公立はこだて未来大学　2018 年度　システム情報科学実習
グループ報告書

Future University Hakodate 2018 System Information Science Practice
Group Report

プロジェクト名
クリエイティブ **AI**
Project Name
Creative AI

グループ名
物語分析班
システム班
視聴覚班
Group Name
Story analysis Group
System Group
Audiovisual Group

プロジェクト番号/Project No.
15

プロジェクトリーダ/Project Leader
1016157 鈴木諒輔 Ryosuke Suzuki

グループリーダ/Group Leader
1016166 松浦史佳 Fumika Matsuura
1016140 高橋翔太 Shota Takahashi
1016137 城田晃希 Shirota Koki

グループメンバ/Group Member
1016157 鈴木諒輔 Ryosuke Suzuki
1016140 高橋翔太 Shota Takahashi
1016166 松浦史佳 Fumika Matsuura
1016200 寺島啓悟 Keigo Terashima
1016214 津沢慎吾 Shingo Tsuzawa
1016226 渡邊広基 Hiroki Watanabe
1016150 山田康貴 Kouki Yamada
1016068 袴田翔 Sho Hakamada
1016144 南部太雅 Taiga Nambu
1016151 吉田拓海 Takumi Yoshida
1016137 城田晃希 Shirota Koki
1014199 佐々木奨之 Shono Sasaki
1016128 田中瑞穂 Mizuho Tanaka
1016131 三浦隆太郎 Ryutaro Miura
1016167 松原千里 Chisato Matsubara

指導教員

村井源　迎山和司　田栁恵美子　平田圭二　角薫　松原仁

Advisor

Hajime Murai　Kazushi Mukaiyama　Emiko Tayanagi　Keiji Hirata
Kaoru Sumi　Hitoshi Matsubara

提出日

2019 年 1 月 16 日

Date of Submission

January. 16, 2019

概要

近年，人工知能分野の目標として挙げられているものに，物語の自動生成がある．物語の表現方法として，言語表現による物語や映像表現による物語，音声表現による物語などが存在し，多種多様である．しかしながら，物語を自動生成できるシステムには，言語や映像，音声などの要素のうち 1 つを自動生成するものがほとんどである．

本プロジェクトの目標は，ホラーとバトルのジャンルでオリジナルの作品を生成できる人工知能システムを開発し評価することである．また，本プロジェクトは既存作品のカメラワークを分析して情景描写に取り入れることで，鑑賞に堪えうる視覚化の実現も目的としている．自動生成の対象となる物語ジャンルとしてはホラーとバトルを選択した．プロットの自動生成においては，物語の典型パターンや物語構造データベースを作成し，それらを組み合わせることで物語の自動生成を行った．初期設定などに基づいてプロットの自動生成と 3DCG による視覚化を行うことで，多種多様なプロットに基づく情景描写を出力するシステム構成となっている．プロジェクトメンバーは，物語分析班，システム班，視聴覚班の 3 つの班に分かれて作業を分担した．プロジェクトとしての目的は，3 つの班が協力してホラーとバトルのジャンルでオリジナルの作品を生成できる人工知能システムを開発することである．

キーワード　人工知能，プログラミング，CG，物語分析，データベース

（＊文責：鈴木諒輔）

Abstract

In recent years, the goal of the artificial intelligence field is the automatic generation of a story. There are many ways of expressing stories, such as languages, images, and sounds. However, research on stories generation is the stage of studying automation methods by separating these elements.

The purpose of this project is to develop and evaluate an artificial intelligence system can create original contents in the genre of horror and battle. This project also aimed to realize appealing visualization based on analysis of the camerawork within existing works. The target genres for the automatic generation were ghost story and martial arts story. In order to generate plots, databases of typical patterns and plot structure were developed and those were combined for automatic story generation. The system automatically generates plots and visualizes that with 3DCG reflecting initial setting, therefore various scene depiction based on plots can be created. The project members are divided into three groups of story analysis group, system group, and audiovisual group. The purpose of a project is to develop an artificial intelligence system that three groups can collaborate and create original contents in the genres of horror and battle.

