

公立はこだて未来大学 2022 年度 システム情報科学実習  
グループ報告書

Future University Hakodate 2022 Systems Information Science Practice  
Group Report

プロジェクト名

クリエイティブ AI

Project Name

CreativeAI

グループ名

システム班

視覚班

音響班

物語班

Group Name

System

Visual

Audio

Scenario

プロジェクト番号/Project No.

5

プロジェクトリーダー/Project Leader

穂積佳 Kei Hozumi

グループリーダー/Group Leader

システム班 新屋拓海

視覚班 坂上翠

音響班 小林祐汰

物語班 坂本珠凜

System Takumi Shinya

Visual Sui Sakagami

Audio Yuta Kobayashi

Scenario Jurin Sakamoto

グループメンバ/Group Member

新屋拓海 Takumi Shinya  
田中悠人 Yuto Tanaka  
本前斗梧 Togo Motomae  
住吉聖也 Seiya Sumiyoshi  
穂積佳 Kei Hozumi  
坂上翠 Sui Sakagami  
畑井梨里衣 Ririi Hatai  
小林祐汰 Yuta Kobayashi  
菊地陵雅 Ryouga Kikuchi  
高橋昂大 Kota Takahashi  
坂本珠凜 Jurin Sakamoto  
幸崎駿祐 Shunsuke Kosaki  
入舩真誠 Masato Irifune  
西村秀幸 Hideyuki Nishimura  
吉井史夏 Fumika Yoshii

### 指導教員

村井源 迎山和司 平田圭二 田柳恵美子 吉田博則

### Advisor

Hajime Murai Kazushi Mukaiyama Keiji Hirata Emiko Tayanagi Hironori Yoshida

### 提出日

2023年1月18日

### Date of Submission

January 18, 2023

## 概要

人工知能に創造性を持たせる研究が盛んに行われている。近年、気まぐれ人工知能プロジェクト「作家ですよ」等といった、人工知能による創造的コンテンツの自動生成の研究が行われており、特に今年度は人工知能によってイラストを自動生成させる研究が大きく進歩した。また、現在ゲーム産業も大きく発展を遂げている。しかし、実際の現場では、ゲームを作る人材が不足しているのに対し、ゲーム制作における作業量は多くなっている。そこで、本プロジェクトでは、ゲームコンテンツ制作を通して、人間と人工知能の協調作業の創造性を示すことを目的とした。ゲームに着目する意義はそこに含まれている多様な創造的要素を活かすことだ。まず、要素数の多さから、人材不足や作業量の多さが課題である。そこに人工知能による自動生成を加えることで、創造的要素の制作作業を代替することができる。次に、要素の種類が多さから、様々なアプローチから人工知能の可能性を検討することができる。この意義のもと、本プロジェクトでは、プロジェクトメンバーを4つの班に分け、活動を行った。1つ目は視覚班だ。ゲームのビジュアルや人工知能によるキャラクターデザイン制作を主に担当する。2つ目は音響班だ。ゲーム内の音楽や人工知能を用いたゲームへの没入感を向上させる BGM を担当する。3つ目は物語班だ。ゲームの世界観の設定やシナリオの執筆、人工知能によるエンディング・背景史の生成を担当する。4つ目はシステム班だ。3つの班の成果物の統合とゲームの基礎システムの開発、マップギミックの自動生成を担当する。その結果、成果物として、各班が担当した自動生成システム及び3D オープンワールドゲーム「Alice in the ideal world」を制作した。今後はプロジェクトの活動過程と成果物に関して評価を行い、成果物のブラッシュアップを図ることが望ましい。

**キーワード** 人工知能, プログラミング, 自動生成

(※文責: 穂積佳)

# Abstract

Research on artificial intelligence with creativity is flourishing. In recent years, research on automatic generation of creative contents by artificial intelligence, such as the whimsical artificial intelligence project "It's a Writer", has been conducted, and especially this year, research on automatic generation of illustrations by artificial intelligence has made great progress. In addition, the game industry is currently making great progress. However, while there is a shortage of human resources to create games in the actual field, the amount of work involved in game production is increasing. Therefore, this project aims to demonstrate the creativity of cooperative work between humans and artificial intelligence through the creation of game contents. The significance of focusing on games is to take advantage of the various creative elements contained in them. First of all, the large number of elements makes the lack of human resources and the amount of work required a challenge. The addition of automatic generation by artificial intelligence can replace the work of creating creative elements. Second, the variety of elements allows us to examine the potential of artificial intelligence from various approaches. With this in mind, the project members were divided into four groups for this project: The first group was the visual group. The second is the sound group, which is in charge of creating music and artificial intelligence for the game. The second group is the sound group, which is in charge of the in-game music and background music that enhances the immersive experience of the game using artificial intelligence. The fourth group is the System Group, which is responsible for integrating the three groups' work, developing the game's basic system, and automatically generating map gimmicks. As a result, each group has produced a 3D open world game "Alice in the ideal world" as well as an auto-generated system as their deliverables. In the future, it is desirable to evaluate the project's activity process and deliverables, and to brush up the deliverables.

**Keyword** artificial intelligence, programing, automatically generation

(※文責: 穂積佳)

# 目次

<b>第 1 章</b>	<b>初めに</b>	<b>1</b>
1.1	背景 . . . . .	1
1.2	目的 . . . . .	1
1.3	課題 . . . . .	1
<b>第 2 章</b>	<b>プロジェクト概要</b>	<b>3</b>
2.1	課題の設定 . . . . .	3
2.2	到達目標 . . . . .	3
2.3	班ごとの課題の割り当て . . . . .	3
2.3.1	システム班 . . . . .	3
2.3.2	視覚班 . . . . .	4
2.3.3	音響班 . . . . .	4
2.3.4	物語班 . . . . .	4
<b>第 3 章</b>	<b>中間発表までの開発</b>	<b>6</b>
3.1	システム班 . . . . .	6
3.1.1	マップ . . . . .	6
3.1.1.1	マップ概要 . . . . .	6
3.1.1.2	小屋 . . . . .	6
3.1.1.3	小屋設計 . . . . .	7
3.1.1.4	小屋内装 . . . . .	8
3.1.1.5	森 . . . . .	9
3.1.1.6	城 . . . . .	9
3.1.2	キャラクター . . . . .	10
3.1.2.1	主人公の制御 . . . . .	10
3.1.2.2	アイテム取得 . . . . .	10
3.1.2.3	キャラクターの効果音 . . . . .	11
3.1.2.4	NPC の制御 . . . . .	11
3.1.2.5	小屋 NPC . . . . .	11
3.1.2.6	カメラ . . . . .	12
3.1.3	ギミック . . . . .	12
3.1.3.1	小屋 . . . . .	13
3.1.3.2	1 つ目のギミック . . . . .	13
3.1.3.3	2 つ目ギミック . . . . .	14
3.1.3.4	森・城 . . . . .	15
3.1.3.5	自動生成 . . . . .	15
3.1.4	シナリオ読み取りシステム . . . . .	16
3.1.4.1	シナリオの読み取りシステムの概要 . . . . .	16

	3.1.4.2	表示画面への実装	16
3.2		視覚班	16
	3.2.1	コンセプトアート	16
	3.2.2	自動生成	17
	3.2.2.1	キャラクターイメージの共有	17
	3.2.2.2	概要	17
	3.2.2.3	3D モデルの自動生成	17
	3.2.2.4	生成目標キャラクターのビジュアル	18
	3.2.2.5	生成アプローチ案	18
	3.2.2.6	MonsterMesh	18
	3.2.2.7	AvatarClip	19
	3.2.2.8	Text2Mesh	19
	3.2.2.9	キャラクターデザイン画の作成	20
	3.2.2.10	顔	21
	3.2.2.11	実写画像の自動生成	21
	3.2.2.12	スタイル変換	22
	3.2.2.13	服	23
	3.2.3	素材作り	24
	3.2.3.1	概要	24
	3.2.3.2	3D オブジェクト	25
	3.2.3.3	デザインの詳細	25
	3.2.3.4	立ち絵	26
	3.2.4	UI	26
3.3		音響班	27
	3.3.1	目標	27
	3.3.2	先行研究	27
	3.3.3	方針	29
	3.3.4	データベース構築	29
	3.3.5	曲決定システム	30
	3.3.6	類似曲検索システム	30
	3.3.6.1	手法	30
	3.3.6.2	ニューラルネットワーク構築	30
	3.3.6.3	曲のベクトル化	31
	3.3.6.4	類似曲の検索	31
	3.3.7	シーンに合った曲の選択システム	31
	3.3.7.1	手法	31
	3.3.7.2	文章からの曲決定	32
	3.3.7.3	フィールドからの曲決定	32
	3.3.8	曲繋ぎ場所検索システム	32
	3.3.9	曲繋ぎシステム	33
3.4		物語班	33
	3.4.1	世界観設定	33

3.4.1.1	世界観決定までの経緯 . . . . .	33
3.4.1.2	世界観設定 . . . . .	34
3.4.1.3	場所設定 . . . . .	34
3.4.1.4	キャラクター設定 . . . . .	35
3.4.2	シナリオ執筆 . . . . .	37
3.4.2.1	シナリオについて . . . . .	37
3.4.2.2	あらすじ . . . . .	37
3.4.2.3	シナリオの流れ . . . . .	37
3.4.3	背景史 . . . . .	39
3.4.3.1	目標 . . . . .	39
3.4.3.2	分析の目的 . . . . .	39
3.4.3.3	作品選定 . . . . .	39
3.4.3.4	分析準備 . . . . .	40
3.4.4	マルチエンディング . . . . .	41
3.4.4.1	目標 . . . . .	41
3.4.4.2	分析の目的 . . . . .	42
3.4.4.3	作品選定 . . . . .	42
3.4.4.4	分析準備 . . . . .	43
3.4.4.5	分析方法 . . . . .	43
<b>第 4 章</b>	<b>中間発表</b>	<b>45</b>
4.1	概要 . . . . .	45
4.2	成果物 . . . . .	45
4.2.1	ポスター . . . . .	45
4.2.2	発表用スライド . . . . .	46
4.2.3	動画 . . . . .	47
4.3	評価 . . . . .	47
<b>第 5 章</b>	<b>前期の反省と後期の方針</b>	<b>50</b>
5.1	システム班 . . . . .	50
5.1.1	前期反省 . . . . .	50
5.1.2	後期方針 . . . . .	50
5.2	視覚班 . . . . .	50
5.2.1	前期反省 . . . . .	50
5.2.2	後期方針 . . . . .	51
5.3	音響班 . . . . .	51
5.3.1	前期反省 . . . . .	51
5.3.2	後期方針 . . . . .	52
5.4	物語班 . . . . .	52
5.4.1	前期反省 . . . . .	52
5.4.2	後期方針 . . . . .	52
5.5	全体 . . . . .	53
5.5.1	前期反省 . . . . .	53

5.5.2	後期方針	54
<b>第 6 章</b>	<b>成果発表までの開発</b>	<b>55</b>
6.1	システム班	55
6.1.1	マップ	55
6.1.1.1	森	55
6.1.1.2	城	55
6.1.1.3	城設計	56
6.1.1.4	城マップのシーン遷移	57
6.1.2	キャラクター	58
6.1.2.1	3Dモデルの差し替え・取り込み	58
6.1.2.2	瞬き機能の追加	58
6.1.2.3	主人公制御スクリプトの改良	58
6.1.2.4	NPCの制御	59
6.1.2.5	小屋NPC	59
6.1.2.6	城NPC	60
6.1.2.7	今後の展望	60
6.1.3	ギミック	61
6.1.3.1	城	61
6.1.3.2	エリア作成	61
6.1.3.3	自動生成	61
6.1.3.4	ギミック自動生成の背景	61
6.1.3.5	ギミックの分類	62
6.1.3.6	ギミックの種類決定	62
6.1.3.7	ギミックデータの収集	62
6.1.3.8	ギミックデータの分析	62
6.1.3.9	Unityでの実装	63
6.1.3.10	今後について	63
6.1.4	シナリオ読み取りシステム	63
6.1.4.1	シナリオの読み取りシステムの概要	63
6.1.4.2	シナリオの読み取りシステムのデータ加工	63
6.1.4.3	表示画面への実装	64
6.1.4.4	メニューの構成と基本機能	64
6.1.4.5	表情差分	65
6.1.4.6	オープニング	66
6.1.4.7	エンディング	66
6.1.4.8	立ち絵制御	67
6.1.5	BGM切り替えシステム	68
6.1.5.1	BGM切り替えシステムの引継ぎ	68
6.1.5.2	ゲーム内への実装	68
6.1.6	ゲームの統合	69
6.1.6.1	各種素材	69

	6.1.6.2	城エリアの統合	70
6.2		視覚班	71
	6.2.1	2D 自動生成	71
	6.2.1.1	2D 自動生成概要	71
	6.2.1.2	キャラクターデザイン案の画像自動生成概要	71
	6.2.1.3	使用技術	71
	6.2.1.4	ワークフロー	72
	6.2.1.5	シナリオを読む、情報抽出	72
	6.2.1.6	デザイン案を出力	73
	6.2.1.7	デザイン案を選定	89
	6.2.1.8	デザイン完成	90
	6.2.2	3D 自動生成	92
	6.2.2.1	3次元モデル自動生成概要	92
	6.2.2.2	使用技術	92
	6.2.2.3	各パーツの生成	93
	6.2.2.4	パーツを組み合わせる (CLIP 利用)	93
	6.2.2.5	リグ付け	93
	6.2.3	アセット制作	94
	6.2.3.1	キャラクター立ち絵	94
	6.2.3.2	キャラクター 3D モデル	100
	6.2.3.3	背景	100
	6.2.3.4	キャラアイテムの絵	105
	6.2.3.5	タイトルロゴ	111
	6.2.3.6	エンディングイラスト	112
	6.2.3.7	3D	113
	6.2.3.8	UV 展開	113
6.3		音響班	115
	6.3.1	データベース構築	115
	6.3.2	ループ素材生成	116
	6.3.3	音源素材集め	118
	6.3.4	前期システム改良	118
	6.3.4.1	入出力データ調整	118
	6.3.4.2	曲遷移場所検索の改良	119
	6.3.4.3	曲繋ぎ方法の改良	119
	6.3.5	オーディオビジュアライザーの作成	119
	6.3.6	システム統合	120
6.4		物語班	122
	6.4.1	メインシナリオ作成	122
	6.4.1.1	前期からの変更・追加点	122
	6.4.1.2	メインシナリオ執筆	124
	6.4.1.3	自動生成したシナリオを組み込むにあたっての調整	125
	6.4.2	背景史	125

6.4.2.1	データ収集	125
6.4.2.2	カテゴライズ	126
6.4.2.3	分析	126
6.4.2.4	自動生成アルゴリズム	129
6.4.2.5	ファインチューニング	130
6.4.2.6	両出力結果の比較	131
6.4.2.7	文章選定	132
6.4.2.8	文章整形	132
6.4.3	マルチエンディング	135
6.4.3.1	分析準備	135
6.4.3.2	相関係数	136
6.4.3.3	分析方法	137
6.4.3.4	分析結果	137
6.4.3.5	アルゴリズム作成	138
6.4.3.6	文章の自動生成	138
6.4.3.7	入力用プロンプト作成	139
6.4.3.8	出力と選定, 文章整形	141
<b>第 7 章</b>	<b>成果発表</b>	<b>142</b>
7.1	概要	142
7.2	制作物	142
7.2.1	ポスター	142
7.3	評価	144
7.4	課題と検討	146
7.4.1	システム班	146
7.4.2	視覚班	146
7.4.3	音響班	147
7.4.4	物語班	147
<b>第 8 章</b>	<b>まとめ</b>	<b>149</b>
8.1	プロジェクト全体の成果	149
8.2	各グループの成果	149
8.2.1	システム班	149
8.2.2	視覚班	150
8.2.3	音響班	150
8.2.4	物語班	150
8.3	今後の展望	152
8.3.1	全体	152
8.3.2	システム班	152
8.3.3	視覚班	152
8.3.4	音響班	154
8.3.5	物語班	154



# 第 1 章 初めに

## 1.1 背景

近年、人工知能を用いた、文章や画像などの自動生成技術が大きな注目を集めている。文章では 2021 年に公開された AI のベリすとや 2022 年の ChatGPT など、画像では Midjourney や Stable Diffusion などの簡単に高クオリティなアウトプットを行えるツールの登場により、研究者だけでなくアーティストや一般の人々にまで生成系 AI が注目をされている。

一方、日本の代表的なエンターテインメント産業にゲーム事業がある。ゲームはシナリオ、ビジュアル、サウンド、システムなど様々な創作的なプロセスを経て 1 つの作品が出来上がる。

本学の前年度から続いている本プロジェクトでは、バトル作品やホラー作品がベースのゲームシナリオの生成、RPG 内のイベント順を自動的につなぎ合わせるシステム、ダンジョンの自動生成、BGM の自動選択システムの考案など、いくつかの創作的プロセスを伴うゲーム開発を人工知能の技術をもとに自動化をめざし進めてきた。

(※文責: 穂積佳)

## 1.2 目的

本プロジェクトの目的は、ゲームを構成するいくつかの要素を人工知能を用いた自動生成技術で実際に制作し、そのプロセスと成果物から、人工知能を用いた創造的な作業の代行を検討、評価を行うことである。今年度はアドベンチャーゲーム×オープンワールドゲームの制作を目標に人工知能システムとゲームの開発を進めていく。アドベンチャーゲームはテキストやグラフィックスによりプレイヤーの置かれている状況が提示され、プレイヤーが思考、選択を行いシナリオを進めていくゲームジャンルである。他のコンピュータゲームジャンルと比較してアドベンチャーゲームはシナリオが明確に存在しその内容をプレイヤーが体験することに趣を置いていることで区別される。同じくプレイヤーの思考によりシナリオを進行させていくロールプレイングゲームやシュミレーションゲームに比べアドベンチャーゲームはプレイヤーが操作するキャラクターやゲーム内のキャラクターやオブジェクトなどに能力値（経験値、攻撃力、好感度）などの数値が非常に少ない、存在しないという特徴がある。オープンワールドゲームはゲーム内世界を構成する仮想空間が移動制限が少なかったり、プレイヤーに操作的行動の自由やシナリオ的行動の自由を体験させるゲームジャンルである。今回目標としているアドベンチャーゲームは不思議の国のアリスをテーマとした世界を舞台にプレイヤーがゲーム内の様々なギミックをこなしていきゲームクリアを目指すものである。

(※文責: 穂積佳)

## 1.3 課題

目標を達成するために解決すべき課題は以下の通りである。

- ・制作するゲームの規模の決定
- ・制作するゲーム内容の大枠な決定
- ・制作するゲームを構成する要素の抽出
- ・自動生成させるゲーム内要素の決定
- ・分析対象とするゲーム作品の大枠の決定
- ・自動生成する際の分析方法の決定
- ・要素を自動生成させる方法の選択
- ・自動生成の実装
- ・ゲームシナリオの制作
- ・キャラクターやワールドに設置するオブジェクトの制作
- ・BGM, SE の選択, 制作
- ・ゲームギミックの実装
- ・ゲーム内要素の統合

本プロジェクトでは、制作するゲームの規模の決定、制作するゲーム内容の大枠の決定、制作するゲームを構成する要素の抽出、自動生成させるゲーム内要素の決定、分析対象とするゲーム作品の大枠の決定、自動生成する際の分析方法の決定、要素を自動生成させる方法の選択、自動生成の実装、ゲームシナリオの制作、キャラクターやワールドに設置するオブジェクトの制作、BGM や SE の選択や制作、ゲームギミックの実装、ゲーム内要素の統合、を課題とした。

(※文責: 穂積佳)

## 第2章 プロジェクト概要

### 2.1 課題の設定

本プロジェクトは、ゲームコンテンツを成果とした、人工知能を用いた創造性の再現を行うことが目的である。初めにどのジャンルのゲームを作るかを話し合った。その結果、謎やギミックを解いていく3Dオープンワールドゲームを制作することとなった。そして、ゲームのジャンルに沿いつつ人工知能を用いて創造性の再現を行うため「マップギミックの自動生成」、「3dモデルの自動生成」、「キャラクターデザインの自動生成」、「マルチエンディングの自動生成」、「現実逃避しているキャラクターの背景史の自動生成」、「曲繋ぎの自動化」の実現を課題に設定した。今年度は課題の設定について新規性を重視し設定した。

(※文責: 入船真誠)

### 2.2 到達目標

本プロジェクトでは、ゲーム要素ごとに4班に分かれ各班で新規性の高い自動生成システムの完成と、完成した自動生成システムが組み込まれたオリジナルゲームの完成を目標としている。システム班は、マップギミックの自動生成システムとゲームシステムの開発、他班の制作したゲームの要素を1つにまとめ、より快適にゲームをプレイさせること、視覚班は3dモデルの自動生成システムとキャラクターデザインの自動生成システムの完成とキャラクターやゲームに必要な小物類のCGアセットの制作、音響班は曲繋ぎの自動化の実現と音響アセットの制作・監督、物語班はマルチエンディングの自動生成システムと現実逃避しているキャラクターの背景史の自動生成システムの完成とメインシナリオの執筆を目標としている。

(※文責: 入船真誠)

### 2.3 班ごとの課題の割り当て

#### 2.3.1 システム班

システム班では他班が作ったものを統合し、1つのゲームとして完成させることを目標としている。大きく分けてマップの作成、マップギミック、キャラクターの操作、UIの作成を班内で分けて作業を行っている。その中でも今年度のシステム班は「マップギミックを自動生成」を目標としている。システム班の中で重要であることは他班とのイメージの共有とその実装である。そのために話し合いを工夫しながら行った。

(※文責: 本前斗梧)

### 2.3.2 視覚班

視覚班は、主にゲームのビジュアル要素のデザイン、制作を担当している。ゲーム制作においてビジュアル要素が必要な理由は2つある。まず、ビジュアル要素はゲームを楽しむために必要不可欠だからだ。人間が受け取る情報のうち8割は視覚からの情報だ。これはゲームにおいても例外ではなく、ビジュアル要素はストーリーやキャラクターの魅力を伝えると共に、没入感のあるゲーム体験を提供する。次に、ビジュアル要素はゲーム制作の進行を助けるからだ。ゲームを制作するにあたり、チーム内で世界観などのイメージの共有は必須だ。そこで、一目でわかる画像を用いたイメージの共有を試みた。活動内容は以下の通りだ。まず、ゲーム内のCGアセットの制作だ。キャラクターの立ち絵と3Dモデルや小物の3Dモデル、UIテンプレートの制作を行った。次にゲームの世界観のイメージ共有のためのコンセプトアートの制作を行った。複数枚制作し、プロトタイプ制作の際のイメージ共有に使った。最後に、最も重要であるキャラクターの見た目の自動生成だ。この活動においてはゲームに登場するキャラクターのビジュアルを自動生成を用いて制作することを目標としている。AIツールを使用し、それに伴う創意工夫を行う。そして、AIを用いて生成したキャラクターデザインをゲーム内に実装できる形に持っていく。この目標の有意性は、キャラクターデザインの時のアイデア出しを簡易化することと、AIにしか作り出せないようなキャラクターデザインの生成をすることである。

(※文責: 坂上翠)

### 2.3.3 音響班

音響班は、ゲームシーンへの没入感を向上させるBGM作成を目標として、具体的には、シーンや場面ごとに切り替わるBGMをより滑らかに、シームレスにつなぐことを検討している。すでにある技術として、インタラクティブミュージックと呼ばれるものがあるが、実装するにあたって専用の曲を用意し、タイミングを制御して切り替える必要がある。そこで作業効率化やインタラクティブミュージック実装のハードルを低くするために、専用の曲でなくともより滑らかに変化するAIを用いたインタラクティブミュージックの実装を目指す。

(※文責: 菊地陵雅)

### 2.3.4 物語班

物語班はゲームの世界観の設定や物語の執筆を担当している。今年度は「不思議の国のアリス」をゲームの世界観として、これをベースに各班でゲーム制作を進める。シナリオは現実世界に辟易した人間が迷い込む「不思議の国」に主人公アリスが迷い込み、同じ境遇であるこの国の住人達と出会うというものだ。また、今年度の物語班の目標は「現実逃避しているキャラクターの背景史の自動生成」と「マルチエンディングの自動生成」としている。物語班内ではそれぞれの自動生成アルゴリズムの構築のために、メンバーを背景史担当とエンディング担当に分けて活動を行う。なお、本プロジェクト内での背景史とは「ゲームシナリオ中では語られないキャラクターのバックストーリー」と定義することにした。キャラクターの背景史は複雑で多岐にわたるため、今年度は世界観と合うように「現実逃避しているキャラクター」に限定して、背景史の自動生成アルゴリズム

CreativeAI

の構築を目指す。

(※文責: 坂本珠凜)

## 第 3 章 中間発表までの開発

### 3.1 システム班

#### 3.1.1 マップ

##### 3.1.1.1 マップ概要

マップは、小屋、森、城の全 3 面構成となっており、オープンワールドとして作成予定である。世界観は、登場キャラクターの現実からの逃げ場となるためポップな世界観であり、それに加えてポップながらも不気味さを感じさせるものとした。また、面同士の行き来を可能にすることで、ステージが徐々に広がり開拓していくようなイメージのマップ構成とした。マップは小屋から始まり、様々なギミックやキャラクターと出会いながら森、城へ進んでいく。小屋、森、城の 3 面構成になった理由としては、他班の成果物の統合やギミックの自動生成システムの開発など、システム班が行う作業と並行して作成できるマップの最低ラインが 3 つであると判断したためだ。今後、余裕があれば森をお茶会・迷路に分割する予定である。また、ステージ構成とした経緯として、システム班内の話し合いでは、オープンワールドとして一つの面に小屋、森、城を実装する計画だった。しかし、シナリオ班との話し合いにより、あまりに自由に探索できてしまうとストーリーが複雑になることや、物語の流れがシステム面で見ても難しくなるという問題点が挙げられた。よって、「ある面をクリアすると別の面に行ける」といった、複数の面で構成することに決定した。

前期中間発表の段階では、プロトタイプを作成した。プロトタイプでは小屋ステージをメインに据え、物語のスタートから小屋を抜け森に出るところまでを作成した。小屋ステージ以降のステージである森や城は、後期に作成予定である。また、マップにはギミックを組み込むため、ギミックと相性の良くなるようなマップ構成や建物造りとした。

(※文責: 住吉聖也)

##### 3.1.1.2 小屋

小屋の設計には、ゲームエンジンの Unity の公式アドオンである ProBuilder を利用した。ProBuilder は、Unity 内でモデリングができるアドオンであり、小物などのオブジェクトから建物などの大きなオブジェクトまで作成することができる。ProBuilder で作成したオブジェクトは自動的にメッシュが付くため、キャラクターなどの当たり判定のあるオブジェクトと干渉することができる。

ProBuilder の仕様として、オブジェクトの面は、片側のみ表示されるようになっている。例えば、四角形の面の板をオブジェクトとして追加するとした時、片側から見れば面があり、それには当たり判定もあるが、その面を裏側から見た時は、面は存在せず当たり判定も無い仕様である。作業していくにあたり、もう一つの仕様の様なものが見つかった。「様なもの」という表現をしている理由は、ある条件が満たされると発生するためである。ある条件とは、まず、ProBuilder で作成した 2 つ以上のオブジェクトのメッシュが重なっている時。次に、その重なっているオブジェクトの面が鋭角を成すように重なっている時である。例として、長方形のオブジェクトを十字にクロスするように重ね、それらが重なることで出来た角などである。これらの条件の時、角になった部分

の「辺」に不具合が起こる。起こり易い例は、部屋の四隅の壁である。この四隅の壁の「辺」に、ある一定以上のスピードでキャラクターが辺に向かって進もうとすると、オブジェクトの当たり判定が消え、キャラクターがオブジェクトをすり抜けてしまう。これはオブジェクトの大きさの変更やメッシュの厚みを変更してもその不具合は改善されなかった。また、アセットに付いているメッシュでも起こる。例えば、本棚を壁際に配置すると、壁のメッシュと本棚のメッシュが重なる。そして、キャラクターが壁と本棚の境目を通ろうとすると壁抜けが発生してしまう。

小屋を作成するにあたって、部屋の四隅、家具と壁の境目、廊下と部屋の繋ぎ目、ドア枠などで壁抜けが起きてしまうバグが多数発生した。このバグを修正するために壁に厚みを持たせたり、家具と壁の間を少し広めにとったりすることを試したが、部屋としてのクオリティが低くなってしまったため、班内で改善案について話し合う機会を設けた。このバグの修正作業をしているうちに、重なったメッシュの角度が緩やかであればバグが発生しないことが判明した。例えば、三角形の部屋があるとしてその隅の角は、すり抜けるバグが起こるが、五角形や六角形になるとバグが起こりにくい。この特性を活かし最も安全かつ作業量の少ない改善案として、空のオブジェクトにメッシュを付けて配置し、すり抜けるバグが起こる部分に対して、からオブジェクトのメッシュ面が斜めになるように設置することにした。空のオブジェクトは見えない透明のオブジェクトである。この方法は、重なることで角になった部分の角数を増やすということであり、直角だった角を緩やかな角度にすることで、壁抜けバグを防ぐ様に改善した。

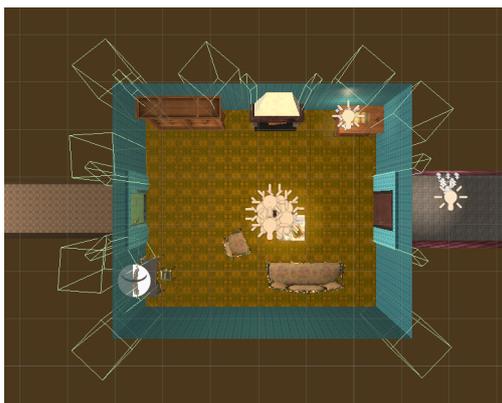


図 3.1 2つ目の部屋



図 3.2 空オブジェクトの配置

(※文責: 住吉聖也)

### 3.1.1.3 小屋設計

小屋は2つの部屋があり、それらが廊下で一列につながっている。1つ目の部屋は、プレイヤーが操作し始める部屋であり、縦穴の底のような形状となっている。そして、この部屋は、筒状のオブジェクトを広げて壁になるように配置した。天井が見えないくらい長く続いている縦穴のように見せるため、壁をある程度の高さまで伸ばし、少しずつ内径を狭くしていくことで永遠に続く縦穴をデザインした。筒状にしたオブジェクトは多角形であり、角数が多いほど処理が重くなってしまふ。処理が重くならず円形に見せるため角数を調節して20角形とした。また、ProBuilderの仕様として、厚みのあるオブジェクトは中がくり抜かれているため、面だけで構成されたオブジェクトとなる。これにより、一部をくり抜いてオブジェクトに穴を開けたり、一部を切り取ってオブジェクトを小さくしたりすることが難しい。そのためドア用の穴を開ける際に、任意のサイズに切り抜けないという問題点が生まれた。これを解決するために、ProBuilderのCut Tool機能を利用

して、2枚で構成された壁の面をそれぞれ作りたいドア穴の大きさに、手動でできるだけ正確に切り抜くことにした。この時、壁の中が空になっているため、2枚の面でできた壁の間は、内側が見えるようになってしまう。そこに Bridge Edges の機能を使って新たに面を加えることで、壁としての厚みを再現する様にした。ProBuilder では、オブジェクトをつなげて家を設計することや、形を変えずにシンプルな作りの建築であれば簡単に制作することができるが、1つ目の部屋で発生した問題点を踏まえ、複雑な形や細かい造形を ProBuilder で再現することは難しいと考えた。よって建物をモデリングするために Blender という 3D モデリングソフトの利用を検討している。

2つ目の部屋は、長方形で作成し、シンプルなものとした。四角いオブジェクトを壁や床として複数配置するのではなく、1つの四角いオブジェクトを Flip Normals という機能を使い、面の表裏を反転させた。通常は、四角形の内側から外側を見た時、面はないが、Flip Normals の機能を使えば内側に面が出来る。そうすることで、四角形の内側から見た時に内壁がある様に設計した。これは、オブジェクトとして当たり判定もあるため、部屋として利用することができる。しかし、四角形の外から反転されたオブジェクトを見た時、面がないため内装が剥き出しになってしまうという欠点がある。小屋として必要なのは、小屋内だけであるので、プロトタイプではこの造りとした。

3つ目の部屋は、森へ出るための出口があり、ドアを利用したギミックを実装するために六角形の部屋とした。1つ目の部屋と同様に、多角形の筒状のオブジェクトを利用し、六角形の壁を配置した。また3つ目の部屋はギミックがメインのため部屋の作りはシンプルなものとした。

それぞれの部屋を繋ぐ廊下もシンプルな長方形として配置し、1つ目から2つ目までの廊下はクラック状にして単調さを回避する様にした。また、廊下の長さは実際にキャラを操作して退屈にならない程度の長さにし、ドアの大きさやドア穴の大きさに合うよう調節した。

(※文責: 住吉聖也)

#### 3.1.1.4 小屋内装

内装には Unity のアセットを利用し、既存の 3D オブジェクトやマテリアルなどをインストールしインテリアとして配置した。不思議の国の世界観は、古い家具や木の家具、西洋風の煌びやかな家具などのイメージがあり、それらに沿ったアセットを Unity Asset Store で調べ、インストールした。家具などを自分達で作成せずにアセットを利用した理由として、アセットはマテリアルやメッシュなどが元々ついている物が多く、作業時間の短縮やクオリティの向上が見込めるため、Unity のアセットを利用した。

1つ目の部屋のデザインコンセプトとしては、不思議の国であることがはっきり分かる様にした。そのために、壁紙を赤色に床を赤黒のチェックにすることで統一感を出した。また、インテリアも系統や大きさを変更して散らばるように配置し、主人公と一緒に部屋に落ちてきたようにした。この部屋にはギミックはなく、イベントが発生するという設定のため、インテリアを複雑にはせず、まず世界観を認識してもらうような作りとした。使用した家具は机や棚、イスなどの小屋内に存在しそうな木製の家具に絞って配置した。照明は、ライトであると世界観と合わずに不自然になってしまうため、蝋燭やランプなどの古いものを配置し、そこに光源があるように Unity の Light を重ねて配置した。この時、炎の色合いを出すために Light の色合いを温かみのある色合いにし、光量も抑えた。

2つ目の部屋のデザインコンセプトは、生活感のある部屋をイメージした。この部屋に用いたアセットは、暖炉やソファ、本棚、開かれた本、ドレッサーなどを配置した。ここに置かれた家具

は、西洋風の豪華な家具にすることで普通とは離れた夢の国らしさを演出した。また、開かれた本やその周りに蝋燭を配置することで、最近まで誰かがいたような雰囲気を感じられる様にした。さらに、2つ目の部屋はギミックを実装するため、ギミック担当と話し合い、ギミックとして活用できそうな暖炉を配置することにした。壁紙や床は1つ目の部屋とは大きく変えて、青や黄色を使用した。そうすることで、本ゲームの世界観に沿ったポップな夢の国となるようにした。

3つ目の部屋のデザインコンセプトは、ギミックを重要視し、出来るだけシンプルなものとした。この部屋は六角形の部屋であり、それぞれの壁の面には1つずつドアがついている。ドアの色は多色でポップさを演出した。3つ目の部屋に入るためのドアは通常のドアと一緒だが、その他の5つのドアにはドア用の穴はなく、開けても壁になっている。しかし、壁に当たるように進むとギミックが作動し、森へ遷移する。そのため5つのドアの壁穴は作らずに作業を減らすことにした。この部屋では、主人公が大きくなったり小さくなったりするギミックのために机が一つ配置され、それ以外は家具を置かない様にした。また、シンプルではあるが何も配置しない部屋では違和感があるため、天井に大きなシャンデリアを配置した。シャンデリアと言っても木製で、ライト部分は蝋燭が配置されている。これを配置することで部屋としての必要最低限なシンプルさを作り上げた。

(※文責: 住吉聖也)

#### 3.1.1.5 森

小屋の次の面として、森を制作予定である。森ではお茶会などのイベントや、迷路などのギミックを実装予定である。また、屋外のためある程度の広さが必要である。プロトタイプでは、UnityのTerrainというアセットを用いて仮の森を作成した。Terrainとは地形を作成することができるアセットである。1枚の大きな地面を生成し、その地面の起伏を変化させ、木のアセットやマテリアルを貼ることで森を作り出すことができる。地面の形状を複雑にするとUnityの処理がかなり重くなり、キャラクターを操作するとUnityが落ちてしまう。また、木や岩などのアセットを大量に配置しても同様に処理が重くなる。そのため、森のマップを作る上で、地面の形状や設置するオブジェクトの量などを考えて作成する必要がある。プロトタイプでは、森のマップに遷移した時に、森であることがプレイヤーに伝わるのが重要であり、森に遷移した時点でプロトタイプは終了する予定であったため、より現実の森に近づけるように地面の起伏を激しく設定し、地面から生える草や木を多く配置した。そうすることで、少ない時間でも森であることが認識できるようにした。今後の予定では、ギミックやイベントなどのオブジェクトなどが森に配置される予定であるため、マップ自体の処理をなるべく軽くするようにしていく必要がある。

(※文責: 住吉聖也)

#### 3.1.1.6 城

城では、主人公が裁判にかけられる裁判場や、探索可能で複雑な作りの城内を実装予定である。また、建物のモデリングにはBlenderを使用することを検討している。

(※文責: 住吉聖也)

### 3.1.2 キャラクター

#### 3.1.2.1 主人公の制御

ここでは、主人公の3Dキャラクターモデルの制御に関して解説を行う。今年度は、視覚班が3Dキャラクター制作ソフトウェア「VRoid Studio」で作成した主人公アリスの3Dモデルを頂き、それをゲーム開発エンジンであるUnityの「Animator」の機能と、Unityの拡張ライブラリである「UniVRM」を利用しUnity上で動かしている。ここで、Animatorは、ゲーム内の3Dモデルにアニメーションを追加するために使用されるコンポーネント、また、UniVRMは、VRoid Studioで作成した3DモデルをUnity上に取り込み、動かすために必要となるものである。キャラクターの身体的な動作は、Unity Asset Storeで無料で配布されている3Dキャラクターのモーションアセットを用いて、待機状態、歩く、走る、ジャンプの4つで構成されている。モーションや動作は、当初、待機状態と歩く状態の2つの行動パターンしかなかったが、3Dオープンワールドゲームならもう少し動作やモーションのパターンがあったら面白いのではないかというメンバーの意見をもとに、新しく走る動作とジャンプの動作を追加した。

4つのモーションの遷移方法については、UnityのAnimatorの機能を使用している。4つの動作ごとにboolean型のパラメータを設定し、キャラクターの状態に合わせてパラメータをtrueやfalseに変更するスクリプトを作成し、主人公のモデルにアタッチすることにより、適切なモーションを行うよう設定を行った。

キャラクターの移動、方向転換に関しては2021年度のキャラクター制御スクリプトの基盤を使用している。移動方向は、キーボードの入力のベクトルとカメラの向きのベクトルを合成して移動方向を決定して移動を行うように作成した。移動に関しては、プレイヤーのxyz座標を示すコンポーネントのtransform内のpositionの値を変更することで実装した。

また、キャラクターのシステム的な制御に関しては、列挙型の変数Statusの中の列挙定数「Stop, Move, Run, Jump, Event, Pause」の6つを使用している。これらは、プレイヤーの状態を表しており、モーションの遷移や会話イベントが発生した際に、状態を切り替えることで、イベント中のプレイヤーの操作など、ゲーム内状態の矛盾の解消と、プレイヤーの動きの管理を行うことができる。

中間発表地点では、主に2021年度の成果物から上で説明した、主人公の制御に関連するシステムの引き継ぎと、モーションの追加など今年度の成果物に合わせた新機能の作成の2つを行った。前者は主人公の移動やモーション遷移、システム的な制御など、うまく引き継ぎができたが、後者は、ジャンプ動作や走る動作の不具合など、改良が必要な点があるため、今後、新たに追加したシステムをメインに、主人公制御システムの改良を行っていく。

(※文責: 新屋拓海)

#### 3.1.2.2 アイテム取得

ここでは、アイテムの取得システムに関して解説を行う。アイテムは小屋や森、城に落ちており、プレイヤーの操作により取得できる。取得システムの仕組みとしては、あるオブジェクトとの接触があった際に動く、Unityの関数の、OnCollisionEnterを利用して、アイテムの近くにプレイヤーがいるかを判定し、存在している場合、アイテムの名前に応じてアイテム名を表示する。その後、キー操作によりアイテムにアタッチされているItemGetスクリプトが動作することで、ア

アイテムを取得し、そのアイテムをプレイヤーから見えないようにして、プレイヤーの持つ List 型の変数、ItemList にアイテムの名前を追加するという仕組みである。

中間発表時点では、ItemGet スクリプトおよびアイテム習得に関するシステムが、小屋ステージに登場する、鍵2つとサイズ変更のアイテム2つの計4つにしか対応していないため、今後制作する森や城のステージに登場するアイテムにも対応できるよう、スクリプトとシステムの改良を行っていく。

(※文責: 新屋拓海)

### 3.1.2.3 キャラクターの効果音

ここでは、キャラクターの効果音について解説を行う。中間発表地点では、キャラクターの歩く・走る動作に応じて鳴る、足音の効果音を実装した。足音については、音響班から頂いた、足音の高さと足音になる速度の異なる2種類の効果音を使用している。この2つの効果音を Unity 上に組み込み、Unity の機能の1つである、「AudioSource」の Play 関数と Stop 関数を使い、主人公のモーションに合わせて足音の効果音を再生・停止することで効果音の制御を行っている。

現時点では効果音は足音のみとなっているため、今後は必要に応じて効果音を追加する作業を行っていく。

(※文責: 新屋拓海)

### 3.1.2.4 NPC の制御

制作中のゲームでは、小屋・森・城の3つのステージがあり、それぞれのステージに NPC が配置されている。以下では、各ステージに登場する NPC の3Dキャラクターモデルの制御に関して説明する。

(※文責: 新屋拓海)

### 3.1.2.5 小屋 NPC

小屋ステージでは、NPC として、白兔の3Dモデルが登場する。この3Dモデルは、主人公のアリスの3Dモデルと同様に、視覚班が VRoid Studio で作成したモデルを Unity の Animator 機能と拡張ライブラリの UniVRM を利用して動かしている。このモデルには大きく2つの機能がある。

1つ目は会話イベント機能である。NPC にはシナリオ読み取りシステムを利用した LoadText という会話イベントスクリプトが組み込まれており、衝突を検知する OnCollisionEnter 関数を利用し、プレイヤーが白兔に近づき、キー操作を行うことで、LoadText が動作する。そして、それにより、会話ウィンドウが表示され、場面に応じた会話が発生するという仕組みになっている。

2つ目は待機モーション機能である。当初は3Dモデルをただ立たせる予定であったが、キャラクターが棒立ちだと見栄えが悪く、さみしいという意見があったため、ゲーム内で一定の時間が経過すると3Dモデルが伸びをするなどのモーションを行うスクリプトを作成し、NPC にアタッチしている。

中間発表地点では、白兔に会話発生機能と1つの待機モーションを付けた。会話イベント機能に関しては問題なく実装できたが、待機モーションに関しては、モーションの途中で動作が終了し

てしまうことや、想定したタイミングで待機モーションを行わなかったことがあるなど、スクリプト自体がまだ未完成のままである。また、小屋ステージ以降に登場する NPC についても、まだ実装予定のスクリプトができていない。そのため、今後は、待機モーション機能の改良および、他の NPC に関するスクリプト制作を視覚班から他のモデルを頂き次第行っていく。

(※文責: 新屋拓海)

### 3.1.2.6 カメラ

メインカメラの動きは、Unity レジストリ内に存在する「Cinemachine」機能の 1 つである「FreeLook Camera」を使い、制御している。FreeLook Camera は、Unity 内のメインカメラに設定することで、TPS ゲームのような三人称視点のカメラワークを実装することができるもので、プレイヤーを追従するようカメラ設定を行うことで、プレイヤーを常に画面の中心に捉えながら、マウスの動きに合わせてカメラの向きと座標を変更することができる。

また、場合によって、小屋の壁や森の木など、カメラがこれらの障害物を貫通してしまい、プレイヤーや NPC が映らないということが起こってしまうため、今年度では、FreeLook Camera に「Cinemachine Collider」という機能を新たに追加している。これは、Unity 内に存在する、FreeLook Camera の拡張機能の 1 つであり、追従対象が障害物に貫通した際に、追従対象がカメラに映るような位置に自動的にカメラを移動させるというものである。カメラが障害物を貫通したかの判定は、Unity の「Collider」という衝突判定システムを使用しているためカメラが貫通して欲しくない障害物には Box Collider などの当たり判定をつけることで Cinemachine Collider の設定を行っている。

また、ゲーム中では、主人公の大きさを変更するアイテムが登場し、主人公のサイズを大きい状態・通常状態・小さい状態の 3 つの状態に変更することができる。しかし、ただサイズを変更するだけだと、FreeLook Camera に設定した、カメラが中心に捉える対象オブジェクトである「LookPos」の座標が変更後のカメラ座標と上手く合わなくなる場合があるため、サイズに応じて LookPos の座標とカメラ座標を適した座標に変更するスクリプトを C# で作成し、実装を行っている。

さらに、メインカメラでは、会話ウィンドウの UI の表示も同時に行っている。Unity 内に配置している GameManager にアタッチされている「WindowController」スクリプトの「ShowTalkWindow」と「HideTalkWindow」の 2 つの関数を使って UI の表示を管理し、会話イベントやアイテム取得が発生した際に、メインカメラで UI を映している。

(※文責: 新屋拓海)

### 3.1.3 ギミック

ギミックはゲームエンジンの一つである Unity とプログラミング言語である C# を使用して作成した。最終目標は自動生成を使って、マップに合ったギミックを生成していく予定である。だが、前期では著名なゲームのギミックの分析を行うことまでできなかった。そのため、前期制作の小屋でのギミックはシステム班のメンバーが考えたギミックを使っている。

(※文責: 田中悠人)

### 3.1.3.1 小屋

小屋には3つの部屋が存在する。そのうち2部屋にひとつずつギミックを制作し配置した。

(※文責: 田中悠人)

### 3.1.3.2 1つ目のギミック

小屋の2つ目の部屋で、「暖炉の灯を消すと鍵が落ちてくるギミック」を制作した。以下はそのギミックの説明である。

#### ・ 暖炉の灯を消す

1つ目の部屋には、火が燃えている暖炉がある。暖炉の前に Unity で生成することができる透明なオブジェクトである CreateEmpty を生成した。それに当たり判定を担ってくれる BoxCollider を付けた。BoxCollider の isTrigger を True にすることで、CreateEmpty への諸突判定をなくし、BoxCollider 内に入ったか入ってないかを OnTriggerEnter という、Unity が持っている関数を利用し、暖炉の前に立っているかどうかを判定できるようにした。CreateEmpty には、FireDel というスクリプトをアタッチした。スクリプトの概要としては、OnTriggerEnter が呼ばれたときに、BoxCollider 内に侵入してきたオブジェクトに付いているタグが Player に等しければ、近いかどうかの Bool 型の変数を True にするという作業を行っている。True であるとき、プレイヤーが E キーを押すと、暖炉の火が消えるようにした。暖炉の灯が消えるとき、一瞬で消えるのは不自然であるため、DoTween という Unity の機能を使用した。それを利用し、火がだんだんと小さくなるのを表現するために、DOScale() という関数を用いた。スクリプトとしては、「transform.DOScale(new Vector3(0.0f, 0.0f, 0.0f), 2.5f)」と記述し、「火の大きさを2.5秒で0にする」という指示をスクリプトで書いた。そうすることで、燃えている火が一瞬で消えるのではなく、自然にだんだんと小さくなっていくというように、表現することができた。火が消えたら、火が消えたことを判定する Bool 型の変数を True にした。

