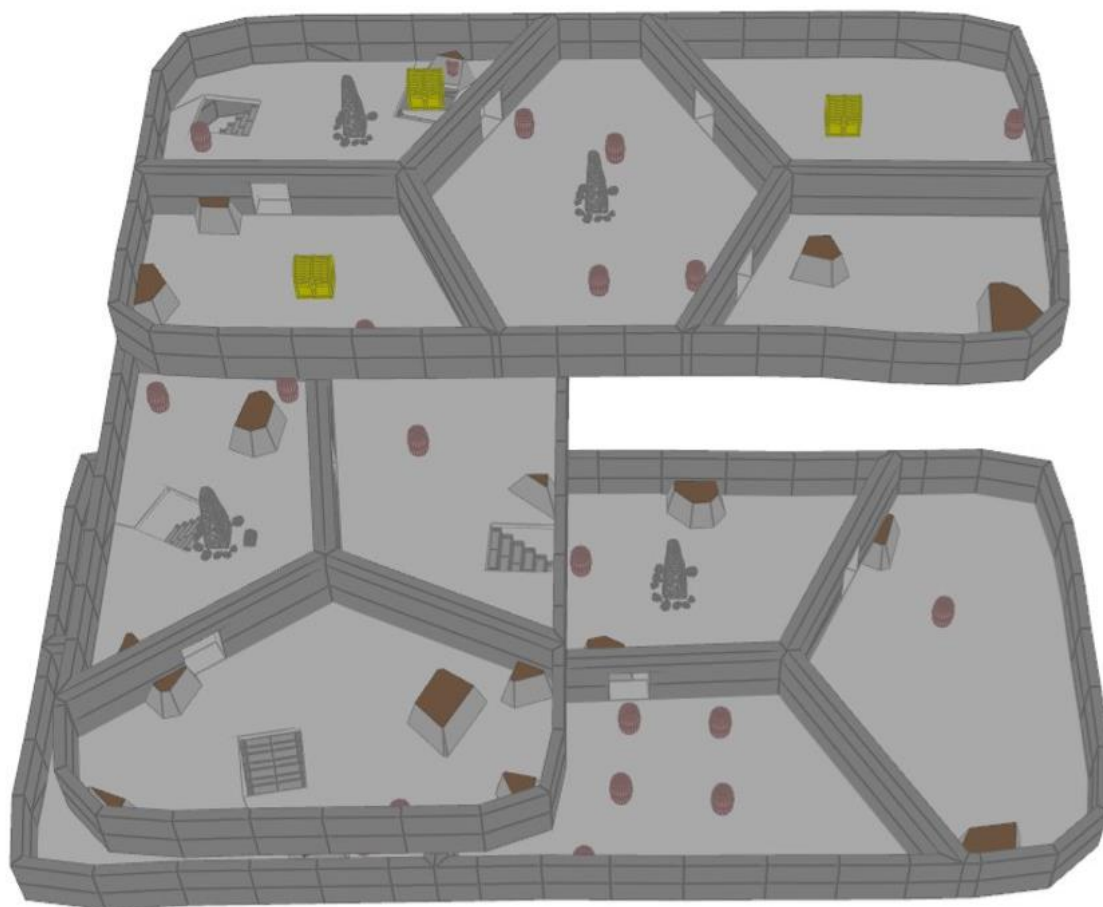


図 40：本システムにより制作したダンジョンの例. 3 階層

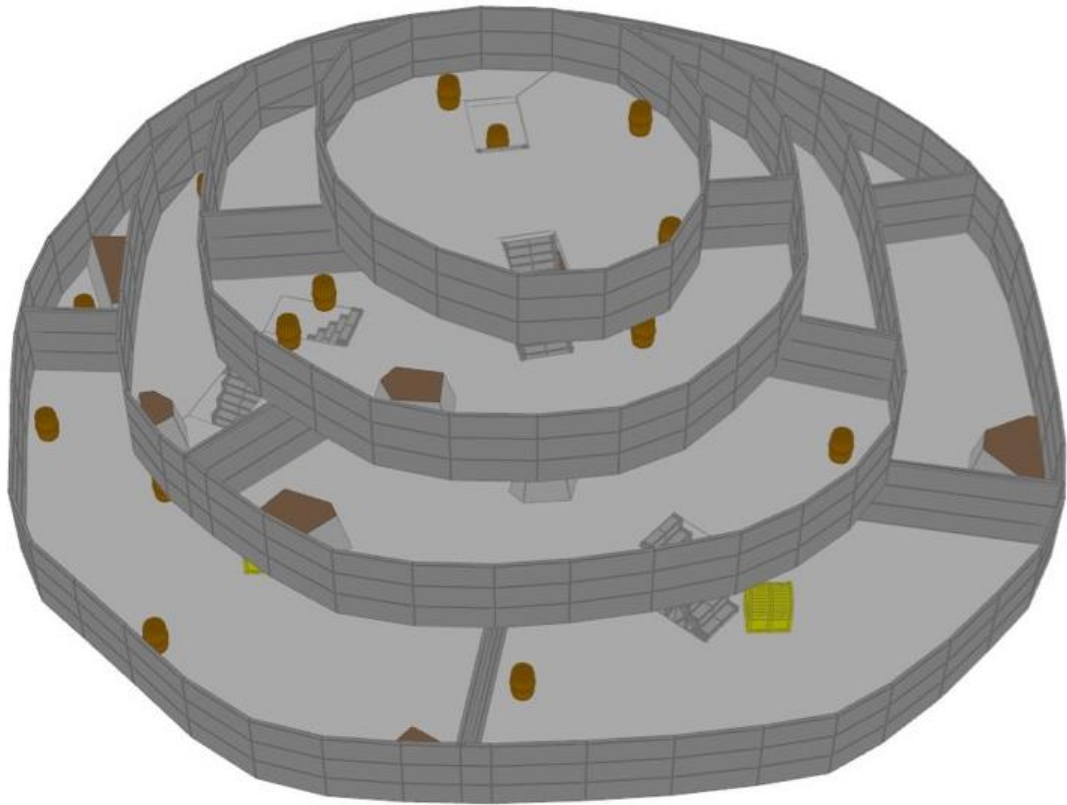


図 41：本システムにより制作したダンジョンの例. 4 階層.

第5章 おわりに

本章では、まず本研究のまとめを行う。そして、本研究の課題と今後の展望について述べる。

5.1 まとめ

本論文では、プロシージャルモデリングによる階層的三次元ダンジョンの制作支援システムを提案した。

パラメータによる制御に加えて、デザイナーを支援する機能として、ダンジョンの外郭形状のスケッチインターフェースおよびインタラクティブな編集機能を実装した。このように、プロシージャル生成と編集機能を組み合わせることで、ダンジョン制作の作業効率化とともに、デザイナーの意図を組み込むことで、特色のあるダンジョンを制作することができるようになると思う。

また、スタートから行き止まりまでの到達しにくさを表す指標を提案した。この指標により、オブジェクトの配置などに役立てることができると思う。

5.2 課題

提案システムの課題として、ユーザインターフェースがある。システムの対象ユーザとして、デザイナーを想定しており、DCC ツールに実装しているが、デザイナー支援のためには、さらに使いやすいシステムになるように、ユーザインターフェースを改善していく必要があると思われる。

もう 1 つの課題として、オブジェクトの配置がある。生成したダンジョン内にオブジェクトを配置するとき、階段や他のオブジェクトと配置場所が重なってしまうことがあるため、今後修正していきたい。

5.3 展望

今後の展望として、編集機能の強化がある。編集機能に関しては、現時点で、部屋の編集・壁の編集・出入り口の編集・スタート、ゴール位置の編集が可能である。これらの編集機能に加えて、さらに多くの編集機能を追加していき、デザイナーが介入できる要素を増やしていきたい。例えば、階段や配置したオブジェクトの配置場所などの位置の編集機能があれば良いと思われる。

謝辞

本研究を進めるにあたって、様々な方々にご指導、ご教授頂き深く感謝いたします。まず、本研究へご指導して頂きました宮田一乗先生、謝浩然先生へ、心より感謝いたします。また、宮田研究室の先輩、同期、後輩の方々にも感謝いたします。