Keyword　Artificial intelligence, Programming, CG, Story analysis, Database

(*Responsibility for wording: Ryosuke Suzuki)

目次

第 1 章 はじめに

1.1 背景

近年，人工知能分野における目標の 1 つとして物語を自動生成する研究に大きな注目が集まっている．これは，人工知能が一般にデータ処理に強みがある一方で，創造性が弱みとなっている点が大きいと考えられる．そこで本プロジェクトでは，人間の「知的活動」である「物語を作る」という行為をシステムで実現することで，作品を自動生成できる創造的な人工知能システムの開発を目指すことにした．現在の物語生成研究の現状としては，言語や映像，音といった物語要素の一部を自動生成する手法を研究する段階である．また，コンピュータが創造性を持てるかという問題に対しては，人工知能も環境とのインタラクションによって試行錯誤による創造性を獲得できるだろうという意見もある[1].

本学の前年度プロジェクトだったインタラクティブ・ストーリーテーリングでは，コンピュータによるインタラクティブな物語自動生成システムの開発を試みていた．役割分担としては，実在作品の分析・データ化を担当する物語分析チーム，物語自動生成プログラムの作成を主に担う物語生成チーム，生成された物語に基づいたアニメーションを作成する映像作成チームの 3 班で分担していた．最終的には，物語の生成に必要なデータをある程度データベース化し，そのデータをもとに Python を用いて物語を生成することが可能となった．また，映像面では 7 体の人物モデルが完成し，メンバーが制作したモーションを用い，自動生成された物語を Unity 内においてアニメーションで表現できるようになったまた，映像面では 7 体の人物モデルが完成し，メンバー制作のモーションを用い，自動生成された物語を Unity 内においてアニメーションで表現できるようになった．

現在の物語生成の方法は，既存の物語構造のパターンに自動生成可能な要素を当てはめていくもの([2,3])と，ある場面の次の展開をデータベースや推論システムなどを用いて自動生成を行うもの([4,5])の二種類に大別できる．物語構造のパターンを用いる場合には，人間が面白いと感じるような一般的な物語構造を維持した生成が行える一方で，自動生成できる話の展開が限られる点に限界がある．また，前の場面の状況から次の場面を生成する場合には，データベースやエージェントの作り込みで多くのパターンを生成することができる一方で，最終的に生成された場面の連続が単なる連続的な行動記録となってしまい，物語としての面白さを担保できないという問題がある．

本プロジェクトは，ホラーとバトルのジャンルでオリジナルの作品を生成できる人工知能システムを開発し評価することを目指している．

（＊文責：鈴木諒輔）

1.2 目的

本プロジェクトはホラーとバトルのジャンルでオリジナルの作品を生成できる人工知能システムを開発し評価することを目標とした．この目標を達成するために，全体的な物語構造のパターンに依拠しつつ前の場面から自然とつながる連続的な展開を生成する手法を試みる．また従来の画像と音声を合わせた統合的な物語生成システムにおいて課題であったカメラワークに関しても，人間が作った既存作品におけるカメラワークの展開のパターンを情景描写に用いることでより自然な映像の作成を目指した．さらに，生成された物語や情景描写に対する評価実験を行うことで本手法の妥当性の検証を行った．

（＊文責：鈴木諒輔）

1.3 課題

目的を達成するために解決すべき課題は以下の通りである．

- 分析対象とする物語のジャンルと作品の決定
- 物語分析方法の決定
- 物語の構造や特徴の分析と体系化
- 体系化フォーマットの確定
- 分析データから物語データを自動生成するプログラムの作成
- 自動生成された物語データから 3D アニメーションを出力するプログラムの作成
- 3D アニメーションの視覚表現に必要なカメラワークの指定
- 3D アニメーションの視覚表現に必要な既存モデルの改変
- 3D アニメーションの視覚表現に必要なステージ及びポーズの制作
- プログラムの統合

本プロジェクトでは，分析対象とする物語のジャンルと作品の決定，物語分析方法の決定，物語の構造や特徴の分析と体系化，体系化フォーマットの確定，分析データから物語データを自動生成するプログラムの作成，自動生成された物語データから 3D アニメーションを出力するプログラムの作成，3D アニメーションの視覚表現に必要なカメラワークの指定，3D アニメーションの視覚表現に必要な既存モデルの改変，3D アニメーションの視覚表現に必要なステージ及びポーズの制作，プログラムの統合を課題とした．また，これらを評価する方法としてはアンケートを取り，そのアンケートの結果から分析した．