#### ・ 鍵が落ちてくる

火が消えたかどうかを判定する変数が True になると同時に、鍵のオブジェクトを表示させ、天井から落下させた。鍵には Rigidbody と呼ばれる、Unity で自動で物理演算をしてくれるコンポーネントを付けることで、鍵に対して重力がかかるようになったため、鍵が落下してくるようになった。しかし、鍵がただ真下に落ちてくるだけでは違和感を感じると話題が上がった。そのため、暖炉の時と同様に、CreateEmpty を作成し、BoxCollider を付け、鍵が落下してくる軌道上にそれを設置し、落下してきている鍵がそれにぶつかることで、鍵が回転するように工夫を行った。軌道上に置いた CreateEmpty には、衝突判定が起こるようにするために、isTrigger を false にしている。鍵に近づき、プレイヤーが E キーを押すことで、鍵を入手することができ、鍵を入手しているかどうかの変数を True にした。それにより、次の部屋に向かうためのドアを開けることができるようになる。

(※文責: 田中悠人)

### 3.1.3.3 2つ目ギミック

小屋の3つ目の部屋にはその部屋に入ってくるための一つのドアと、5つのドアが六角形の部屋の各辺に配置されている。その部屋では、「六角形の部屋にある5つのドアの中から、ランダムに選ばれた1つの正しいドアを選ばないと次に進めない」というギミックを作成した。以下はそのギミックの説明である。

#### ・ドアのランダム

5つのドアには、Unity上で1～5タグをつけてナンバリングを行っている。それぞれ5つのドアの前には、Unityで生成することができる透明なオブジェクトである CreateEmpty を生成し、それに当たり判定を担ってくれる BoxCollider を付けた。CreateEmpty につけられている、BoxCollider の isTrigger を True にすることで、CreateEmpty への諸突判定をなくした。また、BoxCollider 内に入ったか入っていないかを OnTriggerEnter という、Unity が持っている関数を利用し、5つのドアそれぞれの前に立っているかどうかを判定できるようにした。ランダムに1つのドアを選ぶためのシステムとして、RandomDoorSelect というスクリプトを作成した。スクリプト内では、文字として1～5を格納している配列を用意した。また、配列の何番目を取り出すのかをランダムで決定するため、Unityの機能である Random 関数を用いて、Random.Range(0,5) と記述することで、1～4の範囲でランダムな整数値が返るようにした。それを、用意した変数に代入し、ランダムでドアの番号を取得できるようにした。その後、プレイヤーが開けようとしているドアがランダムに選ばれた正しいドアであるかどうかを確かめるために、ナンバリングされたドアの番号と、ランダムに選ばれたドアの番号が等しいかどうかを調べることで、正しいドアであるかどうかの判定を行った。判定を行う際に、Unity のオブジェクトにつけられたタグを呼び出す gameObject.tag という関数を利用して、判定を行った。この機能を使った時に、1つ問題が発生した。それは、gameObject.tag を利用した時の戻り値は String 型で、Random.Range(1, 5) の戻り値は int 型であるため、単純に「gameObject.tag==Random.Range(1, 5)」という風に判別ができないという問題であった。その問題を解決するために、配列に String 型の 1～5 を格納し、配列の値を取り出すという風に判定を行った。

#### ・間違えたドアを選んだ時

上で説明した動作を行うことで、プレイヤーが選んだドアが正しいかどうかの判定をすることができる。プレイヤーが選んだドアが正しければ、まだ実装はされていないが次のステージである森のエリアに進むことができる。もし間違えたドアを選んだのであれば、キャラクターは3つ目の部屋に入ってくるためのドアがあった位置に移動させられ、正しいドアを選ぶまで同じ動作を繰り返すことになる。プレイヤーが間違えたドアを選んだ時に、キャラクターを3つ目の部屋に入ってくるためのドアに移動させるという動作は、スクリプトで管理している。もし、プレイヤーが間違えたドアを選んだ時には、キャラクターの位置を Unity 上で制御している、Transform の値を3つ目の部屋に入ってくるためのドアの位置に強制的に変更させることで、キャラクターの位置を制御している。変更するために、ドアの空間座標を取得する必要があるため、取得するために Vector3 型の変数 DoorPos を用意し、ドアの x, y, z 座標の値を格納している。プレイヤーが間違えたドアを選んだ時、キャラクターの位置を DoorPos の値に変更することで、強制的に移動させることが可能となっている。ただ単に移動させるだけでは、機械的で不自然な移動になってしまうという意見が班内であがった。そのため、画面を一瞬暗転させ元に戻すという表現をするこ

とで、ドアに入ったかのような演出を作成した。その演出を作成するために、Unity の機能である Canvas を作成し、その中にある Panel を使った。Panel の色は黒にし、スクリプトで alpha 値を変更することで、自然な FadeIn・FadeOut を表現した。

- ・体の大きさを変えるアイテム

3つ目の部屋にある5つのドアは、普通のキャラサイズでは通過することが出来ないようになっている。そのため、プレイヤーはドアを通るためにキャラクターの身体の変更するアイテムを入手し、鍵を取得しなければならない。身体を大きくするためにはケーキ、小さくするためには小瓶を入手しなければならない。部屋に入ると、部屋の真ん中にテーブルとその上には小瓶が置いてある。その小瓶を手に入れることで、キャラクターは小さくなることができ、小さくなると大きいときには見えなかった箱がテーブルの足元に置いてあるのを見つけることができ、テーブルの上に鍵が出現する。小さい時には鍵は入手できず、元のサイズに戻す必要が生まれる。その箱を開けるとケーキを入手することができ、元のサイズに戻すことができるようになる。元のサイズに戻すことで鍵を入手することができるようになる。その後鍵を入手することで、ドアを開けられるようになる。

- ・ドアを開ける

ドアの前に Unity で生成することができる透明なオブジェクトである CreateEmpty を生成し、当たり判定を担ってくれる BoxCollider を付けた。BoxCollider の isTrigger を True にすることで、CreateEmpty への諸突判定をなくし、BoxCollider 内に入ったか入っていないかを OnTriggerEnter という、Unity が持っている関数を利用し、ドアの前に立っているかどうかを判定できるようにした。CreateEmpty には、DoorOpen というスクリプトをアタッチした。スクリプトの概要としては、OnTriggerEnter が呼ばれたときに、BoxCollider 内に侵入してきたオブジェクトに付いているタグが Player に等しければ、近いかどうかの Bool 型の変数を True にするという作業を行っている。True であるとき、プレイヤーが E キーを押すことで、ドアが開くようにした。ドアが開く時には、一瞬で開くのは不自然であるため、DoTween という Unity の機能を使用し、キャラクターが開けるかのように、DORotate という関数を用いて、「transform.DORotate(new Vector3(0.0f, 130f, 0.0f), 0.5f)」というように、「ドアの回転を 0.5秒で 130度回転させる」というスクリプトを書いた。そうすることで、ドアが瞬間移動して開くのではなく、キャラクターが開けているかのように、表現することができた。ドアが開いたら、ドアが開いているかを判定する Bool 型の変数を True にし、通過できるようにした。

(※文責: 田中悠人)

### 3.1.3.4 森・城

前期では触れることができなかつたため、後期に行う。

(※文責: 田中悠人)

### 3.1.3.5 自動生成

ドラゴンクエストやバイオハザードなどの著名な作品をプレイし、その中に出てくるギミックをリストアップし、要素分けを行い、それらの組み合わせを変更することで、新たなギミックを生成

する予定である。

(※文責: 田中悠人)

### 3.1.4 シナリオ読み取りシステム

#### 3.1.4.1 シナリオの読み取りシステムの概要

シナリオ読み取りシステムとは、物語班が人工知能により自動生成を行ったテキストデータをもとに、Unity, C#で作られたスクリプトを使用し、各イベントで起こるイベントシーンの読み取り、シナリオに合った自動生成された BGM の挿入、シナリオに沿ったデータをテキストボックス内へのテキストの表示等を行うシステムである。今回、中間発表までの開発では、物語班が自動生成を行わずに作成したテキストデータを読み込みテキストボックスに表示するシステムを作成した。

(※文責: 本前斗梧)

#### 3.1.4.2 表示画面への実装

表示画面は大きく分けて 2 種類に分けられていて、キャラクターを操作する画面と会話やイベントを表示する画面に分けられている。キャラクターを操作する画面ではマップ内に存在するキャラクターに話しかけると会話ウィンドウ用のスクリプトが実行され、画面が切り替わる。会話やイベントを表示する画面では会話ウィンドウの立ち絵や背景、テキストボックス等は視覚班に依頼して素材を作成して頂き、それを 2D オブジェクトとして表示させた。シナリオデータのテキストの読み取りに関しては、テキストを改行文字でスプリット操作を行い、現在話しているキャラクターの名前とセリフを配列に格納している。プレイヤーに話しかけたことを判定する関数である OnCollisionEnter を使い、この関数を用いてプレイヤーへの会話を行ったうえで、UI の表示や表示画面の遷移を行う。後期からはこのプログラムを利用して、物語班が自動生成を行ったテキストデータを読み込めるようなプログラムの修正を行っていく。シナリオ読み込みとは別に会話ウィンドウの表示に関して、ただシナリオを読み取るだけではなく、シナリオに応じてキャラクターの立ち絵が変化したり、BGM が変更したりといった、動きがついた会話ウィンドウを作成することによりゲームとしての完成度が上がるという改善案がグループ内の話し合いで上がった。他にもアイテムデータベースやキャラクターの背景史を図鑑のように表示するという話も上がり、アイテムを拾った時の UI やテキストを視覚班や物語班に依頼して作ってもらったが、前期では他の実装に時間を費やしてしまったため、アイテムデータベースや背景史一覧の UI 作成には至らなかった。

(※文責: 本前斗梧)

## 3.2 視覚班

### 3.2.1 コンセプトアート

班全体で決定したゲームの世界観やそれに伴うキャラクター等のイメージを、絵として表現することは重要である。また、プロジェクトメンバー内で世界観やキャラクターに対するイメージに違和感が出ないように、全体への共有は重要である。そこで、世界観となる「不思議の国のアリス」

に関連する参考資料を調べ、それを元にいくつかのコンセプトアートを制作した。これらは最終制作物に流用する意図は無く、あくまでイメージの共有のために制作したものである。

(※文責: 畑井梨里衣)

### 3.2.2 自動生成

#### 3.2.2.1 キャラクターイメージの共有

キャラクターデザインの自動生成にあたり、はじめに物語班とキャラクターイメージの共有を行った。ここで決定したイメージを軸として、各キャラクターの自動生成に取り組んでいく。決定内容としては、まずメインキャラクターを人型と異形の生物の二種類に分類した。分類する際は、3D/2D モデルの制作コスト、シナリオやマップにおけるキャラクターの出現頻度などを参考に、人形と異形のキャラクターの数とその内訳を決定した。さらに人型のキャラクターについては、いくつかのキーワードで見た目の特徴を決定した。見た目の特徴は、モチーフとなる「不思議の国のアリス」における描写や、世界観、統一感などを考慮したうえで話し合った。詳細は以下の通りである。

##### ・人型のキャラクター

アリス：女性. 8,10 才. 140cm くらい. 金髪. ワンピース. エプロンドレス.

女王：女性. 6,8 才. 110cm くらい. イメージカラーは赤と白. 王冠. 偉そう. ドレス.

チェシャ猫：男性. アリスと同年代. 135cm くらい. イメージカラーは紫とピンク. パーカー, 短パン. 猫耳, 尻尾.

白うさぎ：男性. 15 才. 165cm くらい. フォーマルな服. 眼鏡. 赤い上着. うさ耳, 尻尾.

##### ・異形の生物

トランプ兵, 青虫, 帽子屋, 三月うさぎ

(※文責: 畑井梨里衣)

#### 3.2.2.2 概要

キャラクターの自動生成は2つのアプローチで行った。それは不思議の国のアリスに出てくるキャラクターが4.2.1で前述した通り「人型のキャラクター」と「異形の生物」の2種類に分けることができたためだ。人型のキャラクターの自動生成は汎用性の高さ、異形の生物はオリジナリティが求められると考えた。このように、この2つの種類のキャラクターの自動生成はそれぞれ要求される能力が異なるため、アプローチも2分するべきだという判断をした。

(※文責: 坂上翠)

#### 3.2.2.3 3D モデルの自動生成

物語班との打ち合わせを通して、ゲームの登場キャラクターのうち「トランプ兵」「青虫」「帽子屋」「三月うさぎ」の4体を自動生成技術を利用して生成することが決まった。これらのキャラクターはアリスやハートの女王のようなヒト型のビジュアルではなく、実際には存在しないクリー

チャーとして造形することが決まった。架空の生物を自動生成の技術を用いて生成する。

(※文責: 穂積佳)

#### 3.2.2.4 生成目標キャラクターのビジュアル

生成目標である4体のキャラクターはどれも、「ヒト型×非ヒト型」の形を有しているのではないのかという意見がでた。トランプ兵ならば「兵隊×トランプ」、三月ウサギならば「紳士服を着た人間×ウサギ」のような人外の姿である。原作の帽子屋は人間のキャラクターとして描かれているが、物語班との打ち合わせの際、人間の姿よりも帽子をモチーフにしたクリーチャーとして描いた方が彼の「気が狂っている」キャラクターにマッチするのではないのかという意見があり、クリーチャーとして帽子屋をデザインすることが決まった。

(※文責: 穂積佳)

#### 3.2.2.5 生成アプローチ案

担当教員の吉田先生との相談を通して3つの方法案が出された。

1. 描いた絵を元に3Dモデルを自動生成する。
2. スタイル変換やテキスト入力から3Dモデルのテクスチャや法線情報、メッシュを変換する。
3. アクセサリーや手足のパーツごとにモデルを作成し、それらをキメラのように組み合わせてモデルを生成するアルゴリズムを考案する。

前期では1と2の案を検討していった。理由としては1と2の方法は、即座に試すことが出来るソースコードやツールが公開されているものがいくつか存在したからだ。3の方法では実際に専用の3Dモデルデータを複数作る必要があり、1や2に比べて検討に時間がかかると判断したためである。

(※文責: 穂積佳)

#### 3.2.2.6 MonsterMesh

MonsterMeshはペンツールで描いた絵を立体的に膨らまし、3Dモデルを作ることが出来るツールだ。写真やイラストなどの画像データを入力すれば、それに基づいてテクスチャも自動で張り付けることが出来る。前期では二本足で立つウサギの画像とトランプの兵隊の画像を元に、3Dモデルを生成した。このツールの利点は平面に描ける形であれば自由に生成することが可能である点と操作が比較的簡単な点である。しかし、2Dから3Dに膨らませてモデルを生成するため、膨らませたモデルに独特なシワや形状が出来てしまう点と、非対称の形を生成できないという欠点がある。また、MonsterMesh内のペンツールの仕様により、細かいデザインの造形物を描くことが難しい。

(※文責: 穂積佳)

### 3.2.2.7 AvatarClip

AvatarClip は OpenAI である Clip を用いた技術である。Clip は自然言語処理技術を用いて画像の分類などに利用されるモデルだ。画像とテキストを結びつけることでラベルに設定できるカテゴリの自由度が高く、事前学習によるゼロショット転移で高い精度を出すことが出来る学習モデルである。AvatarClip は Clip を用いて、文章を入力することでヒト型のモデルとモーションを出力することが出来る。例えば「A tall and fat Iron Man that is running.」と入力すれば、太ったアイアンマンのモデルが走っているデータを出力することが出来る。AvatarClip は Google Colab をつかってブラウザ上でも動かすことが出来る。これを使いトランプ兵と三月ウサギの生成を試みた。

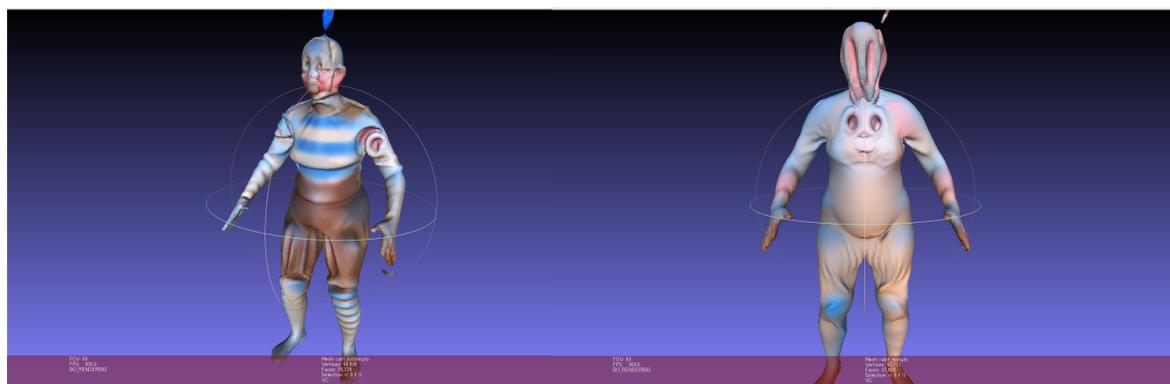


図 3.3 生成されたモデル

トランプ兵は兵隊のような造形ではなくカラフルで道化師のような見た目のモデルが出来上がった。三月ウサギはヒト型のモデルの胴体にウサギの顔が浮かび上がり、ヒト型の顔に当たる部分にはウサギの耳が浮かび上がった。AvatarClip はベースになるヒト型のモデルがあり、その形を少し変形させたり、テクスチャを貼ることでモデルの生成を行っている。なのであまりにもヒト型から離れている造形の物を無理やり生成しようとする、三月ウサギのように人間の身体にウサギの顔をボディペインティングしたようなモデルが生成されてしまうと考えられる。

(※文責: 穂積佳)

### 3.2.2.8 Text2Mesh

Text2Mesh は文章の入力により 3D モデルのスタイルを変えることが出来る学習モデルだ。文章から元となる 3D モデルのメッシュを変形させ造形を変えたり、適切な色をモデルに付けることが出来る。Text2Mesh は AvatarClip とはべつにベースとなるモデルをこちらが用意出来るため、ヒト型以外の様々な形のモデルも生成することが可能だ。MonsterMesh で生成したウサギとトランプ兵のモデルをベースに Text2Mesh での 3D モデル生成を試みた。

三月ウサギは「a 3D rendering of Rabbit in a suit in unreal engine」という文章を入力した。結果では確かにウサギの胴体のパーツに赤いネクタイと灰色のスーツの襟のラインと思われる形が浮かび上がっており、耳にもピンクのラインが浮き出ており、しっかりとウサギの耳とわかる造形になっている。顔も、歪ではあるがほっぺが膨らんで元の 3D モデルデータよりもウサギに近い造



図 3.4 生成されたウサギのモデル

形になっているように見える。しかし、身体の至る所に青の斑点の模様が浮かび上がっており、何を意図している描写なのかは不明だ。メッシュが荒く、形が歪んでしまっている箇所がいくつかあり、恐らくベースになる MonsterMesh で生成したモデルがローポリゴンであるのが原因だと考えられる。Text2Mesh のベースとなる 3D モデルは、頂点が極力均等で細かく設置されている物を使うのが望ましい。頂点の均等化は 3D データ編集ソフトの MeshLab を使って行った。



図 3.5 生成されたトランプ兵のモデル

トランプ兵は「a 3D rendering of Playing cards knight in unreal engine」という文章を入力した。結果は体中に様々な模様が浮かびあがり、色は AvatarClip の結果にカラーリングは似ているモデルが生成できた。身体に浮かび上がった模様は数字やアルファベットのような文字が浮かび上がっており、トランプの要素がうかがえる。また鎖帷子のような凸凹模様も随所に表れており、恐らく兵隊の服装などから造形された要素だと思われる。

(※文責: 穂積佳)

### 3.2.2.9 キャラクターデザイン画の作成

ここでは、2D のキャラクターデザイン画を生成することを目的とする。物語班との打合せを通して、「アリス」、「女王」、「チェシャ猫」、「白うさぎ」のキャラクターデザイン画を生成することになった。ゲームに実装するまでの流れとしては、まずキャラクターの服装と顔のデザインをそれぞれ自動生成し、それらを組み合わせたデザイン画を班員の手で整える。そしてそのデザイン画を元に、実際にゲームで使用する立ち絵や 3D モデルを班員が制作する、という方針で行う。

キャラクターデザイン画の自動生成を服装と顔の二つの要素に分ける理由としては、キャラクターの全身画像を一括で自動生成するよりも、顔と服装それぞれの出力結果を組み合わせたほうが、より密度の高いデザイン画が制作できると期待したからだ。特に顔は、キャラクターの印象に関わる重要なパーツである。また、人間の顔の自動生成は研究が盛んな分野でもある。そのため、顔に焦点を絞ることで、より印象的で精度の高い結果が得られるとも考えた。

(※文責: 畑井梨里衣)

### 3.2.2.10 顔

キャラクターの顔の自動生成については以下の通りだ。まず、テキストからキャラクターのイメージに合った実写の顔画像を生成する。さらにその実写の顔画像を、班員が描いたイラストのデータセットを元に、イラスト調に変換する。このように二段階に分ける理由は、実写画像の自動生成のほうがデータセットや研究が豊富且つ精度が高いため、先に実写画像を生成してしまったほうがやりやすいと考えたからだ。

(※文責: 坂上翠)

### 3.2.2.11 実写画像の自動生成

実写画像の自動生成は StyleGAN+CLIP を利用した。

StyleGAN+CLIP はテキストを入力すると、そのイメージに合った存在しない実写の顔画像が出力されるモデルだ。まず、GAN とは画像生成でよく使用される機械学習の一手法だ。教師なしで学習したデータの特徴を元に実在しないデータを生成したり、データを変換することができる。そして、StyleGAN は派生 GAN モデルの 1 つだ。GAN とは異なり、特徴ベクトルを多次元に増やしたことにより、従来よりも高精度な画像の生成を可能としている。また、CLIP は画像の分類に使用される、ニューラルネットワークを用いたモデル (事前学習方法) だ。インターネットを利用して構築された 4 億組の画像と画像を説明する自由テキストのペアのデータセット (WebImageText) を利用している。画像とテキストの類似度を出力できる。

このモデルを採用した理由は以下の通りである。まず、キャラクターの自動生成についてを行うにあたって顔画像の自動生成について詳しく解説している cedroBlog を参考にした。そこでキャラクターの顔画像を自動生成する方法として、転移学習する方法と StyleGAN+CLIP を利用する方法が挙げられた。転移学習する方法は膨大なデータセットが必要となる上に PC のスペックも要求される。一方、StyleGAN+CLIP は膨大なデータセットを作る必要がなく、ネット上で実行できる GoogleColaboratory を利用できる。これらのことから、利便性と簡易性が高い StyleGAN+CLIP を採用した。

今回の自動生成では 3.2.2.1 でシナリオ班と共有したキャラクターのイメージを元に、テキストを作成して DeepL 翻訳を用いて英語に翻訳して入力した。例えばアリスなら、「She is a pretty English girl between the ages of 8 and 10. She has white skin, big blue eyes, and long blonde hair that is loosely wavy.」と入力した。ここではキャラクターの顔のパーツの情報を具体的に入力するように留意した。アリスと赤の女王の自動生成は特に問題なく実行できたが、白うさぎとチェシャ猫は生成された画像に青いノイズが出現したり、歪んだ顔画像が生成されたりした。そこで、以下の対策をとった。まず入力するテキストに形容詞を加え、詳細にした。例えば、「He is English boy」を「He is a handsome English boy」にする。そうすると、ハンサムな印象の顔画像が生成されるようになった。次に、学習回数を増やした。そうすると、よりテキストに適した画像が生成されるようになった。例えば、「White hair」と入力したとき、学習回数が 101 のときはどちらかという金髪の顔画像が生成されるが、700 にすると顔画像はより白い髪になった。

(※文責: 坂上翠)

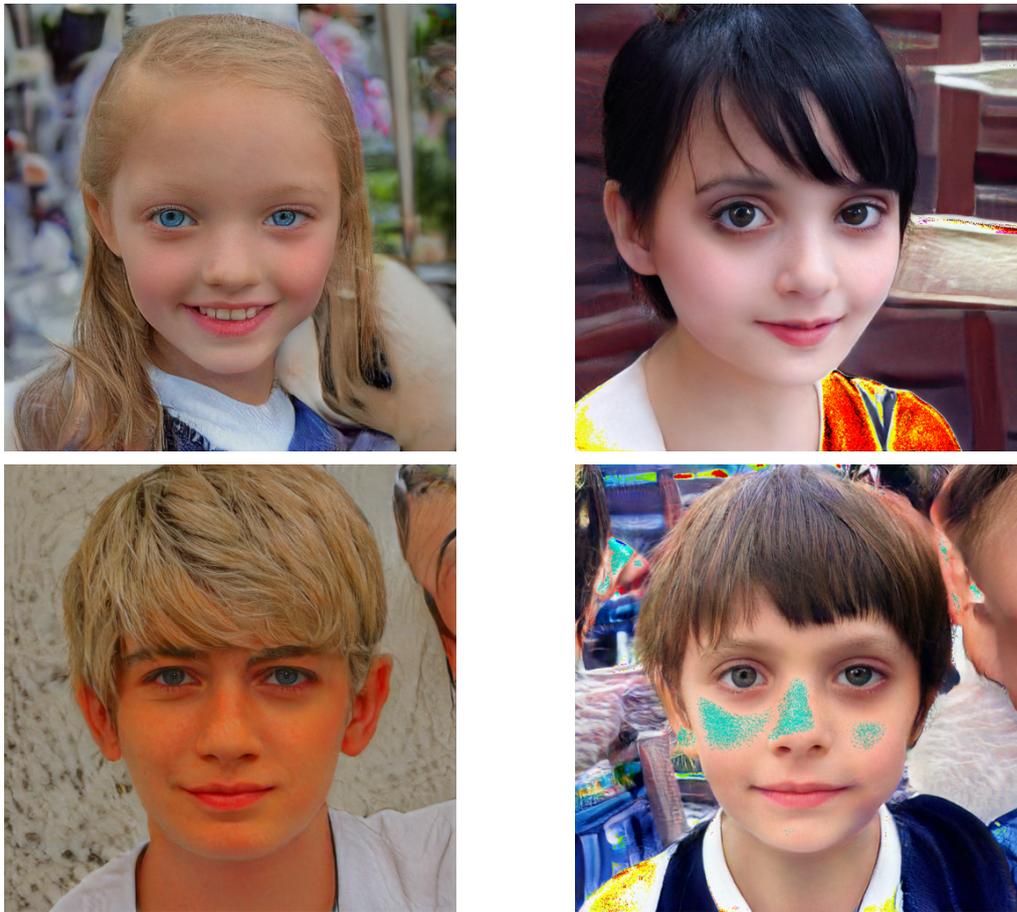


図 3.6 生成された画像

### 3.2.2.12 スタイル変換

StyleGAN\_NADA を用いてスタイル変換を行った。

StyleGAN\_NADA では絵柄のデータセットとテキストを入力すると、実写画像をテキストのイメージに合わせつつデータセットの絵柄に変換する。まず、変換する絵柄の設定のためにデータセットを作成した。1人1日2枚を目標に、イラスト調の正面を向いている胸部から上の人間を描いた。現在までに45枚のデータセットを作成した。データセットをもとにStyleGAN\_NADAでスタイル変換を行った。StyleGAN\_NADAはStyleGANの一種だ。

このスタイル変換の内容は以下の通りだ。まず、テキスト入力を初期設定のまま、グループ全員のデータセットで写真からイラスト調にスタイル変換を行った。すると全体としては輪郭や顔のパーツ、髪に太めの主線が出現し、色が淡い色調になった。目は、瞳の色に赤みがかかり目の輪郭が大きくなった。また、描いた人ごとに画像を分類したデータセットでスタイル変換を行った。そうすると描いた人の絵柄の特徴が出力にも現れた。例えば、色彩や前髪の形、表情などの特徴だ。

次に、入力するテキストに変更を加えた。よりゲームに適した絵柄にスタイル変換する目的のためだ。「Japanese anime cute girl face」と入力するとより幼い印象の顔になり、彩度がテキストを変更する前と比べて上がった。また、特定のゲームを指定するとどうなるか試した。今回は株式会社miHoYoが提供する原神というゲームを指定することにし、「Genshin character cute girl face」と入力した。すると表情が険しくなった。これは原神はキャラクターが戦闘をするゲームであるので凛々しい表情をしていることが多いことによるものだと考えられる。更に、特定のキャラクターを指定するとどうなるか試した。今回は原神のパイモンというキャラクターを指定し、「Genshin

character paimon face」と入力した。すると瞳は大きく、口は笑ったようになり、優しい印象の顔画像が出力された。これはパイモンが大きな目を持ち、笑っていることが多いキャラクターであることが理由に挙げられる。



図 3.7 生成されたモデル

(※文責: 坂上翠)

### 3.2.2.13 服

キャラクターの服装デザインの自動生成については、BigGAN と CLIP を組み合わせることでテキストから画像を生成するモデルを利用した。BigGAN とは、画像生成でよく用いられる機械学習の一手法である GAN の、派生モデルのひとつである。Imagenet という、1400 万枚以上もあるほど大規模な「カラー写真」の教師ラベル付き画像データベースの巨大なデータセットを学習したもので、1000 カテゴリーの画像を最大 512 × 512 ピクセルの高解像度で生成することができる。一方 GAN モデルは、入力パラメータにランダムノイズを使用するため、基本的にランダムな画像しか出力できない。そこで、3.2.2.4.1 の顔の自動生成でも説明のある、CLIP というモデルと組み合わせることで、テキストの内容に沿った画像を生成できるようにしたのがこのモデルである。

実際に自動生成を行う際は、ディープラーニング技術に関するブログ「cedro-brog」内で掲載された、BigGAN と CLIP を用いた画像生成の手法を説明する記事で紹介されているコードを、Google が提供する Colaboratory という環境で実行するという形をとった。Colaboratory は、環境構築が不要、GPU へ無料アクセスできる、等の利点があることからこの方法を採用した。

4.2.1 で物語班と共有したキャラクターのイメージを元にテキストの内容を考え、DeepL 翻訳を用いて翻訳した英文を入力テキストとして、服装デザインの自動生成を試みた。

まず、'An Alice in Wonderland Dress', 'Cheshire Cat Clothing' といった、キャラクターの名前を直接織り交ぜたテキストを入力し、計 8 枚の画像を生成した。これは、世界的に有名な「不思議の国のアリス」というモチーフ作品が存在するために可能な手法でもある。しかし、キャラクターの名前の影響が強いためか、衣装の形と解釈できるような画像はあまり出力されなかった。これは、白うさぎやチェシャ猫といった動物モチーフのキャラクターで特に顕著な傾向であった。一方で、生成された画像では、それぞれのキャラクターを連想させるようなテクスチャが表現されているようにも感じた。そのため、衣装の形として解釈できないような画像は、服装デザインとして直接利用はせずとも、装飾の一部や柄として流用できる部分もあるのではと考えた。

次に、以上のような結果を受けて、衣装の形と解釈できるような画像を出力するためのテキストを模索し、計 15 枚の画像を生成した。例えば、'a doll of ', 'fashion model in ' といった形式のテキストで人型のシルエットの出力を狙ったり、' in white background' といった形式のテキストで背景の雑味の解消を狙ったりした。結果、狙った効果は強くは得られなかったが、見た目の要素を具体的に描写することでより整然とした画像が出力できる傾向があることが分かった。

(※文責: 畑井梨里衣)



図 3.8 生成された画像



図 3.9 生成されたアリスの衣装

### 3.2.3 素材作り

#### 3.2.3.1 概要

視覚班が制作を担当する素材は大きく、3D モデル、イラスト、UI コンセプト・素材の 3 つである。これらのアセットはゲームの世界観を表現するために大切な要素になる。視覚班は物語班とのやり取りをメインにキャラクターや世界観のイメージ共有を行い、ゲーム実装を行うシステム班と相談し素材データを制作していった。前期では、キャラクターモデル×2、アイテムモデル×4、

立ち絵×2、会話シーンの UI 素材を作成した。

(※文責: 穂積佳)

### 3.2.3.2 3D オブジェクト

3D モデルは2つの制作ツールを利用した。アイテムなどの小物類は 3DCG 制作・編集ソフトである blender を用いて作業をすすめた。blender ではゲーム内アイテムや、ステージオブジェクト、キャラクター衣装などを作成することが活動のかなり初期の段階で決まっており、blender の扱いに慣れつつ素材作りを行うため、比較的制作難易度の低いと思われるシンプルな造形物から制作を始めていった。前期では「ケーキ」「カギ×2」「瓶」を制作した。ヒト型のモデルは blender で制作を行うのは難易度が高いと判断したため VRoid Studio というヒト型の 3D モデルアバターを作成できるツールを利用した。VRoid Studio では、ヒト型モデルにパーツを加えたりパラメータを調節することで比較的容易にヒト型の 3D モデルを作ることが出来る。前期ではプロトタイプに実装する「アリス」と「白うさぎ」のモデルを作成した。

(※文責: 穂積佳)

### 3.2.3.3 デザインの詳細

#### ・カギ1

プロトタイプに登場するキャラクターの一人である白うさぎをモチーフにデザインした。登場人物に共通しているトラウマを抱えているという特性や、「不思議の国のアリス」のどこか不条理な世界観を、いびつに曲がった鍵の形で表現した。

#### ・カギ2

プロトタイプに登場するキャラクターの一人であるアリスをモチーフにデザインした。鍵1のモノトーンなデザインと差別化を図るために、カラフルな色使いにした。また、アリスの衣装の色をイメージした、ライトブルーのリボンを結んだデザインにした。

#### ・ケーキ

”EAT ME”と書かれたアップルクーヘンである。「不思議の国のアリス」における、身体を大きくする作用のあるケーキをモチーフにデザインをした。ふっくらしておいしそうに見えるようにこだわった。

#### ・瓶

”Drink Me”と書かれたタグが付いている小瓶である。「不思議の国のアリス」における、身体を小さくする作用のある飲み物をモチーフにデザインをした。不思議の国の世界観に合わせて、怪しげな見た目になるように、ピンクと水色が混じった独特な色合いにした。

#### ・アリス

顔の造形については、6,8才という年齢設定を基準にして適度に若い顔にするために、頬をもったりとさせる、目の位置や鼻の高さを低くする、などの調整を行った。髪型については、自動生成したアリスの実写画像をもとに、前髪はさっぱりとした印象のものにし、毛先は軽くウェーブのか

かったものにした。色味については、肌に関しては健康的で血色よく、髪や目の色に関してはコンセプトアートを参考にやや褪せた色合いにした。服装に関しては、「不思議の国のアリス」に登場するアリスの服装を参考に、水色のエプロンドレスのような服装を VRoid Studio のプリセットから用いた。

#### ・白うさぎ

基本的に、プロトタイプ用に制作した立ち絵のデザインに近いものになるように造形した。よりフォーマルな印象になるように、髪型はあまりはねない整った印象のものにする、手袋を付ける、などの調整を行った。色味については、肌に関しては立ち絵の印象をもとにやや不健康で薄い色にし、髪や目の色に関してはアリスと同様にやや褪せた色合いにした。また、小物としてうさぎの耳を VRoid Studio のアセットから用いた。

(※文責: 畑井梨里衣)

### 3.2.3.4 立ち絵

プロトタイプの会話シーンで使用するキャラクターの立ち絵を、MediBang Paint というデジタルイラストツールを使って制作した。ここでは、アリスと白うさぎの二人を用意した。アリスについては、コンセプトアートとして描いたキャラクターイラストを使用した。デザインについては、アリスは68才の幼い少女であるため、等身は低く、顔立ちは幼くなるように意識した。髪や目の配色・服装などは、「不思議の国のアリス」に登場するアリスの描写を参考にし、金髪碧眼で水色のエプロンドレスを着たデザインにした。白うさぎについては、シナリオ班と決定したキャラクターイメージや、大人しそうな性格という情報などから、新たにキャラクターをデザインして立ち絵を描いた。等身については、15才の青年という設定から、アリスと適度な身長差を感じさせる身長にした。うさぎをモチーフとしたキャラクターであるため、耳は白い毛の生えたうさぎのものとし、それに合わせて髪色も白に統一した。服装は、「不思議の国のアリス」に登場する白うさぎの描写を参考に、タキシードのようなデザインにした。また、眼鏡、蝶ネクタイなどの小物を追加した。立ち絵のポーズとしては、白うさぎの性格に合わせて、伏し目がちに片腕を抑えて立っている様子にした。また、アリスのデザインと統一感を出すために、全体的に彩度をやや抑えた配色にした。最後に、画面上でキャラクターを印象づけるためにキャラクターのシルエットで影をつけるなどの調整を行った。

(※文責: 畑井梨里衣)

### 3.2.4 UI

プロトタイプの会話シーンの UI を Figma で制作した。使用した素材は以下の通りである。世界観に合うように古紙のような素材を用意し、会話内容やキャラクター名、メニューボタンのテキストボックスに適用した。また、フォントは丸みを帯びたフォームと文字の末端のデザインが世界観に適していると判断し、Kaisei Opti を全体の文字に適用した。フレームは世界観に適していて且つ、使用フォントと似たようなデザインのものを選んだ。デザインは以下の通りである。まず、会話シーンにおいてクリックで会話の内容が切り替わるのを示唆するための逆三角形の記号を取り入れた。更にドロップシャドウを入れ他オブジェクトとの区別をした。



図 3.10 テキストウィンドウ・キャラクター立ち絵・背景画像を配置した画像

(※文責: 坂上翠)

## 3.3 音響班

### 3.3.1 目標

2.3「課題の割り当て」にて音響班は、インタラクティブミュージックの自動化を目標としている。他にも目標を立てるにあたって、登場人物のテーマソングを職業や年齢などの感情以外のパラメータを用いて、すでにある曲から自動選択するという目標や、ストーリー場面やエリア画面において、雰囲気にあった曲を自動選択するという目標もあったが、インタラクティブミュージックの自動化を目標にした理由としては、ボス HP の数値によって戦闘 BGM が変化するインタラクティブミュージックを昨年度実装しており、この技術を応用してエリア移動や画面遷移などでも実装できるのではないかと考えた。

また、選択しなかった目標は、どちらも曲を自動選択する技術を用いていることから、インタラクティブミュージックの自動化を実装するにおいて、曲を自動選択するシステムも含まれているため、今年度の目標はインタラクティブミュージックの自動化とした。

(※文責: 菊地陵雅)

### 3.3.2 先行研究

音響班が取り組むものと似ている技術として、以下の三つが挙げられる。

- pioneer 「MIXTRAX」
- AI DJ Project - A dialogue between AI
- インタラクティブミュージックの基本的な手法

まず一つ目のパイオニア株式会社が制作した PC 用楽曲管理アプリケーション「MIXTRAX」は、PC 内にある音楽ライブラリーからテンポや曲調を分析し、自動で楽曲を選択、再生を行い、永遠と再生できる「ノンストップミックス再生」の機能がある。この技術は、インタラクティブミュージックを実装するにあたって、曲の類似度を把握するために必要なモノである。

また、分析結果から、Beat, Chill, Thump, Distortion, High の五つの曲の雰囲気分類することもできるが、ジャンルが EDM やヒップホップなどのダンスミュージックのみとなっているため、オーケストラや J-Pop などの幅広いジャンルには対応していない。

しかし、この技術を応用して曲の雰囲気を幅広く取得し、先の自動選択システムに取り込むことができれば、可能性があると考えられる。

さらに「MIXTRAX」では、曲を切り替える際にリバーブやフィルタなどのエフェクトを用いて、切り替える機能も実装している。この技術は、音響班が取り組むシステムにおいても、曲を切り替える際のアクセントになるのではないかと考える。

二つ目の「AI DJ Project - A dialogue between AI」は、本来、二人の DJ がお互いに曲を掛け合うプレイスタイルである「Back to Back」を、片方は AI で行うというプロジェクトである。このプロジェクトでは、実機のターンテーブルを使用し、物理でかけているレコードを制御する。

システムの特徴として、曲の選択、かけている曲のビート合わせ、会場の雰囲気を読み、曲テーマを選択するクラウドリーディングの三つの要素から出来ており、それぞれ異なる人工知能を用いている。

今回、音響班が開発するシステムに近いものとして、曲の選択を重視した。このプロジェクトでの曲の選択では、スペクトログラム画像から音楽ジャンル、使用している楽器、ドラムマシンを推定でき、それぞれに畳み込みニューラルネットワーク (CNN) を用いている。

音楽ジャンルの推定では、EDM やヒップホップなどのダンスミュージックに限定されており、幅広いジャンルには対応していない。しかし、楽器の推定ではありとあらゆる楽器を推定でき、ドラムマシンの推定においても、メーカーやモデルまで推定することが可能である。

推定した音源データから、あらかじめ用意した全音源データと比較し、似たような音楽的トーンやムード、テクスチャをもった特徴量が最も近い曲を選択する。

この技術は、学習させるデータが膨大になってしまうが、類似精度を向上させることができるのではないかと考える。

また、選択する際のセンスも学習するために、ジャンルやアーティスト、レーベルなどの音楽メタデータを使用せず、オーディオデータのみ使用している。理由として、選曲する際の意外性を持たせるためである。

この技術は、音響班が取り組む曲を切り替えるシステムにおいて違和感がなく、かつ斬新な曲選択を実現させるには、参考になる技術である。

三つ目の「インタラクティブミュージック」は、実装する手法として書き記す。インタラクティブミュージックには、「縦の遷移」と「横の遷移」の二種類の方法がある。

「縦の遷移」では、曲の展開は変わらないが演奏する楽器が変化する、いわゆる「曲のアレンジ」が変化する。これを実装するには、アレンジが異なる曲を同時に再生し、音量変化させることで曲を切り替えることができる。システム面では容易に実装できるが、アレンジした曲が何種類か必要で、かつ、すでに出来ている曲から実装するには難しいため、一から作曲する必要もあり、労力に負担がかかる。

一方、「横の遷移」では「縦の遷移」とは異なり、曲のアレンジは変化しないが曲の展開が変化する。これを実装するには、現在再生している曲の再生位置を検知するシステムや次に再生する曲を違和感なく繋げるためのタイミングを計算するシステム、計算したタイミングから違和感なく曲を繋げ、次に再生する曲をあらかじめ再生するシステムが必要で、システム面での実装はかなり難

しくなる。しかし、必要となる曲数は立った一曲で、すでに出来ている曲からも実装できるため、労力の負担が軽減できる。

(※文責: 菊地陵雅)

### 3.3.3 方針

初めに、インタラクティブミュージックの実現方法について音楽的な手法で解決するのか、技術的な手法について解決するのかを検討した。その過程として、自分たちで具体的でかつ限定的なインタラクティブミュージックを生成することで、実際の実現方法の模索を行った。その結果、目標であるインタラクティブミュージックの実現のために最低限必要な要素を2つに設定した。一つはインタラクティブミュージックの素となるデータベースの構築である。これは、主に著作権に反さないBGMを集めたデータベースであり、実際のゲーム上で利用するBGMとして使われる。もう一つはインタラクティブミュージックを実現するシステムである。このシステムは様々な階層ごとに分類でき、大別して4つのシステムからなっている。前期では、この2つの制作物を基礎としてインタラクティブミュージックの実現を目指していく。

(※文責: 小林裕汰)

### 3.3.4 データベース構築

データベース構築では、学習させる音源データからBPM、クロマグラム、スペクトルセントロイドの三つの音響特徴量を抽出するシステムを開発した。それぞれの特徴を以下に記す。

#### ・BPM

BPMは、いわゆる曲のテンポを取得し、この特徴量によって曲の速さを推定することができる。

#### ・クロマグラム

クロマグラムは、全周波数帯域のパワーを[C,Db,D,Eb,E,F,Gb,G,Ab,A,Bb,B]の12音階に落とし込み、ある区間の時間における音の成分を可視化したものである。この特徴量から分散をとり、調(Key)を推定する。具体的には、特定の調を基準とし、分散が小さければ基準とした調が曲の調であると推定できる。

#### ・スペクトルセントロイド

スペクトルセントロイドとは、スペクトル重心とも言い、音響波形から周波数分析によって取得したスペクトルの midpoint のことである。すなわち、その周波数を分割点として、上と下でエネルギーが拮抗する。この特徴量から音の明るさを推定でき、値が高ければ甲高い音が、低ければ重い音ということが分かる。

また、取得した音響特徴量を可視化できるように、データベースからCSVファイルを出力できるように実装した。これにより、学習モデルへのデータ参照も簡略化することもできる。

これからの実装予定としては、取得した音響特徴量を学習モデルへ受け渡すために、変数を数値化する必要があるため、One-Hotベクトルに変換するシステムを実装する。また、音響特徴量も

曲の雰囲気をより正確に推定するにあたって、実装している三つの音響特徴量では不十分であるため、昨年度用いた音響特徴量も参考にしつつ、新たな音響特徴量を追加する必要がある。今現在、検討している音響特徴量は「音量」であり、音量によって、静かな曲や騒がしい曲の判別ができるのではないかと考える。

(※文責: 菊地陵雅)

### 3.3.5 曲決定システム

このシステムでは、流れている曲の特徴やシーンの雰囲気などを参照して、次回に流すべき曲を決定する。現在、ゲームなどに実装されているインタラクティブミュージックを調べていたところ、それぞれに専用の曲が用意されていることが多く、その専用曲同士を行き来することで、自然な曲の遷移を可能にしていたことがわかった。それらの専用曲同士は、メロディーラインを共有していたり、遷移する場所同士の音の響きが非常に似ていることなどがわかった。これらのことを踏まえて、私たちはAIを用いることで、これらインタラクティブミュージックをリアルタイムに自動生成することが可能なのではないかと考えた。以下の4.6の類似曲検索システムでは、曲同士の繋がりを意識した曲選択、4.7のシーンに合った曲の選択システムでは、場面との相性を意識した曲選択を行い、これらを相互的に作用させることでインタラクティブミュージックに適している次回曲の選択を行う。

(※文責: 高橋昂大)

### 3.3.6 類似曲検索システム

#### 3.3.6.1 手法

曲をそのままデータ化したもの同士の比較は、データが2次元（音程・時間）のため扱いづらいことや処理に時間を要することから、シームレスに曲を繋ぐインタラクティブミュージックの実装には不向きであると考えた。インタラクティブミュージックでは、ゲームへの没入感を上げるために、シーン遷移やフィールドの遷移時にBGMも大きな遅延がなく遷移する必要がある。そのため、曲の選択を行う際の処理を減らし、処理時間の高速化を行う必要もある。そこで、ニューラルネットワークを用いて事前に曲データを数値化し1つの曲を1つの1次元数列で表すことで、曲の選択を行う際の処理を数列同士の比較に置き換えることができる。それに伴い、類似曲を見つける処理の高速化も可能になる。上記より、事前処理としてニューラルネットワークを用いた、曲の1次元数列化を最初の目標とした。

(※文責: 高橋昂大)

#### 3.3.6.2 ニューラルネットワーク構築

ニューラルネットワーク構築を始めるにあたって、学習用の入力データと出力データを定める必要がある。まず、曲ごとの正解データをジャンルとして、EDMや和風、ロックなどを含んだ8ジャンルをワンホットベクトル（1つの成分が1で残りの成分が全て0であるベクトル）としてタグ付けを行った。そして入力データとして、任意の曲から200msごとに取り出した128次元のクロマベクトルとした。これは200msごとに取り出すことで大体の曲の特徴を概ね捉えることが

できるからである。これらの数値データを元に、128次元のクロマベクトルを入力として、8つのジャンルの中から1番特徴を捉えているジャンルを推定するニューラルネットワークを構築する。結果として、128次元の入力層、25次元の中間層が2層、8次元の出力層としてニューラルネットワークと構築を行い、入力層・中間層のそれぞれにバイアス（他の数値に依存しない層に与えられたデータ）を1つずつ付与した。また活性化関数として出力層にはソフトマックス関数、他の層にはシグモイド関数を適用し、最適化アルゴリズムとして Adam を用いた。