宮田先生、謝先生には、研究テーマの方針や実装のアイデアなど、様々なアドバイスを頂きました。また、関連研究や事例など紹介いただき、大変参考になりました。さらに、ゼミにおける進捗報告や論文紹介、また、SIGGRAPH や Eurographics など最新の論文を読む機会もあり、今までより視野が広がったと思います。

先輩方には、研究や就活についてアドバイスいただきました。また、同期や後輩の方々とは、学生生活を一緒に過ごし、とても楽しく過ごすことができました。感謝いたします。

最後に日頃よりお世話になりました、宮田研究室の皆様に感謝いたします。2年間という短い期間でしたが、多くのことを学び成長できたと思います。ありがとうございました。

参考文献

- [1] M. Hendrikx, S. Meijer, J. V. D. Velden , A. Iosup, “Procedural content generation for games: A survey,” ACM Transactions on Multimedia Computing, Communications, and Applications, 2013.
- [2] R. v. d. Linden, R. Lopes , R. Bidarra, “Procedural Generation of Dungeons,” IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 6, NO. 1, 2014.
- [3] E. C. Sheffield , M. D. Shah, “Dungeon Digger: Apprenticeship Learning for Procedural Dungeon Building Agents,” CHI PLAY’18 Extended Abstracts, 2018.
- [4] R. v. d. Linden, R. Lopes , R. Bidarra, “Designing Procedurally Generated Levels,” Artificial Intelligence in the Game Design Process 2: Papers from the 2013 AIIDE Workshop, 2013.
- [5] T. Roden , I. Parberry, “From Artistry to Automation: A Structured Methodology for Procedural Content Creation,” Entertainment Computing - ICEC 2004, Third International Conference, Eindhoven, The Netherlands, 2004.
- [6] B. v. R. Lipinski, S. Seibt, J. Roth , D. Abé, “Level Graph - Incremental Procedural Generation of Indoor Levels using Minimum Spanning Trees,” IEEE Conference on Games, 2019.
- [7] J. Xu , C. S. Kaplan, “Image-Guided Maze Construction,” ACM Transactions on Graphics, 2007.
- [8] H. Pedersen , K. Singh, “Organic labyrinths and mazes,” Proc. of the 4th international symposium on Non-photorealistic animation and rendering, 2006.
- [9] 井上光平 , 浦浜喜一, “重心ボロノイ分割と最小木に基づくランダム迷路パターンによる諧調表現,” 映像情報メディア学会誌 Vol. 62, 2008.
- [10] P. H. Kim, J. Grove, S. Wurster , R. Crawfis, “Design-Centric Maze Generation,” PCG Workshop, 2019.
- [11] 水澤雅高 , 栗原正仁, “迷路問題における難易度指標の導入と実時間探索アルゴリズムの性能解析,” 人工知能学会, 2006.

- [12] 横田和幸, 船瀬新王, 藤原清悦, 内匠逸, “ヒトを対象にした迷路課題の難易度を定量的に決定するパラメータの検討,” 生体医工学, 2019.
- [13] D. Marples, D. Gledhill, P. Carter, “The Effect of Lighting, Landmarks and Auditory Cues on Human Performance in Navigating a Virtual Maze,” Symposium on Interactive 3D Graphics and Games, 2020.
- [14] Y. I. H. Parish, P. Müller, “Procedural Modeling of Cities,” Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001.
- [15] P. Muller, P. Wonka, S. Haegler, A. Ulmer, L. V. Gool, “Procedural Modeling of Buildings,” ACM Transactions on Graphics, 2006.
- [16] J. -D. Genevaux, E. Galin, E. Guerin, A. Peytavie, B. Benes, “Terrain Generation Using Procedural Models Based on Hydrology,” ACM Transactions on Graphics, 2013.
- [17] R. Smelik, T. Tutenel, K. J. d. Kraker, R. Bidarra, “Integrating procedural generation and manual editing,” Proceedings of the 2010 Workshop on Procedural Content Generation in Games, 2010.
- [18] E. W. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” Numerische Mathematik, 1959.
- [19] V. Jarník, “O jistém problému minimálním,” Práce Moravské Přírodovědecké Společnosti, 1930.
- [20] R. L. Shreve, “Statistical Law of Stream Numbers,” Journal of Geology 74, 17-37, 1966.