（＊文責：鈴木諒輔）

第 2 章 プロジェクト学習の概要

2.1 課題の設定

本プロジェクトは，コンピュータによる創造的でインタラクティブな物語自動生成システムを開発することを目標にしている．前年度には，本学のインタラクティブ・ストーリーテーリングというプロジェクトが物語分析や常識知識の大規模なデータを用いて，物語生成やアニメーション表示を行うインタラクティブ・ストーリーテーリングの研究を行っていた．その他にも，これまでには物語のストーリーを自動生成する研究などが行われているが，生成された物語の面白さの向上や物語創作の効率向上，ユーザの発想をより促進・刺激するための UI の改良などの課題が挙げられている[6]．よって，物語分析や大規模データの処理，多様な視覚化表現，機械学習などを取り入れたシステム開発を行うことで，新規性や有効性のある研究ができると考えたからである．物語自動生成システムを開発するにあたって，本プロジェクトでは物語のジャンルとしてホラーとバトルを選択した．ホラーとバトルを選んだ理由として，どちらのジャンルも特徴のあるストーリーが期待できる点や視覚化表現が比較的容易な点からこれらのジャンルを選択した．具体的なシステム開発の流れとしては，ホラーとバトルの作品を分析して，体系化した物語データを作成する．作成した物語データから Python を用いて物語データベースを構築し，チャートを自動生成する．さらに，自動生成されたチャートを Unity に出力し，チャートに映像やモーションを付与し，コンテンツを Unity から表示するようにシステムを開発する．以上がシステム開発の概要である．

本プロジェクトでは課題解決のために，既存の物語構造を分析しデータを収集する物語分析班，物語を生成するのに必要なプロットを生成するためのシステムやプロットに沿って映像化するためのシステムを開発するシステム班，プロットから映像を生成するのに必要となるカメラワークやモーション，モデルを作成する視聴覚班の 3 つの班に分かれて活動した．

（＊文責：鈴木諒輔）

2.2 到達目標

本プロジェクトでは，物語プロットを自動生成し，そのプロットに基づく映像表現を 3DCG で視覚化することで，鑑賞に堪えうる統合的な物語の生成を行うシステムを開発することを到達目標に設定した．物語自動生成システムによって，自動生成された物語が面白いかどうか，使いやすいシステムであるかどうかについてアンケートに回答してもらい，その結果を分析する予定である．ここで，物語自動生成システムの完成版とは，分析データから生成された物語データと 3D モデル等の視覚表現を組み合わせて，矛盾のない物語を自動生成できるシステムを指す．そして，人間の「知的活動」である「物語を作る」という行為をシステムで実現することで，作品を自動生成できるクリエイティブな人工知能システムの開発を目指す．

（＊文責：鈴木諒輔）

2.3 課題の割り当て

2.3.1 物語分析班

2.3.1.1 ホラー班

ホラー系統の物語では，物語によって登場人物の性別，年齢や，発生する怪奇現象，物語の流れやオチなどの属性が大きく変化する．また，物語の自動生成をするために必要なものを分析する必要がある．そこで，物語分析・ホラー班では，物語から主人公や怪奇現象，流れやオチに関する属性を抽出し，それらの属性から，ホラー系統の作品らしい物語の構造を求めることを目標に設定した．

したがって，分析を行う作品の決定，分析する作品の傾向や流れ，オチに対する仮説の設定，分析方法の決定，実際に分析に使用する属性の取捨選択の 5 つを目標達成のための課題とし，物語の分析を行った．

（＊文責：渡邊広基）

2.3.1.2 バトル班

バトル系統の物語では登場する各キャラクターによって戦闘スタイル・性格が複数存在し，また戦う人数・お互いの戦闘スタイルの優劣で展開が変動する．バトル班では物語の自動生成をする際に主人公に設定を付与し，その設定に合わせて物語が変化するようにすることを目標とした．