（※文責: 高橋昂大）

### 3.3.6.3 曲のベクトル化

上記のニューラルネットワークをそのまま用いてしまうと、任意曲のクロマベクトルをジャンル分けするシステムになってしまう。そのため、ジャンルが確定する1つ前の中間層に注目してみた。8次元の出力層に比べて25次元の中間層は単純にベクトルの数が多く、細かく曲の特徴が反映されているのではないかと考えた。何より出力層データは、曲同士が似ているかよりもジャンルが同じかどうかを優先してベクトル付けを行ってしまう。現に、オリジナルと和風アレンジされたオリジナルの曲は、中間層時点では、一番類似度が高い曲として認識されるが、出力層では、和風アレンジされたオリジナルは、他の和風曲との方が類似度が高いとみなされた。インタラクティブミュージックでは、前者を類似曲としてみなすことが多いため、曲のベクトル化を行う際には出力層の1つ手前である中間層を用いてベクトル化を行うことにした。また、1曲ごとにベクトル化を行うため、曲をクロマベクトルに分解し、それぞれのクロマベクトルを4.6.3で構築したニューラルネットワークを適用する。そしてそれぞれの出力結果を合算し、平均をとることで曲全体としての特徴を得ることができる。これらを任意の曲に対して同時に行うことで、1つの曲を25次元の特徴ベクトルとして表すことができる。

（※文責: 高橋昂大）

### 3.3.6.4 類似曲の検索

4.6.3で得ることができた、曲ごとの25次元の特徴ベクトルを比較することで、曲同士の類似度を求めることができる。この類似度は、ベクトル同士のコサイン類似度で求めている。そのため類似度の値として、0～1の間で出力されることになり、0なら全く類似していない、1なら完全に一致と言うように1つの値のみで類似度を出力できるようになる。上記のシステムの流れを一方の曲を固定して行うことによって、固定した曲との類似度を各曲ごとに得ることができる。

（※文責: 高橋昂大）

## 3.3.7 シーンに合った曲の選択システム

### 3.3.7.1 手法

昨年度までに開発されていたBGM選択システムを、今回の目的に対して扱いやすいように改良を行う。昨年度では、Hevnerによる8つの印象語群を用いた感情特徴量からクロマベクトル、スペクトル重心、テンポの推定を行い曲を決定していた。今年度はそれらを元に文章からの曲決定とフィールドからの曲決定の大きく2つに分けて、曲決定を行っていく。

(※文責: 高橋昂大)

### 3.3.7.2 文章からの曲決定

文章からの曲決定では、昨年度と同様の手法をとる。加えて、今年度は文章に対しての感情分析の自動化を検討している。pythonのML-Askライブラリを使用することでテキストデータを「喜・怒・昂・哀・好・怖・安・厭・驚・恥」の10種類の感情で表すことができる。これらをHevnerによる8つの印象語群に落とし込むことで、昨年度のシステムや学習データを生かしながらも、より自動化の範囲が広がったシステムを構築することができる。また、曲ごとにシーンに合っている適合度づけを行うことで、類似曲システム類似度との相互的な曲決定を行うことも可能になる。

(※文責: 高橋昂大)

### 3.3.7.3 フィールドからの曲決定

フィールドからの曲決定では、あらかじめフィールドごとに振り分けられた曲の中から類似曲を探す処理を行う。また、曲の振り分けには4.6.2で構築したニューラルネットワークを用いることで、ニューラルネットワークを書き換えることなく、正解データをフィールドのワンホットベクトルとするだけで大きなデータ数でも瞬時に振り分けられる。

また、類似曲を探す処理でも4.6.4のシステムを流用することで新しくシステムを構築することなくフィールドからの曲決定を実装することができる。

(※文責: 高橋昂大)

## 3.3.8 曲繋ぎ場所検索システム

曲繋ぎ場所検索システムでは、前述までのシステムで決められた曲同士を元に、曲の切り替わるタイミングを決めるシステムである。タイミングの決定方法は、切り替わり時に違和感を与えないように、まず拍を合わせることを優先する。またテンポについては次項で説明を行うためここでは省略する。切り替わる前と後で拍のずれが生まれてしまうと、同時に違和感も生じてしまうため、切り替わるタイミングを小節の頭に限定して行うことにした。また、8小節を切り替えるタイミングにすることが音楽的には好ましいが、曲の切り替えに時間がかかってしまう可能性が高いため今回のインタラクティブミュージックの実装には向いていないと判断した。加えて、今回の曲同士はテンポが違う場合があるため、曲同士のテンポ合わせやより滑らかな遷移を実現したいという観点から、1小節かけて曲同士を調整し小節線で曲を完全に遷移させる方針に決定した。

上記の方針に合わせた曲同士の切り替わるタイミングを決定するために、4.6.3の類似曲検索システムを流用することにした。1小節を局所的な短い曲と見て類似曲検索システムを適用することによって1小節のフレーズをベクトル化することが可能になる。この操作を切り替える前の1小節に適用しベクトル化する。その後、後続曲に対して全ての小節に対し同じ操作を行うことで小節それぞれのベクトルを得ることができる。この後続曲のベクトル群と切り替える前のベクトルで類似度の測定を行う。この操作の中で1番類似度の高い1小節同士が曲の特徴が最も近いと見なすことができ、最も適切な切り替わるタイミングとすることもできる。

### 3.3.9 曲繋ぎシステム

上記全てのシステムを通して、曲の確定や切り替わるタイミングを含めたデータを得ることができた。本システムではこれまでの全てのデータを用いてインタラクティブミュージックの実装・出力を行う。具体的な手法としては現在考察中ではあるが、現状としてこのシステムでは3つの処理を行うこととしている。1つ目は曲のオーバーレイ(重ね処理)である。4.9の曲繋ぎ場所検索システム曲同士の繋がりが1小節であるとして、類似している1小節をそれぞれの曲中から探し出した。これらの小節をオーバーレイすることで単調なフェードイン・フェードアウトではない曲遷移を実現する。また、オーバーレイによって曲の切り替わりが行われるため、曲同士の切れ目がなくより一層インタラクティブミュージックに近づくことができ、没入感の向上にも期待できる。2つ目は曲同士のテンポを合わせる処理である。それぞれの曲にテンポアップやダウンの処理を適当に施すことで、曲同士の元テンポが異なっている場合でもオーバーレイ部分のテンポの相違を解消することができる。付随して、切り替わり後に元テンポに戻す処理も行う。これらの処理をすることでオーバーレイ部分の違和感を解消しつつ次の曲に繋ぐことが可能になる。もう1つの手法は音量の調整処理である。曲の切り替わり中にそれぞれの曲に相互的な音量調整を行うことで自然な遷移が可能になる。シンプルな方法を挙げるとクロスフェードが挙げられる。フェードインとフェードアウトをそれぞれ同時に行うことによって急な音量変化がなく、音量的に自然な曲遷移をすることが可能になる。これらそれぞれ適応することでインタラクティブミュージックを実装する。

(※文責: 高橋昂大)

## 3.4 物語班

### 3.4.1 世界観設定

#### 3.4.1.1 世界観決定までの経緯

プロジェクト開始直後、プロジェクト全体でゲームの大まかなジャンルを、「謎解き要素を含む探索ゲーム」「ファンタジー」と定めた。また、出来れば盛り込みたい要素として「ホラー」「SF」が候補に挙がった。これらの決定事項をもとに、物語班で具体的なゲームの世界観を考え、各班に提案することとなった。世界観のアイデアとして、遺跡、近未来、魔法世界、神話世界、「不思議の国のアリス」等が提案された。これらの中から、物語班で、「不思議の国のアリス」モチーフの世界観と、エジプトなどの古代の神話世界をベースとした世界観の2つに絞り込んだ。詳細を以下に示す。

#### 不思議の国のアリス

舞台は主人公が迷い込んだ「不思議の国」である。「不思議の国のアリス」にちなんだ世界で探索を行い、元の世界への帰還を目指す。

#### 古代の神話の世界観

舞台は古代の神話をモチーフにした世界。かつて王であり殺害されたファラオのオシリスがなぜ亡くなったのか、冥府を探索し関係者の靈魂と会話することで謎を解いていく。

これら二つの世界観は、「不思議の国のアリス」やエジプト神話など具体的な作品が元になっており、キャラクターやギミックなどの設定において一から新しく決めなければならないものが少ないこと、メンバー間で世界観の共有が行いやすいことから選出された。その後、プロジェクト全体に対しこの二つの世界観を提案し、多数決を行ったところ、より具体的であり、世界観の共有が行いやすいとして「不思議の国のアリス」モチーフの世界観が選ばれた。大まかな世界観が決定した後は、すでにゲームに盛り込むことが決定していた要素である「ギミックつきマップ」「キャラクターの背景史」を考慮しつつ「不思議の国のアリス」をもとに細かな世界観設定を行った。世界観は全ての班の作業に関係する重要な要素であるため、物語班の中で設定を確認した後はプロジェクト全体に共有し、不明な点や改善点等の意見を貰った。ゲームの舞台となるマップについては、「不思議の国のアリス」に登場する場所をベースに、システム班と面の数を確認しながら決定した。キャラクター設定については、キャラクターを「不思議の国のアリス」の登場キャラクターとするか、オリジナルのキャラクターにするかという選択の余地があったが、イメージの共有がしやすいこと等から、「不思議の国のアリス」のキャラクターをもとに、動物のキャラクターを人型にするなど、新たな要素を付け加える形で設定を行うこととした。本作の登場キャラクターは全員、不思議の国において演じている性格と、現実世界における本来の性格の二つを持つが、主に描写されるのは前者であるため、不思議の国における性格のみ設定を行った。主に視覚班と情報共有を行いながら、物語班が主体となって設定を行った。以下に決定した世界観の設定とキャラクター設定を示す。

(※文責: 吉井史夏)

### 3.4.1.2 世界観設定

ゲームの舞台である「不思議の国」は、ハートの女王が作り出した現実とは異なる世界である。何らかのトラウマを抱え、現実から逃げ出したいと願う人々が、この世界に迷い込む。迷い込んだキャラクターたちは、現実での自分の記憶を持ったまま、女王に与えられた「白うさぎ」「帽子屋」などの役割を演じることでこの世界の住人となる。「不思議の国」に長く留まると現実世界での記憶を徐々に失う。古くからいる「不思議の国」の住人たちの中には、既に現実での記憶を完全に失った者もいる。現実から逃げてきた人たちの心の拠り所としての世界であり、辛い記憶を持つ彼らを楽しませたいことから、「不思議の国」はカラフルで全体的に明るい雰囲気である。しかし、どこもなく不安定で、何かが狂っていることを本能的に感じさせる。「不思議の国」には「小屋」、「森」、「ハートの女王の城」が存在する。「森」では帽子屋と三月うさぎによって「お茶会」が開かれている他、ハートの女王の城へとつながる「薔薇の迷路」がある。「ハートの女王の城」には「裁判場」が併設されている。「不思議の国」は始めは平和だったが、次第にハートの女王が暴走し支配的になっていき、住人たちは女王に不満を持つようになる。けれど女王に首を斬られたものはこの世界から追放され、現実世界へと帰されるという噂があり、現実におびえる住人たちは女王に歯向かうことができない。

(※文責: 吉井史夏)

### 3.4.1.3 場所設定

#### ・小屋

ゲーム序盤においてアリスが迷い込むことになる家。現実と「不思議の国」との間に存在する、「不思議の国」における入り口のような場所である。実は白うさぎの家。全体的にカラフルで明るい

が、その明るさが逆に訪れた人を不安にさせる。ケーキや飲み物などが置かれているが、これらを口にする则食べた人の体が大きくなったり、小さくなったりする。これは、白うさぎがここが現実ではないことを新しい住人に手っ取り早く説明するために用意しているアイテムである。

(※文責: 吉井史夏)

#### 3.4.1.4 キャラクター設定

##### ・アリス

このゲームの主人公。外見は10歳ほどの少女である。一人称は「私」。

「不思議の国」に迷い込んだばかりであり、女王からまだ役割を与えられてはいないものの「不思議の国のアリス」のアリスように好奇心旺盛な性格である。自分の意見をはっきりと口にするが、基本的には見知らぬ人には敬語を使うなど礼儀正しい。独り言が多く、自分に対して話している言葉と他人に対して話している言葉の区別が付きにくい。これがもつで周囲を混乱させてしまうこともある。

不思議の国に来て間もないため、現実での出来事ははっきりと覚えており、それゆえ現実に戻りたくないという思いを強く持つ。「不思議の国」では現実とは違う自分になりたい、変わりたいと思っている。現実のことは覚えているが、不思議の国にきた当初から自分の名前は忘れてしまっている。

##### ・白うさぎ

外見は15歳ほどの、長身で眼鏡をかけた男性。アリスを「不思議の国」へと誘う。

女王から「不思議の国」の事務的な管理と、住人となる資質を持っている人(トラウマを持ち、現実から逃げたいと願っている人)を勧誘する役目を与えられている。加えて、「少々臆病で気が弱いところがあり、それによって女王を困らせる」という役割も持つ。与えられた役割として女王を困らせているので、首をはねられることはない。臆病な性格は不思議の国にいるとき限定のもので、現実世界でアリスを勧誘した際には、極めて冷静でドライな本性を見せる。一人称は基本「私」だが、「僕」が時々混ざる。口癖は「一応」「多分」。

不思議の国の住人としては古い方で、自分が現実世界から来たことは覚えているものの、どのような人間だったかははっきりと覚えていない。また、過去を窺わせるような発言もしない。

##### ・チェシャ猫

外見はアリスと年の近い少年。森でアリスに「不思議の国」のことを教えた後、少しの間共に行動することになる。

飄々とした態度でいつもにやにやと笑っているが、アリスが森を抜けるためのヒントをさりげなく教えてくれるなど、優しいところもある。女王から与えられた役割は、「人を道に迷わせる」。実は不思議の国での自分の役割に飽きていて、現実世界に戻ろうとしている。

不思議の国では古参の住人に当たり、現実世界での記憶の大部分を失っている。だからこそかつての現実世界での自分に興味をもち、記憶を取り戻すためにハートの女王に逆らおうとしている。

##### ・芋虫

人間より虫に近いが完全に虫ともいえない外見を持つ住人。森でアリスと出会い、話をするようになる。

難しい言葉やことわざなどを多用し、年寄りくさい話し方をしているが、本人は意味をきちんと理解できていない。そのため、答えのわからない質問をされたときは無茶苦茶な理論で押し通し、怒り始める。時々タバコを吸ってはむせている。女王から与えられた役割は「賢そうにふるまう」。

不思議の国に来たのは割と最近で、現実世界でのことも覚えている。ハートの女王の圧政に対し不満を持っているが、現実には戻りたくないため、表立って逆らうことはない。

・帽子屋

どう見ても人間ではない外見をしているが、歩き回り、人間のように話すこともできる。

三月うさぎと共に森でお茶会を開いている。言動が奇妙で、唐突になぞなぞを出すこともあれば、喚いた後、急に静かになることもある。女王から与えられた役割は「終わらないお茶会を開き、狂ったように騒ぐ」。

不思議の国に来た当初は役割として仕方なく狂っているように演じていたが、次第に本来の性格を忘れ、演じていた性格が本当の性格になりつつある。

・三月うさぎ

外見はノウサギ。人間のように話し、飲食もできる。

帽子屋と共に森でお茶会を開いている。帽子屋と同じくおかしな言動をし、アリスとは会話がかみ合わない。人の言葉に茶々を入れたがる。女王から与えられた役割は帽子屋と同じく「終わらないお茶会を開き、狂ったように騒ぐ」。

帽子屋よりも現実世界のことを覚えている。次第に現実世界のことを忘れ、三月うさぎとしての自分が当たり前になっていくことに恐怖を感じているが、現実には戻りたくない。

・トランプ兵

森と城をつなぐ薔薇園で庭師を務めている住人達。手際が悪く、くだらないことで言い争ってばかりいる。また、女王のことを非常に恐れている。女王から与えられた役割は庭の整備と、「女王に忠実である」。

古参の住人ではないが、完全に現実世界での記憶をなくしている。そのため、女王に首を切られることを過剰に恐れているが、そもそも現実世界を認識していないため、首を切られたらどうなるのかについてはわかっていない。

・ハートの女王

外見は、アリスよりも幼い少女。

強気な言動で、傍若無人な性格である。意味のないことでいつも怒っている。自分の周りにたくさん人がいて、そのうえ自分の思い通りに動いてほしいという欲望を持つ。癩癩持ちで衝動的に動いてしまう部分がある。「首をはねろ！」が口癖だが、癩癩の結果思わず口に出してしまう言葉というだけであり、これまで実際に首をはねたことはない。

不思議の国を作り出した張本人。迷い込んだ住民たちに役割を与え、その通りにふるまわせている。不思議の国においては自分が絶対だという考えを持っており、支離滅裂な考えであっても押し通してしまう。

(※文責: 吉井史夏)

## 3.4.2 シナリオ執筆

### 3.4.2.1 シナリオについて

今年度のプロジェクトでは、「登場キャラクターの背景史」の自動生成に重きを置くことが決定していた。そのうえで、メインシナリオに自動生成要素を入れるか否かについては議論があったが、最終的にエンディングの自動生成のみを行い、それ以外の部分のメインシナリオについては全て人の手で執筆することとした。人間の手で執筆することで、自動生成した背景史やマルチエンディングに対し矛盾が無いようにメインシナリオを調節できることが期待される。さらに、先生から、メンバー自らシナリオを執筆する経験は、AIに物語を作成させるという課題に対して、クリエイティビティの発揮の面で非常に有意義となると助言をいただいた。ゲームの世界観決定後、まず物語班内でそれぞれメインシナリオの簡単な起承転結の案を出し合った。執筆担当者がそれらの案を参考に大まかな流れを考案し、物語班の中で共有した。システム班との話し合いにより、舞台となる場所は決まっていたので、考慮しながら作成した。さらに大まかな流れを起承転結に落とし込み、班内で了承を得た後、全体に共有した。現在は、プロローグと、プロトタイプ作成のために行った第一章の執筆が終了している。今後は物語の流れをブラッシュアップするとともに、自動生成を行うエンディングの前まで執筆を進めていく予定である。また、自動生成に適応できるよう内容の細かな調節も進める予定である。具体的には、マルチエンディングの自動生成に矛盾なくつなげられるように分岐を適切に配置するとともに、背景史の自動生成が完了次第、作成された背景史をメインシナリオ内に組み込むことを想定している。

(※文責: 吉井史夏)

### 3.4.2.2 あらすじ

トラウマを抱え、ままならない現実に思い悩むアリスは、白うさぎに導かれて「不思議の国」へと迷い込む。「不思議の国」の住人たちは、女王から与えられた役割をこなすことで不思議の国にとどまり、現実世界から逃げ続けていた。アリスは「不思議の国」の住人となることを望むが、「不思議の国」に長く居続けた場合、現実での記憶を失ってしまうことを知る。女王の圧政を目の当たりにし、自らと境遇の近い住人たちと関わる中で、アリスは徐々に現実と向き合うことを考え始める。そんな中、アリスが関わった住人の一人であるチェシャ猫が裁判にかけられて、斬首されるかもしれないということを耳にする。裁判に乱入したアリスは、チェシャ猫から女王に逆らって共に現実へ戻らないかと誘われる。アリスは、現実に戻るか、不思議の国にとどまるか決断を迫られる。

(※文責: 吉井史夏)

### 3.4.2.3 シナリオの流れ

シナリオは、全5章を想定している。以下にその大まかな流れを記述する。

#### プロローグ

- ・アリスは現実世界で何かしら問題を抱え、思い悩んでいる。
- ・アリスの前に白うさぎが現れる。白うさぎはアリスを不思議の国へと誘う。
- ・アリスは白うさぎを追ってうさぎ穴へ落ちる。その後、小屋へとたどり着く。

## 1 章

舞台：小屋

登場人物：アリス, 白うさぎ

- ・アリスは、白うさぎを追って小屋から出ようとする。
- ・小屋の探索。

## 2 章

舞台：森

登場人物：アリス, チェシャ猫, 芋虫

- ・アリスが小屋を出ると、森へ出る。そこでチェシャ猫と出会う。
- ・チェシャ猫から、ここがハートの女王が管理する世界である「不思議の国」だということと、この世界の住人になりたいのなら、まず女王に会いに行くべきだということを知る。アリスは住人となるため、女王に会おうと城を目指すことにする。
- ・森を探索し、芋虫に出会う。
- ・芋虫から女王の圧政に対する不満を知る。アリスはなぜ女王に不満を伝えないのか聞かすが、現実から逃げたい住人たちは女王には反抗できないのだと教えられる。

## 3 章

舞台：森, お茶会

登場人物：アリス, 帽子屋, 三月うさぎ, 白うさぎ

- ・アリスは森の中でお茶会を見つける。そこで、なんでもない日を祝っている帽子屋と三月うさぎに出会う。
- ・帽子屋と三月うさぎは、現実の記憶を忘れつつある。アリスはこれまでの記憶が無くなり、狂っていくことに恐怖を感じる。
- ・お茶会に白うさぎが通りかかり、アリスはチェシャ猫が裁判にかけられることを耳にする。
- ・アリスは、裁判場のある城に急いで向かうことにする。

## 4 章

舞台：森, 薔薇の迷路

登場人物：アリス, トランプ兵

- ・バラの迷路の探索中、アリスはバラを赤く塗るトランプ兵に出会う。
- ・トランプ兵たちは、現実のことを忘れ、ただ女王のために動くようになっていた。彼らの様子を見たアリスは、この世界の住民にはならず、現実と向き合うことを考え始める。

## 5 章

舞台：城, 裁判場

登場人物：アリス, 白うさぎ, チェシャ猫, 女王

- ・城を抜けたアリスは裁判場に到着する。裁判場では、ちょうどチェシャ猫の裁判が始まるころだった。
- ・アリスは、女王に対面し、チェシャ猫の弁護を名乗り出る。
- ・アリスは女王の無茶苦茶な意見に対して反論するが、女王は自分に逆らうとこの世界にいられなくなると脅す。現実にトラウマがあるアリスは悩む。

・チェシャ猫は、現実に戻りたくてわざと裁判にかけられたとアリスに言う。女王に首を切られたら現実世界に戻れるかもしれないことが判明する。

・アリスは、自分の意見を貫いて女王に首を切られるか、女王に従って現実から逃げるか決断を迫られる。

エンディングは分岐によって自動生成を行う予定である。

(※文責: 吉井史夏)

### 3.4.3 背景史

#### 3.4.3.1 目標

近年、物語の自動生成分野は目覚ましい発展を遂げている。物語には主人公をはじめとして、ヒロインや道を行き交う町人などの多種多様なキャラクターが登場する。彼らは物語の展望を左右する重要な存在であることもあれば、ただの通りすがりの一般人であることもある。特に、キャラクターの過去や生い立ちといった経歴は、物語にも大きく関与することがある。これらを踏まえて、物語においてキャラクターの存在は必要不可欠であると考えた。しかし、物語に登場するキャラクター固有の経歴の自動生成は先行研究がない。そこで、キャラクターの背景史を自動生成することで人工知能を用いた（物語の自動生成分野に）新たな創造性の再現を試みようと考えた。なお、ここでは表現の簡略化を図るため「キャラクター固有の経歴」を「背景史」と定義した。現実の世界に存在する全ての人間はそれぞれが唯一無二の背景史を持っており、物語に登場するキャラクターも同じく背景史を持っていると考えたためだ。また、今年度はゲームのキャラクターが「現実世界に辞易している」という共通点があるため、「現実逃避しているキャラクターの背景史の自動生成」を目指す。

(※文責: 坂本珠凜)

#### 3.4.3.2 分析の目的

自動生成アルゴリズムの構築において、想定される出力には入力させる値との有意性が求められる。現実逃避しているキャラクターの背景史を自動生成するにあたって、既存の作品の登場人物がなんらかのトラウマを想起する過程の膨大なデータアセットを用意する必要がある。これは背景史の自動生成が新規性のある研究課題であり、適切な既存のデータアセットがないためだ。そこで、背景史担当では登場人物のトラウマの想起に関するデータアセットを作成することにした。このデータアセットから各データの相関性を分析することで、「どんな人物がどのような経緯を経てトラウマを抱えるに至ったか」をパターンとして抽出する。このパターンをもとに「現実逃避しているキャラクターの背景史」の自動生成アルゴリズムの構築が可能になると考えた。

(※文責: 坂本珠凜)

#### 3.4.3.3 作品選定

背景史を自動生成するアルゴリズムの生成のために必要なデータを収集した。現実逃避しているキャラクターの背景史を自動生成する為に、作中で現実逃避をしている人が多く登場する作品を分

析対象とした。分析作品は何らかの賞を受賞している作品または、アニメ化されていて一般的に完成度が高いと認識されていることを収集基準とした。これらの基準から前期の段階では、名探偵コナン、おジャ魔女どれみ、ハリー・ポッター、笑ゥせえるすまんを分析対象に決定した。そのほかにも闇金ウシジマくんも分析対象となっていた。しかし、登場するキャラクターの多くが、お金に関する現実逃避をしていることが予測された。お金に関する現実逃避は世界観とマッチしておらず、データの偏りを防ぐために闇金ウシジマくんは分析対象から外された。

・コナン

2001年に第46回小学館漫画賞少年部門で受賞している。また、アニメ化や映画化もされている。この作品では、殺意を抱くほどのトラウマを抱えるキャラクターのデータが収集できることに期待し分析対象に決定した。

・おジャ魔女どれみ

1999年にアニメ化された。また、映画化もされ、主人公らの16歳以降の物語が書かれた小説も出版された。この作品では、小学生年代のキャラクターが多く登場する。そのため、この作品に出てくる現実逃避の原因の多くが大人にとっては問題にもならないが、小学生の年代にとってはとても深刻な出来事の場合が多い。また、小説版では、高校生年代のキャラクターが多く登場する。小説版では、思春期特有の現実逃避の原因が多く存在する。そのため、子供特有の現実逃避と思春期特有の現実逃避をしているキャラクターのデータが収集出来ることに期待し分析対象に決定した。

・ハリー・ポッター

2018年にハリーポッターシリーズの全世界累計発行部数が5億を突破した。また、映画化もしており、映画シリーズは8本ある。この作品は、魔法世界の物語なので、キャラクターの持つ現実逃避の原因が魔法などのファンタジー要素である現実逃避をしているキャラクターのデータが収集できることに期待し分析対象に決定した。

・笑ゥせえるすまん

1989年にアニメ化された。また、2017年には原作エピソードにオリジナルエピソードを追加した笑ゥせえるすまんNEWが放送された。この作品では、人間の愚かさや醜さ、人間関係が原因で現実逃避をしているキャラクターのデータが収集出来ることに期待し分析対象に決定した。

(※文責: 入船真誠)

### 3.4.3.4 分析準備

分析を行うにあたり、まずデータの収集を行いそのデータを用いて解析を行う。ここでは、データの収集についてと解析を行う前の前準備に分けて記述する。

・データの準備

分析の為のデータ収集は、分析の目的で述べた「どんな人物がどのような経緯を経てトラウマを抱えるに至ったか」という点をデータとして記述する必要がある。そのため、まずどのような物をトラウマとして判定するかを決定した。今回のデータ収集では、ある人物が現実世界において「辛い目に遭うこと」や「尾を引くような問題を抱えている」ことを対象にデータを収集した。その際に、問題の大小は考慮しない物として収集している。以上のような人物の属性を以下に示すように

分けた。人物の年齢や職業を示す「人物ステータス」、人物のトラウマの内容をテキストで記述した「トラウマ」、人物がトラウマに対してどのような認識をしているのかを示す「現状認識」、トラウマによって人物に表れた影響を示す「影響(言動・感情)」、第三者がトラウマを負った人を見た際にトラウマを推測する手がかりとなる「手がかり」、トラウマの要因を環境、人間、自己の3つに示した「分類」。これらの6つの属性に分けた。上記のデータの属性で選定した作品全ての対象者をデータとして記述した。

#### ・解析の準備

解析を行うにあたり、収集したデータを解析に適した形に変更する。そのためにKJ法を用いた分析データのフォーマットの作成を行った。今回想定している現実逃避の要因の属性を共起分析や因子分析のような解析を行うためには、必要な情報としてベクトル化した情報が必要である。分析前に収集したデータでは、属性ごとに記述しているデータである「現状認識」と「影響」をベクトルで表すフォーマットが無い状態である。またトラウマおよび現実逃避の要因をデータとして示した物は、テキストの状態であり、ベクトルの表記がしやすいよう必要な属性に細分化するフォーマットの再考が必要である。データのフォーマットの決定の為に、KJ法を用いてデータの属性を類似な物にまとめ、それらをベクトルで表記できるよう決定を行った。以上の操作を「影響」という指標においてどのように決定したのかを見ていく。「影響」の指標では、トラウマによる影響を感情・行動の二つを混ぜて示している。そのため手順として、まず、現実逃避をするような事象による感情への影響と行動への影響を二つに分け、それぞれにKJ法を適用する。さらに詳しく分別した感情への影響を見ていく。感情への影響は、エクマンの感情モデルである、楽しみ・嫌気・悲しみ・恐れ・怒りを利用して分割を行った。感情への影響を分割した後、各感情を10で表される5次元のベクトルとして表現するように決定した。例えば、恐怖であれば、(0,0,0,1,0)というように恐れに該当する部分に「1」が入りそのほかには「0」が入る。先に示したようなベクトルとして表現する方法を決定した。上記で挙げた例のような手順を、他の属性およびテキストにあてはめ、ベクトルとして表現するフォーマットを作成している。

(※文責: 西村秀幸)

### 3.4.4 マルチエンディング

#### 3.4.4.1 目標

エンディングは、全ての作品に必ず含まれており、作品に大きな影響を与える部分でもある。実際に、エンディング以外が素晴らしい完成度でも、エンディングに何かしらの失敗が見られた場合、その作品のイメージは大きくダウンする。また、昨今では、マルチエンディングという、プレイヤーの選択や行動により、エンディングが良い方向、悪い方向などに変化するものを採用するゲーム作品が多くなっている。このマルチエンディングを入れることにより、物語に幅を持たせることができる。また、プレイヤーに選択させ、緊張感を持たせることで、ゲーム作品に良い影響を与えることができる。そこで、今回はマルチエンディングの自動生成システムを作成し、人工知能の新たな視点から、創造性を再現することを試みる。この自動生成システムには、先行研究がないため、新規性が高い。また、物語の幅を従来の作品より広げることが可能になると考えている。

(※文責: 幸崎駿祐)

#### 3.4.4.2 分析の目的

自動生成システムの作成において、どのような値を入力をするかを考えなければならない。加えて、エンディングのパターンを導き出し、それをもとに出力するプログラムを構築する必要がある。そのため、入力の値とエンディングのパターンを探し出すために、条件に適した作品を探し、その作品のエンディングを分析する必要がある。

(※文責: 幸崎駿祐)

#### 3.4.4.3 作品選定

エンディングを自動生成するアルゴリズムを生成するため、エンディングデータを多く集める必要がある。本項では、エンディングデータを収集する作品の選定について記述する。データを収集するとき、良質なデータを集める必要がある。このことを踏まえて、選定対象作品に2つの条件を設けた。

1つ目の条件は、対象作品が、日常から非日常に遷移する物語であることである。今回、クリエイティブ AI が作る作品は「不思議の国のアリス」をモチーフにした作品である。また、シナリオも「不思議の国のアリス」を元に、手動で作成する。そのため、日常から非日常に遷移する物語のエンディングを取らなければ、全体のシナリオと自動生成したエンディングを結合した際に、矛盾が生じる可能性があるためである。

2つ目の条件として、何かしらの賞を受賞している、またはノミネートされている作品であることである。エンディングデータを収集する際、良質なデータを収集する必要がある。良質なデータでなければ、目標として出したい出力に相違が生まれる可能性があるため、必須である。ここでこのことを踏まえて、評価の高い作品、つまりはアカデミー賞、日本ゲーム対象といった賞を受賞している作品、もしくはそれらの賞を取るまでには行かなかったがノミネートまで残った作品を選定するのが良いと考えた。

上記の条件をもとに、映画、ゲーム、小説等の作品を調べ上げた結果、38本の作品が見つかった。作品例として、4つの作品を紹介する。1つ目は、「千と千尋の神隠し」である。あらすじを紹介する。主人公の千尋は不思議な町に迷い込み、両親が豚にされ、湯婆婆の元で働くことになる。その後、湯婆婆からの難題をクリアし、両親と日常の生活に戻ることができたが、異世界のことは忘れてしまう。この作品は、映画作品であり、条件の1つ目である日常から非日常に遷移する物語を満たしている。2つ目の条件では、2002年に日本アカデミー賞 最優秀作品賞を受賞している。以上の点から、選定対象作品とした。2つ目は「アンダーテール」である。この作品は、人間世界に住む主人公が、魔物が住む地下世界へと落ちてしまい、脱出する方法を探すというストーリーだ。主人公の行動によって4つのエンディングに遷移するという特徴がある。この作品はゲーム作品であり、1つ目の条件の日常から非日常に遷移する物語を満たしている。2つ目の条件では、2018年に日本ゲーム大賞優秀賞を受賞している。以上の点から、選定対象作品とした。3つ目は「ゼロ・グラビティ」である。あらすじとして、主人公のライオンが地球から宇宙へ行き、スペースシャトルで作業をしていたときに事故に巻き込まれ、宇宙空間に吹き飛ばされてしまう。その後、様々な困難を乗り越え、無事に地球へと帰還する。この作品はSF映画であり、地球から宇宙へと日常から非日常に遷移しているため、1つ目の条件を満たす。2つ目の条件では、2014年に英国アカデミー賞 英国作品賞を受賞している。以上の点から、選定対象作品とした。4つ目は「十二国記」である。あらすじとして主人公の陽子が、異世界に送られ、困難に直面する。陽子は現実に戻るべきか

十二国で王となるべきか迷うが、荒れた国で苦しむ民を救うことが現実に戻ることよりも自らのやるべきことだと感じたため、現実には帰らないことを選ぶ。この作品は、小説作品であり、日常から非日常に遷移しているため、1つ目の条件を満たしている。2つ目の条件では、2014年に本屋大賞を受賞しているため、条件を満たしている。以上の点から、選定対象作品とした。このような条件をもとに選定対象作品を探した。なお、作品は今後追加する可能性がある。

(※文責: 幸崎駿祐)

#### 3.4.4.4 分析準備

分析可能な状態にするため、属性を決める作業を行った。まず、条件に当てはまる作品の中から数十作品を抽出し、エンディングの抽出を行った。エンディングのあらすじを書きだし、属性の共通点を洗い出した。そして属性の共通点を元に仮のフォーマットの作成を行った。その後、仮のフォーマットに、収集したエンディングのデータを当てはめた。

全てのエンディングを仮のフォーマットに当てはめた後、入力する値の統一とデータフォーマットを決定するため、KJ法を用いた。実際に、当てはめた属性の入力の値にばらつきが見つかった「帰れる or 帰れない理由」と「感情」にKJ法を用いて、入力する値を統一することができた。加えて、KJ法を用いることで、属性内の対比関係があることが明らかになった。例として、「感情」では、「興奮」と「安心」といった対比関係があることが分かった。この関係性をもとに、後に述べるベクトル化を行った。

入力する値を統一することを行ったため、仮のフォーマットを元に、正式なフォーマットを作成した。各属性を「主語」、「帰る or 帰らない」、「何人で帰えるのか」、「帰える or 帰らない理由」、「感情」、「後日譚があるか」、「後日譚の主語」、「状態」、「行動」と決定し、選定した作品をこれらに当てはめてデータフォーマットを作成した。このとき、「作品名」と「エンディングのあらすじ」についても記入をした。なお、これらのフォーマットは次に記述するフォーマットの再考により、変更する可能性がある。

各属性を記入後、フォーマット全体を見直し、修正する部分が無いかを確認した。現時点で、属性に、「実際どうなったか」「なぜ帰れるのか(帰れないのか)」「なぜ帰ること(帰らないこと)を望んだのか」と最初に作成したフォーマットよりさらに詳細に書くことで出力結果に良い影響を与えるのではないかと考察した。このフォーマットの再考は試作段階であり、適応は現段階では行っていないため、後期のプロジェクト学習で行う予定である。

(※文責: 幸崎駿祐)

#### 3.4.4.5 分析方法

KJ法を用いたことにより、属性の対比関係が明らかになった。この対比関係を行列のベクトルとして可視化表現した。「帰れる or 帰れない」は3つのベクトル、「感情」は5つのベクトル「帰れる or 帰れない理由」は8つのベクトルで表現し、 $[1, 0, -1]$ の3つの数値で表現した。この数値は相関係数を出し、属性の強い関係性を表すために使用した。

属性間の相関係数を出すために、ベクトルとして可視化表現した表をExcelでCSVファイルに置き換え、Rという統計解析向けのプログラミング言語に入れた。結果として、「確保、喪失」と「可能、不可能」が0.60、「恋愛あり、恋愛無し」と「決意、迷い」が0.50、「救う、見捨てる」と「他人、自身」が-0.45のように、相関係数を出すことができ、関係性が強い属性も明らかになった。

## CreativeAI

しかし、データフォーマットの再考、作品データの追加により、これらの相関係数は変わってくると予測する。後期では最終決定したデータフォーマットと追加分の作品データをもとに相関係数を出す作業を行う。

(※文責: 幸崎駿祐)

# 第 4 章 中間発表

## 4.1 概要

2022年7月8日、公立はこだて未来大学において対面形式での中間発表会が行われた。事前準備として、プロジェクトの活動内容を説明するためのポスター1枚と、実際の発表で使用する、プロトタイプのデモ映像を含めた14枚の発表用スライドを作成して当日に臨んだ。当日は前半発表と後半発表毎に発表者を分け、10分のスピーチを行い、その後5分間の質疑応答時間を設けた。15分間の一連の流れを前半後半それぞれ3セット、合わせて6セット行った。また、聴衆に発表内容と発表技術についてのアンケートに答えてもらい、10段階での評価とコメントを回答してもらった。

(※文責: 小林裕汰)

## 4.2 成果物

### 4.2.1 ポスター

本プロジェクトの今年度のポスターはA1サイズで日本語と英語のバイリンガルとなっており、Adobe Illustrator 2022を使用して制作した。全体は白と黒を基調としており、班ごとにイメージカラーを設定して各班の領域が視覚的に明確になるように配色した。下図に中間発表用に制作したポスターを記載する。



図 4.1 中間発表用ポスター

図 7.1 のポスターの上部にはプロジェクト名・プロジェクト番号と班員・担当教員名を記載し、本プロジェクトが4つの班で構成されていることを明確にした。また、ポスターの本文には「プロジェクト概要」「グループ活動」「進捗と課題」「今後の展望」について記載した。プロジェクト概要

には本プロジェクトの目的や班構成、制作するゲームのコンセプトについて記載した。ポスター制作当初は「ゲームのコンセプトに沿って新たな物語を自動生成する」ことをプロジェクト概要として記載していたが、これは手段であって目的は前述の通りであるためそのように修正した。グループ活動には前期での各班の活動について、各班の領域に分割して記載した。こちらもポスター制作当初はダイアグラムを用いて各班の連携を表現しようと努めたが、矢印等を使うと本文を記載するスペースがなくなるなどしてうまくいかなかった。そこで、本プロジェクトの担当教員からご提案いただいた Puzzle Radial List というダイアグラム案を参考にした。このダイアグラムはパズルを基調としており、ピースが噛み合う様子が各班の強固な連携を表現できると考え採用した。また、グループ活動の本文は各班に記載してもらうように要請したが、提出された本文の分量が班ごとに異なったため、それらを日本語で 200 字程度に収まるように添削した。その後添削した本文をポスターの各班の領域に記載したが、本文が日本語と英語だけでは見栄えが悪く見づらいという意見が担当教員から出された。そこで、本文の近くに添付することを想定して、各班に現状の活動内容がわかるような画像をいくつか提出してもらうように要請した。その結果、比較的進捗が成果物として反映されているシステム班や視覚班から、数枚の画像が提出された。音響班と物語班は成果物が音楽やテキストであるため、自動生成アルゴリズムの構築に際し生じた画像や表を画像形式で提出してもらった。これらを活用することで、各領域のスペースをうまく活用して見やすいレイアウトを実現できた。進捗と課題には前期の活動を振り返り、プロジェクト全体で得た進捗と残った課題についてまとめたものを記載した。プロジェクト全体で得た進捗としては「プロトタイプの完成」が挙げられたが、課題としては以下の 3 点が挙げられた。

- ・自動生成アルゴリズムが未実装
- ・データセットが未完成
- ・各班の成果物の統合

これらは前期の活動では実現には至らなかったが、後期を含めた 1 年間で達成する目標であるため、今後は実現に向けて活動を進める。今後の展望には各班で具体的な後期の活動方針についてまとめたものを記載した。上述した課題はプロジェクト全体での課題だが、ここでは各班で取り組むべき課題について詳細に記載した。このように「プロジェクト全体での課題」と「各班の課題」を分別して記載したことで、各班の課題解決がプロジェクトの達成と直結していることがわかった。ポスターに関しては、レイアウトやダイアグラムといったデザインはおおむね見やすく作ることができた。特に白と黒を基調とした配色や、班ごとに設定したイメージカラーは後期の成果報告会でも継続して採用する。しかし、成果報告会では今回と同じ構成やダイアグラムを使用しない可能性があるため、今回のポスター制作に際しアドバイスをいただいたことを活かして、プロジェクトの活動と並行してポスター制作を円滑に進めることができるように努める。

(※文責: 坂本珠凜)

## 4.2.2 発表用スライド

今年度は対面形式の発表だったため、聴衆に活動内容を理解してもらうべく、14 枚のスライドを作成した。作成時は予め各班の進捗を確認後、スライド制作担当者が 10 分で行うスピーチの内容を先に決定する方式を選択した。そこから各班でスライドの中身を作成してもらい、最終的にス

ライド制作担当者がそれらを統合し、一つのスライドとして完成させた。スライドの内容は三つに大別でき、それぞれプロジェクトの概要、制作ゲーム内容、AIを用いた自動生成で構成されている。最初にタイトルと目次を載せたスライドを一枚使用し、プロジェクトの概要ではスライドを二枚作成した。最初のスライドでプロジェクトの目的と目的達成の意義について、次のスライドでプロジェクトの進め方について説明した。制作ゲーム内容のスライドでは、制作予定ゲームのコンセプト説明とプロトタイプの動画の二枚のスライドを作成した。自動生成要素の説明のスライドは8枚作成した。各班でメインとなる一つの自動生成要素が存在し、最初一枚を用いて各班で扱った自動生成を簡単に紹介した。その後システム班一枚、視覚班、シナリオ班、音響班で二枚のスライドを使用して各班の自動生成要素の解説を行った。最後のスライドは最初のスライドと同じものを用いて、全体のまとめを行った。スピーチではプロジェクトの概要で2分、制作ゲーム内容で2分、自動生成の内容で6分強となるよう設定していた。自動生成要素の説明では各班で発表者を選出した。時間がかかりギリギリで、伝えたいことが多く取捨選択を行ったが詰めきれない部分が存在した。そのため実際の発表では10分でおさまらないことが多々あった。

(※文責: 小林裕汰)

### 4.2.3 動画

中間発表までの作業内容を視覚的に表現するためにプロトタイプを作成し、それをプレイしている動画をデモ動画とした。当初、デモ動画の時間は30秒であったが、中間までに完成させた見せたい部分をすべて動画に収めると、既定の時間からオーバーしてしまうことが分かった。そのため、一度録画したものを再編集し、デモ動画に充てる時間を30秒から45秒に調整した。デモ動画の最後では、小屋ステージの続きである森ステージを、主人公が駆け抜けるアニメーションをUnityで作成し編集した。これは今後制作予定であるステージの1つに森のステージが含まれており、システム班の今後の展望を伝えるために、動画に加えた。

(※文責: 田中悠人)

## 4.3 評価

前述した聴衆からのアンケートで47名の評価を受けた。47名の内、42名が本学の学生を占め、残りの5名が本学の教員であった。評価の平均は前年度よりも高い数値となった。

発表技術についてコメントは38件の回答をいただいた。参考になるコメントの一例を以下に示す。

- ・全体的にグループ毎で内容がまとめられており分かりやすかった。ただし、ポスターは細すぎて読めない。また、スライドの1ページの量が多いためもっと分けてもいいかもしれない。
- ・原稿は見ないで発表をした方が良いと思います。
- ・文字が少し小さく、分かりづらかった。
- ・声がよく聞こえず、よく分からない部分があった。

表 4.1 中間発表会の評価

評価点	発表技術	発表内容
1	0	0
2	0	0
3	0	0
4	1	0
5	0	1
6	6	3
7	11	4
8	13	17
9	11	13
10	10	9
平均	8.94	8.38

- ・発表資料の前に発表者が立っていたり、扇風機が置いてあったりして、資料が見えなかったです。
- ・スライドの文字が小さく見づらい部分があると感じた。
- ・スライドや説明が詳しくもあり、端的でもあって聞きたい部分をよく聞けたなと思います。
- ・せっかくポスターがあるのだから、うまく連動したらよいのでは？(メインポスター)
- ・動画の使用によりイメージがしやすかった。スライドも必要なことだけ記載されていて見やすくわかりやすかった。
- ・全体的にグループ毎で内容がまとめられており分かりやすかった。ただし、ポスターは細すぎて読めない。また、スライドの1ページの量が多いためもっと分けてもいいかもしれない
- ・スライドに動画を埋め込んだりして、飽きないような工夫を感じられた

全体としてプロトタイプの動画に関しては好評であった。スライドについては賛否両論であった。わかりやすいと意見があれば文字数が多い、この量では理解しきれないなどの率直な感想が多かった。ポスターについては文字が小さいとの意見が多かった。発表では主にスライドを利用し、直接ポスターについて言及せず、より詳細が知りたい人のために用意していたため、近づかないと見えないという齟齬が生まれてしまったと考える。また話し方については声が小さい、原稿を見て発表しているなどの評価が多くみられた。

発表内容についてコメントは39件の回答をいただいた。参考になるコメントの一例を以下に示す。

- ・非常に面白い発表だと思います。AIのどの技術を用いて実装するかについて説明してもらえる

とより良いと思いました。

・プロジェクトの紹介や概要，目標の有意性等を具体的に説明しており，何をやるプロジェクトなのかを深く理解できた。

・AIを用いた創造性の再現とは？目的と作っているアプリの関係は？どうなったらAIを用いた再現性を検討できたと言えるか。

・千と千尋とアリスを比べていた図がイマイチよくわからなかった。帰る帰らない等の説明が欲しかった。

・自動生成のダンジョンなどを冒険するゲームはすでに存在するので，どう差別化するかが気になります。でも顔やキャラクター，テキストまで自動生成するものはなかなかないと思うので完成が楽しみです。

・グループごとにはではなく，プロジェクトで内容ごとに担当を分けて一つの成果物を作るという点が自分のプロジェクトと違って興味深いと思いました。デザインのポスターが個人的に好きです。どんなシナリオになるのか楽しみです。もっと今後の活動を知りたいです。実際にゲームしてみたいです。

・現在の状況と今後の展望がしっかり書かれているので，わかりやすい発表だなと感じました。AIとの関連性をもっとスライドに組み込んでくれるともっとわかりやすいかなとも感じました。

・内容は興味深かったがあえてAIを使う理由がいまいち伝わらなかった。

・深層学習の量に注目したい。期待しています。

・自動生成がAI要素？マップギミックの自動生成なら、ローグライクが参考になる？キャラクターのアイデアだしのための自動生成？実際のゲーム中のキャラクターの自動生成？自動生成でそのアルゴリズムを取りあげた理由は？背景史はなにに使う？エンディング生成との関係は？マルチエンディングはどう作る？（最後はテキスト出力？それはゲームでどう使う？） BGMの自然なつながり？（切り替えはどのようなタイミングを想定？ゆるくつなぐでよいの？）

全体として好意的な意見が多く，今後の成果に期待するような内容が大半であった。また，少量ではあるが今回の発表で伝えきれなかった内容についての疑問や，根本的な課題についての指摘もあり，後期の活動で参考になる意見も見られた。

(※文責: 小林裕汰)

## 第 5 章 前期の反省と後期の方針

### 5.1 システム班

#### 5.1.1 前期反省

前期では、最初に Unity の使い方についてメンバーで確認した後、制作ゲームのプロトタイプである小屋ステージの制作を目標に、メンバーを、ギミック担当・キャラクター担当・マップ担当・UI 担当の 4 つに分け、作業を行った。メンバーの中には、初めて Unity を使う人もいたため、本格的な制作や話し合いを行う前に、Unity のチュートリアルを通して使い方を学ぶことで、今後の制作のイメージづくりを行うことができた。プロトタイプの制作では、主要な要素ごとに担当を分けたことで、それぞれの役割と成果物を明確にできた。作業中は他班と、世界観やゲーム内の要素などについてまめに話し合いや情報共有を行い、制作しようとしているゲームの方向性がぶれてしまわないようにした。また、担当ごとの成果物を 1 つに統合する際に、データの競合が起きてしまったため、成果物の統合に手間取ってしまった。また、それに伴い、メンバー 1 人に作業量が偏ってしまった。そのため、今後、データの受け渡しや、作業量の分担について工夫が必要である。また、前期はプロトタイプの制作をメインに行っていたため、ギミックの自動生成システムの開発に着手することができなかった。

(※文責: 新屋拓海)

#### 5.1.2 後期方針

後期では、主に、プロトタイプ以降のステージ制作と、ギミックの自動生成システムの 2 つの作業を主として行う。プロトタイプ以降のステージ制作では、森・城ステージの制作をメインに行う予定である。また、それと同時に、他班からの成果物や自動生成システムをゲーム内に統合する作業を同時に行う。また、作業を行うに当たり、前期の反省点であった、作業量の偏りとデータ受け渡し時のデータ競合の改善のために、後期では、メンバー 1 人 1 人の作業量とデータの共有方法の見直しをする。ギミックの自動生成システムでは、前期に着手することができなかったギミックのデータ収集および分析を行う。そして最終的に、ゲーム内ギミックを自動生成するシステムの実装を目指す。また、前期に引き続き、他班との話し合いや情報共有をまめに行うことで、最終的な成果物の方向性がぶれないよう努める。

(※文責: 新屋拓海)

### 5.2 視覚班

#### 5.2.1 前期反省

前期の活動から、分かったことは以下の通りだ。まず、CLIP を利用する際に入力するテキストによって出力されるデータが異なることが分かった。そこで、出力をより狙ったものにするため、

入力するテキストの調整が必要になる。どのようなテキスト入力がより効果的なのか模索していかなくてはならない。次に、イメージ共有の重要性が分かった。イメージを共有することでゲーム制作全体の流れの方針を示すことができる。また、前期の活動の問題点もある。それはデータセット作成の方針を変更する必要があるということだ。現在の方針は最低 100 枚のデータセットを作成する予定だ。その内容は年齢は 6～18 歳くらいの見た目の男性キャラクターのイラスト 50 枚、女性キャラクターのイラスト 50 枚だ。しかしデータセット作成は滞っており、入力するデータが 45 枚の今でも十分にスタイル変換が可能だ。そして、今後データセットを増やすことによって出力に変化はあるのか、データセットは各キャラクターでそれぞれ用意しなくていいのかという課題が浮かび上がってきた。これらのことからデータセット作成の方針の変更を検討しなくてはならないという結論に至った。

(※文責: 坂上翠)

## 5.2.2 後期方針

後期のキャラクターの自動生成の方針は以下の通りだ。まず、画像やモデルを生成する際に、ある程度ゲームに適した形で出力できるように入力するテキストを調整する。そして調整の仕方に一定の法則を見つけ出す。次に、キャラクターの顔のデザインの自動生成で用いるデータセットの作成についての方針を検討する。最後に、キャラクターを自動生成した後は、自動生成されたものを実際にゲームに組み込める形まで人の手でブラッシュアップをしていく。素材づくりの方針は以下の通りだ。ゲーム実装を進めるにあたり、UI やキャラクターの立ち絵などの 2D 素材、キャラクターや小物・オブジェクトなどの 3D 素材の制作作業が増えると予想できる。まず、キャラクターや小物・オブジェクトの見た目や雰囲気、立ち絵の表情差分は、シナリオの描写に関わる要素だ。このことから特にシナリオ班との連携が重要である。また、2D 素材/3D 素材共に、ゲームシステムに問題なく組み込める規格に合わせる必要がある。これはシステム班との連携が必要である。このように、適宜他の班と情報共有を行いながら、必要な素材の種類やフォーマットなどを把握管理し、各種素材の制作に取り掛かっていく予定だ。

(※文責: 坂上翠)

## 5.3 音響班

### 5.3.1 前期反省

前期では音響班による AI による自動生成要素について検討し、様々な案をだすことができた。そして早い段階で前期の目標を設定することで作業への取り掛かりも早めることができた。5 月では音響分析についても勉強を行いつつ、具体的な作業の検討を行った。具体的には、前期の目標である状況に応じたシームレスに BGM が変化するインタラクティブミュージックの実現方法について検討した。実現方法の模索段階では、音楽的な手法で解決するのか、技術的手法で解決するのかで難航し、少なくない時間を有した。その後のインタラクティブミュージックの実現のための作業分担がうまくいかず、班員間の負担がバラバラになってしまう問題もあった。また、他班との情報共有の意識が高くなく、進捗報告の際に正しく活動内容について伝えることができていなかった。後に、グループ全体での活動内容について共有するのではなく、個人の活動内容をそれぞれ報告す

ることで、他の班との情報共有の精度を改善することに成功した。終盤の中間発表会の際には、他の班の制作物をまとめたプロトタイプの作成が必要になった。この際に音響班全体ではなく、一人の班員にその作業を一任してしまった。

(※文責: 小林裕汰)

### 5.3.2 後期方針

後期では、前期での反省を踏まえて作業分担を見直し班員の負担を均一化する。それに加えて、インタラクティブミュージックを制作を行い、実際のゲーム上の動作環境でのインタラクティブミュージックの実装を行う。具体的には、作成したインタラクティブミュージックをプレイヤーの行動やシナリオ展開に応じて正しく動作させるシステムを新たに生成する。それ以外にも、前期の段階で検討していた、キャラクターごとの専用 BGM の自動選定にも取り掛かる。具体的には、キャラクターと使用できる BGM にパラメータを設定し、そのパラメータに近いキャラクターと曲を紐づける。昨年も同じことを行っているため、新たなパラメータを追加したりなどして精度を上げていく。そして、最終的に他班の制作物と合わせ、完成品を組み立てることを視野にいれて、早い段階から他班との情報共有を行い、ゲームを完成させる。

(※文責: 小林裕汰)

## 5.4 物語班

### 5.4.1 前期反省

物語班は背景史担当とエンディング担当に分かれて活動しているため、中間発表会やポスター、中間報告書の制作に向けて物語班全体の進捗をまとめるのが大変だった。背景史担当はデータ収集に時間がかかった。これは、制作するゲームに必要な「現実逃避しているキャラクター」の要素や「一般的に完成度が高い作品」の選別について話し合う時間が長くなってしまったことと、データの収集量の目標を 100 としていたためだ。また、データ収集後のフォーマットの作成にも時間がかかった。これは、フォーマットをデータの収集量が増えるごとに繰り返し改良を重ねているためだ。これらの要因から、前期はデータ分析に着手することができなかった。現状としては収集したデータをカテゴライズし、データ分析に向けてフォーマットを調整している段階だ。エンディング担当はデータ分析に着手しているが、収集したデータ量が少ない。これは作品の収集基準が「日常から非日常に遷移する物語」かつ「一般的に完成度が高い作品」であり、基準を満たす作品が少ないためだ。データ数の少なさは抽出したパターンの有意性の程度に直結するため、データ収集の強化が必要だ。

(※文責: 坂本珠凜)

### 5.4.2 後期方針

引き続き、背景史班とマルチエンディング班の二班の形式で活動を行っていき、ストーリー作成については、担当者が進めていく。まず、物語班全体の後期方針としては、完成したメインシナリ

オに背景史とマルチエンディングの自動生成の成果物を統合することを方針とする。メインシナリオは、完成するコトを基本に、マルチエンディングに合わせた物語の執筆を行い、背景史の生成ができしだい、その背景史を匂わせる文章を作成して行くことを後期の具体的な方針として定める。背景史では、シナリオ班全体の方針に向けて自動生成を行うコトが大きな方針となる。そのため、自動生成を行うアルゴリズムを生成することが基本の方針となる。以上を満たすためにまず、前期でも行っていたデータの分析を続行する。現在、分析が途中である作品を分析することで作品数をさらに増やし出力データの安定を計る。また、収集したデータにおいてテキストの部分があるため、この部分の属性を分割を行いベクトルでの表現をしやすようにする。また、既存のデータのデータフォーマットが決定していない部分があるため、その部分の決定を行っていきたいと考えている。同時進行で、データが有意かどうかを統計的な処理をしながら判断し適宜データアセットの増強とデータフォーマットの改良を行う。以上のようなデータから背景史を出力するアルゴリズムを構築していく。マルチエンディングでは、物語のマルチエンディングを自動生成していくことを後期の主な方針とする。現在、出力している相関係数などから得た結果からデータフォーマットの改良を逐次行っていく。また、改良したフォーマットを用いてそのほかにもデータの解析を行う予定である。また、データ数との兼ね合いからデータとして採用する作品の基準を大幅に下げるかどうかを決定するコトも後期方針として盛り込まれる。特にアニメ化している作品を分析データに入れるかどうかは、後期において決定する内容となる。データの分析においては、相関分析の他にもいくつか検討中の段階であり、データ数の兼ね合いによって背景史班と同様の分析手法をめぐりに分析活動の決定とそれを元にした分析活動を行う。以上で得たデータを用いてマルチエンディングを自動で生成するアルゴリズムを完成させるコトを後期の具体的な方針とする。

(※文責: 西村秀幸)

## 5.5 全体

### 5.5.1 前期反省

全体での前期の反省点をあげていく。今期では制作するゲームの大まかな方向性が決まりメンバーをグループに分けた後、各グループにプロジェクトの大まかなグループレベルでのスケジュールを計画してもらった。この作業は各グループの今後の方針をより具体的にすると同時に、別グループの作業計画も理解してもらいスムーズなプロジェクト進行を図るものであった。前者のグループの活動方針の具体化は比較的どの班もうまく行っていたよううかがえた。

教員からの指摘があった点で、個人やグループの週報、議事録の内容が詳細に書かれていないことが挙げられる。その日行ったことをただ書くのではなく、具体的にその作業に至った経緯や活動状況についてより具体的に記載すべきだった。週報や議事録のクオリティをあげることは、プロジェクトの進捗管理と報告書の質を上げることにつながる。

各班、完璧に計画道理の進行が行われているわけではないが、中間発表の段階である程度、計画目標としていた段階まで作業を進められていたよううかがえた。前期では各グループがスムーズに作業が進み、別グループの進捗が遅れることによる作業の中断などは比較的少なかった。しかし、後期になると、実際にゲームとして各班が担当している要素を組み込んでいく作業が増えていき、グループ同士の進捗が大きくプロジェクト全体の進捗に関わってくる。その点などを踏まえると、グループやそのメンバーのタスクがどれだけ残っているのかをメンバー一人一人が理解しておくのが望ましいだろう。そのためにもメンバーの進捗をいつでも確認できるツールや、より状況把

握がしやすい議事録を記録していく必要があるだろう。

(※文責: 穂積佳)

### 5.5.2 後期方針

前期の反省を踏まえ、後期に向けて作業スケジュールの見直しを行いたい。大まかなスケジュールから、月レベルの細かなスケジュールを決めていき、班同士で共有を行う。この作業は1度で終わるものではなく、定期的に行うことで、各班の進捗のズレやスケジュールの荒を見つけ出す作業に繋げていきたい。そのため、Google カレンダーやエクセルシートなどのツールをうまく活用していきたい。

また、前期では連絡や、オンラインでの会話に Slack を利用していたが、Slack の仕様が9月以降に変更される予定が発表されており、今後のプロジェクトにそぐわない可能性があるため、Slack の代替ツールを探す必要がある。

ほかにも後期の活動を行っていくにあたり、様々な問題点、反省点が新たに生まれることが予想されるが、それらをうまく改善して円滑なプロジェクトの運営に努めていく。

(※文責: 穂積佳)

## 第 6 章 成果発表までの開発

### 6.1 システム班

#### 6.1.1 マップ

##### 6.1.1.1 森

小屋マップを攻略すると森マップに進むように初期は想定していた。しかし、製作にかかる時間と工程を改めて見直し、物語班と話し合った。その結果、森マップよりエンディングに関わる城マップを先に製作することに決定した。また、城マップを製作していくにあたり、森マップの製作に着手することが難しいと判断し、森マップの製作は行わず、イベントのみで進めることとした。今後の予定として森マップを新たに作り、イベントと差し替えることで作品の完成度を高めていきたい。

(※文責: 住吉聖也)

##### 6.1.1.2 城

城マップに関して、前期の小屋マップ製作の反省を活かして Unity 内での建物製作ではなく、モデリングソフトの Blender を利用して製作した。そして、城マップは 1 階、2 階、そして地下や謁見の間など複数の部屋が存在する。そこをキャラクターが行き来するため、同一のシーンに部屋を配置するとアセットの関係などで動作が重くなってしまう恐れがある。よって今回作成した城マップは複数のシーンにそれぞれ 1 階や 2 階などを配置し、シーンを分けて管理する様にした。

城マップ製作の手順として、まず部屋の形の仮モデルを作成し、Unity 内に仮配置する。次にキャラクターとの縮尺を確認して問題がなければ、Blender で再度ディティールアップをする。形が完成し次第、順次 Unity に取り込み、Unity 内で家具などのアセットを配置した。また、壁紙や床に関しては小屋マップでは Unity 内でテクスチャを貼り付けていたが、城マップは Blender 内で UV 展開をしてテクスチャを貼り付けることとした。

城マップを製作するにあたり、小屋マップで発生したような壁をすり抜けてしまうバグや、アセットと壁の境目をキャラクターが登ってしまうようなバグは発生しなかった。また、Blender で製作したオブジェクトは.blend 形式で Unity に取り込むことができ、製作したオブジェクトを Unity に合うように変換しなくてよいため、かなり効率よく進めることができた。



図 6.1 城の部屋の内装

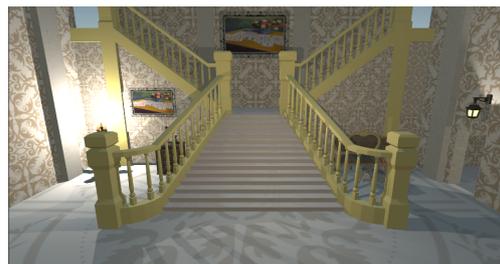


図 6.2 城の一階の内装

### 6.1.1.3 城設計

城の形は西洋の城のような作りを想定した。小屋とは作りを大きく変えて、階段や部屋を増やし豪華な城を作り上げた。また、イベントに必要な牢屋に関しては鉄格子を作り、地下であることをイメージしたため階段を下るような設計とした。

外観に関して、視覚班と話し合い余裕があれば製作してもらう予定であった。アセットで最低限の城オブジェクトが見つかったため、現在はそれを設置している。また、視覚班が新たな城オブジェクトを製作した場合、置き換えることは容易であるため、アセットの城は仮設置としている。地面は前期に引き続き Unity の Terrain を利用した。シナリオの関係上、城へ向かう道とは別に裁判所へ向かう道があるような設計とした。これは、地下牢にさえ行っていれば、城を全て探索しなくてもエンディングを迎えられるようにするためである。外の世界観は落ち着いた印象で、明るさを抑えるようにした。今回のストーリーにとって城というものは、重要なアイテムやキャラクターがいる場所である。また、シナリオ的にも真相が分かる場所でもあるので、それに合わせた外観を製作した。

城内部の設計として、まず城に入ると正面には大きな階段が設けられている。これは西洋の城や宮殿のイメージとして合っているものを調べ、画像を元に設計した。階段の中間程まで登ると、一度踊り場があり、そこから左右両方に階段が枝分かれしているような設計である。1階は部屋というより城のエントランスというイメージが強かったため、階段以外のオブジェクトは造形していない。また、1階に隣接するように料理部屋を作り、1階の階段横の壁にドアを設置して行き来できる様にした。キッチンが地下牢に行くために必要だったため製作した。料理部屋は、階段のある1階の大きさを目立たせるためあまり大きな作りにはせず、キッチンと戸棚があるといったシンプルな作りにした。階段を登ると2階へ遷移するようになっている。2階は、部屋が2部屋ある造りである。特に部屋数の指定はなかったが、造りやすさと探索のしやすさを考慮して2部屋製作した。2階の壁に付いているドアを開けて入ると謁見の間に行けるようにした。謁見の間では特にイベントはなく城を設計する上で、大きな謁見の間を設置すると世界観がディティールアップできると考えた。また、探索してアイテムを集めることが重要であるため、広い部屋などはアイテムの置き場所としても有効活用できると考えた。謁見の間を奥に進むと女王の部屋に行ける様になっている。女王の部屋は、女王に関する重要なアイテムを置く予定だったため、部屋を作った。城の造りとしては以上であり、階段のある一階、料理部屋、地下牢、2階(2部屋)、謁見の間、女王の部屋といった部屋数となった。小屋は3部屋しかなく、またキャラクターと比較しても小さかったため、城はかなり大きく、そして広く作ることが出来た。

城の製作方法として、Blender を利用して部屋の大体の形を造り、そこからキャラクターの大きさと比較して細かく調整していった。前期作った小屋の最初の部屋は円形であったが、円形はドア用の穴を壁に開けるのがかなり難しく手間がかかってしまったので、城製作では、なるべく四角形などの平面を多用した作りとすることにした。また階段は Blender の機能である階段を生成できるツールを利用したのだが、形がシンプルになり過ぎてしまった。しかし、視覚班の Blender に詳しいメンバーに部屋のモデルを渡し、形を調節してイメージに沿った造りに改良することを手伝ってもらった。



図 6.3 城の女王の部屋の内装



図 6.4 城の外装

#### 6.1.1.4 城マップのシーン遷移

前期製作した小屋マップは、全ての部屋が廊下で繋がっているため、マップに関してのシーンは一つであり、遷移はシナリオに移る場合のみであった。後期製作した城マップは、部屋ごとにシーンを分けて作ったため、例えば、「1階から2階に上がる時」や「2階から謁見の間に進む時」などでシーン遷移を行う必要性が生まれた。通常シーン遷移をすると遷移先のシーンが初めからロードされる。そこで、キャラクターの開始位置が固定されてしまう問題が発生した。今回作りたいゲームでは、部屋を行き来することや、一度訪れた部屋に何度も来られることを想定している。例えば、2階は1階とも謁見の間とも繋がっている。そのため、「1階から2階に遷移した時」と「謁見の間から2階に遷移した時」では、キャラクターが出現する位置が違わなければいけない。通常のシーン遷移では、シーンを保存した時の情報がキャラクターの初期位置となるため、階段にキャラクターがいる状態で保存した場合、謁見の間から2階にシーン遷移すると、ドア前ではなく関係のない階段の上にキャラクターが出現することとなる。WebサイトなどでUnityのシーン遷移について調べても、一方向のシーン遷移は多く記事になっているが、行き来するようなシーン遷移の情報はほぼ無いといった結果だった。よって、シーン遷移に関するスクリプトを自分で考え、実装することにした。問題は、キャラクターの出現位置である。つまり、何らかの形でキャラクターの位置を保存したり、指定したりすることができれば、遷移が上手くいくと考えた。

シーン遷移のスクリプトに関して具体的に記述する。まず、遷移に必要なオブジェクトを2つ用意し、これを一つのセットとして考える。一つはキャラクターの位置を指定するためのオブジェクトである。ここでは仮に位置オブジェクトと呼ぶ。もう一つは、スクリプトが付属したシーン遷移をするためのオブジェクトである。ここでは、遷移オブジェクトと呼ぶ。まずゲームが始まると、スクリプトが起動し、キャラクターを初期の位置へ移動させる。キャラクターが移動し、遷移オブジェクトに触れると、指定したシーンへ遷移する。遷移先では、再度スクリプトが起動し、キャラクターをそのシーンの位置オブジェクトの位置へ移動させる。これは、戻ってくる場合でも同じように適用される。これらのオブジェクトはプレイヤーが現在いるシーンから行くことが可能なシーンの数だけ、このオブジェクトセットを配置する。例えば、2階の場合は一階に行くためのオブジェクトセットと、謁見の間に行くためのオブジェクトセットが必要となる。そして、2つ以上オブジェクトセットを置く弊害として、遷移先のシーンに位置オブジェクトが2つあるのでキャラクターの移動先が被ってしまう問題が発生した。この問題には遷移前のシーンを記録し、それに合わせた位置オブジェクトを選択するようにスクリプトを改良することで解決した。このシーン遷移のオブジェクトセットとスクリプトさえあれば、どのようなマップでもシーンの行き来が可能になり、汎用性のあるシステムが完成することができた。

(※文責: 住吉聖也)

## 6.1.2 キャラクター

### 6.1.2.1 3Dモデルの差し替え・取り込み

成果物の統合に当たり、前期で使用していた仮3Dモデルを、視覚班が後期に制作した正規3Dモデルへ差し替える作業と前期に統合できなかった3Dモデルの取り込みを行った。差し替えを行ったのは、アリスと白うさぎのモデル、また、新たにUnity内に取り込んだのは、チェシャ猫と女王のモデルである。これらのモデルについては、前期と同様に、視覚班が3Dキャラクター制作ソフトウェア「VRoid Studio」を用いて作成を行った。また、Unity内での動作も、前期と同じく、Unityの「Animator」機能と、Unityの拡張ライブラリである「UniVRM」を利用した。また、トランプ兵の3Dモデルも仮モデルからの差し替えを予定していたが、モデルの統合作業が発表会直前になってしまった影響で作業を終わらせることができず、成果発表会の時点では、統合を行うことが出来なかった。

(※文責: 新屋拓海)

### 6.1.2.2 瞬き機能の追加

後期では、トランプ兵のモデル以外のアリス・白うさぎ・チェシャ猫・女王の4つのモデルに対して瞬き機能の追加を行った。この機能は、各キャラクターに組み込まれた「AutoBlink」スクリプトによって実装されている。VRoid Studioで作られたモデルには顔の部位や、喜怒哀楽などに対応したパラメータが設定されており、これにより顔の表情を自由に変わることができる。AutoBlinkスクリプトでは、そのうちの目の開閉を管理するパラメータの「fcl\_EYE\_Close」を制御しており、ゲーム内の経過時間が、予め設定した瞬きの発動時間である「blinkTriggerTime」変数よりも大きくなった時に、「CloseEye」関数と「OpenEye」関数を用いて制御し、機能を実装した。また、瞬きを行う際、関数を呼び出すごとに、blinkTriggerTimeの値に5.0から10.5までのランダムな実数を代入することで、一定の感覚ではなく、不規則に瞬きを行うようになり、より自然な瞬きを実現した。

成果発表会時点では、予定していた全てのキャラクターに機能を実装できたが、一部キャラクターで、目をつぶった際に表情が不自然になってしまうことがあった。

(※文責: 新屋拓海)

### 6.1.2.3 主人公制御スクリプトの改良

後期では、前期に引き継ぎ・改良を行った「PlayerContoller」に対し、さらに何点か改良を行った。まず、ジャンプ機能の廃止である。前期の地点で森ステージにアクション要素のあるギミックの作成が予定されており、このことから、主人公のアクションの1つとして「ジャンプ」が実装されていたが、制作工程に置いて、城ステージを優先的に制作し、森ステージを会話イベントのみで進めるということをメンバーとの話し合いで決定したため、ジャンプ機能が必要では無くなり、結果的にジャンプ機能が廃止された。また、それに伴い、Animatorにあったモーションや、プレイヤーの状態を表す列挙型変数であるStatusの中での列挙定数「Stop, Move, Run, Jump, Event, Pause」のうち、機能として無くした「Jump」の定数を消去した。次に、走る動作の修正である。前期の地点で既に走る動作は実装されていたが、走る動作中、マウスの右クリックボタンを押した

まま、入力していた W キーを離すと、その場でキャラクターが走り続けてしまうというバグが残っていた。そのため、後期では、条件文を追加することによって、走る動作のバグを修正することが出来た。さらに、変数の整理も追加で行った。前期の地点では、会話ウィンドウの表示を管理するための「WindowController」の関数を「PlayerController」から参照するために「myWindow」変数が宣言されていた。しかし、統合の際に変数に紐付けされていたオブジェクトの参照が無くなったことによるエラーの多発や、会話イベントの仕組みその物が、会話ウィンドウの表示・非表示による切り替えから、シーンそのものの切り替えによる会話イベントの実装に変わったことから、変数が必要では無くなり、消去した。他にも、機能として必要では無くなったものや、変数の型の変更が必要になったものに対しては、変数の消去や整理を行った。これにより、スクリプトひとつでプレイヤーの動きが完結するようになり、統合を行う際にも、前期よりもスムーズにプレイヤーオブジェクトの受け渡しができるようになった。成果発表会時点では、中間発表時から作成を続けていたものをゲーム内に統合し、実際に動かすことに成功した。一方で、ゲームの進行上問題はないが、プレイヤーの挙動やモーションに軽微なバグが残ってしまった。

(※文責: 新屋拓海)

#### 6.1.2.4 NPC の制御

中間発表後、新たに城ステージに配置する NPC を制御するスクリプトを制作した。もともとは森ステージ及びそこに登場する NPC も制作予定であったが、前述の通り、制作工程に置いて城ステージを優先的に制作し、森ステージを会話イベントのみにすることになったため、森ステージに登場する NPC の制作は行わなかった。成果発表会では、白うさぎ・チェシャ猫の 3 D モデルをゲーム内の小屋や城に配置し、実際に動かすことができた。しかし、女王に関しては、モデルや機能は完成したがゲーム内に配置することはできなかった。また、トランプ兵に関しては、機能は完成したが、6.1.2.1 で記述した通り、3 D モデルの差し替えができなかったため、こちらもゲーム内への配置はできなかった。以下では、小屋・城ステージに登場する NPC の 3D キャラクターモデルの制御及び改良点についての詳細を説明する。

(※文責: 新屋拓海)

#### 6.1.2.5 小屋 NPC

小屋ステージには、中間発表時と同様に、白うさぎが NPC として登場する。中間発表の地点では、会話イベント機能と待機モーション機能の 2 つの機能がついていたが、会話イベント機能の仕組みが、会話ウィンドウの表示・非表示による切り替えからシーンそのものを切り替えになったことで、実装方法を変更することになり、NPC についていた LoadText スクリプトが不要になったため、会話イベント機能を廃止した。成果発表会地点では、待機モーション機能に加え、キャラクターの自動瞬き機能を実装した。待機モーション機能に関しては、前期同様、ゲーム内で一定の時間が経過すると 3 D モデルが伸びをするなどのモーションを行うスクリプトを作成し NPC に紐づけしている。前期の地点で、待機モーションが途中で終わってしまうなどの不具合があったが、後期では、アニメーション遷移を見直し、狙ったタイミングでモーションが発生、また、モーションが途中で終わらないようにすることができた。自動瞬き機能に関しては、6.1.2.2 で記述した通り、設定した時間ごとに瞬きをする機能を追加した。

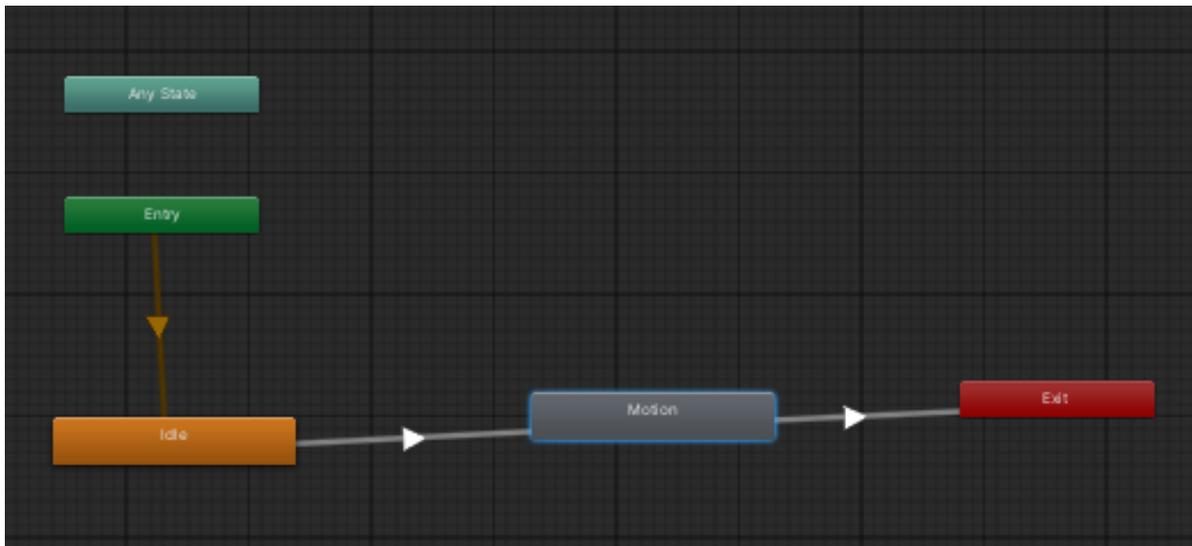


図 6.5 改良後のアニメーション遷移図

(※文責: 新屋拓海)

#### 6.1.2.6 城 NPC

城ステージにはチェシャ猫・女王・トランプ兵が NPC として登場する。チェシャ猫，女王に関しては，白うさぎと同様に，待機モーション機能・自動瞬き機能が付いており，ゲーム内で一定の時間が経過するとモーションや瞬きを行うようになっている。また，トランプ兵に関しては「CardSoldierContoller」スクリプトが組み込まれている。このスクリプトでは，キャラクターをある一定の地点まで移動させることができる「NavMeshAgent」という Unity の機能を利用し，設定した複数の地点を障害物を避けながら巡回する機能を実装している。具体的には，巡回させたい座標を保存しておくための List 型変数「pointList」に座標を何箇所か入れておく。その後，スクリプトを動かすと，「GotoNextPoint」関数により，pointList に入った座標が目的地に設定され，その場所に向かってトランプ兵が動くようになる。目的地に付くと，再度 GotoNextPoint 関数が呼び出され，目的地が再設定される。これを繰り返すことにより，トランプ兵が城内を見回っているような動きを実装することができた。また，このスクリプトには，プレイヤーが近づくことで，動きを一時停止させる機能がついている。この機能は，衝突を感知する「OnCollisionEnter」関数を用いて実装しており，プレイヤーが近づいた時に，トランプ兵の目的地をプレイヤーの座標にすることで，その場にトランプ兵をとどまらせるような挙動を実装することができた。

(※文責: 新屋拓海)

#### 6.1.2.7 今後の展望

今後の展望として，まず，主人公制御システムに関しては，残ったバグの修正を行うと同時に，システムのブラッシュアップ作業を行う。次に，瞬き機能に関しては，一部キャラクターで発生した瞬きの不自然さを解消できるよう，スクリプトの改良や視覚班と協力してモデルの改良を行う。最後に，NPC に関しては，トランプ兵がゲーム内で NPC として機能するよう，3D モデルの統合作業やスクリプトの改良を行っていきたい。また，白うさぎ，チェシャ猫，女王の3キャラクターについては，モーションパターンの追加や表情の追加などを行い，ゲーム内への実装を行って

いく.

(※文責: 新屋拓海)

### 6.1.3 ギミック

ここでは、ギミックについての説明を行う。

(※文責: 田中悠人)

#### 6.1.3.1 城

後期は城エリアの作成を行った。森エリアは時間の関係上作成することができなかった。そのため、森エリアを飛ばして城エリアを作成した。城エリアでは、会話イベントや階移動のシステムを組み込んでいる。以下では、城エリア作成の細かい内容について説明を行う。

(※文責: 田中悠人)

#### 6.1.3.2 エリア作成

システム班では、「UI」「ギミック」「キャラクター」「マップ」という役割で分担している。城エリア作成では「マップ」の担当者が Blender という 3D モデルを作成するアプリケーションを用いて作成を行い、「城の前の庭」「城の一階部分」「城の二階部分」「謁見の間」「女王の屁鵜や」「牢屋」の 6 個の部屋を行き来できるように統合した。部屋を行き来するためのシステムは、マップ制作担当者が作成していたものを組み込んだ。組み込んだ方法として、特定の位置にプレイヤーが来ると特定の位置に移動するというスクリプトを C# で作成されていたものを Unity に組み込んだ。

(※文責: 田中悠人)

#### 6.1.3.3 自動生成

ここでは自動生成について説明する。後期ではギミックのアイデアをもとに、ギミックの出現場所を提案するシステムの構築を行った。

(※文責: 田中悠人)

#### 6.1.3.4 ギミック自動生成の背景

ギミックはゲームの中で最も面白い要素である。しかしそんな中、マップに沿ったギミックの自動生成は過去に事例がない。そこに新規性を感じギミック自動生成システムの構築を目的とし、制作に取り組んだ。

(※文責: 田中悠人)

### 6.1.3.5 ギミックの分類

ギミックには様々な種類がある。その中で、今回の自動生成でどのギミックを作成して用いるのかを、KJ法という方法を用いて考えた。KJ法とは、「断片的な情報・アイデアを効率的に整理するために用いられる手法」である。大きめの模造紙と、付箋を用いてKJ法を行った。KJ法ではまず、付箋に思いつくマップギミックを書き出していった。それはチームメンバー全員で行い各々が思いつかなくなるまで、書き出しを続けていった。全員が書き出し終わったら、付箋に書いてある内容ごとに分類分けをした。分類分けしたもののの中から、どのマップギミックを選択し作成していくのかを決定するために話し合いを重ねた。マップギミックの分類を進めたことによって、「アイテムなどを集める、収集系」「物を移動させる・壁を壊すなどの、アクション系」「スイッチギミック・物の大きさ操作などの、操作系」「落とし穴・動く床などの、トラップ系」「パズル・暗証番号などの、謎解きパズル系」の五つの系統に分けることができた。

(※文責: 田中悠人)

### 6.1.3.6 ギミックの種類決定

ギミックの種類決定にあたり、先ほどあげた5つの系統の中からどれを選択するのが良いかを思考した。それぞれの系統での良い点や悪い点についてそれぞれ出し合い決めていった。その結果、様々なゲームを基に収集できるデータ数が多く、ギミックの要素分解や組み合わせる等のギミックの派生が行いやすいという点が利点として挙げられたため、収集系ギミックの中の特にお使いギミックを自動生成の対象に使用という事に決まった。お使いギミックとは、アイテム等を収集し依頼人に渡す・仕掛けに使う等することによってゲームを進行させる仕組みのことをいう。

お使い系ギミックの例としては、「宝箱から鍵を入手し、鍵を使って扉を開ける。」「町の住民の悩み(物を届ける等)を解決し、ゲームの進行に必要なヒントをもらう。」等といったものがある。このように決定したことで自動生成としてどのようなデータを集めなければならないのかという話し合いも重ねて行った。

(※文責: 田中悠人)

### 6.1.3.7 ギミックデータの収集

ギミックの種類を決め、それに伴ってどのようなデータが必要になってくるのか再び話し合った。データの収集においてどのようなデータが、どのくらい必要になってくるのかを分析の方法を伺うとともにプロジェクトの担当教員と確認を行った。データ数については、最低でも100データほど必要になってくるという風に説明を頂いた。システム班のメンバーで分担して、実際に著名なゲームをプレイしデータをGoogleスプレッドシートにまとめるという風に決定しデータを集めていった。しかし、実際に集めたデータ数は100もいかず、約30データほどしか集まらなかった。

(※文責: 田中悠人)

### 6.1.3.8 ギミックデータの分析

実際に集めたデータを、[いつ][どこで][誰が][誰に][どこで][何を][どうする]という7つの要素に分解した。これにより、データの関係性を見極めることができた。分かった関係性や傾向を、今回

扱うデータにも反映させる事ができるように Python を用いてさらに詳しくデータ分析することになった。Python で行った事としては、集めたデータから傾向のある組み合わせを見極めて、ある程度のパターンを作成した。その組み合わせを決定するために、Python のランダム機能を用いてパターンの作成を行った、拾うアイテムには [拾える場所][どのようにして入手するか][どのアイテムを入手するか] の3つの組合せを作成した。

(※文責: 田中悠人)

#### 6.1.3.9 Unity での実装

先ほど作成したデータを Unity で利用しようとするとうまく扱うことができなかった。そのため、集めたデータを再整形しなければならなかった。そのため、先ほど用いたデータを数値データで変更し、Unity で動かせるようにした。Unity では、テキストデータを収集してアップロードを行い、読み込んだうえで一区切りに置き換え、利用していった。

(※文責: 田中悠人)

#### 6.1.3.10 今後について

今後は、ギミックデータを更に集めていきたい。現在自動生成システムとして行っているのはある程度指定された位置の中からランダムに選択しその位置にランダムに選ばれたアイテムを置くというようになっている。完全な自動生成を目指すためには、三次元の X, Y, Z 軸の位置を自動で決めて配置されるように新しいシステムを作成していきたいと考えている。ゲーム自体の完成度を向上させていくために、自動生成システムの向上を目指して行きたいと思う。

(※文責: 田中悠人)

### 6.1.4 シナリオ読み取りシステム

#### 6.1.4.1 シナリオの読み取りシステムの概要

シナリオ読み取りシステムとは、物語班が人工知能により自動生成を行ったテキストデータをもとに、Unity, C# で作られたスクリプトを使用し、各イベントで起こるイベントシーンの読み取り、シナリオに合った自動生成された BGM の挿入、シナリオに沿ったデータをテキストボックス内へのテキストの表示等を行うシステムである。中間発表までの開発では、物語班が自動生成を行わずに作成したテキストデータを読み込み、テキストボックスに表示するシステムを作成した。全ての会話シーンで適用可能にするためにスクリプトを変更するのではなく、メモ帳等のテキストデータの変更をすることでシナリオの変更をしている。シナリオの読み取りシステムは LoadText.cs というスクリプトで実装した。

(※文責: 本前斗梧)

#### 6.1.4.2 シナリオの読み取りシステムのデータ加工

シナリオの読み取りシステムは渡されたテキストデータをそのまま使用することができない。また、文字がテキストボックスに収まりきらない場合や現在話しているキャラクターが誰か、シナリ

オがどこで終わりなのか等の判定はテキストデータ上の文字列で行っていた。このことから、物語班から素材データとして受け取った文章を加工する作業を行った。加工をする際に当初はスクリプトにシナリオを埋め込もうとしたが、ソースコードの量が多く修正する時に大変であると考えたため、元のテキストデータを加工することでシナリオをそのまま読み取りシステムに適用することにした。また、テキストデータを分割しそのまま文字列として利用することができると分かったため、それを利用して、現在話しているキャラクターと現在話している内容に分け、加工を行ったデータを表示した。

また、データの加工を行う上でキャラクターの表情差分を、テキストデータ上でこのセリフが読まれた場合は、この表情を表示させるというように、セリフと表情を対応させて表示するシステムを開発予定だったが、開発の際、スクリプトを分けて独立的させた方が修正や作業をしやすいため、データ加工にキャラクターの表情差分を組み込まなかった。今後の展望としてはシナリオの読み取りシステムが現在選んだセリフに応じて表情を AI が読み取ってくれる自動システムの開発をすることで、よりゲームの創造性を広げることができると考えられる。

(※文責: 本前斗梧)

#### 6.1.4.3 表示画面への実装

表示画面は大きく分けて2種類に分けられていて、キャラクターを操作する画面と会話やイベントを表示する画面に分けられている。キャラクターを操作する画面ではマップ内に存在するキャラクターに話しかけると会話ウィンドウ用のスクリプトが実行され、画面が切り替わる。会話やイベントを表示する画面では会話ウィンドウの立ち絵や背景、テキストボックス等は視覚班に依頼をして素材を作成して頂き、それを2Dオブジェクトとして表示させた。シナリオデータのテキストの読み取りに関しては、テキストを改行文字でスプリット操作を行い、現在話しているキャラクターの名前とセリフを配列に格納している。プレイヤーに話しかけたことを判定する関数である `OnCollisionEnter` を使い、この関数を用いてプレイヤーへの会話を行ったうえで UI の表示や表示画面の遷移を行う。後期からはこのプログラムを利用して、物語班が自動生成を行ったテキストデータを読み込めるようなプログラムの修正を行っていく。シナリオ読み込みとは別に会話ウィンドウの表示に関して、ただシナリオを読み取るだけではなく、シナリオに応じてキャラクターの立ち絵が変化したり、BGM が変更したりといった、動きがついた会話ウィンドウを作成することによりゲームとしての完成度が上がるという改善案がグループ内の話し合いで上がった。他にもアイテムデータベースやキャラクターの背景史を図鑑のように表示するという話も上がり、アイテムを拾った時の UI やテキストを視覚班や物語班に依頼して作ってもらったが、前期では他の実装に時間を費やしてしまったため、アイテムデータベースや背景史一覧の UI 作成には至らなかった。後期では前期での反省を踏まえて、アイテム管理のメニューや物語班からもらった背景史の統合から背景史一覧画面の実装ができた。UI に関しては前期で視覚班が作成したフレームやテキストバーを流用した。

(※文責: 本前斗梧)

#### 6.1.4.4 メニューの構成と基本機能

メニューはマップ上で開くことができ、アイテム、BGM、背景史欄の三つのタブから構成される。メニューはタブキーから開くことができ、閉じるときは画面内の UI のバックボタンから閉じ

ることができる。メニューのタブを切り替える際に効果音を設定している。これはゲームのクオリティを上げるためゲーム体験の向上につながる。また背景の画像の明瞭度を薄くすることによりメニューのUIをより際だてられるようにしている。

- ・アイテムメニュー

アイテムメニューではキャラクター操作のシーンで拾った背景史のアイテムが表示されるようになっている。まだ未取得のアイテムに関しては別のイメージを配置し、アイテム取得の bool 値を設定し true にすることでアイテム欄に背景史アイテムの表示を行っている。

- ・図鑑メニュー

背景史メニューではキャラクター操作中に取得した背景史アイテムを取得することで表示または閲覧することが可能となっている。取得する前の段階では未取得用の画像を表示している。キャラクターの立ち絵と背景史は視覚班と物語班からの成果物を統合している。

- ・基本設定メニュー

予定では SE と BGM の音量調節とゲーム終了を行えるようにするものだったが、バグ取りの時間の調整を行った結果、この機能はゲーム終了だけ行えるようになっており、SE と BGM の音量調節は未実装となっている。今後の展望としてはこの細かい実装できなかった機能を実装していればいいと考えている。

(※文責: 本前斗梧)

#### 6.1.4.5 表情差分

表情差分システムに関してはストーリーを進めるにつれてキャラクターのセリフに合った表情を表示、切り替えをするシステムを開発した。このシステムはシーンごとに作成を行った。シーンは 1-1, 1-2, 1-3, 2-1, 2-2, 2-3, 2-4, 2-5, 4-1, 4-2 に加えてオープニングとエンディングの計 12 個のシーン分あったので 12 個のスク립トを用意した。

スク립ト名: FaceController

機能: 会話を進行のための左クリックした際に、表示している画像を切り替える。会話やセリフにあったそのキャラクターの表情を表示する。

引数: 配列に格納したテキスト番号

返り値: なし

表情差分に用いた素材としてアリスが `alice_disappointed`, `alice_doubt`, `alice_panic`, `alice_sad`, `alice_shadow`, `alice_sleep`, `alice_smile`, `alice_standard`, `alice_standard_smile`, `alice_surprised_1`, `alice_surprised_2` の計 12 個, チェシャ猫が `cheshirecat_cool`, `cheshirecat_disappointed`, `cheshirecat_smile`, `cheshirecat_standard`, `cheshirecat_standard2`, `cheshirecat_surprised` の計 6 個, 女王が怒, 泣, 笑, 通常, 絶望の計 5 個, 白うさぎが `white_rabbit_cool`, `white_rabbit_disappointed`, `white_rabbit_standard`, `white_rabbit_standard2`, `white_rabbit_surprised` の計 5 個で、これらすべてをシーン上で切り替えを行った。

#### 6.1.4.6 オープニング

オープニングではタイトル画面の次にオープニングに入るようにし、タイトル表示のフェードアウト、フェードイン機能を実装し、会話が少し進んでからゲームの始まりを意識させるようにした。グループの中で、オープニングをどういう風にタイトルを出すかということ話し合いフェードインとフェードアウトを利用することで緊張感を持たせることができ、ゲームの完成度があがるのではないかという意見の基、フェードイン、フェードアウトの実装に至った。フェードアウトとフェードインの中身の部分に関しては、黒の image を Unity 内で配置し、FadeController というスクリプトを Panel に付けて Unity 内の image である Panel を変数の fadeimage に代入することで fadeImage.color.a から Panel の  $\alpha$  値を変更することでフェードイン、フェードアウトの処理を行っている。

FadeController.cs

関数名：StartFadeIn

機能：このメソッドが呼び出されたときにフェードインを開始する。fadeSpeed でフェードの速度を変更する。

引数：なし

FadeController.cs

関数名：StartFadeOut

機能：このメソッドが呼び出されたときにフェードアウトを開始する。fadeSpeed でフェードの速度を変更する。

引数：なし

FadeController.cs

関数名：SetAlpha

機能：このメソッドが呼び出されたときに image の色の初期化を行っている。

引数：なし

(※文責: 本前斗梧)

#### 6.1.4.7 エンディング

今回作成したゲームのエンディングは分岐式となっており、エンディング A、エンディング B、エンディング C、エンディング D、エンディング E、エンディング F の 6 つに分岐される。この分岐は物語班と話し合いで決まったものである。分岐することで本プロジェクトの目標である、ゲームの創造性を広げるといった趣旨を考慮した結果、自動生成と合わせて何度プレイしても飽きないゲームに近づけることができた。分岐方法としてはゲーム内アイテムである、背景史アイテムといわれているアイテムの収集率とシナリオ 4 - 2 時の会話の選択によって決まる。まずゲーム内の背景史アイテムは 4 つ存在している。そして選択肢は UI で上と下の二つの選択肢を提示される。一回目の提示の時に上の選択肢を選ぶとエンディング A、下の選択肢を選んだ後は二回目の提示を

されその時に上の選択肢を選ぶとエンディング B, 二回目の提示時に背景史アイテムが 1 つ以下で下の選択肢を選ぶとエンディング C, 三回目の提示で上の選択肢を選ぶとエンディング D, 三回目の提示で背景史アイテムが 3 以上で下の選択肢を選ぶとエンディング E, 2 つであればエンディング F に進むという場合分けとなっている。尚, この処理は SceneChange というスクリプトにまとめられている。SceneChange のスクリプトは Unity 内での SceneManagement というライブラリですべてのシーン遷移の処理を行っている。以下より操作画面から会話画面への遷移の方式や SceneChange のスクリプトの関数を説明する。

SceneChange.cs

関数名: OnCollisionEnter

機能: オブジェクトに当たったときの関数。イベントというタグがついたオブジェクトに触れているかつ, Unity のヒエラルキー上のオブジェクト名でどのキャラクターに話しかけたかでシーンの遷移先を決定している。

引数: 話しかける対象

SceneChange.cs

関数名: Ending1

機能: シナリオ 4 - 2 で上の選択肢として提示されるボタンの処理。エンディング A, エンディング B, エンディング D への遷移。

引数: なし

SceneChange.cs

関数名: Ending2

機能: ナリオ 4 - 2 で下の選択肢として提示されるボタンの処理。エンディング C, エンディング E, エンディング F への遷移。

引数: なし

(※文責: 本前斗梧)

#### 6.1.4.8 立ち絵制御

会話ウィンドウ内の立ち絵の制御に関しては, LoadText.cs というスクリプトで制御している。立ち絵の切り替えタイミングは物語班の方に委任し, システム班が編集したテキストデータ上で切り替えを行った。最初は切り替えを行っていなかったが, それだとゲームの完成度としてあまりいいものではなかったため, 急遽切り替えを行った。その際に, 効率よくまた使いまわしがしやすいようにテキストデータ上のセリフの前に特定の文字列を入れることによって立ち絵の制御をしている。LoadText.cs では物語班のシナリオのテキストデータを改行ごとに分けし, 分けした文字列を配列に格納し Regex.Match メソッドを使用しテキストデータとその文字列を正規表現で特定する処理を行った。会話が終わったかどうかの判定と全シナリオの立ち絵を切り替えるタイミングや切り替えパターンを合わせると 32 パターンとなった。このパターンの詳細としては, 白うさぎが消える, 白うさぎが出現, チェシャ猫が消える, チェシャ猫が出現, トランプ兵 A が消える, トランプ兵 A が出現, トランプ兵 B が消える, トランプ兵 B が出現, アリスが消える, アリスが出現, 三月うさぎが消える, 三月うさぎが出現, 女王が消える, 女王が出現, 女王が消える, トラン

プ兵 A と B が消える, トランプ兵 A と B が出現, シナリオ 4 - 2 の特別制御 1, シナリオ 4 - 2 の特別制御 2, シナリオ 4 - 2 の特別制御 3, シナリオが終わったかどうか, エンディング選択肢出現, 帽子屋が消える, 帽子屋が出現する, シナリオ 2 - 4 の特別制御 1, シナリオ 2 - 4 の特別制御 2, エンディング A 背景, エンディング B 背景, エンディング C 背景, エンディング D 背景, エンディング E 背景, エンディング F 背景, の計 32 パターンの処理を正規表現として処理している. 反省点として立ち絵の制御と背景 (Background) の制御を一緒のスクリプトに記述してしまっているが, 分けた方が見やすいコーディングにつながるという点ともう一つ, 今回は 32 パターンを正規表現で立ち絵の制御を行ったが, これだと制御するソースコードだけで 400 行以上のコード量となってしまうため改善できる点として挙げられる.