そのために物語のシーンのデータを抽出するにあたり, シーンを構成する各カットや場所の情報, 自動生成で特定のシーンが選ばれる条件，特定のシーンを通過した後の主人公のステータスの値の変化が必要と考えた．また主人公のデータを抽出する場合には主人公のステータスの種類や付与した設定がステータスに与える影響を考慮する必要があると考えた．

以上のことを踏まえ，バトル班では物語を展開によってパート分けをし，各パートを一文で説明したものをシーンとした．そして，そのシーンを登場するキャラクターのセリフや動作ごとに分けて 1 カットとしてエクセルシートのフォーマットに合わせて抽出した．

（＊文責：袴田翔）

2.3.2 システム班

2.3.2.1 Python 班

Python 班の目的は，物語分析班から受け取ったデータを取り込み，Unity 側で処理のしやすい形式に変換して渡すためのシステムの開発を行うことである．ただ変換して渡すだけではなく，大量のデータを用いて自動生成が実現できるように，大量のデータを処理できる機構を用意しておかなければならない．よって，データの集計にデータベースを用いることにした．

したがって，システムの流れとしては物語分析班から受け取った Excel のデータを 1 度データベースに格納する．その後，データベースから必要なデータを抽出し，物語の自動生成を行い，最終的にテキストデータ形式の物語データの生成を行うといった流れになる．

（＊文責：山田 康貴）

2.3.2.2 Unity 班

得られたデータから Unity 上で物語を再生するためにはオブジェクトの設定や Script での制御など Unity の様々なシステムを使う必要があることが分かった．そのため，システム班では班員全体で大まかなシステムの概要について考えを統一した. 課題は, Python 班から得られたデータを Unity で読み取ることのできる形式に変換すること，ユーザが読み取りやすい形にセリフを読み込んで画

面に表示させること，得られたデータからモデルとポーズを読み込んで適切な形でキャラクターを表示させること，これらのシステムを全て制御してデータの受け渡しやシーンの遷移を行うクラスを作ることの四つとした．中間発表以降は反省を踏まえ各担当の仕事を拡大し，上記の仕事に加えモデルに武器を持たせられるようにする，物語の舞台であるステージデータを作成するなどの活動も並行して行った．

（＊文責：高橋翔太）

2.3.3 視聴覚班

視聴覚班は物語に必要なビジュアルの部分とオーディオの部分を受け持つグループである．視聴覚班は第一にビジュアルの部分を重要と考え，中間発表まではカメラとモデルの二つの部分を重点的に制作した．

（＊文責：城田晃希）

2.3.3.1 カメラ班

カメラワークの自動生成においてイメージを具現化するために映画をカットで割りカメラワークの分析をはじめに視聴覚班全員で行った．結果として典型的なカメラワークを割り出す事ができ，カメラワークのそれぞれには象徴的な意味合いが存在する事が分かった．分析から割り出したカメラワークを適切な場面に適応することで，物語に没入感や，補助的な見えない説明を加える事が可能であると考えた．一方ではカメラワークを視覚的に落とし込む必要があり，そのための課題を下記のように分類した．

（＊文責：城田晃希）

2.3.3.1.1 映像分析

映像分析は私たちに既存のカメラワークの知識がないため，既存のカメラワークがどのようなものかを割り出すために行った．手法は，映画を場所，人物が変わらないシーンに分けシーンからカメラが切り替わるまでのカットに分けることから始めた．分けられたカットで用いられているカメラの位置や動きを一つずつまとめ，類似したカメラワークを統合することで割り出されたカメラワークを典型的なカメラワークとして分類した．

（＊文責：三浦隆太郎）

2.3.3.1.2 カメラワークの実装

割り出されたカメラワークは Unity のシステム上でカメラワークアルゴリズムとして実装をする必要があった．割り出された典型的なカメラワークは主体を基準とした相対的なカメラの位置を計算することで実装することができた．

（＊文責：三浦隆太郎）

2.3.3.2 モデル班

Unity 上で物語を再現するためには，人物や背景が必要である．さらに様々なアングルで撮影するためには，人物は3D である必要がある．そこでモデル班では，様々な3D モデルの制作を行うことになった．前期は，『人物モデル』『ステージ』『ポーズ』を重点的に制作及び拝借した．