(※文責: 本前斗梧)

### 6.1.5 BGM 切り替えシステム

ここでは、BGM 切り替えシステムについて説明を行う。本システムは、音響班が制作を行った、「インタラクティブミュージックの自動化システム」を使用し、成果物のゲームにマッチするよう調整を行うことで実装を行った。以下では、音響班からのシステム引継ぎとシステムの調整について説明する。

(※文責: 新屋拓海)

#### 6.1.5.1 BGM 切り替えシステムの引継ぎ

中間発表後、音響班が制作を行っていたインタラクティブミュージックの自動化システムの引継ぎを行った。このシステムは、現在流れているフィールドの曲から、最も類似度が高い曲と、最適な曲繋ぎの場所を検索し、シームレスに曲を切り替えるシステムになっている。システム自体は「NextMusic\_Search」と「NextPoint\_Search」という 2 つのスクリプトが Python によって構築されていたが、そのままでは、Unity で制作しているゲーム内に実装することが難しいと予想されたため、作成したプログラムを実行ファイル形式に変換し、それを Unity 内で読み込み、「Cross\_Music」という類似度の高い曲と最適な曲繋ぎの場所を検索するスクリプトを動かすことで実装を行うことを考えた。しかし、システム作成者と実行環境が異なっていることや、成果物のゲームにおいて、実行ファイル形式でプログラムを動かした場合にゲームが重くなってしまうことがあった。このことから、既存の Python のプログラムを Unity でスクリプトを記述する際に使う C# 言語を用いて再構築を行い、Unity に組み込むことで、音響班のシステムの引継ぎを行うことができた。

(※文責: 新屋拓海)

#### 6.1.5.2 ゲーム内への実装

音響班からのシステム引き継いだ後、成果物のゲームにマッチするよう、Cross\_Music スクリプトの機能をいくつか調整した。まずは、シーンの選択方法である。元々は、矢印キーを用いて行い、入力があった際にそのキーに応じて小屋・森・お茶会・城の 4 つのどれかに切り替えるというものであった。調整後では、「Change\_Music」スクリプトを新たに作成し、各シーンの名前が付いたオブ

ジェクトに紐づけを行った。このスクリプトには、衝突を感知する関数である「OncollisionEnter」を使用し、プレイヤーがオブジェクトに触れることで、シーン名を Cross\_Music スクリプトに受け渡すことで、シーンの切り替えを行った。次に、曲の切り替えフラグの立て方である。元々はスペースキーを押すことで曲の切り替えフラグが立ち、曲を切り替えることができた。調整後では、「Change\_Music」内で、オブジェクトから離れたことを検知する関数である「OncollisionExit」を使用し、プレイヤーがオブジェクトから離れることで、曲の切り替えフラグが立ち、曲を切り替えることができた。これらの機能をまとめることで、プレイヤーがステージの床に埋め込まれているオブジェクトを通過することにより、シーンに沿った曲をシームレスに切り替えるシステムをUnity内に実装することができた。

成果発表会の時点では、音響班から引継いだシステムの調整はできたが、引継ぎ・調整に時間がかかってしまい、ゲーム本編に統合することはできなかった。今後の展望としては、ゲーム本編へのシステム統合及び、ゲーム内でシステムが正常に動作するかの確認を行っていく。また、音響班と協力し、システムの改良も並行して行っていく。

(※文責: 新屋拓海)

## 6.1.6 ゲームの統合

視覚班、音響班、物語班、システム班の4班のそれぞれの成果物を一つのゲームとするために、Unityで統合作業を行った。以下では、統合素材の詳細と後期に統合を進めた城エリアについての説明を行う。

(※文責: 田中悠人)

### 6.1.6.1 各種素材

ここでは、システム班以外の3つの班から頂いたゲームの素材について説明する。

#### ・視覚班の素材

視覚班からは、キャラクターの立ち絵やキャラクターモデル、背景イラスト、タイトルロゴなどデザインに関する素材を提供して頂いた。

#### ・音響班の素材

音響班からは、BGMやSE等のゲーム中の音楽に関する素材とBGM切り替えシステムを提供して頂いた。

#### ・物語班の素材

物語班からは、ストーリー文章や個々のキャラクターの背景史の文章、マルチエンディングの分岐条件、アイテムの説明文等提供して頂いた。

(※文責: 田中悠人)

#### 6.1.6.2 城エリアの統合

マップ作成の担当者からもらった Blender のデータを用いて、城エリアの統合を行った。城エリアには階層ごと、部屋ごとにモデルを分けて作成して、その部屋間を行き来するシステムはマップ作成者が制作を行った。そのため各システムを城エリアに組み込む作業を担当した。それに加え、ギミックの作成も行った。アイテムを回収するためのギミックを何種類か作成した。アイテムを入手するために、[扉を開けると中からアイテムが出現する]というギミック、[柱を壊すとその中にアイテムが入っている]というギミック、[牢屋に行くために体のサイズを変更する]というギミックを作成した。ギミックは自分で考えて作成した物を利用した。ギミックを利用するにあたって、[設置する場所][それを利用して何を得れるのか][どうしてそれをしないといけないのか]という三点について考えながら、ギミックの作成を行った。今回は自作のギミックを利用してアイテムを入手するという方針を立ててたが、今後の展望としてはギミック自体を自動で生成できるように新しいシステムとして組み込んでみたいと思っている。

(※文責: 田中悠人)

## 6.2 視覚班

### 6.2.1 2D 自動生成

#### 6.2.1.1 2D 自動生成概要

目標はキャラクターデザインの自動生成である。前述した前期の方向性に基づき、自動生成は2つのアプローチで行った。1つ目はキャラクターデザイン案の画像、2つ目は異形キャラクターの3次元モデルだ。これは2次元イラスト、3次元モデルの自動生成の特徴と、人型かそれ以外かのキャラクターデザインの特徴を考慮した結果である。

(※文責: 坂上翠)

#### 6.2.1.2 キャラクターデザイン案の画像自動生成概要

目標はキャラクターデザイン、つまり三面図の制作である。今回は、「アリス」「チェシャ猫」「女王」「白うさぎ」の4体のキャラクターの三面図を制作した。このために、AIにはたくさんのキャラクターデザイン案を出力する役目を担ってもらった。しかし、AIが出力する全身のキャラクターイラストは顔の情報が曖昧であることが多い。今回は顔のパーツと服装の情報がどちらも重要となっているため、2つに分けてデザイン案を出力させることにした。

(※文責: 坂上翠)

#### 6.2.1.3 使用技術

キャラクターデザイン案自動生成で主に使用した技術は Stable Diffusion と textual inversion である。まず、Stable Diffusion(安定拡散)とは Latent Diffusion Model(潜在拡散モデル)を採用した訓練済みモデルである。Latent Diffusion Model(以下 LDM とする)の中身は以下の通りである。

1. 元画像をピクセル空間からエンコーダで潜在空間に潜在変数(特徴量のベクトル)として出力する。
2. 潜在空間で潜在変数に拡散モデルを適用して、元画像からノイズ画像になるまでの過程を学習させて、ノイズ画像から元画像に復元するまでの過程を予測させて、画像を生成させる。(上記の元画像、ノイズ画像、画像はすべて潜在変数である)
3. CLIP というモデルを使い、上記で生成した画像(潜在変数)と入力テキストとの類似度を評価する。
4. 2と3を繰り返す、類似度がより高くなるように画像(潜在変数)を生成する。
5. 生成した画像(潜在変数)を潜在空間からデコーダでピクセル空間に画像(ピクセル画像)として出力する。

今まではピクセル空間に拡散モデルを適用していた。そうすると、人間が知覚できないレベルの部分も処理するため、膨大な GPU と時間が必要になる。しかし、LDM は潜在空間のみに拡散モデルを適用する。そうすると、人間にとって意味のあるレベルの部分のみを処理する。また、拡散モデルで処理する情報が今までより低次元になる。このようなことから前者よりも少ない GPU と時間で済むようになるのだ。

次に、textual Inversion は35枚の画像を使用し、ファインチューニングを行う方法だ。自分が用意した画像でニューラルネットワークのごく一部を訓練し、新しい画像を生成するときの結果を

使用できる。こうすることで、StableDiffusion に独自のオブジェクトや画風を覚えさせることができる。

(※文責: 坂上翠)

#### 6.2.1.4 ワークフロー

2次元イラストの生成を取り入れたキャラクターデザインができるまでの過程をワークフロー図にした。人間とAIの協調作業の過程を明確化するためだ。

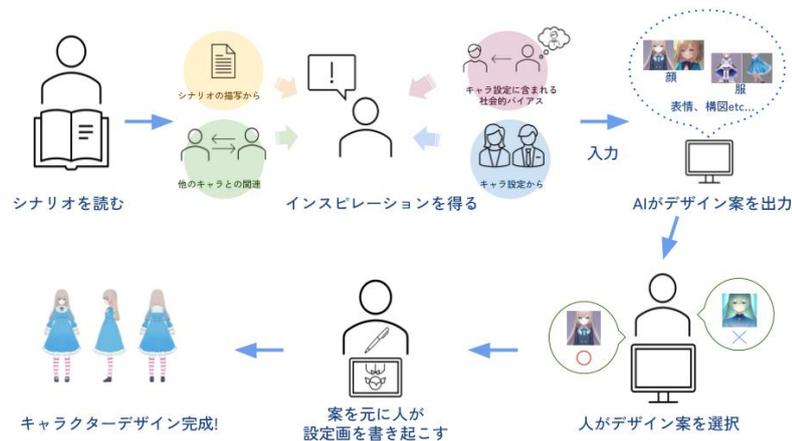


図 6.6 ワークフロー図

(※文責: 坂上翠)

#### 6.2.1.5 シナリオを読む、情報抽出

シナリオやキャラクター設定から、キャラクターの情報を抽出した。抽出した各キャラクターの情報は以下の通りである。

##### ・アリス

抽出したアリスの情報は2つある。

1つ目は以下の通りだ。キャラクター設定「基本的に礼儀正しい」、シナリオのチェシャ猫の台詞「外見で勝手に判断させてもらおうと、キミは礼儀正しそうだったのだけどねえ」から、アリスは実際に礼儀正しそうな表情、行動をしていることが読み取れる。そこから、会話の時に相手をしっかり見ていると解釈し、背筋を伸ばし、目をしっかり開け、程よく口角をあげ微笑んでいるという身体的特徴を導き出した。

2つ目は以下の通りだ。キャラクター設定「自分の意見をはっきりと言う」から、アリスは比較的気が強い性格をしていると解釈した。そこからつり目という身体的特徴を導き出した。

##### ・チェシャ猫

抽出したチェシャ猫の情報は3つある。

1つ目は以下の通りだ。キャラクター設定「いつもにやにやと笑っている」から、チェシャ猫は何かを企んでいそうな笑みを浮かべていると解釈した。そこから、にやけた顔という身体的特徴を導き出した。

2つ目は以下の通りだ。このキャラクターは猫である。現実の猫は特徴的な耳と尻尾を持っていることから、このキャラクターも猫の特徴を持った耳と尻尾、まるい釣り目を持っていると解釈した。

3つ目は以下の通りだ。キャラクター設定「飄々とした態度で」、シナリオのチェシャ猫の台詞「言い忘れていたけど、オレは女王から森にいる人を迷わせる役目を受けてるんだ」から、チェシャ猫は森で活動できるような身軽で動きやすい恰好をしていると解釈した。そこから、スニーカー、短パン、ゆったりした服、ショートカットの髪型という身体的特徴を導き出した。

#### ・白うさぎ

抽出した白うさぎの情報は3つある。

1つ目は以下の通りだ。キャラクター設定「少々臆病で気が弱いところがあり、それによって女王を困らせる」という役割から、白うさぎは臆病で気が弱い行動をとっていることが読み取れる。そこから、人目を避け、人と目を合わせないと解釈した。そして、たれ目で伏し目という身体的特徴を導き出した。

2つ目は以下の通りだ。キャラクター設定「現実では極めて冷静で、ドライな本性を見せる」、シナリオ「白うさぎの表情は不思議の国の時と違って冷静で無感情」から、白うさぎは役割を演じているとき以外は冷めた表情をしていることが読み取れる。そこから、冷めた目をしている身体的特徴を導き出した。

3つ目は以下の通りだ。このキャラクターは白うさぎである。現実の白うさぎは白い体毛と赤い目を持っていることから、白い髪に赤い目をしていると解釈した。

#### ・女王

抽出した女王の情報は3つある。

1つ目は以下の通りだ。キャラクター設定「意味のないことでいつも怒っている。」から、女王はいつも怒っていて、目がつり上がっていると解釈した。そこから、つり目という身体的特徴を導き出した。

2つ目は以下の通りだ。キャラクター設定「不思議の国においては自分が絶対だという考えを持っており」、シナリオのトランプ兵 A の台詞「そんなふうになんか女王様を疑うお前が裁判にかけられるべきだ！」やトランプ兵 B の台詞「そ、そうさそうさ！女王様を疑う者の首は刎ねるべきだ！」、トランプ兵たちの台詞「…女王陛下の仰せのままに」から、女王はトランプ兵を従えており、それを顕示するような性格であると解釈した。そこから、トランプに使われる配色（黒・赤基調）や大きな王冠、豪華なドレス、ハイヒールなどの身体的特徴を導き出した。

3つ目は以下の通りだ。キャラクター設定「外見は、アリスよりも幼い少女。」、シナリオのアリスの台詞「え、この子が女王…？」から、女王は幼さが一つの特徴的な要素であり、子供っぽい髪型をしていると解釈した。そこから、髪は短めであるか、高い位置で結んであるという身体的特徴を導き出した。

(※文責: 坂上翠)

### 6.2.1.6 デザイン案を出力

ここでは、3.2.2.2.1 でシナリオやキャラクター設定から抽出した情報を元に、AI にキャラクターデザイン案を出力してもらう。使用した AI モデルは、「Stable Diffusion TrinArt

Characters model v1」(以下「TrinArt v1」)という、アニメ・マンガ的なイラスト表現に特化したモデルだ。これは、3.2.2.2.1.1 で説明した Stable Diffusion という AI に、1920 万枚の画像を追加で学習させたモデルであり、2022 年 10 月に公開され、無料で利用できるようになった。” TrinArt V1 を AUTOMATIC1111 + Google Colab で手早く動かす”(URL:https://note.com/uchidama/n/n34bb62cf0bf6)を参考し、この「TrinArt v1」モデルを Google Colab 環境で実行するという形で利用した。



図 6.7 実際に TrinArt V1 を Google Colab で動かす画面

実行画面は、プロンプト入力欄・ネガティブプロンプト入力欄・生成画像欄・各種設定欄等から構成される。各種設定欄では、自動生成する画像のサイズや枚数などを設定することができる。

プロンプトの基本的な入力方式としては、英単語をカンマ記号で区切って羅列する。重みづけに関しては、プロンプトの各要素を () やなどの記号で囲むことで強調することができる。また、基本的に、AI はステップごとに左から順番にプロンプトを読んで適用しようとする。従って左の方にある要素ほど最初の段階(トピックや画風が決定づけられる段階)で優先されやすく、右の方にある要素ほど後から画像をその方向に「引っ張る」役割を果たす。よって、強調したいトピックがある場合、プロンプト中に何度も割り込ませるようにして重複して書くという方法も有効である。さらに、ネガティブプロンプトを利用して、極力除外する要素を記述することが可能である。ネガ

タイププロンプトの記述に関しては、絵の細やかさクオリティの高さを求めるにあたって、主に元素法典（中国語圏コミュニティによって制作された、画像生成 AI に対する入力プロンプト・ネガティブプロンプト等の設定がまとめられた PDF ファイル）を参照した。加えて、各キャラクターによってデザインの観点から適宜記述した。例えば、チェシャ猫の入力プロンプトに cat\_ears を加えたところ、猫耳だけに収まらずもはや猫のような見た目が出力されたため、それを防ぐためにネガティブプロンプトに cats を加えた。

入力プロンプトを決める際は、「絵の構図」「画風や品質」「抽出したキャラクターの情報」という三つの主な観点から複数の単語を選出し、様々な組み合わせを試すという方針で設定した。以下に、それぞれの観点から採用した入力プロンプトを記す。

「絵の構図」デザイン案の自動生成にあたって、今回は「顔」と「服装」に着目し、それらを別の画像で出力するという手法を取った。「顔」については、正面から見た一人のキャラクターのバストアップの構図が欲しいという点から、「standing, portrait, perfect symmetrical pretty face, frontispiece, 1girl, 1boy」等を採用した。「服装」については、正面から見た一人のキャラクターの体全体が映った構図が欲しいという点から、「standing, stand on the ground, Full body, costume design, character design, 1girl, 1boy」等を採用した。

「画風や品質」クオリティの高い作品が望ましいという点から、「illustration, masterpiece, backlighting, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper」等を採用した。また、現代のイラスト風の出力をしたいという点から、「virtual youtuber, GenshinImpact, kawaii, 2022 Artstyle preset」等を採用した。

#### 「抽出したキャラクターの情報」

##### ・アリス

背筋を伸ばし、目をしっかり開け、程よく口角をあげ微笑んでいるという身体的特徴から「smiling face」、つり目という身体的特徴から「slant or almond-shaped eyes」等を採用した。

##### ・チェシャ猫

にやけた顔という身体的特徴から「smirking face」、耳と尻尾、まるい釣り目を持っているという身体的特徴から「cat\_ears, cat\_tail, big round suspended eyes」、スニーカー、短パン、ゆったりした服、ショートカットの髪型という身体的特徴から「sneaker boots, short pants, loose-fitting clothing, loose-fitting jacket」等を採用した。

##### ・白うさぎ

たれ目で伏し目という身体的特徴から「drooping eyes, downcast look eyes, sad face」、冷めた目をしている身体的特徴から「dark eyes」、白い髪に赤い目をしているから「white hair, red eyes」等を採用した。

##### ・女王

つり目という身体的特徴から「big-glaring-upturned-suspended-eyes, irritated look」、トランプに使われる配色（黒・赤基調）や大きな王冠、豪華なドレス、ハイヒールなどの身体的特徴から

「black\_hair, red, big queen's crown, red dress, beautiful detailed dress, gorgeous dress, high - heeled shoes」, 髪は短めであるか, 高い位置で結んであるという身体的特徴から「short\_hair, high twin-tail」等を採用した。

また今回は, 世界観のモチーフとして「不思議の国のアリス」という作品を設定していることから「Alice in Wonderland」を各キャラクターに採用した。加えて各キャラクターの年齢設定によって「6 to 8 years old, 8 to 10 years old, 15 years old, loli, shota」等を採用した。

以下に, 後述の 3.2.2.2.3 でデザイン案として選定された各画像を出力する際に用いた, 実際のプロンプトを記す。

- ・アリス/ 顔/ 1 枚目



図 6.8 生成された画像

入力プロンプト↓

standing,(best quality),((illustration)),(portrait),masterpiece, 1girl, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, beautiful detailed eyes, extremely detailed face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed girl, cute girl, loli, child girl, Alice, Alice in Wonderland, 8 to 10 years old, smiling face, (slant or almond-shaped eyes), big round suspended eyes, (2022 Artstyle preset), Alice, Alice in Wonderland, bold line drawing

ネガティブプロンプト↓

nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, big nose, retro style, 1980s, 1990s, 2000s, 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019, hat, monochrome picture

\* ( ) やは強調表現として使われている。(以下のプロンプトについても同様)

- ・アリス/ 服/ 1 枚目

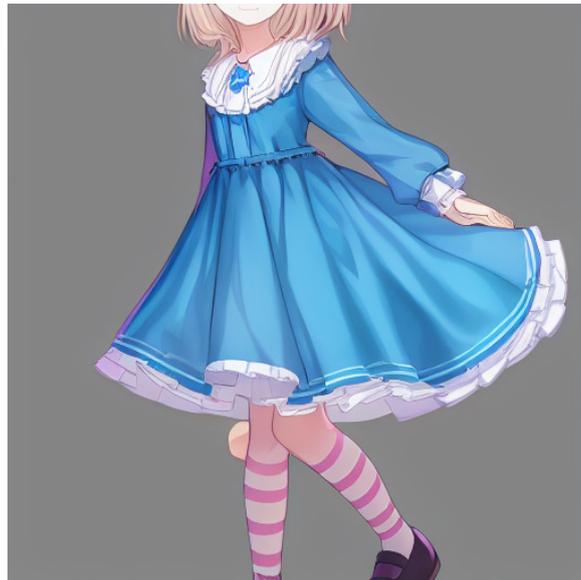


図 6.9 生成された画像

入力プロンプト↓

standing, stand on the ground, Full body, costume design, character design, (best quality),((illustration)),(portrait),masterpiece, 1girl, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed girl, cute girl, loli, child girl, Alice, Alice in Wonderland, 8 to 10 years old, smiling face, (slant or almond-shaped eyes), big round suspended eyes, (2022 Artstyle preset), shoes, Alice, Alice in Wonderland, bold line drawing

ネガティブプロンプト↓

nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, big nose, retro style, 1980s, 1990s, 2000s, 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019, monochrome picture

・チェシャ猫/ 顔/ 1 枚目



図 6.10 生成された画像

入力プロンプト↓

standing, (best quality), ((illustration)), (portrait),masterpiece,1boy,Amazing,virtual youtuber,GenshinImpact,perfect symmetrical pretty face,beautiful detailed eyes,finely detail,Depth of field,extremely detailed CG,original, extremely detailed wallpaper,bishonen,cool boy,beautiful detailed boy,child boy,shota,purple\_hair,cat\_ears,cheshire cat in Alice in Wonderland,short\_hair,8 to 10 years old, masterpiece,best quality,1boy,Amazing,beautiful detailed eyes,finely detail,Depth of field,extremely detailed CG,original, extremely detailed wallpaper,child boy,shota,purple\_hair,(smirking face),big round suspended eyes,cat\_ears,(2022 Artstyle preset)

ネガティブプロンプト↓

nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry,cats,aqua eyes,red eyes

・チェシャ猫/ 服/ 1 枚目



図 6.11 生成された画像

入力プロンプト↓

standing, stand on the ground, Full body, costume design, character design, (best quality),((illustration)),(portrait),masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed boy, bishonen,cool boy, child boy,shota, sneaker boots,short pants,detailed fine clothes,loose-fitting clothing,purple\_hair,cat\_ears,cat\_tail,cheshire cat in Alice in Wonderland,short\_hair,8 to 10 years old, (smirking face), big round suspended eyes, }, (2022 Artstyle preset), sneaker boots,short pants, bold line drawing

ネガティブプロンプト↓

nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry,cats

・チェシャ猫/ 服/ 2 枚目

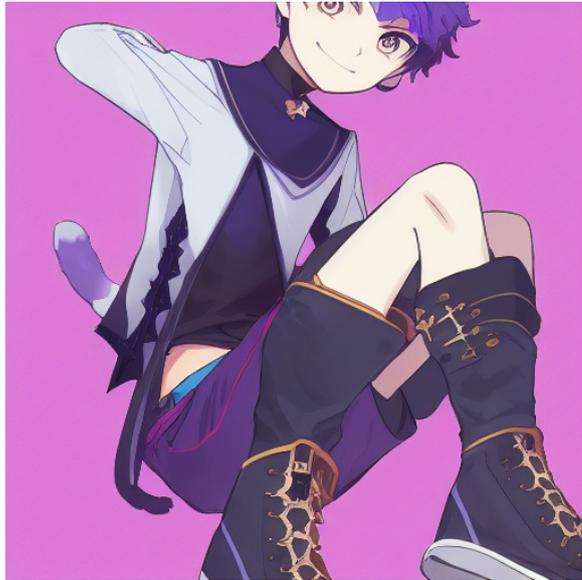


図 6.12 生成された画像

入力プロンプト↓

standing, stand on the ground, Full body, costume design, character design, (best quality),((illustration)),(portrait),masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed boy, bishonen,cool boy, child boy,shota, sneaker boots,short pants,detailed fine clothes,loose-fitting jacket,purple\_hair,cat\_ears,cat\_tail,cheshire cat in Alice in Wonderland,short\_hair,8 to 10 years old, (smirking face), big round suspended eyes, }, (2022 Artstyle preset), sneaker boots,short pants, bold line drawing

ネガティブプロンプト↓

(チェシャ猫/ 服/ 1枚目のものと同一)

・チェシャ猫/ 服/ 3枚目・4枚目

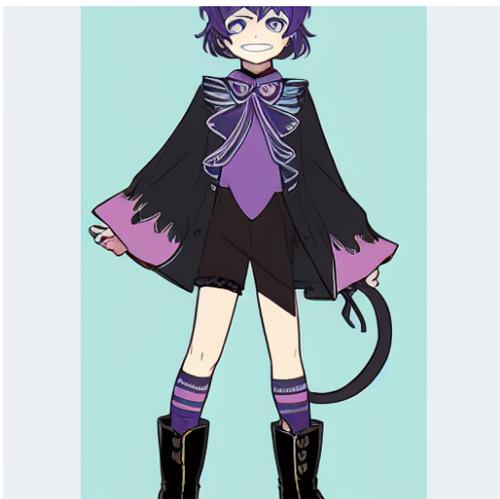


図 6.13 生成された画像



図 6.14 生成された画像

入力プロンプト↓

standing, stand on the ground, Full body, costume design, character design, (best qual-

ity),(illustration),(portrait),masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed boy, bishonen,cool boy, child boy,shota, sneaker boots,short pants,detailed fine clothes,lightweight clothing,purple\_hair,cat\_ears,cat\_tail,cheshire cat in Alice in Wonderland,short\_hair,8 to 10 years old, (smirking face), big round suspended eyes, }, (2022 Artstyle preset), sneaker boots,short pants, bold line drawing

ネガティブプロンプト↓

(チェシャ猫/ 服/ 1枚目のものと同一)

・チェシャ猫/ 服/ 5枚目



図 6.15 生成された画像

入力プロンプト↓

standing, stand on the ground, Full body, costume design, character design, (best quality),(illustration),(portrait),masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed boy, bishonen,cool boy, child boy,shota, shoes,sneaker boots,short pants,detailed fine clothes,lightweight clothing,purple\_hair,cat\_ears,cat\_tail,cheshire cat in Alice in Wonderland,short\_hair,8 to 10 years old, (smirking face), big round suspended eyes, (2022 Artstyle preset), shoes,sneaker boots,short pants, lightweight clothing,bold line drawing

ネガティブプロンプト↓

(チェシャ猫/ 服/ 1枚目のものと同一)

・白うさぎ/ 顔/ 1枚目

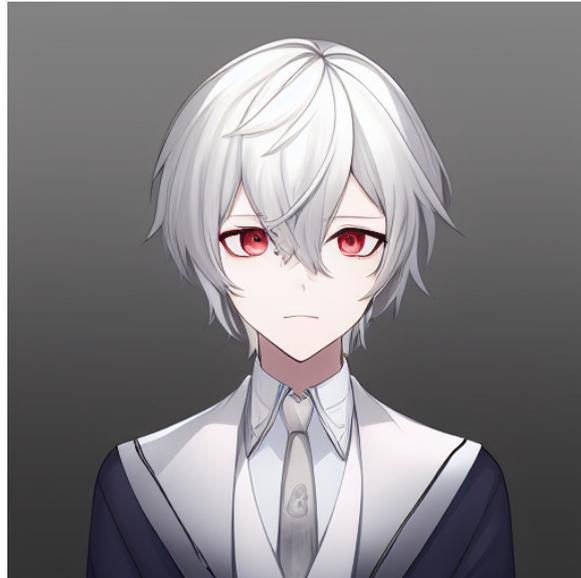


図 6.16 生成された画像

入力プロンプト↓

standing,(best quality),((illustration)),(portrait),masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face,beautiful detailed eyes, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed boy, ikemen, cool boy, rabbit\_ears, The White Rabbit in Alice in Wonderland, 15 years old,red eyes, white hair, sad face, drooping eyes, big round suspended eyes, downcast look eyes, masterpiece, best quality, 1boy, Amazing, beautiful detailed eyes, finely detail, Depth of field, extremely detailed CG, original, extremely detailed wallpaper, rabbit\_ears, (2022 Artstyle preset), dark eyes

ネガティブプロンプト↓

nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, wide-eyed eyes, slant or almond-shaped eyes, rabbits, strong man, cats, big nose, retro style, 1980s, 1990s, 2000s, 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019, girl

・白うさぎ/ 顔/ 2 枚目・3 枚目



図 6.17 生成された画像



図 6.18 生成された画像

入力プロンプト↓

standing,(best quality),((illustration)),(portrait),masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face,beautiful detailed eyes, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed boy, ikemen, cool boy, rabbit\_ears, The White Rabbit in Alice in Wonderland, 15 years old,red eyes, white hair, sad face, drooping eyes, big round suspended eyes, downcast look eyes, masterpiece, best quality, 1boy, Amazing, beautiful detailed eyes, finely detail, Depth of field, extremely detailed CG, original, extremely detailed wallpaper, rabbit\_ears, (2022 Artstyle preset)

ネガティブプロンプト↓

nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, wide-eyed eyes, slant or almond-shaped eyes, rabbits, strong man, cats, big nose, retro style, 1980s, 1990s, 2000s, 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019, girl

・白うさぎ/ 服/ 1 枚目

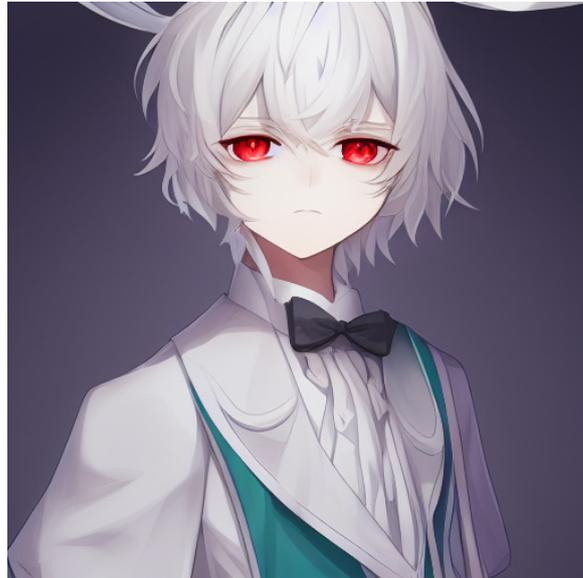


図 6.19 生成された画像

入力プロンプト↓

standing,(best quality),((illustration)),(portrait),masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face,beautiful detailed eyes, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed boy, ikemen, cool boy, rabbit\_ears, The White Rabbit in Alice in Wonderland, 15 years old,red eyes, white hair, sad face, drooping eyes, big round suspended eyes, downcast look eyes, rabbit\_ears, (2022 Artstyle preset), dark eyes

ネガティブプロンプト↓

nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, wide-eyed eyes, slant or almond-shaped eyes, rabbits, strong man, cats, big nose, retro style, 1980s, 1990s, 2000s, 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019, girl

・白うさぎ/ 服/ 2枚目



図 6.20 生成された画像

入力プロンプト↓

standing, stand on the ground, Full body, costume design, character design, (best quality),((illustration)),(portrait),masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, beautiful detailed eyes, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, beautiful detailed boy, ikemen, cool boy, rabbit\_ears, shoes, The White Rabbit in Alice in Wonderland, 15 years old,red eyes, white hair, sad face, drooping eyes, big round suspended eyes, downcast look eyes, rabbit\_ears, (2022 Artstyle preset), dark eyes

ネガティブプロンプト↓

(白うさぎ/ 服/ 1枚目のものと同一)

・白うさぎ/ 服/ 3枚目

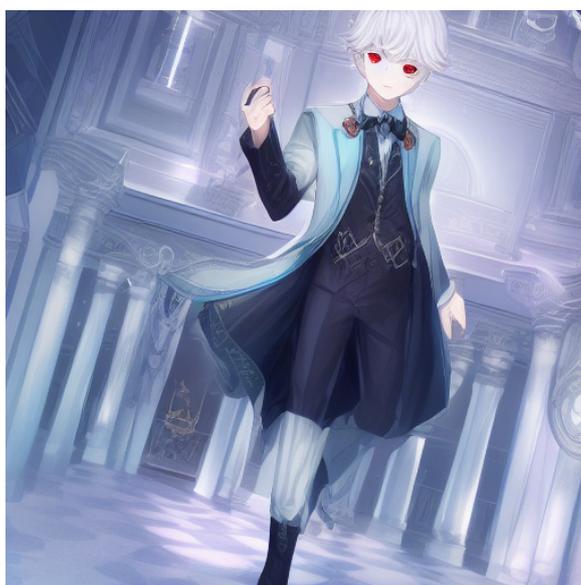


図 6.21 生成された画像

入力プロンプト↓

standing, stand on the ground, full body, costume design, character design, floor, the ground, (best quality), ((illustration)), (portrait), masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, shoes, beautiful detailed boy, ikemen, cool boy, rabbit\_ears, The White Rabbit, Alice in Wonderland, 15 years old, red eyes, white hair, poor face, drooping eyes, big round suspended eyes, downcast look eyes, shoes, (best quality), ((illustration)), (portrait), masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, rabbit\_ears, (2022 Artstyle preset), shoes, The White Rabbit, Alice in Wonderland, bold line drawing, Single-color background, simple background

ネガティブプロンプト↓

girl, nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, wide-eyed eyes, slant or almond-shaped eyes, rabbits, strong man, cats, big nose, retro style, 1980s, 1990s, 2000s, 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019, girl

・白うさぎ/ 服/ 4 枚目



図 6.22 生成された画像

入力プロンプト↓

standing, stand on the ground, full body, costume design, character design, floor, the ground, (best quality), ((illustration)), (portrait), masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, shoes, beautiful detailed boy, ikemen, cool boy, rabbit\_ears,

The White Rabbit in Alice in Wonderland, 15 years old, red eyes, white hair, sad face, drooping eyes, big round suspended eyes, downcast look eyes, shoes, (best quality), ((illustration)), (portrait), masterpiece, 1boy, Amazing, virtual youtuber, GenshinImpact, perfect symmetrical pretty face, finely detail, Depth of field, extremely detailed CG, extremely detailed wallpaper, rabbit\_ears, (2022 Artstyle preset), shoes, The White Rabbit in Alice in Wonderland, bold line drawing, Single-color background, simple background

ネガティブプロンプト↓

(白うさぎ/ 服/ 3枚目のものと同一)

・女王/ 顔/ 1枚目



図 6.23 生成された画像

入力プロンプト↓

standing, (best quality), ((illustration)), (portrait),masterpiece,1girl,Amazing,virtual youtuber,GenshinImpact,perfect symmetrical pretty face,beautiful detailed eyes,finely detail,Depth of field,extremely detailed CG,original, extremely detailed wallpaper,cute girl,beautiful detailed girl,child girl,loli,black\_hair,big queen's crown,queen of hearts in Alice in Wonderland,short\_hair,6 to 8 years old, masterpiece,best quality,1girl,Amazing,beautiful detailed eyes,finely detail,Depth of field,extremely detailed CG,original, extremely detailed wallpaper,child girl,loli,black\_hair,(irritated look),big\_glaring\_upturned\_suspended\_eyes,big queen's crown,(2022 Artstyle preset)

ネガティブプロンプト↓

nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry, nsfw, lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry,aqua eyes

・女王/ 服/ 1 枚目



図 6.24 生成された画像

入力プロンプト↓

standing,whole body,character design, (best quality), ((illustration)), (portrait),masterpiece,1girl, Amazing,virtual youtuber,GenshinImpact,perfect symmetrical pretty face,beautiful detailed eyes,finely detail,Depth of field,extremely detailed CG,original, extremely detailed wallpaper,cute girl,beautiful detailed girl,child girl,loli,black\_hair,big queen's crown,red,beautiful detailed dress,gorgeous dresses,high\_heeled shoes,queen of hearts in Alice in Wonderland,queen of hearts,trump card pattern,short\_hair,6 to 8 years old,standing, whole body,masterpiece,best quality,1girl,Amazing,beautiful detailed eyes,finely detail,Depth of field,extremely detailed CG,original, extremely detailed wallpaper,child girl,loli,black\_hair,(irritated look), big\_glaring\_upturned\_suspended\_eyes,big queen's crown,red,beautiful detailed dress, gorgeous dress,high\_heeled shoes,queen of hearts,standing,whole body,character design,(2022 Artstyle preset)

ネガティブプロンプト↓

(女王/ 顔/ 1 枚目のものと同一)

・女王/ 服/ 2 枚目



図 6.25 生成された画像

入力プロンプト↓

standing,whole body,character design, (best quality), ((illustration)), (portrait),masterpiece,1girl, Amazing,virtual youtuber,GenshinImpact,perfect symmetrical pretty face,beautiful detailed eyes,finely detail,Depth of field,extremely detailed CG,original, extremely detailed wallpaper,cute girl,beautiful detailed girl,child girl,loli,black\_hair,big queen's crown,red dress,beautiful detailed dress,gorgeous dresses,high-heeled shoes,queen of hearts in Alice in Wonderland,queen of hearts,trump card pattern,short\_hair,6 to 8 years old,standing, whole body,masterpiece,best quality,1girl,Amazing,beautiful detailed eyes,finely detail,Depth of field,extremely detailed CG,original, extremely detailed wallpaper,child girl,loli,black\_hair,(irritated look), big glaring\_upturned\_suspended\_eyes,big queen's crown,red dress,beautiful detailed dress, gorgeous dress,high - heeled shoes,queen of hearts,standing,whole body,character design,(2022 Artstyle preset)

ネガティブプロンプト↓

(女王/ 顔/ 1枚目のものと同一)

(※文責: 畑井梨里衣)

### 6.2.1.7 デザイン案を選定

AI が出力した多数のデザイン案を選定するにあたり、まず以下の 2 点を条件とした。1 つ目は人体の構造が著しく破綻していないという点だ。足が長すぎたり、肩から足が生えていたりするものなどが却下された例だ。2 つ目は複数のキャラクター間でデザインの被りを少なくするという点だ。例えば、白うさぎは髪の色が「白」と予め決まっているので、アリスでは白系の髪色を避けた。この条件で前述の「デザイン案を出力」で紹介した画像を選出した。

(※文責: 坂上翠)

### 6.2.1.8 デザイン完成

選定したデザイン案を元に、「アリス」「チェシャ猫」「女王」「白うさぎ」の4キャラクターの三面図を制作した。使用ソフトはMedibangPaintだ。三面図を制作する際、以下の2点について考慮した。1つ目はAIが出力した画像の不鮮明な点を補完するということだ。AIが出力した画像には整合性がとれていない箇所や曖昧になっている箇所がある。例えば、一見問題ないように見えるフリルでも、よく見ると現実には再現不可能なものになっていることがある。そのような点を人間の手で補完、修正する必要がある。2つ目はできるだけゲームに実装しやすい形に落とし込むということだ。1つ目の点を考慮したとしても、限られた期間で完成させるには細かすぎるデザインが残っていることがある。このような点を人間の手で簡略化し、今持っている時間と技術に見合うデザインに修正した。

- ・アリス

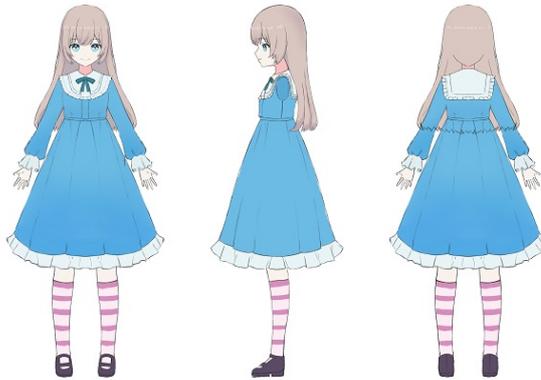


図 6.26 アリスの三面図

- ・チェシャ猫



図 6.27 チェシヤ猫の三面図

・女王



図 6.28 女王の三面図

・白うさぎ



図 6.29 白うさぎの三面図

(※文責: 坂上翠)

## 6.2.2 3D 自動生成

### 6.2.2.1 3次元モデル自動生成概要

後期は、前期の活動を踏まえて使用技術の検討を行った。前期では、トランプ兵、三月うさぎ、帽子屋の3キャラクターを生成する予定であったが、後期に入りシステム班を中心にスケジュールの見直しを行った結果、一部の予定していたゲームステージの実装を変更した。その結果、森ステージに使用する予定だった三月ウサギと帽子屋の3Dモデル素材が必要なくなり、トランプ兵のみの生成を行うことになった。

(※文責: 穂積佳)

### 6.2.2.2 使用技術

CLIP-Mesh はテキスト入力から、ゼロショットで3Dモデルを生成する技術である。テキストを入力すると、テクスチャマップ、法線マップ、メッシュデータを生成、3Dモデルをobjで出力する。CLIP-Meshは、メッシュ、テクスチャマップ、法線マップの3種類のデータを生成します。このデータを多角度から複数枚レンダリングしCLIPが評価、入力テキストとの類似度をより高くなるようにデータを変化させていきます。CLIPは4億組の画像と画像を説明する自由テキストのペアデータセットを自然言語教師データとして学習させており、自由なテキストと画像の類似度をゼロショット転移で出力できるモデルです。CLIP-Meshを使用する利点は2つあり、1つ目の利点は生成時間が短いことである。前期に検討したtext2Meshや、AvatarCLIPと比較して数時間かかった生成がGoogle Colaboratoryでの実行で15分から20分程度でCLIP-Meshは生成が完了する。2つ目の利点は、生成されるモデルデータがゲームの素材として流用しやすいことにある。モデルデータは、メッシュデータのOBJファイル、テクスチャマップと法線マップのPNGファイル、そしてメッシュとマップデータを紐づけるMTLファイルが生成される。このデータセットは、ゲームのモデルデータとしてそのまま流用することができる。一方、人体、動物などの凹凸が激しいモデルや、槍やトランプなど、極端に薄い形状や細長い形状のものは生成の制度が低いことが、何度かの生成で分かった。以上のことから、CLIP-meshによるキャラクター(トラン

プ兵) モデルの生成は一度の生成で完成させることが難しいと判断し、キャラクターをパーツに分け (トランプの胴体や顔など), それを組み合わせることで生成するようにした。

(※文責: 穂積佳)

### 6.2.2.3 各パーツの生成

トランプ兵のパーツは, 顔, トランプの体, 手, 靴を CLIP-Mesh で生成した。上記で述べたように, 手のような複雑な形のものや, トランプのような極端に細長い形状のものは CLIP-Mesh で生成が難しかった。したがって, 手やトランプは通常 CLIP-Mesh の機能を一部変更して生成した。通常, CLIP-Mesh が 3 D モデルを生成する際は, 球体のメッシュデータの頂点を移動させ形を生成していくが, オプションを設定することで, 前もって制作したモデルを元にスタイル変換やメッシュ変形させることが出来る。これを利用し, 本来 CLIP-Mesh で生成しづらい形状の 3 D モデルを生成した。

(※文責: 穂積佳)

### 6.2.2.4 パーツを組み合わせる (CLIP 利用)

パーツを組み合わせる作業の自動化を CLIP を利用して行う検討をした。想定する仕様は, 自由テキストと複数の 3 D モデルデータを入力, テキストに応じて 3 D モデルを配置するというものである。例えばシルクハットのモデルデータと石膏像の顔のモデルデータと「シルクハットをかぶった男」というテキストを入力すると, 顔のモデルの上にシルクハットを移動させ OBJ ファイルとしてエクスポートする。配置した複数のオブジェクトを移動させてその都度レンダリング画像を CLIP で評価, 類似度が高くなるようにさらに移動をさせて任意のオブジェクト配置を自動で行う。2つの 3 D モデルを 1 次元移動のみで制御するまでは実装できたが, それ以上の実装はゲームの統合作業に間に合わないと判断し断念した。結果的には, blender を利用しパーツを組み合わせてトランプ兵を完成させた。

(※文責: 穂積佳)

### 6.2.2.5 リグ付け

生成した 3 D モデルアニメーションなどの動きを付けるために, リグ付け作業を行った。リグ付けは, ボーンの配置, ボーンとモデルの関連付け (スキニング), ウェイト調整, などの作業を行うが, いくつかのアプリケーションではこの過程を自動的に行うことが出来る。アニメーション制作が出来る web サービスの Mixamo には人型のモデルに, 手, 腰, 頭, など数点の位置を指定するだけで人型のリグ付けが自動的に行われる機能がある。この機能を使用してトランプ兵のモデルのリグ付けを行おうとしたが, 正常にリグ付けが完了しなかった。これは, トランプ兵の関節ごとのパーツが分離していることや, 人型からやや離れた形状をしていることが考えられる。こちらも結果的には blender を利用して手動でリグ付けを行うことになった。

(※文責: 穂積佳)

## 6.2.3 アセット制作

### 6.2.3.1 キャラクター立ち絵

三面図を元に「アリス」「チェシャ猫」「女王」「白うさぎ」を、自動生成した 3D モデルを元に「トランプ兵」を、オリジナルで「帽子屋」の立ち絵を MedibangPaint を用いて制作した。「三月ウサギ」の立ち絵は StableDiffusion で自動生成した。また、物語班で必要と判断された表情の差分を用意した。制作の際、ラフや色塗りの途中の段階でグループメンバー内で以下の 2 点を確認した。1 つ目はポーズだ。キャラクター設定に合っているか、人体の構造としておかしくないかなどの点を確認し、必要があれば修正した。2 つ目は色合いだ。ゲームの世界観に合っているか、自動生成された背景に合っているかなどの点を確認した。この時、実際に自動生成された背景と立ち絵を合わせるなどの作業も行った。結果、世界観に合わせて立ち絵の彩度は落とし、キャラクターを目立たせる目的で右側にグレーの影を付けることになった。

#### ・アリス

アリスは 11 種類の差分を用意した。

ポーズは服の自動生成の際に出てきた画像を元に設定した。

「通常の表情」は三面図に忠実に、キャラクター設定も踏まえて芯が強そうに表現した。

「驚いた表情」は主に目を見開き、眉を上にはずらすことで表現した。口を閉じた差分と開けた差分を用意することで驚きの程度の差を付けた。

「笑った表情」は主に口元の微笑みで表現した。目が開いているものと目が笑って糸目になっているものを用意して笑顔の程度の差を付けた。

「悲しい表情」は困り眉と閉じた口、伏せた目元で表現した。また、目線を斜め下にすることで目の前の状況から目を背けたくなるような悲しさを表現した。

「疑う表情」は困り眉と汗で困惑と疑念の感情を表現した。

「焦る表情」は困り眉と見開いた目、汗で表現した。

「残念に思う表情」は、じとっと相手を睨むような目つきや困り眉、汗、少しだけ開けた口で表現した。

「眠る表情」は瞑った目で表現し、眉の形状でなるべく穏やかな表情になるようにした。

「陰る表情」は顔から前髪にかけて影を付けることで表現した。



図 6.30 アリスの「通常表情」の立ち絵

・チェシャ猫

チェシャ猫は6種類の差分を用意した。

ポーズは手を猫のようにし、体の向きをトレードマークである尻尾がよく目立つような後ろ向きに設定した。

「通常表情」は三面図に忠実に、キャラクター設定も踏まえてにやにやとした表情にした。また、口を閉じた差分と口を開けた差分を用意して、にやにやとした表情に程度の差を付けた。

「笑った表情」は眉を下げることで表現した。「通常表情」のにやにやとした表情とは打って変わって穏やかにほほ笑んだ表情にした。

「残念に思う表情」は、困り眉や汗、苦笑いのように開けた口で表現した。

「冷たい表情」は閉じた口元で表現した。

「驚いた表情」は見開いた目とハイライトの消えた瞳、閉じた口元で表現した。



図 6.31 チェシャ猫の「通常表情」の立ち絵

・女王

女王は5種類の差分を用意した。

ポーズは胸を張り、右手は腰に、左手と両足は大きく外に広げて、女王の威厳を誇張するように設定した。

「通常表情」は三面図に忠実に、キャラクター設定も踏まえてツンとした表情にした。

「笑った表情」は笑った口元と細めた目で表現した。特に口元は口角を片方だけ上げることによって得意げな表情にした。

「怒る表情」は吊り上がった眉の目、大きく開けた口で表現した。

「泣く表情」は困り眉と涙、開けた口で、見た目年齢相応に幼い子供が大きく声を上げて泣く様子表現した。

「絶望の表情」は見開いた目とハイライトの消えた瞳、小さく開けた口で表現した。



図 6.32 女王の「通常表情」の立ち絵

・白うさぎ

白うさぎは5種類の差分を用意した。

ポーズは、手を後ろに組み足を肩幅に広げることで、女王の忠実な側近であることを表現するように設定した。

「通常表情」は三面図に忠実に、キャラクター設定も踏まえて困ったような表情にした。また、口を閉じた差分と開けた差分を用意し、困った様子の程度に差を付けた。

「残念に思う表情」は、困り眉や汗、閉じた目、開けた口で呆れて溜息をついている様子表現した。

「冷たい表情」は閉じた口元で表現した。

「驚いた表情」は見開いた目とハイライトの消えた瞳、閉じた口元で表現した。



図 6.33 白うさぎの「通常表情」の立ち絵

・トランプ兵

トランプ兵は1種類の差分を用意した。ポーズは3Dモデルを自動生成した時に入れた間接の場所が分かるようなもの、出力されたモデルの雰囲気に沿った無機質さを感じるものに設定した。

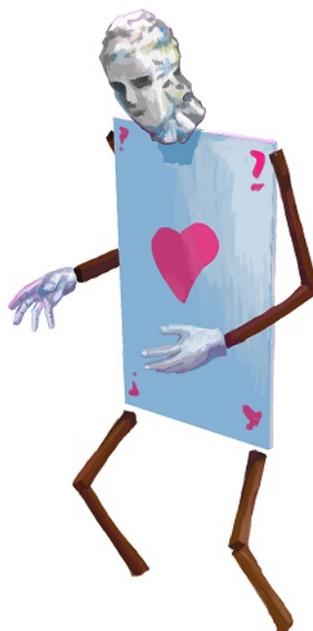


図 6.34 トランプ兵の立ち絵

・三月ウサギ

三月ウサギは2種類の差分を用意した。

他キャラクターとは異なる実写のようなタッチにすることで、三月ウサギがいるお茶会会場やそもその不思議の国全体が狂った空間であることを表現した。

「通常表情」は透過素材に編集しやすいように、背景が青い茶色い西洋のうさぎの画像を生成した。

「楽しげな表情」は前述の「通常表情」を imagetoimage にかけて楽しげな表情のうさぎの画像を生成した。



図 6.35 三月ウサギの「通常表情」立ち絵

・帽子屋

帽子屋は1種類の差分を用意した。

人型ではない異形の生き物というコンセプトで、人間とは異なる身体の構造をしているということ表現した。



図 6.36 帽子屋の立ち絵

(※文責: 坂上翠)

### 6.2.3.2 キャラクター 3D モデル

三面図を元に、「アリス」「チェシャ猫」「女王」「白うさぎ」の3次元モデルを、VRoid studio を用いて制作した。まず、顔・体型・服装についてパーツの選択・編集を用いて大まかに設定した後、テクスチャ編集で詳細なデザインを再現した。テクスチャ編集は、VRoid studio のテクスチャ編集機能と MedibangPaint を併用しながら行った。さらに「チェシャ猫」「白うさぎ」については、尻尾のパーツを髪型のメッシュを複製するという方法で制作したため、VRoid studio の設定では尻尾が頭の動きに追従してしまう。そのため、尻尾を身体の動きに追従させるために blender を用いて調整を行った。調整にあたって、解説動画”【VRoid 正式版】Blender を使って尻尾をちゃんと動かす！ ウェイトとボーン親子変更の解説” (URL: <https://www.youtube.com/watch?v=tLTOV15o5UA>) を参考にした。



図 6.37 キャラクター 3次元モデル

(※文責: 畑井梨里衣)

### 6.2.3.3 背景

ゲーム内背景を Trinant\_v1 を用いて自動生成した。「夜の都会」「森」「お茶会会場」「薔薇の迷路」「白うさぎの小屋 1」「白うさぎの部屋 2」「城外観」「裁判場 1」「裁判場 2」「地下牢」の 10 枚を自動生成した。1つのテーマにつき数枚画像を自動生成し、グループ内で話し合って選定した。画像と入力プロンプトは以下の通りである。

- ・夜の都会



図 6.38 夜の都会の画像

入力プロンプト↓

Concept art of people in intersection in Skyscraper district at night, Shinjuku in Japan, Medium shot angle, masterpiece, Unreal Engine, Game Background, realistic photo, highly detailed, realistic, photorealistic

ネガティブプロンプト↓

helicopter shot, Overhead-angle

・ 森



図 6.39 森の画像

入力プロンプト↓

Illust of dark forest in a fantasy world with colorful mushrooms, many trees, old wooden signboard, magical world, Anime background, Long shot angle, Granblue Fantasy, Low key Light, Game Background

・ お茶会会場



図 6.40 お茶会会場の画像

入力プロンプト↓

fantasy vivid colors, sweets and teapots and tea sets on colorful cloth-covered tables and many chairs in dark forest, Alice in Wonderland, colorful, fantasy, magical world, Anime background, Long shot angle, Game Background, perfect lighting

・ 薔薇の迷路



図 6.41 薔薇の迷路の画像

入力プロンプト↓

masterpiece, concept art of very beautiful white rose garden with several narrow stone paths, many white rose shrubs, labyrinth, stepladder, paint and brushes, vivid color, Alice in Wonderland, colorful, fantasy, magical world, Anime background, close-up angle, Game Background, perfect lighting

ネガティブプロンプト↓

big tree, buildings

・ 白うさぎの小屋 1



図 6.42 白うさぎの小屋 1 の画像

入力プロンプト↓

masterpiece, concept art illustration of the small simple round room, pink checkered tile floor and pink wallpaper, antique small brown color sofa, small bottle on small antique table, Alice in Wonderland, concept art, colorful, Granblue fantasy, magical world, Anime background, Game Background

ネガティブプロンプト↓

low-quality

・白うさぎの小屋 2



図 6.43 白うさぎの小屋 2 の画像

入力プロンプト↓

masterpiece, concept art illustration of the small simple round room, pink checkered tile floor and pink wallpaper, antique small brown color sofa, small bottle on small antique table, Alice in Wonderland, colorful, fantasy, magical world, Anime background, Game Background

ネガティブプロンプト↓

low-quality

・城外観



図 6.44 城外観の画像

入力プロンプト↓

front of the red and white castle of the Queen of Hearts in Alice in Wonderland, from front side extreme close-up angle, highly detailed, Alice in Wonderland, colorful, fantasy, magical world, Anime background, Game Background, perfect lighting

・ 裁判場 1



図 6.45 裁判場 1 の画像

入力プロンプト↓

masterpiece, concept art of the outdoor parliament space in garden, red floor, The presiding judge's seat is high in the center, and the witness stand is across from it, illustration of the courtroom in garden, highly detailed, Medium shot angle, concept art, Alice in Wonderland, colorful, fantasy, magical world, Anime background, Game Background, perfect lighting

ネガティブプロンプト↓

low-quality, blue floor

・ 裁判場 2



図 6.46 裁判場 2 の画像

入力プロンプト↓

masterpiece, concept art of the outdoor parliament room in garden, red floor, The presiding judge's seat is high in the center, the witness stand, highly detailed, concept art, Alice in Wonderland, colorful, fantasy, magical world, Anime background, Game Background, perfect lighting

ネガティブプロンプト↓

row-quality

・地下牢



図 6.47 地下牢の画像

入力プロンプト↓

concept art of In front of the jail cell, dimly lit and narrow stone dungeon, jail, iron grill, dim light, concept art, fantasy, magical world, Anime background, Game Background, perfect lighting

(※文責: 坂上翠)

#### 6.2.3.4 キャラアイテムの絵

各キャラクター(「アリス」「白うさぎ」「チェシャ猫」「女王」)の背景史と関連したアイテムの絵(「猫のキーホルダー」「時計」「食べかけのチェリーパイ」「絵本」)を、Trinart v1 を用いて自動生成した画像を元に、人の手で作成した。以下に、各アイテムの完成画像、デザインの元となる自動生成された画像、画像を自動生成する際に用いたプロンプトを記す。

- ・猫のキーホルダー



図 6.48 完成画像

<デザインの元となる画像 1 >



図 6.49 自動生成された画像

入力プロンプト↓

a mascot keychain, cat, cute, a plush keychain, For Girl

<デザインの元となる画像 2・3・4・5 >



図 6.50 完成画像

入力プロンプト↓

a mascot keychain, cat, cute, a plush keychain, For Girl, Alice in Wonderland

・時計



図 6.51 完成画像

<デザインの元となる画像>



図 6.52 自動生成された画像

入力プロンプト↓

A pocket watch design, master piece, Alice in Wonderland, A pocket watch, classical design, Belongings of the White Rabbit in Alice in Wonderland, rabbit ears

- ・食べかけのチェリーパイ



図 6.53 完成画像

<デザインの元となる画像>

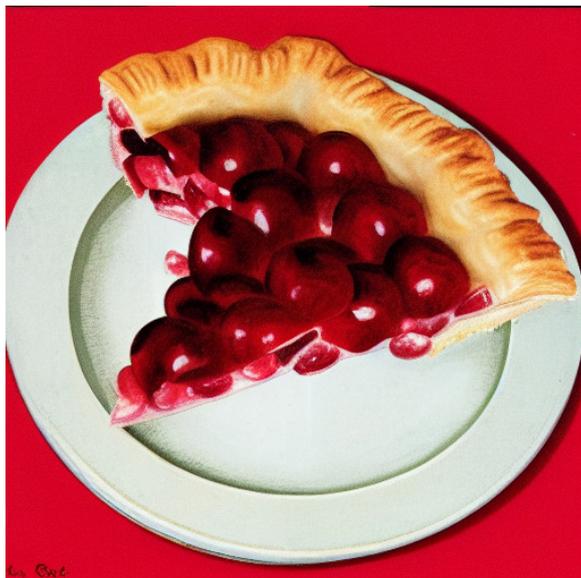


図 6.54 自動生成された画像

入力プロンプト↓

uneaten cherrypie, dirty dish.

・絵本



図 6.55 完成画像

<デザインの元となる画像 1 枚目>

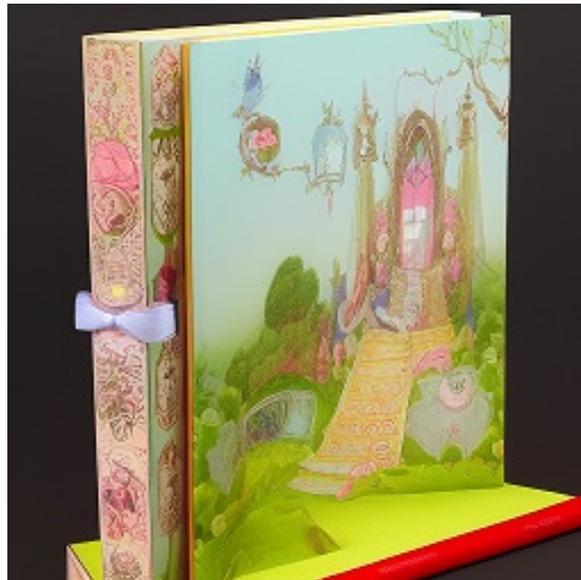


図 6.56 自動生成された画像

入力プロンプト↓

Thick three-dimensional picture book decorative design from an angle, masterpiece, cute, gothic, Alice in Wonderland.

<デザインの元となる画像 2 枚目>



図 6.57 自動生成された画像

入力プロンプト↓

Thick, three-dimensional picture book decorative design seen from an angle, cute, gothic, Alice in Wonderland

(※文責: 畑井梨里衣)

### 6.2.3.5 タイトルロゴ

「Alice in the ideal world」のタイトルロゴを作成した。作成手順として、はじめに手書きで案出しを行った後、いくつかのフォントやカラーバリエーションを試しながら、以下の画像のような複数のロゴを作成した。制作にあたり、少女のシルエット部分の作成は MediBangPaint, その他の編集は Adobe Illustrator を用いて行った。



図 6.58 タイトルロゴ案

さらに、候補を三つに選んだ（画像の 1,2,3）。各デザインの意図を以下に記す。

#### ・1

プロジェクトのテーマとして人工知能が大きく関わるため、Alice の文字における AI の部分を白抜きにして強調することで、同時に人工知能 (AI) の意味合いを含めた。また、不思議の国のアリスという童話をモチーフとしていることから、絵本を連想させるような手書き風の味わいのあるフォントを選んだ。the ideal world の文字については、物語の舞台は理想郷であると同時に、逃げ出したいくなるような過去を抱えた者が集まる逃避の地であり、女王の圧政に支配されているという歪んだ印象を与えるため、文字の並びを少々崩した。さらに世界観を表現するために、主人公アリスをイメージした少女のシルエットを配置した。

#### ・2,3

まず、各キャラクターの背景史には現実逃避という共通項がある。さらに、物語の一つの軸として異世界からの脱出という要素がある。作品全体に関わるこれらの「逃避・脱出」という要素を、「逃げ出す」という意味を持つ非常口のデザインに結び付けた。非常口らしい特徴的なデザイン要素として、矢印記号を A の文字に取り入れ、ピクトグラムを主人公アリスをイメージした少女のシルエットに置き換えた。配色は従来の非常口になぞらえ、白地に緑とした。また、1 と同様の意図で AI を強調するために、Alice の文字の AI と ice の部分を隔てた。

最終的に、タイトル画面で使用するにあたって、背景画像やメニューボタンのデザインと合わせて画面全体の統一感などを考慮した結果、1 の案を採用した。

(※文責: 畑井梨里衣)

### 6.2.3.6 エンディングイラスト

自動生成されたエンディングシナリオを元にエンディングイラストを6枚制作した。使用ソフトは MediBangPaint である。

A.

エンド A は、アリスが現実へ帰ることを決め、チェシャ猫は現実へ希望を託すもう一人の自分であることに気付くエンドだ。鏡越しに向かい合う二人を描くことで「もう一人の自分」を表現した。

B.

エンド B は、アリスがチェシャ猫と共に現実へ帰るエンドだ。互いに見つめあい、ほほ笑む二人を描くことで、自分の意思で希望を持って現実へ帰ることを表現した。

C.

エンド C は、まだここでやらなければならないことがあると、不思議の国に残ったアリスを先に現実世界に帰ったチェシャ猫が迎えに来るエンドだ。逆光の中、優しく微笑みながらこちらに手を差し伸べる姿を描くことで、これから希望のある現実へ導いてくれるのだということを表現した。

D.

エンド D はアリスの目の前でチェシャ猫が処刑されてしまい、ショックで記憶喪失になったアリスが不思議な渦に飛び込み現実へ帰るエンドだ。アリスがチェシャ猫の血を踏んで渦へと向かう様子を描くことで、アリスがチェシャ猫を本当に忘れてしまったということを表現している。

E.

エンド E は処刑されそうなチェシャ猫をかばってアリスが殺されてしまうエンドだ。大切な人を守ることができた安心感でほほ笑むように息を引き取るアリスを抱えてうなだれるチェシャ猫の姿を描くことで、悲壮感溢れる状況を表現した。

F.

エンド F は、チェシャ猫を処刑されたショックで、アリスが女王を殺してしまい、次の不思議の国の女王になるというエンドだ。処刑されたチェシャ猫やそれを見てほほ笑む白うさぎ、アリスが殺してしまった女王など、アリスにとって衝撃的な出来事を背景にしつつ、血濡れの王冠を被ったアリスを中央に描くことで、絶望的な状況を表現した。

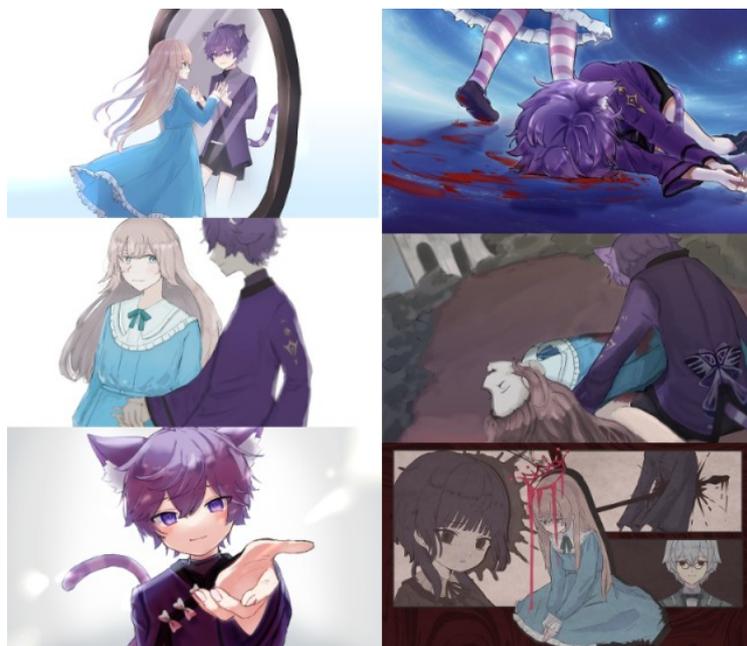


図 6.59 エンド A B C D E F のイラスト

(※文責: 坂上翠)

### 6.2.3.7 3D

後期では主に城ステージの内装に使用する 3D モデルを制作した。内装に使用するモデルはインターネットでダウンロードしたフリー素材をメインに用意をしたが、適切な素材が見つからない際は、視覚班が制作した。後期には城のシャンデリア、女王の玉座を制作した。制作には blender を使用した。城のステージの 3D モデルを生成した。城ステージは 1 階、2 階、謁見の間、女王の部屋、牢屋の 3 つに分かれており、システム班が大まかな城の構造をスケッチ、それを参考に 3D モデルを生成した。シンプルな構造の部屋は、unity 内で単純なモデルを合わせて作ることが出来るが、複雑で細かなモデルは blender で制作することになった。制作の段取りとしては、まずシステム班が部屋のサイズやドアの位置がわかる程度の大まかなモデルを生成、それをもとに、視覚班がより細かい装飾を施すといったように進めていった。一部の部屋やテクスチャは、システム班や物語班のメンバーに協力をしてもらい完成させた。

(※文責: 穂積佳)

### 6.2.3.8 UV 展開

Blender で作製したマップを Unity 側に移動させ、内装を施すため壁や床などにマテリアルを付け加える作業を行った。しかし、マテリアルの表示にいびつが発生し、その修正作業のために UV 展開を行った。UV 展開は Blender, Unity どちらでも可能であったが、操作性の点から Blender を使用した。UV 展開をするにあたり、Blender 上で、マテリアルがどのように表示されているか確認したところ、UV 座標の頂点や辺が線状または点状、1 部分のみだけ表示といった問題が多々あることが分かった。この問題を解決するため、Blender の UVeditor を開き、UV の表示される範囲の変更を行った。まず、1 部分のみ表示されている部分の修正を行った。UV を確認した際に、2 つの直角三角形で長方形、または正方形となり、マテリアルを表示している場合が殆どであった。

そこで、2つの三角形をBlenderの機能である「溶解」を行い、UV展開がしやすい長方形、正方形に変更した。次に、線状、点状になっているUVを修正するために、Blenderの機能の「スマート展開」を行った。これは本来では、モデルに適したUV展開を自動で行う機能である。しかし、UV展開をするモデルに問題があったのか、自動で適したUV展開にならず、マテリアルを表示する範囲が微小ながら展開された。これにより、手動ながらUV展開を修正が容易に行えるようになった。最後に、ドア付近のUV展開はBlenderの機能である「ビューから投影」を行った。これはBlenderの視点からUV展開をする機能である。そのため、ドア付近などのUV展開が複雑になっている部分の修正に適している。また、マップの細部の部分にマテリアルを合わせる際に、全体を表示するようにUV展開をすると、表示が細かくなり、逆効果になってしまう。この場合も「ビューから投影」を行い、壁の模様にあう形で修正を施した。以上のような方法でマテリアルに合わせてUV展開を行い、マテリアルを正しく表示した。その後、Blender状態でUV展開をしてマテリアルの調節をしたマップをUnityに移動させ、マテリアルのデータもUnity上に反映させ、正しく表示されていることを確認した

(※文責: 幸崎駿祐)

## 6.3 音響班

### 6.3.1 データベース構築

曲の雰囲気をより詳細に学習させるために、学習用の音源データベースを構築した。

音源データベースの構築では、学習させる音源データから BPM, クロマグラム, スペクトルセントロイド, ZCR, バンド幅, RMS の六つの音響特徴量を抽出するシステムを開発した。それぞれの特徴を以下に記す。

- ・BPM

BPM は、いわゆる曲のテンポを取得し、この特徴量によって曲の速さを推定することができる。

- ・クロマグラム

クロマグラムは、全周波数帯域のパワーを [C, Db, Eb, E, F, Gb, G, Ab, A, Bb, B] の 12 音階に落とし込み、ある区間の時間における音の成分を可視化したものである。この特徴量から分散をとり、調 (Key) を推定する。具体的には、特定の調を基準とし、分散が小さければ基準とした調が曲の調であると推定できる。

- ・スペクトルセントロイド

スペクトルセントロイドとは、スペクトル重心とも言い、音響波形から周波数分析によって取得したスペクトルの midpoint のことである。すなわち、その周波数を分割点として、上と下でエネルギーが拮抗する。この特徴量から音の明るさを推定でき、値が高ければ甲高い音が、低ければ重い音ということが分かる。

- ・ZCR

ZCR とは、ゼロクロッシングレートとも言い、信号波形が正からゼロへ、または負からゼロへ変化する割合で、主にメロディとパーカッションの音を分類するなどに使用され、今回においては、楽曲の中でノイズがどれくらい入っているのかを示す指標、要するに ZCR の数値が大きければ騒がしい曲、反対に数値が小さければ静かな曲と分類することではないかと考え、採用した。

- ・バンド幅

バンド幅とは、信号の周波数において最高周波数と最低周波数の幅を示すものである。前述の ZCR を用いて高音のみで構成された楽曲を分析すると、ノイズが多い楽曲の波形と近似するため、ZCR 単体では分類しにくいデメリットがある。

そこでバンド幅を採用することによって、あらかじめ全体的な楽曲の高低音を把握し ZCR でノイズの割合を分析することができるのではないかと考え、採用した。

- ・RMS

RMS とは、実効値とも言い、音響特徴量としては音量を指し示す。この特徴量から、音量が大きいか小さいかを推定することができる。

また、取得した音響特徴量を可視化できるように、データベースから CSV ファイルを出力できるように実装した。これにより、学習モデルへのデータ参照も簡略化することもできる。

さらに、取得した音響特徴量を学習モデルへ受け渡すために、変数を数値化する必要があるため、One-Hot ベクトルに変換するシステムを実装した。

(※文責: 菊地陵雅)

### 6.3.2 ループ素材生成

今回、ゲーム内でBGMを実装させるためにフリーBGMを利用した。利用したフリーBGMのサイトを(6.3.3 音源素材集め)に記す。フリーBGMを利用するにあたって、利用する価値が高いもののループ再生できない楽曲が多く、無理やりループ再生させるとゲームの雰囲気に見合わなくなってしまう問題があった。

そこで、ループ再生できない楽曲を違和感なくループ再生できるように、ループ楽曲素材を生成するシステムを、昨年度のシステムを改良して開発した。大まかな概要としては、ループ再生できない楽曲から曲構成を基に楽曲を分割し、分割した曲をループ楽曲素材として生成する。以下に具体的なシステムの流れを記す。(図 6.60)

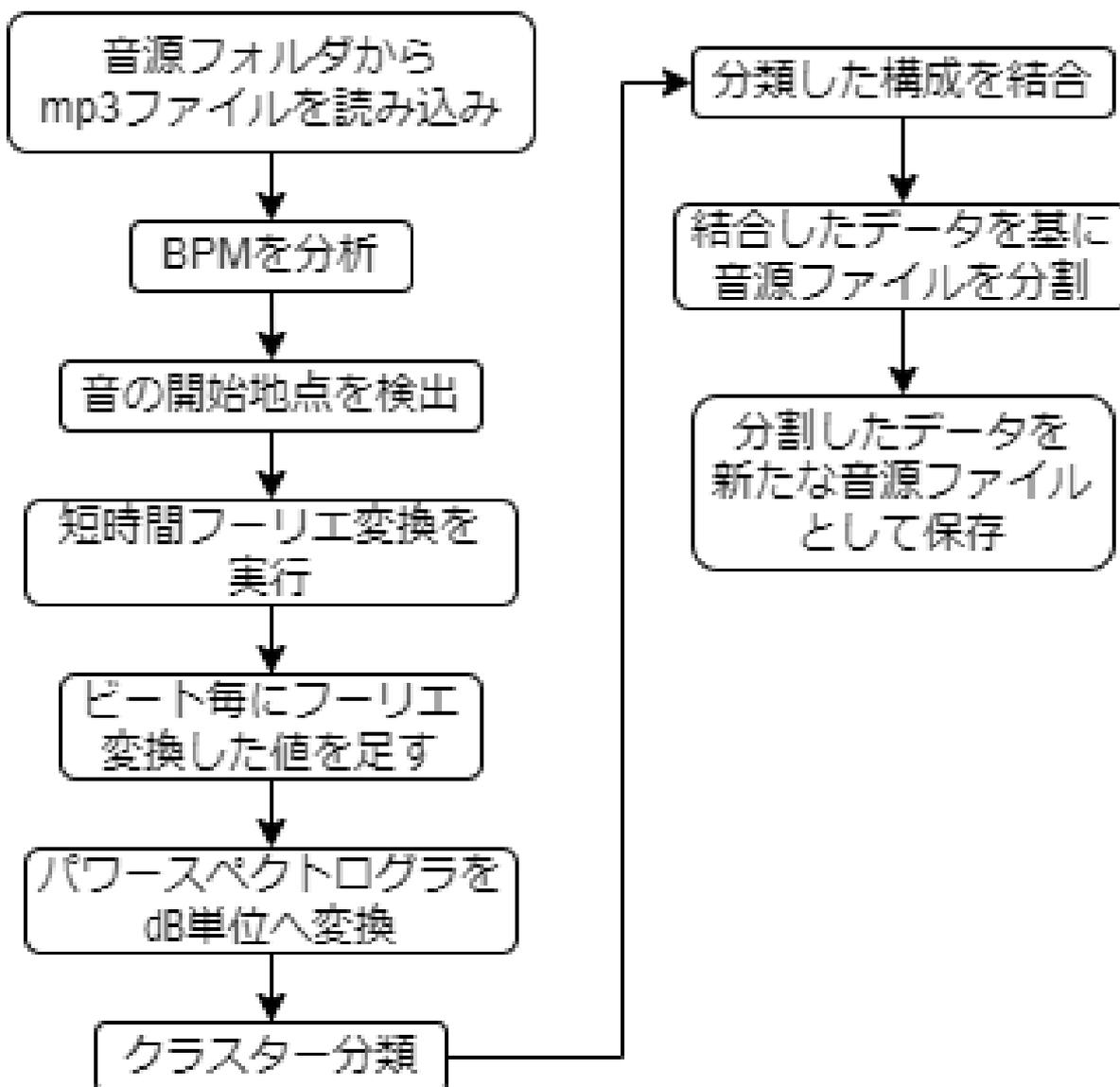


図 6.60 システムの流れ

楽曲構造の分析には、Python の音楽情報処理ライブラリと機械学習ライブラリ scikit-learn を用いた。対象はゲーム内に用いるフリー BGM 全 170 曲で、すべて mp3 形式のファイルである。楽曲を読み込む際のサンプリングレートは 22050 である。

まず、音源フォルダから mp3 ファイルを読み込み、ビートを取得し、ビート間に含まれる周波数の値の合計を算出した。ビートとは拍のことであり、楽曲のまとまりに大きく関係する。よって、決まった時間間隔ではなくビートごとにまとめ直すことによって、後のクラスタリングでより自然な区切りで楽曲を区切ることができるようにする。まず Librosa 内の関数 librosa. onset. onset\_strength で音の開始点を表す Onset の時間変化を計算し、Librosa 内の関数 librosa. beat. beat\_track を用いることで位置を検出した。

次に、Librosa 内の関数 librosa. stft を用いて楽曲を短時間フーリエ変換し、周波数スペクトルの時間経過の行列を算出した。窓の長さは 2048 で、窓の移動幅は 512 である。これにより、1 秒を約 43 分割した範囲で、0Hz から 11025Hz を 1025 分割したそれぞれの周波数成分を得ることができる。時間分解能は約 23. 26ms で、周波数分解能は約 10. 76Hz である。また、短時間フーリエ変換によって得られる周波数成分は約 23. 26ms ごとに分けられているため、それぞれのビート間の時間に該当する部分のデータを加算した。

最後に整理したデータに対し、クラスタリングを行った。昨年度ではクラスタリングに scikit-learn 内の K-Means++ クラス (図 6.61) を用いていたが、今回は scikit-learn 内の Agglomerative Clustering クラスを用いた。Agglomerative Clustering クラスとは、階層的クラスタリング (Hierarchical Clustering) であり、個体からクラスターへ階層構造で分類する分析方法の一つである。(図 6.62) このクラスタリングの長所として、特定の数のクラスターを仮定する必要がなくなる。そして、楽曲構造分析によって分類されたまとまりごとに楽曲を分割し、ループ素材を生成した。

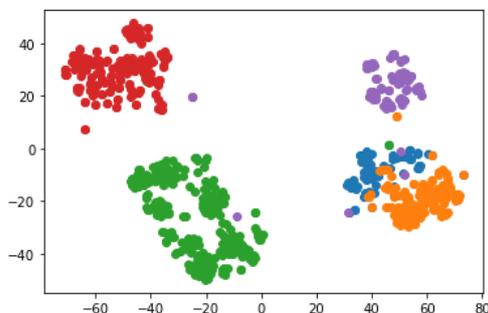


図 6.61 K-Means++ クラスによるクラスタリング

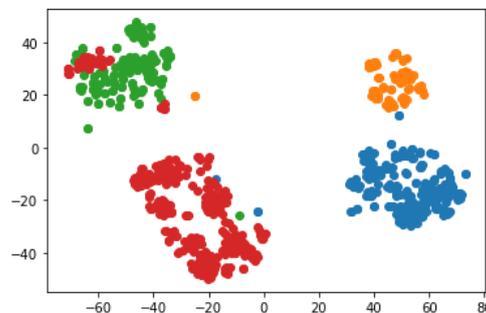


図 6.62 Agglomerative Clustering クラスによるクラスタリング

昨年度のシステムに加えた機能として、曲の指定再生時間以下の音源ファイルを生成しない機能である。従来のシステムでは、楽曲を分割すると再生時間が極めて短い音源ファイルも生成されてしまい、質の高いループ素材の厳選に時間がかかってしまっていた。そこで、違和感なくループ再生できるように小節単位で時間指定し、時間指定以下の音源ファイルを生成しないようにした。今回においては、分割したループ素材が 8 小節以下だった場合生成しないように設定した。また、曲の分割が細かすぎ、分割したループ素材がすべて時間指定以下だった場合には、原曲をそのままコピーするようにした。

結果として、生成されたループ素材は音符一個程度の誤差に収まった。また、まれに区切りの悪いループ素材や分割できなかった楽曲が存在した。これらには三つの原因が考えられる。

一つ目は、楽曲に「アフタクト」という表現方法が用いられた場合である。「アフタクト」とは別名弱拍とも言い、一拍目 (強拍) 以外の弱拍から曲、またはフレーズが始まる表現方法で、今

回のシステムでは強拍を基準に楽曲を分割していたため、区切りが悪くなってしまったと考えられる。

二つ目は、分割前の楽曲そのものがループ素材だった場合である。元々ループ素材の楽曲は、曲の展開が穏やかな楽曲が多かった。今回のシステムでは、曲の展開がはっきりしているほど分割する精度が良く、曲の展開が穏やかに変化していく楽曲に対しては展開がつかみにくく、区切りが悪くなってしまったと考えられる。

三つ目は、拍子の問題である。先述した曲の指定再生時間以下の音源ファイルを生成しない機能において、拍子を考慮する必要があった。今回は全ての楽曲に対して4/4拍子で8小節をカウントしたため、3/4拍子や6/8拍子といったような楽曲には、不自然な区切りのまま生成されたと考えられる。

(※文責: 菊地陵雅)

### 6.3.3 音源素材集め

フリーBGMはDOVA-SYNDROMEや甘茶のお茶工房、BGMer, MusMus, 魔王魂, PeriTuneのサイトから、フリーBGMを約200曲ダウンロードし、それぞれ「森」「タイトル」「小屋」「城」「その他」の雰囲気に合わせて、フォルダ分けした。物語班とシステム班と事前に曲の雰囲気を決めていたため、「森」は暗い雰囲気、「タイトル」はオルゴール調の雰囲気、「小屋」は明るめの曲調だが奇妙さがある雰囲気、「城」は高貴な雰囲気にあうBGMを中心に収集した。

SEは、効果音ラボというフリー効果音サイトを用いて策定した。SEも物語班やシステム班と事前に相談していたため、決定音、キャンセル音、アイテム取得音、衝撃音、ドアの開閉音、草を踏む音、水滴の音、都会の喧騒音を収集し、足音に至ってはプレイヤーの移動に合わせてSEを鳴らす要望があったため、足音のSEを加工した。

(※文責: 菊地陵雅)

### 6.3.4 前期システム改良

#### 6.3.4.1 入出力データ調整

前期の音響システムに対して、統合に向けた入力・出力形式の調整を行なった。具体的には、入力データを曲名にしていたものを候補曲の格納されている場所のファイルパスに変更した。この変更によって、使用楽曲を指定ファイルに置く操作のみで使用楽曲をシステムから呼び出せるようになった。また、出力データに対しての調整も行なった。出力データに各楽曲のテンポデータを追加した。この変更によって、前期で行なっていた自動テンポ取得に比べて、処理時間の短縮だけではなく、曲の拍や小節の把握がより容易・明確になり、結果的に楽曲同士のずれの解消にも繋がった。出力データの増加に伴い、楽曲とテンポのを紐付けたCSV形式のファイルを用意し、そのファイルからデータを抽出する形をとった。

(※文責: 高橋昂大)

#### 6.3.4.2 曲遷移場所検索の改良

現在の曲から後続曲に繋ぐ際に現在の曲の次の小節を参照して、参照した小節に似ている部分を後続曲の中から検索する手法を実装することにした。前期開発では、構想段階で止まっていたため後期の開発ではこのシステムがメインとなった。まず、全ての曲を前期と同様に全てデータ化を行い、曲名と現在の再生位置を指定するといつでもデータを得られるように下準備しておく。この下準備をしておくことで、曲の再生を行いつつ次小節の楽曲データを参照することができる。このデータ化を行なっているプログラムは構築済みのニューラルネットワークを用いている。また、全ての楽曲に対して同じものを使用しているため、曲が似ていると出力データも似ている。そのため、現在曲の次小節データを抽出し、後続曲の中から似ているデータを探してその位置を記録することで、後続曲の似ている部分への遷移が可能になる。

(※文責: 高橋昂大)

#### 6.3.4.3 曲繋ぎ方法の改良

前期の曲繋ぎでは、後続曲のテンポに合わせて現在流れている曲のテンポを少しづつ近づけていく動作を python で実装した。しかし、最終の統合作業は unity で行うため曲繋ぎシステム自体を移行する必要があった。unity と python では言語が違うこともあり、音響系の操作についても多少の違いがあり解消する必要があった。具体的には、テンポ変更をした際に unity では曲の音程も同時に変わってしまうことである。unity では、楽曲を 1.2 倍すると周波数も 1.2 倍されるため曲自体の音程が高くなってしまう。また、楽曲を 1 倍未満にした際も同じ原理で曲自体の音程が低くなってしまう。この楽曲の音程が変わってしまう現象を対処するために unity のオーディオミキサーを導入し、その中のエフェクトである PitchShifter を使用した。PitchShifter を利用することで、unity のスクリプト上からオーディオミキサーに格納されている楽曲の音程を逐次変更することが可能になる。今回の曲繋ぎシステムでは、楽曲のテンポを少しづつ変えていくため音程変更もそれに合わせて変えていく必要がある。PitchShifter の音程変更は倍率で操作するものであったため、常に周波数×音程倍率 = 1.0 になるように周波数の数値に合わせて、音程倍率も同時に変化するようにスクリプトを変更した。

(※文責: 高橋昂大)

#### 6.3.5 オーディオビジュアルライザーの作成

最終発表会に向けて、オーディオビジュアルライザーの作成にも取り組んだ。音声データのみで発表を行う予定であったが、視覚的にデータ化されたものがあつた方がより理解も容易で説明もしやすいと感じたため、音響システムを可視化するオーディオビジュアルライザーの作成に取り組んだ。また、制作・実装は unity 上で行なった。

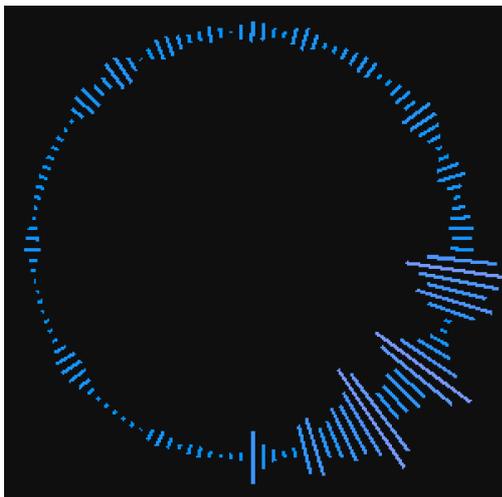


図 6.63 ビジュアライザー寒色系

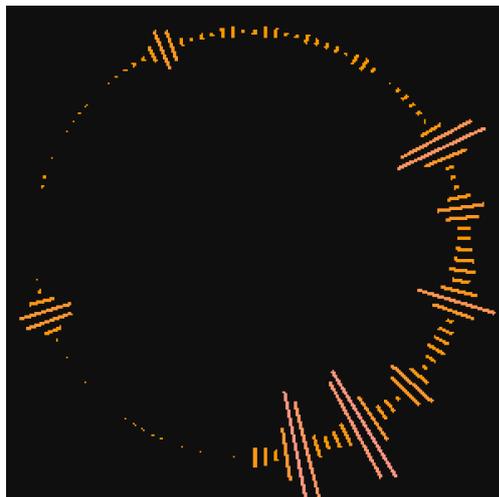


図 6.64 ビジュアライザー暖色系

上記の図のように寒色と暖色、2 パターンの色を準備し、曲の切り替わる瞬間に色も切り替わるように表現した。また、周波数によってサイズの変わるバーを円形に配置することで音楽が流れている様子がわかり易く表現できるし、音楽が切り替わる際に互いの音量関係なども可視化することができた。アンケートでも、音声データの可視化が見やすくわかりやすかったという意見も見られた。結果的に音声データのみで行うよりも効果的に表現を行うことができたといえる。

(※文責: 高橋昂大)

### 6.3.6 システム統合

前期で開発したシステムや後期で改良を行なったシステムを全て統合し、unity に移行する作業を行なった。初期の段階では、unity 側では全体の流れのみをスクリプトで記述し、必要になった際に、python で記述した実行ファイル呼び出して実行する方法をとった、この実行ファイルは、PyInstaller を使用して exe 化を行なった。しかし、システムの受け渡しの際に OS の違いがあり PyInstaller で exe 化したファイルがうまく読み込めなかったりと多くのエラーが発生した。そのため、これまで python で実行していたファイルを unity のスクリプトで再構築する手法に切り替えた。

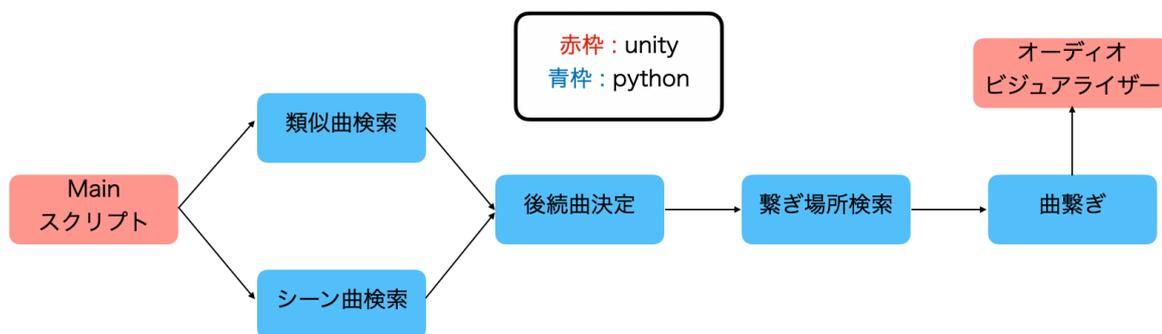


図 6.65 プログラムの流れ

最終的なシステムの流れを上記の図として再構築を行なった。また、入力データは、ゲームにおける次のシーンや場所などの文字データ (お茶会・森など) と曲を切り替えるタイミングの 2 つ

とし、出力データは、ゲームで流れている音楽とした。具体的な流れとしては、シーンや場所などの文字データを入力で受け取った際に、現在の曲との相性と後続のシーンとの親和性を考慮して、音響システムが次回曲の選定を行い決定する。この決定した地点から現在曲の小節毎に後続曲への遷移場所を検索する。そして曲を切り替えるタイミングを受け取った際には曲の切り替えを行い、後続曲のあらかじめ決められていた小節に対して曲繋ぎの処理を行う。また、裏ではこのような動作を常に行うようにしているが、表では絶えず音楽が流れているように設定し、全ての楽曲をループ可能にしているため楽曲の終わりで曲がと途切れることもない。この一連の流れを下図にまとめた。

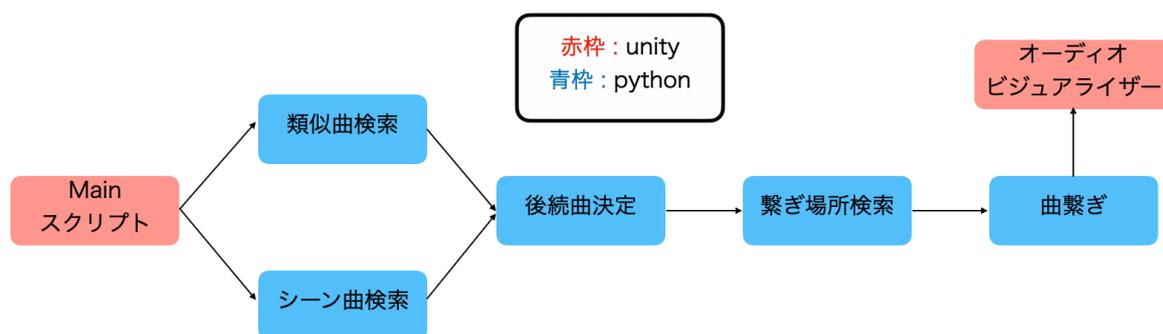


図 6.66 処理の流れ

最終的には、ゲーム中に楽曲が途切れることない音響システムの構築を行うことができた。しかし、曲の繋ぎ方や繋ぎ目の荒さが気になることも少なくはなかった。今後は、楽曲同士のテンポ差や雰囲気の違い、音程の差などの違いによって曲同士の繋ぎ方を工夫していくなどの改良を行なっていきたいと思う。

(※文責: 高橋昂大)

## 6.4 物語班

### 6.4.1 メインシナリオ作成

#### 6.4.1.1 前期からの変更・追加点

メインシナリオ担当は、前期の活動で世界観の設定やメインシナリオの構成の決定、第1章の執筆を行った。後期の活動において、考案していたシナリオ全てを実装するには時間が足りなくなると判断したことや、執筆上での都合から、幾つか設定の変更や追加を行った。以下に具体的な変更点を記述する。

第一に、シナリオ構成の変更を行った。前期では全5章を想定していたが、システム面での実装が間に合わないことが懸念されたことから、全4章に変更した。また、2章と3章はステージの実装が間に合わず、シナリオの表示のみとなったため、その形式に合うようにシナリオを変更した。具体的なシナリオ構成の変更点を以下に記述する。

#### 2章

- ・芋虫が登場しなくなり、アリス、チェシャ猫のみ登場
- ・女王の圧政についての情報は芋虫からではなく、チェシャ猫から聞く

#### 3章

- ・4章の薔薇の迷路の内容が3章に組み込まれる
- ・登場人物にトランプ兵を追加

#### 4章

- ・前期の活動で作成したシナリオ構成の5章の内容が4章になる
- ・裁判の前に、城での探索と、城の地下牢でチェシャ猫と遭遇するイベントが発生する
- ・地下牢でアリスはチェシャ猫が現実に戻りたいと考えていることを知る
- ・地下牢でアリスとチェシャ猫が話していると、女王が登場し、チェシャ猫を裁判に連れていく。その後、裁判に繋がる

第二に、世界観設定の変更を行った。シナリオ展開上、「女王に首を斬られたら現実に帰ることができるのではないか」と推測しているのはチェシャ猫のみとするため、「女王に首を斬られたものは不思議の国から追放され、現実世界へと帰されるという噂がある」という設定を無くした。また、実際に現実世界に帰る条件は、マルチエンディングで行った自動生成の結果に委ねることとした。