（＊文責：田中瑞穂）

2.3.3.2.1 人物モデル

Unity 上で動作させるモデルは 3D で表現する必要がある．モデルそれぞれは，物語を進行させる際に混乱を生まないために，個別に判断がつくように外見の種類を豊富にする必要がある．

（＊文責：松原千里）

2.3.3.2.2 ステージ

システムが物語を再生している最中，物語がどこで展開しているかを示す情報は，一部セリフを除いてステージ以外には存在しない．そのため，作成するステージは屋外，屋内が理解できることに加え，システム利用者が見ただけで，キャラクターがどこにいるのかわかるくらいに明確にデザインする必要がある．また，カメラワークの実装により，場面を様々な角度から移す可能性があるため，モデルの可動領域などを考慮して，全体を取り囲むようにステージを作成する．

（＊文責：松原千里）

2.3.3.2.3 ポーズ

物語の中で登場人物は様々な動作を行う．モデルで物語内のキャラクターのそれぞれの動作を表現するためには，そのモデルは棒立ちではなく，ポーズをとらせる必要がある．物語が進行することや，ホラー，バトルなどの展開をより良くみせるためには，日常的によく見られるポーズから，ホラーやバトルにみられる特有のポーズまでを網羅的に実装することが必要となる．

（＊文責：松原千里）

第 3 章 中間発表までの開発

3.1 物語分析班

3.1.1 ホラー班

中間発表までの課題として，分析を行う作品の決定，分析する作品の傾向や流れ，オチに対する仮説の設定，分析方法の決定，実際に分析に使用する属性の取捨選択がある．ここではこれらの課題を達成するために我々が行った方法について記述する．

（＊文責：渡邊広基）

3.1.1.1 分析作品の検討

物語を自動生成するためには元となるデータが必要となる．そのデータを用意するために，物語をよく知る必要があると考えた．本項では物語分析班がデータ収集および分析に取り組むまでに行った作業について記述する．

物語分析・ホラー班はどの作品を対象に分析を進めるか討議した．作品のジャンルは数多くあるが，ホラーの作品を対象とした理由は，人間ではない何かが出現し主人公を脅かすという物語の流れが大体決まっていて，パターンが分かりやすいと考えたからである．

分析する際，データを多くとるために話数や巻数が多くなければならない．また，分析したデータをもとに物語を生成するため，分析する作品の物語が破綻していてはいけない．そのため広く知られていて高い評価を受けている作品を対象にするのが良いと考えた.この 2 つの条件を満たすホラー作品を討議した結果,「怪談レストラン」という小説を分析対象として定めることとなった．怪談レストランはホラーを題材としたオムニバス形式の児童文学で，小説は全部で 50 巻出版されている．

（＊文責：松浦史佳）

3.1.1.2 分析作品の傾向

分析を始める前に，まず数話ほど作品を読み，ホラー作品の流れを整理した．物語の全体背景や主人公の説明などが書かれている導入部分があり，お化けが出現する予告があり，そのあとお化けが出現する．ここで物語が解決するかしないかで分岐し，解決する場合はお化けについての解説があり，解決しない場合は解説なしでエピローグにつながる．ほとんどの作品がこのような流れにあてはまった．我々は，ホラー作品はこのような流れで進むと定義し，分析を進めた．

次に分析作品の傾向をつかむために，28 話分の作品を読みながら，作品に登場する主人公の性別や世代，どんな行動をとるのか，お化けの目的や形態，物語が解決するかしないか，またそれぞれの場合に事態が好転するか，暗転するかなどのデータをエクセルに入力した．データを集計したところ，物語が解決するのは 13，未解決で終わるのは 15 であった．解決する 13 のうち，事態が暗転する場合が 7，そのうちお化けに目的がありそれが成功するのは 5 であった．また未解決で終わる 15 のうち事態が暗転する場合が 9, そのうちお化けに目的がない場合は 6 であった. このことより，お化けに目的がありそれが成功した場合物語は解決し暗転する，また，お化けに目的がない場合物語は未解決で暗転するという傾向があることが分かった．解決にせよ未解決にせよ事態が暗転するという傾向が見られた．

しかし，手動でデータを集計したため，完全に物語の傾向がこうであるとは言い切れない．目に見えて分からなくても主人公の性別や世代，お化けの形態などが作品の傾向にかかわっているかも

しれない．次項では物語にかかわる隠れた要因を探るために行った作業について記述する．

（＊文責：松浦史佳）

3.1.1.3　分析方法

まずは，分析方法について記述する．我々は，分析作品の傾向から，物語に登場する人物やお化け，ストーリーの途中で主要人物がとる行動などの属性が物語の流れや結末に，大きな影響を与えるのではないか，また分析に用いたこの作品らしい物語の構造があるのではないかという 2 つの仮説を立てた．そこで，これらの仮説を数学的に検証するために，因子分析という手法を用いた．因子分析は，多変量解析の手法の一つで，ある観測された変数がどのような潜在的な変数から影響を受けているか探る手法のことである．この手法は，複数の変数の関係性を基にした構造を探る際に頻繁に用いられる．そこで，我々はこの手法を用いることにより，物語上の様々な属性から，それらを基にした物語の構造を探ることにした．

次に，因子分析に使用した属性について記述する．因子分析に使用した属性は，3 つに分類することができる．主人公の属性，お化けの属性，そして物語の属性である．主人公の属性は，男女どちらかという”性別”, 成人か未成年かという”世代”, 物語に家族が登場するかという”家族”がある．お化けの属性は，お化けが”人”，悪魔などの人以外の人型生物を表す”人型”，人の手や足，髪の毛などを表す”人の一部”，その場所自体がお化けであることを表す”場所”，そのもの自体がお化けであることを表す”物”のどれに当てはまるかという”形態”，お化けが主人公の救出を目的にしていることを表す”救出”，お化けが主人公に何らかの依頼をすることによって目的を達成しようとすることを表す”依頼”, お化けが主人公の手を借りずに自分で目的を達成しようとすることを表す”自発”, 主人公の殺害を目的とすることを表す”殺害”のどれを目的に行動するかという”目的”がある．物語の属性は，主人公がお化けに一人で遭遇するか，複数人で遭遇するかということを表す”遭遇”，物語が解決して終わった時に主人公が生きているかどうかということを表す”解決・生死”，物語が未解決のまま終わった時に主人公が生きているかどうかということを表す”未解決・生死”がある．

（＊文責：渡邊広基）

3.1.1.4　分析結果

まずは，上記の属性を用いた因子分析の結果について記述する．因子分析は，R 言語を用いて行った．R 言語を用いた因子分析に関する参考文献として[3]がある．最尤法を用いて因子を抽出し，回転はバリマックス回転を用いた．また，各属性の相関行列を算出し，その相関行列の固有値を求めた．その時に 1 を超えている固有値の数を因子数とした．表 3.1.1-1 は，因子分析の結果である．数値は，因子分析によって求められた各属性の因子に対する因子負荷量を表している．この因子負荷量の絶対値が大きいほど，因子に対する各属性の相関が強いということが言える．今回は，因子負荷量の絶対値が 0.35 以上のものを相関があるという基準とした．また，因子負荷量が 0 の部分に関しては，空欄となっている．

表 3.1.1-1 より，因子 1 は主に形態・人，形態・人型，目的・殺害からなり，殺害を目的とするお化けの形態を示す因子であることが分かる．

因子 2 は，主に未解決・生死，形態・人，形態・人の一部からなり，物語が未解決で終わり主人公が死亡する場合は，お化けの形態が人になりやすく，お化けの形態が人の一部になりにくいことを示し，物語が未解決で終わり，主人公が生存している場合は，お化けの形態が人の一部になりやすく，お化けの形態が人になりにくいことを示す因子であることが分かる．

因子 3 は，主に性別，世代からなり，主人公が成人の男性もしくは未成年の女性になりやすいことを示す因子であることが分かる．

因子 4 は，主に目的・依頼，目的・自発からなり，お化けの目的が依頼の場合には目的が自発になりにくいことを示す因子であることが分かる．

因子 5 は，主に解決・生死，形態・人，形態・物からなり，物語が解決して主人公が死亡する場合には，お化けの形態が人になりにくく，お化けの形態が物になりやすいことを示し，物語が解決して主人公が生存している場合には，お化けの形態が人になりやすく，お化けの形態が物になりにくいことを示す因子であることが分かる．