第三に、具体的な場所の設定の追加を行った。追加した場所は森、お茶会、薔薇の迷路、女王の城、裁判場の5箇所である。以下に詳しく記述する。

#### ・森

不思議の国の面積のほぼ大半を占める森。カラフルな他の場所と違い、鬱蒼としていて薄暗い。キノコが数多く生えていて、現実では考えられないほど大きなものもある。キノコは食べると不思議な効果をもたらす。整備されているとは言えないものの一応道は存在する。しかし、道がたくさんあり、かつ入り組んでいるのと、道のそばにはいろいろな方向を示す看板が立てられていて、道行く人を迷わせる。道のそばにある大きな木の上にはチェシャ猫がいて、道に迷ったものを上から

見下ろしてにやにやしている。森のはずれには帽子屋と三月うさぎが開くお茶会が、森と女王の城の間には薔薇の迷路がある。

#### ・お茶会

帽子屋と三月うさぎが「なんでもない日」を祝ってほぼ毎日開催しているお茶会。開けた草原に大きなテーブルと椅子がセットされ、テーブルの上にはティーセットとたくさんのお菓子が置かれている。鬱蒼とした森の中でのカラフルなお茶会は非常に奇妙に見える。

#### ・薔薇の迷路

森の出口、女王の城の入り口に位置する、白と赤の薔薇の木で構成された迷路。常に庭担当のトランプ兵が常駐し、整備を行っている。女王は気分が薔薇の色を指定するため、すぐに薔薇の色が変更できるようにペンキと刷毛が用意されている。

#### ・女王の城

薔薇の迷路の先にある、ハートの女王が住む城。トランプ兵が常駐している他、白うさぎも仕事上よく訪れる。横には裁判場も併設しており、外から入ることができる。地下を除き、全体的に赤、白、金を基調とした豪華なデザインが特徴的である。一階、二階、地下が存在する。一階は入ってすぐに大きなホールがあるほか、厨房もある。厨房には、地下にごみを捨てるためのダクトがある。二階には女王の部屋があり、玉座と、女王のお気に入りの小物が置かれている。地下には裁判前の罪人を収容するための石造りの地下牢があり、薄暗いその雰囲気は上の二つの階とは全く異なる。一階にある地下への入り口は、トランプ兵によって厳重に警備されている。

#### ・裁判場

女王の城に併設されている裁判場。「裁判場」といっても、屋内ではなく外にある。裁判長の席は数段高くなっており、ハートの女王が座っている。裁判長席と向かい合うようにして被告人の席があるほか、横には白うさぎが座るための席もある。周りはトランプ兵によって厳重に警備されている。

第四に、キャラクター数の変更、キャラクター設定の変更を行った。キャラクター数については、シナリオを短縮したことにより芋虫が登場しないこととなったため減少した。キャラクターの設定については、キャラクターの背景史の自動生成との関係も考慮し、各キャラクターの現実での基本的な設定を追加した。加えて、帽子屋、三月うさぎ、トランプ兵についてはこれらのキャラクターが登場するシナリオを前期時点ではまだ執筆していなかったこともあり、実際の執筆に伴って不思議の国でのキャラクター設定も追加した。以下に具体的に追加した設定を記述する。

#### ・アリス

現実では、20代の女性である。

#### ・白うさぎ

現実では、10代後半の男性である。

#### ・チェシャ猫

現実では、30代半ばの男性である。

・帽子屋

アリスのことは、帽子屋にとっての常識が通用しないことから見下している。  
現実では、40代前半の男性である。

・三月うさぎ

帽子屋と同じくおかしな言動をするが、多少の常識は残っているため、意味の通じる会話は可能。語尾に「ねぇどうして?」とつけるのが口癖。不思議の国に来てもお現実を捨てきれないアリスに対して苛立ちのような、うらやましいような複雑な感情を持っている。しかし自分は不思議の国側であるため、帽子屋の態度にのっとなってアリスのことは見下している。  
現実では、20代後半の男性である。

・トランプ兵

二人のトランプ兵が登場する(以下トランプ兵A, Bとする)。トランプ兵Aは態度が大きく話し方も厳しい。トランプ兵Bは気が小さいが調子に乗りやすい性格である。  
現実では、30代から40代の男性である。

・ハートの女王

現実では10代前半の女性である。

(※文責: 吉井史夏)

#### 6.4.1.2 メインシナリオ執筆

中間発表までの活動の項目でも述べた通り、エンディング以外のメインシナリオは全て人の手で執筆することとなった。また、中間発表までの時点ではプロローグとプロトタイプのための第1章の執筆が完了していたが、前述したようなシナリオ構成の変更もあり、プロローグと第一章の内容も修正しつつ、エンディング直前までの全4章の執筆を完了した。第一稿の執筆後はプロジェクトのメンバーに意見を貰って修正を行った。加えて、自動生成されたマルチエンディングや背景史と合わせたときに矛盾が無い状態となるようにシナリオを調整したほか、キャラクターの言動にそのキャラクターの背景史を窺わせるものものを追加し、伏線を張るなどしてメインシナリオがより重厚となるよう意識して執筆を行った。具体的な内容は次の章に記述する。

前期でメインシナリオを人手で執筆することが決定した際、自らシナリオを執筆する経験はAIに物語を作成させるという課題に対して、クリエイティビティの発揮の面で非常に有意義となると先生に助言をいただいた。実際執筆しているときには、「この展開は本当に面白いのか」と悩むことが何度もあり、自らの発想力の限界にも向き合うことになった。しかし、だからこそ、AIが自動生成した結果に対して、自分にはなかった発想を得られることに価値を感じることができた。自動生成した内容にあわせて調節を行うための人手での執筆作業だったが、この過程を通して、AIの価値を再認識することができた。

(※文責: 吉井史夏)

### 6.4.1.3 自動生成したシナリオを組み込むにあたっての調整

物語班では、キャラクターの背景史に関するテキストと、マルチエンディングを自動生成した。具体的な自動生成については後述の各章に譲るが、自動生成したこれらの内容に対して矛盾が無いようにメインシナリオの調整を行った。

背景史とメインシナリオの調整については、先述の通り、メインシナリオ中のキャラクターの言動にそのキャラクターの背景史を窺わせるものを追加した。具体的な一例を紹介する。主人公であるアリスの背景史に、友人が自殺してしまうという描写があった。そこで、アリスが決意する場面の台詞を

アリス「それに、ここで引き下がったら、私はたぶんずっと後悔する…だから」

というものから

アリス「それに、もう二度と恩人を死なせたくない…。だから」

と変更することで、アリスは過去に親しい人間を無くしているのではないかという考察につながるができるように工夫し、アリスの背景史への伏線とした。マルチエンディングとメインシナリオの調整については、メインシナリオが自然にマルチエンディングに繋がるように工夫した。エンディングは6つあり、それぞれのシナリオが、アリスが現実へ帰ることを決意するのかそれとも不思議の国に留まろうとするのか、また、実際にどういう結果となるのかという内容を含んでいる。マルチエンディングの項目で詳しく触れるが、エンディングはエンディング直前に示される選択肢とアイテム入手率によって決定する。「現実へと帰る手段を持っていると思われるチェシャ猫に、アリスが現実へと帰ることを提案される」という場面でメインシナリオを終了させ、分岐条件となる選択肢に、そしてその先のエンディングの内容に自然と繋がるように執筆を行った。

(※文責: 吉井史夏)

## 6.4.2 背景史

### 6.4.2.1 データ収集

前期の活動では、背景史の自動生成のために「現実逃避しているキャラクター」が登場する完成度が高い4作品を分析対象として取り上げた。プロジェクト学習開始当初より、背景史の自動生成に必要なデータは100個程度と見積もっていた。後期の活動開始時点ではデータ量が100個を超えていたため、新たなデータ収集の優先度は自動生成アルゴリズムの構築に比べて低かった。しかし、必要なデータ量が想定よりも多かった場合に備えて、データ分析や自動生成アルゴリズムの構築と並行してデータ収集を行うことにした。分析対象は、ハリー・ポッターシリーズの「ハリー・ポッターと賢者の石」を取り上げた。選定理由は、前期の活動で収集したハリー・ポッターシリーズのデータ量が他の3作品よりも少なく、同作品が若年層のキャラクターが多く登場するファンタジー作品で、収集済みのデータの補完という観点でも最適であると判断したためだ。

プロジェクト学習の終盤では、背景史のデータ総数は122個となった。データ量の内訳は、名探偵コナンが34個、おジャ魔女ドレミが25個、ハリー・ポッターが25個、笑ゥせえるすまんが38個である。

(※文責: 坂本珠凜)

### 6.4.2.2 カテゴリライズ

収集したデータは、前期に作成したフォーマットに従ってデータの特徴ごとにカテゴリライズして Google スプレッドシートに入力した。フォーマットに設定したカテゴリのうち、「現状認識」、「影響」、「分類」はベクトルの表記に直す必要があった。これは、収集したデータを分析しやすい形に置き換えるためである。

上記のカテゴリのデータをベクトルに直すにあたって、細分化されたデータの統合を図った。これは、分析に使う個々のデータが想定以上に細分化されてしまい、そのままの状態だと分析に適さなかったためだ。ここで、「影響」のカテゴリのデータを例に挙げて、細分化と統合の過程を見ていく。「影響」のデータには、「逃走」や「忘却」、「忌避」というものがあった。これらは1つ1つのデータ量が乏しかったため、「逃避」という小カテゴリに統合して分析に使える水準のデータ量にした。この作業を「現状認識」と「分類」のカテゴリにも行い、最終的に全6個のカテゴリを13個の小カテゴリに分割した。

このとき、13個の小カテゴリのデータは全てテキストで保存されていたが、分析に使うデータはベクトルである必要があった。そこで、データ分析前の最終準備として、データのベクトル化を行った。ここで、13個の小カテゴリのうち「逃避」を例に挙げて、ベクトル化の過程を見ていく。上記にあるように、「逃避」は「影響」に属する小カテゴリであり、トラウマによって現れた「影響」のうち、トラウマの対象からの「逃避」を意味するデータを持つ。このデータのベクトル化には、トラウマの対象に相対したキャラクターが、逃避したか否かを1と0の2値で判別する必要がある。つまり、キャラクターが逃避した場合は1、逃避しなかった場合は0を Google スプレッドシートに入力した。この作業を全13小カテゴリに行い、データ分析の準備を終えた。

なお、後期より本格的にデータ収集を開始したハリー・ポッターは、データのベクトル化時点ではデータ数が少数だったため、データ分析に用いたデータの総数は101個となった。

(※文責: 坂本珠凜)

### 6.4.2.3 分析

背景史を自動生成する為に、収集したデータの分析を行った。分析手法としては収集したデータの中でトラウマとして多くみられる要素の発見のため、またデータ数とカテゴリライズしたカテゴリー数の関係から因子分析を採用した。因子分析とは外的規準の無い大量データから共通因子を見つけ出し、データの背後にある少数の要因を統計的に推測する方法である。これにより、背景史やマルチエンディングのよくあるパターンを探索できると考えた。そこで、カテゴリライズしたデータを因子分析に適用できるように、カテゴリライズした項目のベクトル化を行った。この際データ数は101個、変数は33個であった。実際の変数の種類と、どのようにベクトルで表現したかを具体的に説明する。

「年齢」では、キャラクターの年齢が20歳未満の場合を-1、60歳以上の場合を1、それ以外は0を割り当てて表現した。「性別」ではキャラクターが女性の場合を1、男性の場合を0と割り当てて表現した。「認識」ではトラウマを認識している場合を1、認識していない場合を0と割り当てて表現した。「行動」ではトラウマに対して解決しようと行動する場合を1、行動しない場合を0、行動していないわけではないが解決に向けての行動ではない場合を-1と割り当てて表現した。「責任」ではトラウマの原因が自分にある場合を1、不明な場合を0、他者にある場合を-1と割り当てて表現した。「恐怖」ではトラウマに対して恐怖をしている場合を1、恐怖をしていな

い場合を0, 恐怖しているが立ち向かっている場合を-1と割り当てて表現した。「落ち込み」ではトラウマに対して落ち込んでいる場合を1, 落ち込んでいない場合を0と割り当てて表現した。「忘却・現実逃避」ではトラウマを忘れようとしている場合を1, 忘れようとしていない場合を0と割り当てて表現した。「逃走・忌避」ではトラウマから逃走している場合を1, 逃走していない場合を0と割り当てて表現した。「怒り」ではトラウマの原因に対して怒りを感じている場合を1, 怒りを感じていない場合を0と割り当てて表現した。「嫌悪」ではトラウマの原因に対して嫌悪を感じている場合を1, 嫌悪を感じていない場合を0と割り当てて表現した。「恐怖」ではトラウマの原因に対して恐怖を感じている場合を1, 恐怖を感じていない場合を0と割り当てて表現した。「悲しみ」ではトラウマの原因に対して悲しみを感じている場合を1, 悲しみを感じていない場合を0と割り当てて表現した。「驚き」ではトラウマの原因に対して驚きを感じている場合を1, 驚きを感じていない場合を0と割り当てて表現した。「逃避行動」ではトラウマに対して逃避行動をしている場合を1, 逃避行動をしていない場合を0と割り当てて表現した。「奮い立たせる」ではトラウマの解決に向かい自分を奮い立たせている場合を1, 奮い立たせていない場合を0と割り当てて表現した。「攻撃的」ではトラウマを想起したときに攻撃的になる場合を1, 攻撃的にならない場合を0と割り当てて表現した。「精神的疲労」ではトラウマが原因で精神的疲労を感じている場合を1, 精神的疲労を感じていない場合を0と割り当てて表現した。「混乱」ではトラウマを想起したときに混乱する場合を1, 混乱していない場合を0と割り当てて表現した。「受容」ではトラウマを受容している場合を1, 受容していない場合を0と割り当てて表現した。「友人関係」ではトラウマの原因が友人関係である場合を1, 友人関係でない場合を0と割り当てて表現した。「家族関係」ではトラウマの原因が家族関係である場合を1, 家族関係でない場合を0と割り当てて表現した。「会社関係」ではトラウマの原因が会社関係である場合を1, 会社関係でない場合を0と割り当てて表現した。「恋人関係」ではトラウマの原因が恋人関係である場合を1, 恋愛関係でない場合を0と割り当てて表現した。「不思議」ではトラウマの原因が不思議な現象である場合を1, 不思議な現象でない場合を0と割り当てて表現した。「理不尽な扱い」ではトラウマの原因が理不尽なものである場合を1, 理不尽なものでない場合を0と割り当てて表現した。「大切なモノ・ヒト」ではトラウマの原因が大切な物や人である場合を1, 大切な物や人でない場合を0と割り当てて表現した。「ネガティブイベント」ではトラウマの原因が運悪く遭遇してしまっただけの場合を1, そうではない場合を0と割り当てて表現した。「劣等感」ではトラウマの原因が自分の劣等感である場合を1, 自分の劣等感ではない場合を0と割り当てて表現した。「自分に非がある」ではトラウマの原因が自分に非がある場合を1, 非がない場合を0と割り当てて表現した。「願望」ではトラウマの原因が自分の願望に関係している場合を1, 自分の願望に関係していない場合を0と割り当てて表現した。「ショック」ではトラウマの原因がショックを受けるような場合を1, ショックを受けるようなものでない場合を0と割り当てて表現した。「いやだ」ではトラウマの原因が自分が嫌だと思っている場合を1, 嫌だと思っていない場合を0と割り当てて表現した。

このようにベクトル化を行い因子分析を行ったが、カテゴライズの時点でデータを細かく分類し過ぎてしまったために、因子を見つけることが出来なかった。そのためカテゴリーの似通っている項目の統合や削除を行った。結果変数は14個になった。実際の変数の種類と、どのようにベクトルで表現したかを具体的に説明する。

「年齢」ではキャラクターの年齢が20歳未満の場合-1, 60歳以上の場合1, それ以外は0と割り当てて表現した。「性別」ではキャラクターが女性の場合を1, 男性の場合を0と割り当てて表現した。「行動」ではトラウマに対して解決しようと行動する場合を1, 行動しない場合を0。行動していないわけではないが解決に向けての行動ではない場合を-1と割り当てて表現した。「責任」

ではトラウマの原因が自分にある場合を1, 不明な場合を0, 他者にある場合を-1と割り当てて表現した。「恐怖」ではトラウマに対して恐怖をしている場合を1, 恐怖していない場合を0, 恐怖しているが立ち向かっている場合を-1と割り当てて表現した。「落ち込み」ではトラウマに対して落ち込んでいる場合を1, 落ち込んでいない場合を0と割り当てて表現した。「逃避」では忘却・現実逃避・逃走・忌避いずれかがある場合を1, いずれかがない場合を0と割り当てて表現した。「負の感情」では怒り・嫌悪・恐怖・悲しみ・攻撃的いずれかがある場合を1, いずれかがない場合を0と割り当てて表現した。「ポジティブな行動」では奮い立たせる・受容いずれかがある場合を1, いずれかがない場合を0と割り当てて表現した。「対人関係」では友人・会社・恋人いずれかがある場合を1, いずれかがない場合を0と割り当てて表現した。「ネガティブイベント」ではネガティブイベント・理不尽な扱い・大切なモノ/人いずれかがある場合1, トラウマの原因が不思議な現象である場合を0と割り当てて表現した。「自己要因」では劣等感・自分に非がある・願望・ショック・拒絶いずれかがある場合を1, いずれかがない場合を0と割り当てて表現した。「精神的疲労」では驚き・精神的疲労・混乱いずれかがある場合を1, いずれかがない場合を0と割り当てて表現した。

Rの関数を使用し因子数を決定する際, 実行したRのバージョンの違いやOSの違いによって出力される因子数に違いが出るのが分かった。WindowsでRのバージョン4.1.3を使用すると因子数が3, WindowsでRのバージョン4.2.1を使用すると因子数が4, MacでRのバージョン4.2.1を使用すると因子数が5と出力された。そこでグラフを出力させて確認を行った。

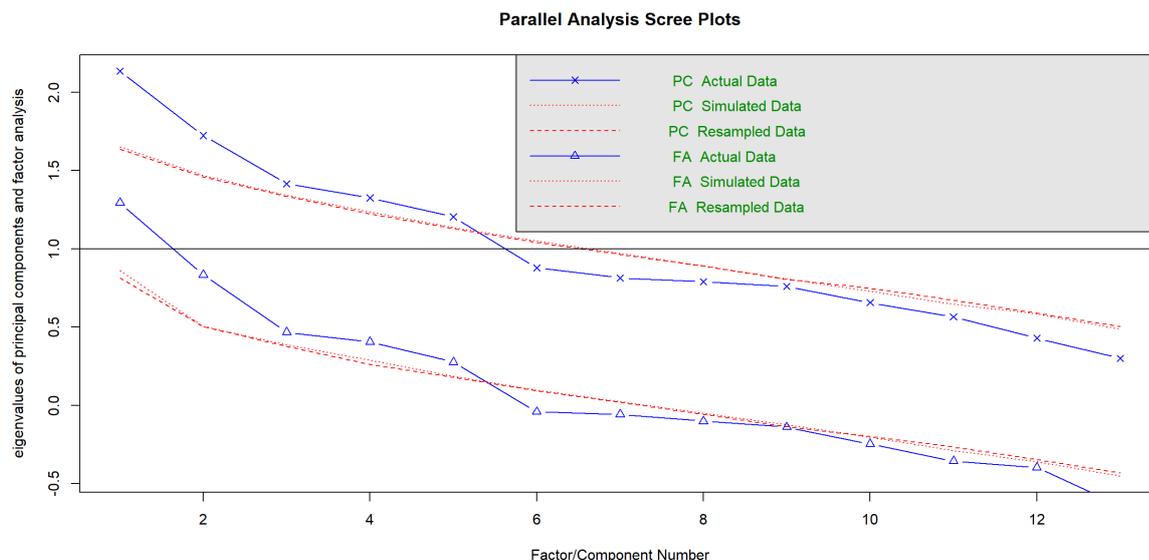


図 6.67 スクリープロット

グラフのように因子数3, 4, 5ではほとんど差がないことが分かった。このことから, 今回の因子分析では因子数3, 4, 5それぞれで分析を行うことにした。

それぞれの分析の結果を見て各因子の解釈を行った。ここでは因子負荷量の絶対値が0.5以上のものを強く働いていると解釈し, それに付随するいくつかの変数も解釈に使用した。因子数3で分析して出た因子の解釈は「対人関係でない場合は自己要因である」「少女は行動する」「精神的疲労があると負の感情はなく逃走する」である。因子数4で分析して出た因子の解釈は「少女は行動する」「対人関係がないときは自己要因である」「精神的疲労があるときは負の感情がない」であった。因子数4で分析を行った際, 因子内の変数の因子負荷量が絶対値0.5を超えるものがない因

子が1つ存在したため解釈を3つとした。因子数5で分析して出た因子の解釈は「精神的疲労があるときは負の感情やポジティブな行動がない」「対人関係でないときは自己要因である」「少女は行動する」「自分に責任があるときは不屈である」「ネガティブイベントがないときは自己要因がある」であった。その後、背景史を作成するキャラクターが4人であったため、各キャラクターの設定にある程度沿っていて背景史に利用しやすい解釈を4つ決定し使用することにした。

(※文責: 入船真誠)

#### 6.4.2.4 自動生成アルゴリズム

自動生成にあたって、大規模言語モデルである GPT-2 の採用を検討した。しかし、本プロジェクトでは背景史を日本語で生成することを目標としていたため、日本語に特化した言語モデルが必要だった。そこで、一般的な日本語テキストの特徴を有した高度な日本語文章の自動生成が可能である GPT2-japanese と、その後継となる GPTSAN を採用した。

GPT-2 は、単語レベルの確率の組み合わせから文の確率を計算する言語モデルである。そのため、背景史を自動生成させるためにはプロンプトと呼ばれる命令文を入力する必要がある。プロンプトは出力したい文章と同じような構成、文量である必要がある。そこで、データ分析によって発見された因子を命令文の形に整形し、GPT2-japanese と GPTSAN に入力する手法をとった。最終的に、因子とプロンプトとして与えた文章の組み合わせは以下の4つである。

因子1：少女は行動する

”彼のことが好きな女の子たちは、彼に近づこうとする私を虐めた。私は女の子たちに無視され、私の家の鍵はゴミ箱に捨てられた。それでも、私は彼の隣にいたかった。”

因子2：精神的疲労があるとき、負の感情はない

”友人が津波に巻き込まれて死んだと遺族から聞いた。あの日、興味本位で津波を見に行くと言った友人を、なぜ俺は止めなかったのだろう。今でもあの時のことを思い出すと動悸がする。”

因子3：精神的疲労があるとき、逃避する

”俺の両親は多額の借金を抱えていた。仕事で精神を病んだ父親に殴られて泣いて謝る母親を、俺はただ見ていた。あのとき母親を助ける勇気を持てなかったことを、今でも後悔している。”

因子4：自分に責任があるとき、不屈である

”私は子供の頃はエレクトーンのコンクールで数々の賞を獲得していた。しかし、徐々に賞を取れなくなり、周囲の大人たちからも期待されなくなっていった。でも、たくさん練習して次こそはコンクールで最優秀賞を取ると決めた。”

各プロンプトは3つの文からなる短い文章として、データ分析によって発見された因子を重複しないように取り入れて作成した。しかし、プロンプトは抽象的であればあるほど生成される文章も表現が曖昧なものとなり、具体的であればあるほどプロンプトに近いものが生成されることが予想された。そこで、プロンプトの適切な情報量を探るべく、最初は因子を簡単な文章にしてプロンプトとして与え、自動生成を繰り返す過程で適宜修正を加えていった。また、プロンプトは「本プロジェクトのゲームキャラクターに与えたい背景史」を自動生成できるように意識して作成した。例

えば、因子1は少女に関する内容であったため、主要ゲームキャラクターの1人である「ハートの女王」に与える背景史としてプロンプトを作成した。因子1のプロンプトは当初「彼女はトラウマの解決に向かって奮起した」だったが、繰り返し生成させた文章にいじめや恋愛に関する記述が比較的多く見られたため、最初からプロンプトとしてそれらの要素を与えることで徐々に具体的な背景史を生成できるように誘導した。なお、「ハートの女王」に与える背景史であるからといって、プロンプトの主語を「ハートの女王」に置き換えることは避けた。これは、ハートの女王という単語が様々な単語を連想させてしまい、原典である「不思議の国のアリス」から設定を引用してしまう恐れがあったためだ。

プロンプトの他にも、GPT-2には自動生成する文章の特徴を設定するパラメータがあらかじめ割り振られている。このパラメータには「文章長の最大と最短」や「生成される文章の無作為性」を設定するものがあり、こちらもプロンプトと同様に自動生成を繰り返しながら各パラメータの値を調整した。上記で設定したパラメータとプロンプトを入力として生成した背景史の一例を図6.68に示す。

父親を殺した犯人を見つけ出す作業を、俺は必死で仕上げていた。  
 \n背景史：あのころの俺は何も出来なかった。俺を支えてくれたのが、母親だった。  
 \n背景史：一人で家に帰った俺は、思い出の家へと帰っていった。その途中、建物の影から怪しげな音が聞こえたことを覚えている。そんなところにいたんだ、と、そう思いながら、私は何も考えなかった。考えることは、考えられなかった。

図 6.68 自動生成結果

(※文責: 坂本珠凜)

#### 6.4.2.5 ファインチューニング

プロンプトの入力で言及される作業によって GPT-2 を用いた自動生成は完了した。しかし、物語班としては、追加の目標として「文章を読むことで解釈の余地がいくつも存在する」ような文章を作成したいと考えた。また、文体を統一することで文章の統一感を出すことも目的となった。

実現の方法として、大規模言語モデルに目標とするような文章に近いデータをファインチューニングすることで、出力を調整する方法を採用した。

ファインチューニングは、自動生成で用いた GPT2-japanese で行った。本モデルの採用理由として、このモデル以外では、言語モデルが大規模でも、外部のサイト等を用いてデータを扱わなければならない作業が複雑になる。しかし、本モデルでは、python 環境のみで解決するため採用に至った。

ファインチューニングを行うデータとしては、2022 年 Game of the year を受賞した Elden Ring や Ultimate Game of All Time を受賞した DARK SOULS シリーズを中心にデータの収集を行った。収集するデータ部分は、ゲーム内のアイテムテキストであった。アイテムテキストは、アイテムそのものの説明の文章とそのアイテムに関係する内容を示した文章部分で主に構成される。データとして扱ったのは、アイテムに関係する文章部分であった。文章として採用されるモノは、人物の行動やゲーム世界の歴史に関係するモノを抜粋し収集を行った。文章は、データとして扱うため改行を消し、1 行の文章の形に変更し、文末に文章の終了を意味する `—endoftext—` を加えた。このようなデータを 300 ほど集めた。

ファインチューニングは、GPT-2 japanese が推奨する Japanese-BPEEncoder を用いてデータ

形式を npz 方式に変更して学習を行った。学習では、gpt-2 が提供する medium モデルを 300 エポックほど学習した。

Python の環境は諸事情からデータの npz 形式の変更と学習の作業を異なる python 環境で行っている。学習は、コマンドプロンプト形式の python3 を使用しているが、データの形式変更は Google Colaboratory を利用している。データの形式を変更する処理をコマンドプロンプトの python で行った場合、分散系の処理に問題が発生しエラーが起きるため、Google Colaboratory を利用している。一方で学習過程は、本来、学習を Ctrl+C で停止することができるが、Google Colaboratory だと、学習を途中で打ち切ることが出来ないためコマンドプロンプト形式の python で学習を行った。

学習を行った GPT-2 に前回と同じプロンプトを用いて入力を行い Google document で文章量が 10 ページになるよう出力を行った。

学習前に示していた出力結果と同様のプロンプトで行った出力結果を図 6.69 に示す。

あのとき、こんな風に思えたのだろうか。「そんな顔をさせるくらいなら、いっそ死んだほうがマシだ」あの日、思ったのだろうか、もう母親の顔を思い出したくないと。  
背景史：私は、せめて海の向こうの人々の思い出の一部になりたかったのだ。私は、あの悲劇の記憶に浸り、安楽椅子に腰を掛けてそう願ったのだろうか

図 6.69 ファインチューニング後の自動生成結果

(※文責: 西村秀幸)

#### 6.4.2.6 両出力結果の比較

ファインチューニングを行う前後を比較すると大きな違いがいくつかあることが分かった。まず、学習前では、一人称が「私」、「俺」等、安定しなかったのがファインチューニング後の出力では「私」の頻度が多くなった。また、文章の接続の自然さも上昇した。学習前の出力結果では、文章間の接続表現が少なく、文章の区切りの後の文章が、文脈上続いているような文章が出力されている。学習前の出力である図 6.68 の「あのころの俺は何も出来なかった。俺を支えてくれたのが、母親だった。」文章では、初めの文章と後の文章は読むことは可能ではあるが、文章間の接続においては、出来なかったという言葉と俺を繋ぐ言葉がないため不自然に感じる。しかし、学習後の図 6.69 では、「もう」、「浸り」というような接続が可能な言葉や動詞を連用形にして接続することで読みやすく自然な文章へと変化している。学習後は、文脈に合わないような表現が減り、言い回しとして許容出来るものが増加した。図 6.68 では、始めの 1 行目の部分は、「見つけ出す作業を必死に仕上げている」というような旨の文章があり、作業という言葉から仕上げるというワードが表れたと考えられるが、自然な日本語とはいえない。しかし、学習後では、「せめて海の向こうの人々の思い出の一部」という文章の接続の関係上自然であり、違和感を感じずに読むことが出来る表現が出力されている、また、「あの」や「その」、「だと」といった抽象的な表現が増加している。学習後の文章では、「だろうか」や「と」という表現が語尾に増えているとともに、「あの日」といった表現も増えている。これらは、ファインチューニングによる大きな効果として感じられる。

(※文責: 西村 秀幸)

#### 6.4.2.7 文章選定

GPT-2 で生成した Google document で 40 ページ分の文章の中から 4 つの文章の選定を行った。選定基準として 4 つの基準を設けた。具体的には、出力された 256 字以下の文章である程度のストーリーの流れが存在し、その内容が妥当であること、入力したプロンプトやファインチューニングで使用したデータにない創造性 (新しい単語や表現の発露が見られる) が見られること、文章の繋がりが著しく破綻していないこと、本人が精神的な快方に向かっている描写がない、泣いているといった負の感情が発露している、否定的なワードが表れているなどトラウマとして判断できること、を基準とした。また、必須ではないが読み手に創造性を与えることが出来る (感情の想像の余地を与えている) 文章も選定の対象とした。下記に選定した文章の一例と選定理由を記す。

だけど、努力が失敗した。彼女は自殺した。背景史: 諦めきれず、私はコンクール上位者に憧憬を見せた。たとえ、その夢が甘かったとしても。背景史: 私は彼女の夢を応援した。美術門を開き、傑作を送り出したあの人の夢を背景史: あの青い瞳は、描けば切り裂かれてしまいそうだ。私は彼女の夢を支えると決めた。彼女が笑いかけてくれるように。

図 6.70 選定した文章例

選定理由としては、「彼女の夢を支える決心をして、応援していたが彼女の努力が実らず自殺してしまった」というストーリーの流れが存在し、その内容が妥当であり、「あの青い瞳は、描けば切り裂かれてしまいそうだ。」という文章はプロンプトやファインチューニングのデータにない創造性があると考えたためである。また、文章の繋がりが著しく破綻している箇所が見られず、彼女の夢を支えることを決心したが彼女が自殺してしまうというストーリーからトラウマとして判断できるため、この文章を選定した。このような選定方法で、23 個の文章を選定した。しかし、今回プロジェクトでは 4 キャラクター分の背景史で十分であるため、メンバーと話し合い最終的に 4 つの文章を選定した。

(※文責: 入船真誠)

#### 6.4.2.8 文章整形

文章の選定を行った際、GPT-2 では日本語として分かりづらい文章やプロンプトとして使用した「\n 背景史」などの意味をなさない物、URL などの文章として成立していないものが出力されることが分かった。そのため、出力される文章をそのままゲーム内に実装することは出来ないと考えた。改善の作業として、わかりづらい日本語の修正、文章として意味を成さないものの削除、より文章のテーマが伝わりやすくなるような文の追加を行った。下記に修正した 4 つの修正前と修正後の文章と、修正した箇所の詳細を記す。

**修正前**

だけど、努力が失敗した。彼女は自殺した。背景史:諦めきれず、私はコンクール上位者に憧憬を見せた。たとえ、その夢が甘かったとしても。背景史:私は彼女の夢を応援した。美術門を開き、傑作を送り出したあの人の夢を背景史:あの青い瞳は、描けば切り裂かれてしまいそうだ。私は彼女の夢を支えると決めた。彼女が笑いかけてくれるように。

**修正後**

私は彼女の夢を応援した。  
美術の道を拓き、傑作を送り出したあの人の夢を。

あの青い瞳は、描けば切り裂かれてしまいそうだった。  
私は彼女の夢を支えると決めた。  
たとえ、その夢が甘かったとしても。  
彼女が笑いかけてくれるように。

だけど、努力が失敗した。彼女は自殺した。  
諦めきれず、私は夢を継ぐものを見つけるためにコンクール上位者に彼女の夢を語った。

図 6.71 アリスの背景史における修正前と修正後の文章比較

図の修正点を挙げる。文章の最初の「だけど、努力が失敗した。彼女は自殺した。諦めきれず、私はコンクール上位者に憧憬を見せた。」という文を文章の終わりの文にすることで、応援していたけれど彼女は努力が実らず自殺してしまったというストーリーが伝わりやすいように変更した。「美術門を開き」という文は、彼女の今までにない美術という分野の新たな挑戦と解釈したが、美術門という単語は存在せずわかりづらいものになってしまうため、「美術の道を拓き」という文に変更した。「たとえ、その夢が甘かったとしても。」という文を「彼女は自殺した」という文の少し前に置くことで「私」が彼女が自殺してしまうことを予感していたのではと考えることが出来るようにした。「\n 背景史:」というプロンプトの入力に使用した意味のなさないものを削除した。

**修正前**

私は生まれ変わりたいと願った。私は、いつか生まれ変わりたいと思った。でも、現実 is 厳しい。たとえ、願いが叶っても、その先に地獄があるのなら、私は死に物狂いで前に進むのみだ。背景史:私はきつと、ライブで歌い踊り、人々の願いを叶える。私はそう信じている。私は彼女たちのライブを支える『凄い人たち』になるのだ。

**修正後**

私は生まれ変わりたいと願った。  
私はきつと、ライブで歌い踊り、人々の願いを叶える。  
だが、現実 is 厳しい。

私は、いつか生まれ変わりたいと思った。  
たとえ願いが叶ったその先に地獄があるのだとしても、死に物狂いで前に進むのみだ。

私は彼女たちとともにライブを支える『凄い人』になるのだ。  
私はそう信じている。

図 6.72 白うさぎの背景史における修正前と修正後の文章比較

図の修正点を挙げる。「私は、いつか生まれ変わりたいと思った。」という文の前に同じような意味の「私は生まれ変わりたいと願った。」という文がある。連続して同じ意味の文が出てくるのは日本語として読みづらいため、間に「私はきっと、ライブで歌い踊り、人々の願いを叶える。」という文を差し込むことで、より読みやすい文になるようにした。この変更によって、何度も「生まれ変わりたい」と願う「私」の気持ちが強調されるようになっている。さらに、「私は、いつか生まれ変わりたいと思った。」の前に「だが、現実には厳しい。」を持ってくることで、私は生まれ変わりたいと願っている。でも現実には厳しく生まれ変わるなどできない、それでも生まれ変わりたいという強い願いを表現した。「私はそう信じている。」と「私は彼女たちとともにライブを支える『凄い人』になるのだ。」という文の順番を逆にすることで、最後まで生まれ変わりたいという願いが変わることがないということを表現した。「\n 背景史：」というプロンプトの入力に使用した意味のなさないものを削除した。

#### 修正前

私の中に残ったあのおぞましいものを、彼女たちは恐れたのだろうか\n 背景史：私は、彼女たちの、そして、彼らの言葉を思い出す。助けて、助けて、助けて、と.\n 背景史：彼女たちは、あなたの言うことを何でも聞くという。その声は、あなたの元に届くのだろうか.\n 背景史：私たちは、元からなんでもできたのだ。女の子たちが、それを教えてくれた\n 背景史：あなたの声が殺し屋にイジメられたとき、偶然通りがかった孤児院の手口が脳裏に焼き付いている。彼女は悪名高い殺し屋であり、その手口は他の人殺しに知られたくないものばかりであ

#### 修正後

私の中に残ったあのおぞましいものを、彼女たちは恐れたのだろうか。  
彼女たちの、そして、彼らの言葉を思い出す。

「助けて、助けて、助けて」と。

彼は、悪名高い殺し屋であり、その手口は他の人殺しに知られたくないものばかりである。私たちは元から何でも出来たのだ。その声はあなたの元に届くのだろうか。

図 6.73 チェシャ猫の背景史における修正前と修正後の文章比較

図の修正点を挙げる。「私は、彼女たちの、そして、彼らの言葉を思い出す。」という文には、読点が多く日本語として読みづらくなっている。さらに、その前の文章で、「私」という主語は登場しているため「私は、」をこの文からは削除した。「助けて、助けて、助けて、と。」では、「助けて」がくり返しになっていて、文が「と。」で終わるため、「助けて、助けて、助けて」はセリフだと解釈した。そのため鍵括弧で囲んだ。「女の子たちが、それを教えてくれた。」、「あなたの声が殺し屋にイジメられたとき、偶然通りがかった孤児院の手口が脳裏に焼き付いている。」の2文を削除した理由としては、入力に使用したプロンプトの性質上\n 背景史：の後に続く文は場面が異なる場合があり、これらの文は文章のテーマと少しずれてしまい、文章がわかりづらいものになっているため削除した。「彼女は悪名高い殺し屋であり、その手口は他の人殺しに知られたくないものばかりである」という文の「彼女は」が「彼は」に変わっている理由は、文章中に登場する殺し屋は「私」であり、今回この背景史が割り当てられたキャラクターがチェシャ猫であった。設定上チェシャ猫の性別は男であるため、「彼女」を「彼」に変更した。また、文末が、「ばかりである」と切れてしまっている。これは、GPT-2で出力をする際の一回の出力で出てくる文字数の上限を設定してい

たためそれを超えてしまい文の途中で切れてしまっている。そのため「ばかりである。」ときれいに終わるように修正した。「私たちは元から何でも出来たのだ。その声はあなたの元に届くのだろうか。」という文を、「私たちは」というのが「彼女たち」で「私」と「彼女たち」が昔から知り合っていたことを示唆して、「その声」が「助けて、助けて、助けて」というセリフであり、殺し屋であり非情な「私」に助けを求める声は届くのだろうか、「彼女たち」が考えてると解釈したため、解釈をわかりやすく伝えるために文章の最後に持ってきた。「\n 背景史:」というプロンプトの入力に使用した意味のなさないものを削除した。

#### 修正前

あのとき、こんな風に思えたのだろうか。「そんな顔をさせるくらいなら、いっそ死んだほうがマシだ」あの日、思ったのだろうか、もう母親の顔を思い出したくないと。n背景史:私は、せめて海の向こうの人々の思い出の一部になりたかったのだ。私は、あの悲劇の記憶に浸り、安楽椅子に腰を掛けてそう願ったのだろうか

#### 修正後

あのとき、こんな風に思えたのだろうか。

「そんな顔をさせるくらいなら、いっそ死んだほうがマシだ」と。

あの日、思ったのだろうか、もう母親の顔を思い出したくないのだと。  
せめて海の向こうの人々の思い出の一部になりたかったのだ。  
私は、あの悲劇の記憶に浸り、安楽椅子に腰を掛けてそう願ったのだろうか。

図 6.74 ハートの女王の背景史における修正前と修正後の文章比較

図の修正点を挙げる。この文章は、あまり修正の必要がなかったため最小限の変更にとどめた。修正前では、「そんな顔をさせるくらいなら、いっそ死んだほうがマシだ」の後に、そのまま文が続いていたが、文が長くなってしまったために「と。」を文の最後につけて文を区切り、セリフらしさを出しつつ文の長さの調整を行った。また、「あの日、思ったのだろうか、もう母親の顔を思い出したくないと。」の文は、前の文で同じような文の終わり方をしてしまっているため、少し修正して、「あの日、思ったのだろうか、もう母親の顔を思い出したくないのだと。」に変更した。「\n 背景史:」というプロンプトの入力に使用した意味のなさないものを削除した。

(※文責: 入船真誠)

### 6.4.3 マルチエンディング

#### 6.4.3.1 分析準備

前期では、正式なフォーマットを作成後、ベクトルを元に数値化を行い、属性間の相関係数を出力した。しかし、フォーマットを詳細に書くことで出力結果に良い影響を与えると考察したため、フォーマットの改定を行った。具体的な改定としては属性の追加、削除を行った。追加した属性として、「実際どうなったか」「なぜ帰れるのか(帰れないのか)」「なぜ帰ること(帰らないこと)を望んだのか」の3つを追加し、「後日譚の主語」、「状態」、「行動」を削除し、「後日譚のパターン」

として纏めた。「後日譚のパターン」として、「帰るパターン」と「帰らないパターン」に分け、「帰るパターン」では、「異世界でのことを忘れる」、「普通の日常に戻る」、「異世界での経験によって日常が変化する」、「異世界の仲間と現実で再会する」、「帰れたことを喜ぶ」、「異世界の経験を誰かに話す」、「もう一度異世界に戻ろうとするが、失敗する」に分け、「帰らないパターン」では「普通に暮らす」、「統治する」、「あきらめずに帰る方法を探す」に分け、当てはまるパターンを要素として記入した。この改定したフォーマットを元に、作品のデータを当てはめ直した。なお、作品は前期のデータに加え、選定条件は変えずに作品を探し、10 作品を追加した。具体的な作品名として、「信長協奏曲」、「クレヨンしんちゃん 嵐を呼ぶ アッパレ！ 戦国大合戦」、「泣くな、はらちゃん」などがあげられる。「信長協奏曲」は、主人公であるサブローが、戦国時代にタイムスリップし、織田信長として生きていく物語である。そして、秀吉に光秀とされ、追われる立場になる。その後、秀吉につかまり、家臣に戦いのない世界を託し、死亡する。気づいたら平原にいたサブローは現代に戻ったを気づく。時がたち、サブローと同じようにタイムスリップした人々が現代に戻り、サブローが皆に託した通りの世界になっており、涙した。この作品は、前期で設定した条件の1つ目である日常から非日常に遷移する物語を満たしている。2つ目の条件では、第80回ザテレビジョンドラマアカデミー賞最優秀作品にノミネートされているため、条件を満たしている。以上から選定対象作品とした。「クレヨンしんちゃん 嵐を呼ぶ アッパレ！ 戦国大合戦」は、主人公のしんのすけとその家族が戦国時代にタイムスリップし、幾多の困難を乗り越える。その後、しんのすけは死んだ又兵衛に託された短刀で金丁をうち、廉姫と別れを告げる。車に乗ると春日部に戻っていた。この作品は、1つ目の条件である日常から非日常に遷移する物語を満たしている。2つ目の条件では、第6回文化庁メディア芸術祭アニメーション部門で大賞を受賞しているため、条件を満たしている。以上から選定対象作品とした。「泣くな、はらちゃん」は、現実世界にいる越前と、漫画の世界にいるはらちゃんの恋愛物語である。物語の終盤で、漫画の世界にいる越前さんは元の世界にいらなくても自分がいなければいいと考えてた。しかしはらちゃんに説得され、はらちゃんと共に現実世界へ出る。その後、はらちゃんはお世話になった人に別れを告げ、自分の世界が好きであるため、漫画の世界へ帰る物語である。この作品は、1つ目の条件である日常から非日常に遷移する物語を満たしている。2つ目の条件は、第76回ザテレビジョンドラマアカデミー賞において、最優秀作品賞を受賞しているため満たされる。以上から選定対象作品とした。このように追加した作品を、フォーマットに当てはめ再度 KJ 法を行った。結果として、「なぜ帰ることを望んだのか」に、対比関係があることが分かった。例として、「自分のため」か「他人のため」や、「主人公の行動が積極的」か「主人公の行動が消極的」、「主人公が感情的になった」か「主人公が何か行動をしたい」のような対比関係が6つ存在することが分かった。属性の関係性を調べるために相関係数を用いるため、「なぜ帰ることを望んだのか」を6つのベクトルで表現し、「-1, 0, 1」の3つの数値で表現した。

(※文責: 幸崎駿祐)

#### 6.4.3.2 相関係数

前期では、属性間の関係性を表すために相関係数で分析することにした。改定前のフォーマットを元に作成したベクトルを表としてまとめ、Excel で csv ファイルに変換した。そして、変換した csv ファイルを R という統計解析向けのプログラミング言語に入れ、実際に相関係数を出力した。後期では、改定したフォーマットを元に作成した新たなベクトルを用いて、可視化表現をし直した。そして同じ手順で相関係数を出力した。しかし、相関係数の数値に変動は発生したが、変動

が微小であり新たに追加したベクトルに関しても、前回の相関係数と似たような出力になった。また、相関係数は属性間の関係を示すため1つの属性に対し1つの属性の出現確立を調べるため、複数の属性に対し1つの属性がどのような関係を持つかは調べることができなかった。

(※文責: 幸崎駿祐)

#### 6.4.3.3 分析方法

先で述べたように、相関係数では、複数の属性と1つの属性との関係性が調べられないことがわかった。そのため、別の分析手法を取る必要があり、因子分析を行うことにした。因子分析は、外的規準の無い大量データから共通因子を見つけ出し、データの背後にある少数の要因を統計的に推測する方法である。因子分析を用いることで、エンディングのよくある属性の組み合わせパターンを模索でき、それぞれのパターンにおいてどの属性が強く影響しているかが分かる。以上の点から因子分析を採用した。分析をするにあたり、可視化表現した表を csv ファイルに変換し、相関係数と同じように R という統計解析向けのプログラミング言語を用いて因子分析を行った。

(※文責: 幸崎駿祐)

#### 6.4.3.4 分析結果

R で因子分析をする方法として、まず、R のバージョンを 4. 1. 3 で使用した。その後、R のパッケージから因子分析用のための「psych」のパッケージをインポートした。パッケージのインストール後、因子分析を行った。R を用いて因子分析を行った結果、図で示す通り、6つの因子があることが分かった。なお、今回は因子分析で出た因子負荷量が 0. 25 以上または-0. 25 以下のものを強く働いているとして採用している。1つ目の因子は「可」、「実際に帰った」、「自身（感情）」、「確保」、「積極」、「感情」の属性が含まれていることがわかった。2つ目の因子は「希望する」、「実際帰った」、「自身（望む理由）」、「行動」、「恋愛なし」の属性が含まれていることがわかった。3つ目の因子は「可」、「正の感情」、「興奮」、「他人（帰れる帰れない理由）」、「他人（帰るまたは帰らないを望む理由）」、「感情」の属性が含まれていることがわかった。4つ目の因子は「決意」、「死」、「他人（帰るまたは帰らないを望む理由）」、「積極的」、「恋愛あり」の属性が含まれていることがわかった。5つ目の因子は、「正の感情」、「自身に向けた感情」、「日常が変わる」、「生きる」の属性が含まれていることがわかった。6つ目の因子は、「興奮」、「目的の達成」、「生きる」、「能動」、「積極的」の属性が含まれていることがわかった。以上の6つの因子にそれぞれ違った属性が存在していることが分かった。それぞれの因子を分かりやすくするために、1つ目の因子から順に「仲間のために帰還するエンディング」、「自分のことを優先的に考え帰還するエンディング」、「仲間を思いやるエンディング」、「恋愛があり、誰かが死んでしまうエンディング」、「日常が変わる幸せ要素を含むエンディング」、「積極的に何かを成し遂げるエンディング」とした。しかしながら、これら6つの因子のうち、4つが現実世界に帰るのかそれとも帰ることなくとどまるのかを示す因子負荷量が強く働いていなかった。そこで、決定木アルゴリズムを作成し、どの因子が決まるのかを明確にするとともに、因子負荷量が強く働いてない属性をいくつか追加するシステムを開発した。

(※文責: 幸崎駿祐)