因子 6 は，主に形態・人，形態・場所からなり，お化けの形態が場所の場合，お化けの形態が人になりにくいことを示す因子であることが分かる．

これらの結果が示す通り，物語に登場する人物やお化け，ストーリーの途中で主要人物がとる行動などの属性が物語の流れや結末に，大きな影響を与えるということは分かったが，この作品らしい物語の構造というものをいまだに十分解析することができていない．また，因子 3 のように，3 つに分類した属性のうち主人公に関する属性の関係しか示していない因子や，因子 4，6 のように形態や目的など，一つの物語において 2 つ以上該当しない属性の関係しか示していない因子が現れた．これらは物語の構造を示すうえでは不必要なものであり，それらが総因子数の半分を占めているのは問題があると考える．そこで，今後はこの作品に対する仮説の再検討，また仮説に基づいた使用属性の検討，各属性に対するより適した分類をする必要がある．

表 3.1.1-1　因子分析結果

	因子 1	因子 2	因子 3	因子 4	因子 5	因子 6
解決・生死			0.340		0.421	
未解決・生死	0.165	0.388	-0.232		-0.171	0.187
遭遇	0.236		-0.325	0.332		
形態・人	-0.441	0.543		-0.112	-0.617	-0.369
形態・人型	0.970	0.197	0.102			
形態･人の一部		-0.981			-0.156	
形態・場所	-0.109	0.123				0.981
形態・物	-0.253	0.193	-0.112	0.108	0.795	-0.122
目的・救出	-0.128			0.218	-0.202	-0.111
目的・依頼	-0.295			-0.940	0.128	
目的・自発		-0.123	-0.111	0.366		-0.105
目的・殺害	0.459			0.109		
性別	-0.207	0.259	-0.450	-0.172		
世代			0.936			0.141
家族			0.102	0.309	0.157	

（＊文責：渡邊広基）

3.1.2　バトル班

3.1.2.1　分析作品の検討

我々が分析する物語のジャンルはバトル系統と呼ばれる，戦闘描写のある物語である(以下，ジャンル:バトルとする)．我々は物語分析を行う作品を決定する前に，どのような自動生成を行うのかについて検討した．検討をしていく中で，ジャンル:バトルの物語で一つの戦闘で物語が完結するものはなく，一つの物語の中に様々な戦闘が点在していることが分かった．そこで，我々が行う自動

生成を方針として二つの案が挙げられた．一つは，物語の完成を重視して，戦闘描写を含んだ整合性の取れた物語の自動生成である．もうひとつは，戦闘描写を重視した，戦闘展開の自動生成である．我々は戦闘展開の自動生成を最終的な目標とした．戦闘展開の自動生成を選んだ理由は，ジャンル:バトルの物語を分析するのであれば，ジャンル:バトル特有のストーリーの展開の自動生成に挑戦したいと考えたためである．次に，分析を行う作品についての検討を行った．初めに班員がそれぞれ興味のある戦闘の傾向を挙げることにした．挙げられた例として，バトルロイヤル，制限付きの戦闘，人対怪獣などがあった．プロジェクト学習の期間と我々の能力を考えた場合，多くの作品を用いた分析ができないと判断し，完結していて多くの戦闘パターンが存在する作品が相応しいと考えた．上記の条件に合致する作品として『Fate/stay night』を分析することを決定した．

（＊文責：吉田拓海）

3.1.2.2 分析方法

バトル班の分析方法を記述していく．ジャンル:バトルの物語には戦闘が行われているシーンと戦闘が行われていないシーンの2種類が存在する．我々はこの2種類のシーンをそれぞれ”戦闘パート”と”非戦闘パート”と位置付けた．そして戦闘パートの前後の非戦闘パートを分割し，非戦闘パートの始まりの部分を”プロローグ”，非戦闘パートの終わりの部分を”エピローグ”とした．しかし，プロローグとエピローグのままでは分析する規模が膨大であるため 1 つの文で場面の説明を与えて分割した．この分割で得られたものを”シーン”とした．さらにシーンを登場するキャラクターごとのセリフや動作・ポーズで細かく分割した．それを”カット”として，1 つ 1 つのカットに登場するキャラクターごとの位置や向き・方角を分析し与えた．そして分析して抽出したデータを Excel シートに記入した．