#### 6.4.3.5 アルゴリズム作成

決定木アルゴリズムを作成するにあたり、最初に因子分析で判明した6つのパターンを出力するアルゴリズムを作成した。なお、このアルゴリズムには、エンディングの要素を配列とした「element」を配置している。このアルゴリズムでは、5つの質問を用意し順番に質問を問い、回答を「Y」か「N」を入力し、「Y」を回答したところで因子を出力するものである。なお全て「N」の場合でも該当する因子を出力するようにした。質問の内容として、「帰ることを希望しますか」、「帰る手段はありますか」、「帰る帰らない時の感情は主人公に対するものですか」、「感情は良い方向のものですか」、「恋愛はありますか」と設定した。「帰ることを希望しますか」が「Y」の場合、「自分のことを優先的に考え帰還するエンディング」。「帰る手段はありますか」が「Y」の場合、「仲間のために帰還するエンディング」。「帰る帰らない時の感情は主人公に対するものですか」が「Y」の場合、「日常が変わる幸せ要素を含むエンディング」。「感情は良い方向のものですか」が「Y」の場合、「仲間を思いやるエンディング」。「恋愛はありますか」が「Y」の場合、「恋愛があり、誰かが死んでしまうエンディング」。全ての質問に「N」と答えた場合、「積極的に何かを成し遂げるエンディング」とした。続いて、6つのパターンを出力するアルゴリズムに、因子負荷量が弱い要素を付け加えるアルゴリズムを作成した。最初に、各パターンに当てはまる作品を分析した作品の中から探し、各パターンで現れた強い因子負荷量以外の要素を比較し、確率を求めるマルコフ連鎖を適用した。しかしながら、パターンに当てはまる作品が極端に少なく、確率を求めることが難しいと判断したため、マルコフ連鎖の適用を断念し、ランダムで因子負荷量が弱い要素を付け加えることにした。ランダムで要素を付け加えるため、パターンとの間に矛盾が生じる場合がある。例として、「自分のことを優先的に考え帰還するエンディング」の要素に「実際に帰った」とあるが、ランダムで要素を出す際に「帰ることは不可」と出力された場合、矛盾が生じてしまう。このような矛盾が生じないように、パターンが決まった際に、パターンと矛盾が生じるような要素を配列のelementから除外するようにした。最後に、現実世界に帰るのか、それともとどまるかがわからない4つのパターンに対して、「帰ることが可能か不可能か」、「帰ることを希望するのか希望しないのか」、「実際に帰ったのかもしくは実際に帰らなかったのか」の3つの要素を必ず出力するように変更を加えた。なおこの3つの要素が1つでも入っている場合は、その他の2つの要素を出力するように調整を行った。以上の形で因子分析で出力されたパターンをエンディングのプロンプトとして入力しやすいように要素を出力する決定木アルゴリズムを作成した。このアルゴリズムを用いて、各パターンごとに10回出力を行った。

(※文責: 幸崎駿祐)

#### 6.4.3.6 文章の自動生成

因子分析結果やアルゴリズムで出力した内容をもとにマルチエンディング文章の自動生成を行った。自動生成の流れを以下の図に示す。

まず、アルゴリズムで出力した属性の組み合わせパターンをもとに、背景史の項目で先述したGPT-2に入力するための文章を執筆した。ひとつの属性の組み合わせパターンにつきひとつのエンディングシナリオを執筆したため、エンディング因子6個×約10個の属性の組み合わせで、約60個のエンディングシナリオを執筆した。入力文章の執筆については次章で詳しく説明する。その後執筆した文章をGPT-2に入力し、各エンディング因子に対応するテキストをそれぞれ数10個出力した。出力したテキストはそのまま使用することができなかったので、いくつかの文章を抽

## GPT-2で文章生成（マルチエンディングの自動生成）

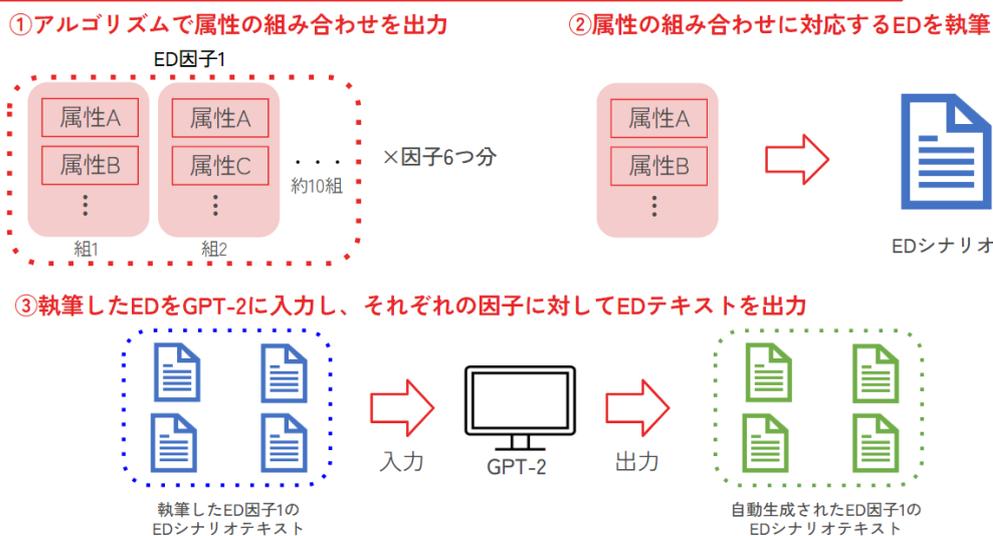


図 6.75 エンディング自動生成の流れ

出し組み合わせで最終的なエンディング文章を作成した。具体的な文章作成については次々章で説明する。このようにして因子分析における各因子につきひとつのエンディングシナリオ、つまり合計 6 つのエンディングシナリオを生成した。

（※文責: 吉井史夏）

### 6.4.3.7 入力用プロンプト作成

マルチエンディング作成においても、背景史と同じように GPT-2 を使用して文章生成を行った。また、求めている文章に近い形の出力を得るため、こちらも背景史と同じようにプロンプトエンジニアリングの手法を採用した。方針として、まずアルゴリズムで出力した属性の組み合わせと、その属性の組み合わせに対応する、人手で執筆したエンディングの内容を入力することを考えた。実際の入力内容の一部を以下に示す。

入力：現実世界に帰ることは可能か: 現実世界に帰ることは可能 \n 現実世界に帰ることを希望するか: 現実世界に帰ることを希望する \n…(省略)…エンディング: アリスは不思議の国で自らが周りの意見に流されず、自分の意思をもって発言し続けたことに対し、自信を持つ。チェシャ猫にも背中を押されたことで、現実世界に帰り、自分自身の力で状況を変えることを決意する。そのはじめの一步として、女王の理不尽な態度にも一步も引くことなく反論し、結果的に死刑を下される。その後、アリスは現実世界で目を覚ます。 \n 現実世界に帰ることは可能か:…

上に示した通り、各属性を「\n」で区切り入力した後、執筆したエンディングの内容を入力した。また、同じエンディング因子の異なる属性の組み合わせと、それに対応するエンディングの内容を続けて入力した。この形式では、エンディング文章の内容を入力しただけで、実際ゲームに使用できる形でエンディングを執筆していないため、出力もそのまま使用できる形ではないものになってしまった。また、出力された内容を解釈し、人手で執筆し直すという方法も考えたが、常識

的ではない内容の出力が多く、執筆し直すのには時間がかかり困難だと考えた。そこで、属性に対応するエンディング文章を、実際に使える形で執筆するという案が考えられた。具体的な入力内容を以下に示す。

入力：現実世界に帰ることは可能か: 現実世界に帰ることは可能 \n 現実世界に帰ることを希望するか: 現実世界に帰ることを希望する \n…(省略)…エンディング: 私はチェシャ猫を信じることにした。彼はずっと私に真実を教えてくれていたから。きっと女王に処刑されることが、現実世界に帰る条件だ。…けれど。「ごめんなさい、チェシャ猫さん。私、現実には帰らない」「私はこの世界を放っておくことができない。自分の意思をなくして女王の言いなりになっている住民たちを助けたいの。だから」…

事務的な内容を示すだけだった後半の文章が、アリス目線の一人称での文章となったことで、実際に使用できるような表現になったことがわかる。この入力を行ったところ、GPT-2 が実際に使用できそうな表現を以前より数多く出力するようになった。ただ、始めに属性の集合を入力すると、出力がその表記に引っ張られてしまうことがあった。その具体的な例を以下に示す。

出力：でも私は、チェシャ猫を信じる。ここにいるのは偽りの彼だけかもしれない。でも私は現実に帰れる。私は現実世界に帰れる \n 現実世界に戻ることは不可能ではない。ここにいるのは偽りの現実であって現実になりえない。

表現としては見えそうだが、「\n」が含まれており入力の表記に引っ張られてしまっていることがわかる。また、「でも私は現実に帰れる。私は現実世界に帰れる」という文章は、「現実世界に帰ることは可能か: 現実世界に帰ることは可能」という属性の入力に影響を受けており、そのまま使う文章としては不自然であるといえる。そこで、属性の集合を入力から取り除き、執筆したアリス目線のエンディング文章のみを入力することとした。また、入力の最後に、ほしい出力に合わせた書き出しの一文を付け加えることで求める出力が得やすくなることを発見した。具体例を以下に示す。

入力：私はチェシャ猫を信じることにした。彼はずっと私に真実を教えてくれていたから。きっと女王に処刑されることが、現実世界に帰る条件だ。…けれど。「ごめんなさい、チェシャ猫さん。私、現実には帰らない」「私はこの世界を放っておくことができない。自分の意思をなくして女王の言いなりになっている住民たちを助けたいの。だから」…(一部省略)…「決めた。私はチェシャ猫さんと一緒に現実に帰る」「…本当にいいのかい?」「うん。

出力：私は現実で、チェシャ猫さんとまた会いたい。だから」「…ありがとう、アリス」…

入力から属性の集合が削除されていることがわかる。また、入力の最後に台詞のような形の文を入力したため、それに続く形で出力がされていることがわかる。このように、ひとつの属性の組み合わせにつきひとつのエンディング文章を執筆したあと、同じ因子の物を続けて入力し、最後に書き出しの一文を付け加えるという形で全6因子分の入力を行った。

(※文責: 吉井史夏)

#### 6.4.3.8 出力と選定，文章整形

前章のような方針で作成した入力用の文章を GPT-2 へ入力し，因子ごとに幾つかの出力を得た。その後，得られた出力から実際に使用する部分を抽出する作業へと移った。抽出する際には，内容の構成として使用するものと，表現として使用するものに分けてそれぞれ抽出した。抽出の条件としては，構成として使用するものは内容が因子に合っているもの，表現として使用するものは内容が因子に合っている，かつ表現が印象的だと判断したものとした。後に人手で解釈し，執筆し直すため，日本語の表現としての間違いは気にしないこととした。以下に「仲間のために帰還する」エンディング因子における抽出例を示す。

抽出例 1（構成として使用するもの）：彼がいてくれたから、私は今現実世界でも生きていける  
\n だから。「決めた。私は私で現実で生きよう」\n 現実世界と現実の扉が開く。そこに居たのは私だった\n 彼だった。

抽出例 2（表現として使用するもの）：あなたは今でも、あなたが信じられないでいた私たちを信じてくれた。

例を見ればわかる通り，属性の集合を入力していた頃よりは自然な文章での出力となったものの，抽出した文章をそのまま使用することはできない。そこで，自動生成された内容を解釈し，書き直す必要がある。抽出例 1 では，「私」は「彼」の影響で現実世界へと帰ることを決心している。そして現実世界への扉が開いたとき，「私」は「彼」と再会する。この「私」をアリス，「彼」をチェシャ猫だと解釈し，この流れをエンディングとして採用した。抽出例 2 では「あなたは今でも，あなたが信じられないでいた私たちを信じてくれた」という言い回しが印象的であったため採用した。このように構成として抽出した部分をもとにシナリオを組み立て，表現として使用するものを組み合わせて最終的なエンディングシナリオを作成した。以下に，実際に作成した「仲間のために帰還する」エンディングシナリオの一部を記述する。

私は不思議の国に来てからのことを思い返す。私はこの国で、チェシャ猫に助けられた。チェシャ猫は初めからきっと私を信じてくれていた。私は彼を信じていなかったのに。…(省略)…「決めた。私は現実で生きる」

気が付くと私たちは真っ白い空間にいた。隣にはチェシャ猫がいた。彼が私に微笑みかける。「ありがとう、アリス」…

抽出例 1 をもとにした流れであること，そして抽出例 2 の表現が「チェシャ猫は初めからきっと私を信じてくれていた。私は彼を信じていなかったのに。」という形に変わって使用されていることがわかる。このように出力を抽出し，人手で解釈してつなぎ合わせ，1つの因子につき1つ，合計6つのエンディングシナリオを完成させた。

(※文責: 吉井史夏)

## 第 7 章 成果発表

### 7.1 概要

中間発表では、プロジェクト全体で 15 分の発表を行った。しかし、成果発表の際には、各班の成果を丁寧に詳しく説明するため、前半数分をプロジェクト全体の説明、後半を班ごとにブースを分けて、視聴者には気になる、班の発表を視聴していただく形式をとった。この形式では、すべての班の発表を視聴するのは困難であるが、各班の成果内容から検討した結果、各班で丁寧に発表していく形式が無難だと判断した。視聴者からのフィードバックでは発表技術が 10 段階評価で、平均 7.7、発表内容に関しては、平均 8.5 という数値であった。フィードバックコメントでは、各班の発表内容が詳しく説明されていることに対して評価が高く、おおむね好評であることから、今回の形式が成功したことがうかがえる

(※文責: 小林裕汰)

### 7.2 制作物

#### 7.2.1 ポスター

本プロジェクトの成果発表用のポスターは、中間発表用に制作したポスターのデザインや大まかなレイアウトを踏襲した。成果発表会では各班が自身のブースで発表を行う方式を採用したため、ポスターは各班のブースでどのような発表がされるのかイメージできる構成にする必要があった。下図に期末発表用に制作したポスターを記載する。

## Creative AI

AI と作るゲームの世界 : Alice in the ideal world

**システム班 / team: System**

- 新屋拓海 Takumi Shinya
- 田中裕人 Yuto Tanaka
- 本新斗悟 Togo Motomae
- 住吉聖也 Seiya Sumiyoshi

**視覚班 / team: Visual**

- 坂上優 Sui Sakagami
- 藤原佳 Koi Hozumi
- 畑井梨里衣 Rirri Hatal

**音響班 / team: Audio**

- 小林祐汰 Yuta Kobayashi
- 菊地隆雄 Ryoiga Kikuchi
- 高橋昂大 Kota Takahashi

**物語班 / team: Scenario**

- 坂本珠濤 Jurin Sakamoto
- 志保結城 Shunsuke Kosaki
- 入舩真誠 Masato Irfune
- 西村秀幸 Hideyuki Nishimura
- 吉井史夏 Fumika Yoshi

**教員 Professors:** 村井 浩 Hajime Murai / 迎山和司 Kazushi Mukayama / 平田 圭二 Keiji Hirata / 田柳 恵美子 Emiko Tayanagi / 吉田 博則 Hironori Yoshida

**プロジェクト概要 Project Overview**

このプロジェクトは人工知能を用いたゲーム制作を通じて、人工知能による創造的なコンテンツの自動生成の可能性を探る。ゲームは「物語」「視覚」「音響」などの要素を統合して、AIが生成したコンテンツを統合してゲームを作成する。このプロジェクトは、AIが生成したコンテンツを統合してゲームを作成する。このプロジェクトは、AIが生成したコンテンツを統合してゲームを作成する。

This project pursues the possibility of automatic generation of creative content by artificial intelligence through the creation of games using artificial intelligence. Games contain various creative elements such as "story", "visuals", and "audio". By introducing artificial intelligence technology suitable for these elements, we will create a game that integrates the automatically generated contents by dividing into four groups: story, visual, audio, and system.

In the first semester's activities, the game concept was decided to be "an open world game based on the theme of Alice in Wonderland" and each group prepared learning materials and answers for the target auto-generated contents and implemented the basic game system in the second semester activities. The game title was decided to be "Alice in the ideal world", and each group prepared the learning material, generated contents, and integrated it into the game system.

**グループ活動 Group Activities**

**今後の展望・活動 Future Prospects, Activities**

システム班: システム班は、ゲームの動作を安定させるための最適化作業を進め、最終的に3Dモデルから生成されたゲームの動作を安定させることに成功した。また、ゲームの動作を安定させるための最適化作業を進め、最終的に3Dモデルから生成されたゲームの動作を安定させることに成功した。

視覚班: 視覚班は、ゲームの視覚的な魅力を高めるために、3Dモデルから生成されたゲームの視覚的な魅力を高めることに成功した。また、ゲームの視覚的な魅力を高めるために、3Dモデルから生成されたゲームの視覚的な魅力を高めることに成功した。

音響班: 音響班は、ゲームの音響的な魅力を高めるために、3Dモデルから生成されたゲームの音響的な魅力を高めることに成功した。また、ゲームの音響的な魅力を高めるために、3Dモデルから生成されたゲームの音響的な魅力を高めることに成功した。

物語班: 物語班は、ゲームの物語的な魅力を高めるために、3Dモデルから生成されたゲームの物語的な魅力を高めることに成功した。また、ゲームの物語的な魅力を高めるために、3Dモデルから生成されたゲームの物語的な魅力を高めることに成功した。

図 7.1 成果発表用ポスター

本プロジェクトでは、各班で制作したゲームコンテンツを1つのゲームに統合する手法で開発を行った。しかし、この開発態勢では各班が最終的にどのような成果物を獲得して、どんなゲームが完成したのか不透明になることが懸念された。そこで、期末発表用ポスター制作過程の初期工程で、各班の成果物や今後の展望をまとめることと並行して互いの成果物の詳細を確認した。これにより、ゲームにおける互いの成果物の関連性を再認識することができた。

図 7.1 のポスターは、図 4.1 の中間発表用ポスターと比べて成果物の画像を多めに配置した。これは、中間発表時に比べて各班の成果物が格段に増えたことと、各班の活動内容や成果を聴衆に視覚的に提示する狙いがあったためだ。また、全体的に文字のサイズも小さくしている。これは、今回のポスターに求められる役割が観衆の関心を惹くことであるためだ。詳細な各班の活動内容についてはポスターのテキストではなく各班のブースで発表することが決まっていたため、テキストの重要性は画像に比べて相対的に低くなった。文字サイズの縮小は可読性を損なわないように意識したほか、英語を日本語よりも小さめの文字サイズに設定した。これは英語の高い可読性を利用した工夫である。

ポスターの最上部にあるプロジェクト名や班名、班員や教員の方々の記述は中間発表用のポスターと同様の記載である。変更点としては、プロジェクト名の下にキャッチコピーとプロジェクトで制作するゲームのタイトル、ポスターの右上にゲームのロゴを追加した点である。これまでは、プロジェクトの活動や方針を一言で表現できるキャッチコピーがなかった。中間発表や HAKODATE アカデミックリンク 2022 での発表時も「Creative AI AI × ゲーム」のような紹介しかなかったことが反省点となり、成果発表用のポスターを制作するタイミングでキャッチコ

ピーを作成した。キャッチコピーの案は複数あったが、メンバー全員の投票により「AI と作るゲームの世界」に決定した。キャッチコピーはポスターのプロジェクト名の下に配置したほか、発表スライドにも記載するなどプロジェクトメンバーと観衆の両方に浸透するよう努めた。また、ゲームタイトルのロゴに関しては視覚班がいくつか候補を挙げて、最もメンバーに好評だったものを採用した。

ポスター上部にあるプロジェクト概要のテキストは、中間発表用ポスターに比べてさらにわかりやすく的確な表現に変更した。また、中間発表時は決まっていなかったゲームタイトルについても言及した。

ポスターのグループ活動を紹介するテキストは、中間発表時の反省を踏まえて、あらかじめ指定した文字数で各班の担当者に依頼した。しかし、指定した文字数でも文章量が多かったため、情報密度を損なわないように添削した。ポスター中央のダイアグラムは中間発表用のポスターと同じく、各班の連携を表現できるように努めた。ポスター制作当初はこのダイアグラムが中間発表用のポスターと同じ色合いで光沢がなかった。その後、ダイアグラムをもっと強調してほしいという要望をメンバーから受けて、光沢や色の彩度を調整することで立体感を演出した。各班のテキストの周りには、各班に対応した成果物の画像を添付した。これによって、各班がどんな活動を通してどんな成果を得たのか一目でわかる構成になった。

ポスター下部にある今後の展望・活動のテキストは、上述のグループ活動のテキストと合わせて依頼した。ここには各班が1年間のプロジェクト学習で得た成果物から考えられる今後の展望や活動方針を記載した。

成果発表用ポスターは、中間発表用ポスターとは異なり、各班の活動に興味を持ってもらえるような構成を意識して制作した。これは、中間発表と成果発表で発表方式が変更されたためだが、ポスター制作に着手する前の決定事項だったため対応することができた。また、中間発表用ポスターと比べて、成果発表用ポスターでは各班の成果物の画像も大量に添付することができた。これは、ポスター制作を始める前に各班に画像を大量に用意してもらうように依頼していたためだ。豊富な画像データのおかげで円滑に制作を進められたほか、採用できなかった画像をプリントして発表会場の装飾に使うことで観衆の興味を惹くことができた。今回のポスター制作では、徹底した事前準備とポスターの意義の再確認を行ったことが最も重要だった。

(※文責: 坂本珠凜)

## 7.3 評価

発表技術についてコメントは49件の回答をいただいた。参考になるコメントの一例を以下に示す。

- ・スライドが見やすかった。説明が詳しくされていた。すぐ隣に音声を流すブースがあるので、発表の声をもっと大きくするか、発表ブースの配置にもう少し余裕を持たせられるとさらに良くなると思った。

- ・全体的に声や音が小さく、聞き取りづらかった。スライドはとても見やすかった。

- ・4つの班がバラバラで一個しか聞けないのが悲しい体育館で周りの声もするのでほとんど聞こ

表 7.1 成果発表会の評価

成果発表会でのアンケートでは 60 名に評価をしてもらった。中間発表会の際のアンケート数は 47 名であったため、中間発表よりも注目度が高かったのではないかと推測する。アンケートに答えてもらった 60 名のうち 4 名が本学の教員であった。また、外部からお越しいただいた評価者は 6 名で、残りの 50 名が本学の学生だった。評価の平均は、発表技術に関しては中間発表時の 8.94 よりも低い 7.73 であった。一方、発表内容に関しては中間発表時の 8.38 よりもやや高い 8.5 という結果になった。

評価点	発表技術	発表内容
1	0	0
2	0	0
3	0	0
4	3	1
5	5	1
6	2	1
7	12	10
8	19	15
9	11	15
10	8	17
平均	7.73	8.5

えなかった

- ・発表に興味のある班に分けて聞けるのはとても良いと思った。ただし原稿を見過ぎな気がしました。

- ・内容が整理されていたので、聞きやすかったです。スライドの切り替えがよりスムーズだと話が入ってきやすいと思いました。

- ・班のブースが近すぎて動きづらくて聞きにくかった

中間発表の時よりも改善点を記入してくれているコメントが多くみられた。指摘にもある通り、体育館という広い場所での発表を活かして切れてない部分が多々あった。中間発表時は全体を通してのみの発表で、狭いスペースでの発表を行っていた。そのため、体育館の広さを考慮できず、全体として声が小さい発表になってしまった。

発表技術についてコメントは 48 件の回答をいただいた。多くのコメントが発表を行なった班に関する内容なので、詳細に関しては後述する。

## 7.4 課題と検討

### 7.4.1 システム班

まず、成果発表について位置的な課題が挙げられる。プロジェクトの発表方法はコの字型で行い、システム班は奥に位置していた。全体の発表から班の発表に移行した時、観覧者はコの字の中心から一番近い視覚班や物語班のブースに行くことが多い印象だった。よって、奥まった位置にあるシステム班の発表ブースまで足を運び辛くなっていると感じた。次に、発表の時間配分と内容に関して課題が見つかった。後期までの作業や成果物について、まとめて発表する予定だったが、話す内容の情報量が多くなってしまい早口で発表することとなった。聞き手側の理解するスピードをあまり考慮せずに発表をする形となってしまったので、聞き手が理解できない部分があったと感じる。また、ゲームを実際にプレイすることで、成果物を見せながらゲームの統合やギミックの自動生成について発表したが、後期に製作した部分を少ししか見せることができなかった。

検討として、発表の位置的な問題に関しては、観覧者が各班の発表ブースへ移動する時間をしっかり設けられていたので、システム班が自ら観覧者を呼び込み、アピールすべきだったと考える。また、コの字形の入り口部分で人の流れが滞留していたので、観覧者をシステム班のブースへと誘導させることも必要だったと感じる。発表内容に関しては、かなり内容を省き簡略化したが、もう少し伝えたいことを取捨選択すべきだった。詳しい説明はスライドなどに書き込み、実際に話す内容の文章量は減らすことで、聞き手にとっても聴き易い発表をするようにすべきだった。ゲームプレイをしながらの発表についても同じであり、プレイする部分を最初から通して実行してしまったので、見せたい部分まで見せることができなかった。プレイする部分をダイジェストにしたり、後期に製作した部分からプレイしたりするなど、ゲームプレイに関しても内容を取捨選択し省略する必要があった。

(※文責: 住吉聖也)

### 7.4.2 視覚班

視覚班の成果発表はパネルとスライドを使用した。パネルには自動生成をもとにデザインしたキャラクターイラストを印刷し、スライドはテレビモニターに表示させた。発表内容は、視覚班全体の目標を説明したのち、2次元画像生成と3次元モデル生成の2つの成果を紹介した。スライドの製作は視覚班のメンバー3人全員でGoogleスライドを使用した。成果発表には、高校生や一般の来客者など、学外の方も参加することが予想されたので、専門的な用語を少なくし、図をメインにスライドと発表内だけでプロジェクト学習の成果が理解できるよう意識をした。成果発表の際に視聴者からいくつかの質問をいただくことがあった。いただいた質問としては「AIが制作したデザインや、AIが学習の際に利用した画像の著作権に関する質問」と「AIをもとに生成したデザインにオリジナリティが存在するのか」というものがあった。この2つの質問は創作AIでは話題にあげられることが多い問題であり、視聴者からの質問が予測できたのにも関わらず、スムーズな受け答えを出来るよう意見共有をメンバー内で行っていなかった。想定される質問の答えを前もって共有をするべきであったらう。

(※文責: 穂積佳)

### 7.4.3 音響班

成果発表では、二つの問題点が挙げられる。

一つ目は、発表における位置に問題があった。今回の発表では、コの字でのブースで行い、音響班はブースの角に配置された。実際に発表してみると、物語班や視覚班のブースに聴衆者が集まりやすく、音響班ブースに聴衆者が集まりにくいことが分かった。また、音響班はデモンストレーションを行うために、スピーカーを用いて発表を行ったが、隣接する視覚班の発表に影響が出ることも考え、度々音量を調整する必要があった。位置の検討として、発表では他のブースに影響がないように配置する必要があると考える。また、聴衆者を集めるために、自身のブースをアピールする必要があったと考える。

二つ目は、発表における仕方に問題があった。発表は10分間で行われたが、前半の発表において約6~7分程度で速めに発表を終わらせてしまった。発表が短くなってしまった原因として、事前に原稿を用意し、発表では原稿の通りに発表していたため、アドリブが少なく早く終わってしまったのではないかと考える。また、発表において成果物のデモンストレーションも行ったが、デモンストレーションを行う時間を詳細に決めていなかったため、発表が短くなってしまったと考える。

発表の検討として、原稿を用意した上でアドリブを加えたり、デモンストレーションにおいても詳細に時間を決めつつ、アドリブを加える必要があると考える。

(※文責: 菊地陵雅)

### 7.4.4 物語班

成果発表における、物語班の課題と検討について、発表の仕方、発表内容の2点を焦点に課題と検討を記述する。

まず、発表の仕方についてである。成果発表評価では、「声が小さい」「原稿を見すぎている」という旨の内容が多く散見された。物語班では、台本を作っていたが、その多くに頼ってしまったことや、台本を見るために聴衆の方向を見ずに声を出す関係で声が届かなかったものだと考えられた。この点について、ある程度の話しの流れを覚えることや、スライドを見ることである程度、聴衆を見て話せるように練習することを検討している。しかし、中心に声が集まってしまうのは、ブースの配置の問題のため、この点については様々な配置を検討している。もう一つ、発表の仕方について、課題として実際に自動生成を行う様子を見てもらう際に多くの手間がかかると共に、一度接続が切れると復旧に時間がかかる点が挙げられる。前半の発表では、うまく機能していたが、後半では接続が切れてしまい、発表の内容を変更するに至った。今後の検討では、接続が一方で操作可能なアプリ等を用いて発表途中で問題が起きても即座にリカバリーできるような体制を整えることを検討している。

次に、発表内容である。成果発表評価では、マルチエンディングがゲーム上でどのような機能を果たすのか、そしてどのようにテキストの組み込みを行ったかを問われる評価が見られた。今回のスライドでは、大規模言語モデルを用いてどのように自動生成を行ったかに焦点を当てている。また、具体的な例も生成物の文章量から背景史の内容を引っ張り出しており、マルチエンディングと成果物の組み込みについて言及しなかったといえる。この点について、具体例の表示に背景史とマルチエンディングを交互に用いて実装例について表現することを反省としている。また、組み込

## CreativeAI

みの説明については、システム班との協力が必要であり、この点についてシステム班と協議する必要性があった。

(※文責: 西村 秀幸)

## 第 8 章 まとめ

### 8.1 プロジェクト全体の成果

人工知能の発達は目覚ましく、今年度は様々な人工知能を用いた自動生成システムが登場した。そのような中で、本プロジェクトでは人工知能を用いた自動生成による創造的な作業の実現可能性をゲーム制作というアプローチで検証した。

5月頭からプロジェクトを開始し、プロジェクトメンバー15人で具体的な成果物の方針を決定した。5月中旬からは以下の4つのグループに分かれて活動を始めた。ゲームのビジュアル部分のデザインや人工知能の自動生成を活用してキャラクターのビジュアル制作を行う視覚班、ゲームへの没入感を向上させるようなBGMを人工知能の自動生成を活用して制作する音響班、ゲームの世界観の設定や物語の執筆を人工知能の自動生成システムを活用して制作する物語班、そしてこの3つの班の成果物を最終的に統合し、ゲームの基本骨子やゲーム進行に必要なマップギミックを人工知能の自動生成を活用して製作するシステム班である。7月の中間発表会形式で行われ、プロジェクトメンバー全員で協力し、プロジェクト全体での発表を行った。夏休み期間中も各自で必要に応じて活動を行い、ゲーム制作に邁進した。後期の活動もグループごとに分かれて活動を行った。11月にはアカデミックリンクという、函館市内の8つの高等学校教育機関の学生が、普段研究している内容や成果などを発表する合同研究発表会にブースセッション形式で参加し、様々な反応を得られた。また、この時期にはゲーム制作が佳境に入り、必要に応じてグループ間で人員の移動を行い成果物の完成を目指した。12月の成果発表会も中間発表と同じ対面形式で行われた。プロジェクト全体での発表と、各グループの発表という分割形式での発表となった。

成果物である制作したゲームに関しては一通りプレイできるような完成度に仕上げることができた。現在細かい挙動の修正やバグの除去、操作性の向上などの最終調整を行っている。

(※文責: 小林裕汰)

### 8.2 各グループの成果

#### 8.2.1 システム班

システム班では、班員がそれぞれ作った「マップ」「UI」「キャラクター」「自動生成」の統合に加え、他班が作成したキャラクターやBGMや自動生成で作成したシナリオ等の成果物を「ゲーム」という形で実装できた。ゲームの完成度としては当たり判定の問題があることやUIの小さな修正が必要なものはあるものの大きなバグがなくストーリーはすべて完成しているので、完璧ではないがゲームとして動かせるものを実現することができた。

(※文責: 本前斗梧)

### 8.2.2 視覚班

視覚班は、物語班が用意したキャラクター像をもとに自動生成技術を用いてキャラクターデザイン、キャラクターモデル生成を行った。当初、物語班からリクエストがあったキャラクターは7体いた。視覚班ではこのキャラクターたちを「バックボーンがしっかりと設定されている主要キャラで人型であるキャラクター」と「人間とは違う造形をしたキャラクター」の2つに分け、それぞれ別のアプローチでキャラクタービジュアルの自動生成を行った。前者はテキスト入力から2次元画像を生成、それをもとに人の手でキャラクターデザインを完成させ、後者はテキスト入力から3次元モデルを生成、それを組み合わせてゲーム素材を制作した。今回の大きな目標は、シナリオをもとに、自動生成技術を用いて、キャラクターの造形をデザインすることであり、全自動とはいかなくとも、大まかな造形やほとんどのモデル生成を自動で行うことが実現できた。

(※文責: 穂積佳)

### 8.2.3 音響班

音響班では、ゲーム内でのイベントやフィールド遷移時に生じるBGM遷移の際に、ゲームシーンへの没入感を向上させるBGMをより滑らかに、シームレスにつなぐことのできる音響システムを作成した、そのために必要なデータやシステムも多数収集、制作した。実際にゲーム上で使用するフリーBGMやフリーSEの素材集めを行い、十分な量のデータを収集した。収集したフリーBGMに関しては、そのままの素材ではループ再生の際に違和感が生じてしまうため、ループ時の違和感を消す加工も行った。

そしてその収集したBGMデータを用いて、音響システムで生成するために必要なデータベースも構築した。音響システムで使用する6つの音響特徴量を抽出できるようシステムもデータベースに実装した。このデータベースの音響特徴量を用いて、BGMをよりシームレスに繋ぐことのできる音響システムを作成した。これにより、従来のゲームよりBGMの遷移を自然に行うことが可能になった。

(※文責: 小林裕汰)

### 8.2.4 物語班

物語班の成果を記述する。物語班では、背景史とマルチエンディングを統計の分析を用いて自動生成し、4つの背景史と6つのエンディングを生成した。加えて、これらの内容を本筋のストーリーに組み込み物語の執筆を行った。これらの成果物から、物語班の成果として、大規模言語モデルで自動生成したものを物語の執筆に組み合わせることが出来るということ、自動生成には未だ人手による調整が必要であることがわかった。また、他のモデルと比較してGPT-2の生成の特徴を明確にした。

まず、物語の執筆に自動生成の要素を組み合わせる点についてである。「6. 1. 3 自動生成したシナリオを組み込むにあたっての調整」にあるように、自動生成の要素によって作成途中の物語に新しい要素を与えていることがわかる。物語の伏線の決定や、物語のキャラクター性をある程度決定することが出来る。この点から自動生成した成果物は、物語執筆において発想の手助けになるほ

か、物語の展開を限定的な範囲で担うことが出来るということが成果としてわかった。

次に、自動生成には未だ人手による調整が必要であることがわかった。背景史、マルチエンディングの成果物作成には、文章の選定・整形の手順を必要としている。両者とも、言語モデルの出力結果がそのままでは文章として利用できないため行う必要がある。これは、大規模言語モデルそのものの問題であると考えられる。

以降では、大規模言語モデルに触れたことによる成果について、生成過程で扱ったモデルとその他の言語モデルを用いて比較する。今回の生成では、GPT2-japanese のモデルを利用して生成した。この GPT-2 では、文脈を持ったストーリーを生成する上でいくつかの問題が見えた。まず、文脈に沿った長文を生成できないことである。100 字を超えるような文章を作成する場合、文脈に関係ない単語や登場人物が登場し、その単語に引きずられる形で文章が変化する。また、瞬間的な単語によって文章を作成している性質上、長文になるほど初めのプロンプトの内容から逸脱した内容になる傾向が見られる。これは、GPT2-Japanese の後継モデルである、GPT3.5 にも同様の傾向がみられた。加えて GPT3.5 の場合は、物語展開よりもニュースやネット記事に近いような内容の出力することがあった、GPT2-japanese の学習データに比べて多くのデータを学習していることがうかがえる。ただし、物語の生成においてはこのバリエーションの高さが時に望む出力の内容からそれることがあり、この観点からはプロンプトの工夫等が必要であった。GPT-2 の他に我々が試した言語モデルとして chat-GPT を試した。こちらは、物語の生成を行う上で GPT-2 よりも高い精度を示した。言語モデルが GPT-2 よりも大規模な GPT-3 を用いている点からも、文章生成においては GPT-2 よりも自然な文章を生成した。長文生成については、途中で文章が止まってしまうことが多いため確認することができなかったが、短い文章であれば自然な文章を作成することが可能であった。例えば、「アリスは隕石に巻き込まれた」という内容から文章を作るよう命令した際に、「地球に落下したとアリスが思った。」という旨の内容から、後の文章で、実は落ちた場所は地球ではないというように物語としてオチの回収までを自然な文脈で行えるというような、高度な生成を行えることを確認した。ただ、物語を生成する上でいくつかの問題があることも確認した。プロンプトを「アリスは地上で隕石に巻き込まれた」という文章に変更する。この文章では、地上が恐らく地球上を指している可能性が高いことを想像できるが、chat-GPT は「隕石の衝突で地球に落ちたとアリスが考えた。」という旨の内容を返す。また、生成された文章は、初めのプロンプトから大きな変化が起きていないことから、文法の能力は向上しているが、文章の文脈の整合性をとれないという GPT-2 から続く問題は解決されていないのではないかと考えている。そのため、物語の作成には短文を組み合わせる手法が今でも現実的である。一方で、プロンプトへの理解はかなり上昇していると言える。今回、私たちが利用したプロンプトは、プロンプト入力で説明があったような生成したい特徴を備えた文章をそのまま入力する形式であったが、chat-GPT では、「この物語の続きを作って」というような自由度の高い形式で命令することができる。私たちの成果物が生成した直後のものを見ると、プロンプトの名残が残っておりこれらを人手で修正する必要があった。これは、プロンプトを入力した際にプロンプトの内容を GPT2 が理解できていないため発生したと考えられる。しかし、chat-GPT では、文章を用いて正確に命令内容を伝えることが出来るため、今回利用した GPT-2 に比べて命令への理解度は高かった。そのため、その点はとても進歩していると考えられる。

(※文責: 西村 秀幸)

## 8.3 今後の展望

### 8.3.1 全体

プロジェクト全体を通しての今後の展望については、いくつかの学会発表を行うことを検討している。プロジェクトメンバーから希望者を募り、それぞれがメインで担当した内容に合致した学会での発表形式で行う予定である。そのためプロジェクト全体の内容を学会発表するのではなく、プロジェクト内での活動のうち、特筆するものをいくつか選んでの学会発表となる。

(※文責: 小林裕汰)

### 8.3.2 システム班

システム班は、ゲーム作品のブラッシュアップとバグの修正・マップに沿ったギミック自動生成システムの改良の2点を行う。

成果発表会の時点では、ゲームを最初から最後まで問題なく遊べるところまで機能の実装と他班の成果物の統合を行うことができた。今後は背景史や音響システム・キャラクターモデルなど、他班から新規の成果物や修正されたものがあつた場合は、その成果物をゲーム内に統合、及び既存の成果物を修正する作業を行っていく。また、ゲーム内システムの機能追加やバグの修正、ゲーム内演出の改善などといったゲーム開発も、担当ごとにそれぞれ継続して行っていく。

また、ギミック自動生成については、設定したギミックの要素をランダムに組み合わせる配置するといったところまでしか開発が進まなかったため、今後は、分析するギミックデータを増加させることや分析手法の改善、Unity側でギミック配置方法の改善を行い、より自由度の高い自動生成システムの構築を目指す。

以上の2点の作業に取り組むことにより、成果物であるゲーム作品やギミック自動生成システムの完成度の向上に努めていきたい。

(※文責: 新屋拓海)

### 8.3.3 視覚班

キャラクターデザイン案の画像に関する今後の展望は、2つある。

1つ目は、シナリオからキャラクターの情報を抽出する作業の自動化だ。今回は人間の手で行ったが、AIがこの作業をできるようになれば、キャラクターデザイン制作はさらに効率化できるだろう。

2つ目は、textual Inversion を用いた独自の画風のファインチューニングを用いて出力された画像をキャラクターデザイン案に使用することだ。ファインチューニングをする目的は、AIが出力した画像に視覚班のメンバーの絵柄を取り入れるためだ。今回は、以下のような方法で試みたが採用までには至らなかった。独自の画像を Google Colab で学習をし、モデルを Stable Diffusion Concepts Library に登録した。「Google Colab ではじめる Textual Inversion | npaka | note」(URL: <https://note.com/npaka/n/n63dc6617a14d>) と「Stable Diffusion Concepts Library に独自のファインチューニングモデルを登録

する | npaka | note」 (URL: <https://note.com/npaka/n/nf7080c6bf4dd>) を参考にした。まず、縦横比が等しい 512 × 512 の jpg 画像を 5 枚用意し、google drive に保存した。前期にデータセットとして描いたもので、比較的配色が似ているものを選択した。Google Colab で Textual-inversion fine-tuning for Stable Diffusion using diffusers (URL: [https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/sd\\_textual\\_inversion\\_training.ipynb](https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/sd_textual_inversion_training.ipynb)) を実行しファインチューニングを行った。この際、モデルを Stable Diffusion Concepts Library に登録するために、Huggingface で書き込みに対応したトークンを取得し、使用した。ファインチューニングは約 3 時間程で終了した。このファインチューニングモデルは「AliceBeta」という名前で登録され、`¡Alice-style¡` というプロンプトで利用できるようになっている。このファインチューニングモデルを使用し、`image-to-image` にかけると独自の画像のような太めの手描きの線が特徴的な画像になった。参考サイトによると画像には共通のコンテキスト情報が含まれている必要があり、同じオブジェクトを異なる視点から移し、画風は配色とアートスタイルを一貫した方が良いとのことだった。しかし、用意した画像は同じキャラクターを描いたものではなく、描いた人も別々である。これらの点を改善してもう一度試したい。



図 8.1 訓練用に用意した 5 枚の画像



図 8.2 ファインチューニングモデルで `image-to-image` にかける前の画像



図 8.3 ファインチューニングモデルで `image-to-image` にかけた後の画像

キャラクターの 3 次元モデルに関する今後の展望は、キャラクターデザイン制作の効率化をという観点から、今回の製作過程で人間が行った箇所を自動化することだ。特に、数千種類もの 3 次元モデル素材から適切な素材の選択や配置を CLIP に判断させるシステムを開発することで、テキスト入力からゲーム素材として利用できるモデルデータ群を生成することが出来る。

(※文責: 畑井梨里衣)

### 8.3.4 音響班

今年度の音響班は、すべてのBGMをゲームイベント、フィールド遷移時などのタイミングにあわせてシームレスに変化していくインタラクティブミュージックを実現するという目標として活動していた。インタラクティブミュージックの実現は昨年度も行っていたが、ゲームの重要なボスとの戦いの際に使用するBGMのみという限られた状況のみをインタラクティブミュージックを実装していた。この昨年度の成果を参考に、今年度ですべてのBGMに対して実装することができた。特に音響特徴量はクロマベクトル、セントロイド、BPM、ゼロクロッシングレートの4つであったが、今年度ではこれに加えて、バンド幅、RMSという2つの要素を追加した。音響特徴量に関してはこれ以上の追加は複雑化するため難しいと思われる。

一方、今年度行いたいものとして、昨年度用いていた感情特徴量の追加を考えていたが、昨年度用いていたHevnerによる8つの印象語群を用いた感情特徴量をそのまま使用した。感情特徴量に関してはまだ追加や改善の余地があるため、来年度に期待したい。

その他にも今年度行いたいものであったが実現できなかったものとしては登場人物のテーマソングの実装、エフェクトの選択、曲の自動選択が挙げられる。

登場人物のテーマソングに関しては、登場人物の職業や年齢などの感情以外のパラメータを用いることで、その場面での登場人物にあったBGMを自動選択を目標としていた。音響特徴量をそのまま用いることもできず、具体的なパラメータの設定を行う余力がなかったため、来年度以降の活動に期待する。エフェクトの選択に関しては、より自動生成したインタラクティブミュージックの質を上げることも考慮していた。安易にエフェクトを実装しても逆に質を下げってしまう可能性もあるため、実装を見送った。曲の自動生成については、昨年度行っていたものであり、今年度もそれを参考に改良したいと考えていた。これも時間が足りず具体的な計画立案もできなかった。まだまだ人工知能の活用案はたくさんあるため来年度の活動に期待する。

(※文責: 小林裕汰)

### 8.3.5 物語班

物語班は、生成した成果物は完成している為、成果物のクオリティー向上を主な今後の展望としていく。

まず背景史は、三つの展望を考えている。一つ目は、キャラクターの属性に合わせた背景史を出力することである。今回の成果物は、プロンプトに文章を入力することで生成している。そのため、キャラクターの属性に合わせた生成は難しく、キャラクターの属性が乖離してしまう可能性が高い。これらを解決するため、今後、キャラクターの属性をプロンプトとすることで背景史を出力できる手法を追求していきたいと考えている。二つ目に出力の幅を増やしたい。今回の因子分析の結果では、大きく分けて4つの特徴が抽出された。しかし、抽出されたものは選定した作品の影響が大きく出ている、新しい発見となるような特徴的な因子の発見には至らなかった。そのため今後は、データベースの拡大を検討するとともに分類カテゴリの増強も考えている。まず、選定作品の影響は、データとする作品を増やすことで影響度合いを少なくするとともに本質的なトラウマキャラクターの因子を抽出したいと考えている。また、カテゴリの増強はトラウマを持つキャラクターの特徴についてもっと詳細にカテゴリを分類することで分析の際の特徴性をさらに細かに表現したいと考えている。三つ目に、さらに大規模な言語モデルを使用した生成を行いたいと考えて

いる。今回使用した GPT-2 には、学習データ数をさらに増加した GPT-3 や Google が発表した BERT 等耐規模言語による表現を用いて生成を行いたいと考えている。試作の段階では、これらのモデルが GPT-2 よりも長い文章に対応できることや、プロンプトへの理解度が高いということが分かっており、これらを用いてさらに長い背景史の生成を行うことを目標とする。以上3点を今後の展望とする。

マルチエンディングでは、二つの展望を考えている。まず、ファインチューニングを行い目標にあった文章が出力されるようにすることである。マルチエンディング班では、後期の反省にもあるように、成果物のファインチューニングが行うことが出来なかった。今回生成した成果物は、ストーリーを作成した本人がプロンプトを作成し入力としたが、それだけでは、雰囲気にあったものが作成されることは難しい。言語モデルそのものを物語の雰囲気にあったものに調整するとより物語に合うような生成物が生成出来る可能性が高まる。そこで今後の展望の一つとして、物語の雰囲気にあった文章の内容になるようにデータを決定しファインチューニングを行うことを展望とする。二つ目に、プロンプトを ED 手前まで入力することで出力が可能となるプロンプトを追求することである。今回のマルチエンディングのプロンプトは、執筆者本人がエンディングの内容を全て書いたものをプロンプトとしている。そのため、以上の作業を各パターン分用意しなければならず膨大な時間が必要になる。また、物語本体の内容を反映させてプロンプトを作成したいと考えており、上記の方法を用いることが良いと判断した。そのためには、どのような展開を経て ED に向かうのかというデータセットが必要となるため、そのようなカテゴリを増やしたデータセット作成の必要がある。以上2点を今後の展望とする。

物語の執筆では、背景史やマルチエンディングの生成結果に関する伏線をはりたいと考えている。今回の作業進捗では、生成した成果物を物語の本体に組み込むことは日程上難しく組み込むに至らなかったため、生成物を組み込んだ作品作りを今後の展望とする。

(※文責: 西村 秀幸)

## 参考文献

- [1] 松原仁, 佐藤理史, 赤石美奈, 角薫, 迎山和司, 中島秀之, 瀬名秀明, 村井源, 大塚裕子. コンピュータに星新一のようなショートショートを創作させる試み. 人工知能学会全国大会, 2D1-1 (2013)