中間発表では中間発表用の分析フォーマットが用意されており，そのフォーマットに合わせて記入した．分析するにあたり，登場するキャラクターのセリフと動作，またセリフと場面の状況に適した位置関係に注意を払い，物語が画面上で表示される時に違和感が無いようにシートを作成した．

（＊文責：袴田翔）

3.1.2.3 分析結果

分析した結果，我々は自動生成の方針を決めることができた．一つのシーンを成立させるための矛盾のないカットの流れは，同一シーンの場合ほぼ変化がなかった．一方，一つのパートを成立させる矛盾のないシーンの流れは様々なパターンが存在した．我々はこの結果から，異なる物語の判別方法として異なるシーンの流れが挙がった．このシーンの流れの自動生成を行うことができれば，非戦闘パートの物語の自動生成ができるのではないかと仮定し，今後の自動生成の方針とすることにした．

また，中間発表に向けた物語分析では，システム-Python 班へ渡すための分析フォーマットに合わせたデータの抽出を行った．分析した項目は，システム班-Unity が使用するデータをもとに，場所・カメラの主体・セリフ・動作主・行動・方向・位置・画面効果・カットの意味(物語上の役割)である．なお，中間発表では画面効果およびカットの意味はプログラムの都合上使用されなかった．

（＊文責：吉田拓海）

3.2 システム班

3.2.1 Python 班

中間発表までの開発では，開発予定のシステム全体の流れを大きく「データの入力」及び「データの出力」の 2 つに分けて分担し，自動生成を考慮しないものとして，1 つのデータから 1 つの物語データを出力するシステムの開発を行った．

（＊文責：山田康貴）

3.2.1.1 データの入力

データの入力処理としては，物語分析班からデータを受け取り，データをデータベースに格納する所までが該当する．

使用したデータベースのテーブルとしては，場所を格納する Scene テーブル．主体と音楽，及びセリフを格納する Cut テーブル．役割や動作，方向や位置を格納する Character テーブルの 3 つを用意した．また，各テーブルのリレーションシップを表現するために，各テーブルに sceneID と cutID を導入した．各テーブルの形式としては，以下のようになっている．

表 3.2.1.1-1　Scene テーブル

sceneID	場所

表 3.2.1.1-2　Cut テーブル

sceneID	cutID	主体	音楽	セリフ

表 3.2.1.1-3　Character テーブル

cutID	役割	動作	方向	位置

プログラムの流れとしては，物語分析班から受け取った Excel 形式のデータを読み込み，システム内で解析をする．その後，必要なデータをデータベースの中に格納していくという流れである．

具体的な処理内容としては，物語分析班から受け取った Excel 形式のデータをシステム上で読み込み，シートの左上のセルから下に向かって順に走査を行う．セルの走査を行っている段階で必要なデータが見つかり次第，データベースに格納を行う．データベースに格納するときは，データそれぞれに対して各 ID を付与する．これらの処理を Excel シート最後まで行い，データの入力処理を行う．

（＊文責：山田康貴）

3.2.1.2 データの出力

データの出力処理としては，データ入力で作成したデータベースからシステム班-Unity に渡す物語データを txt ファイルとして出力するところまでが該当する．

物語データの形式の一例としては，以下の図の用になっている．

表 3.2.1.2-1　物語データ例

A さん，女性_通常
B さん，男性_幽霊
p 森
r1，a 歩く大人，dW，f4
r2，a 立つ 2 男，dE，f6
g1，m，s こんにちは
z

まず，先頭に登場人物の名前とそれに対応するモデル名を順に表示している．次に，シーン情報として p の後に場所の名前を表示した．次に，キャラクター情報として r の後にモデルに割り当てられた数字，a の後にモデルのモーション，d の後にモデルの向き，f の後にモデルの位置を表示した．次に，カット情報として g の後にカットの主体となるモデルに割り当てられた数字，m の後に音楽，s の後にセリフを表示した．最後に物語データの終わりを表すために z が有る．

プログラムの流れとして，まずプログラムを実行するとデータベースのファイル名を入力する欄が表示され，入力するとそのデータベース内に出現するモデルの一覧が出力される．次に，登場人物の名前を入力し，その名前に対応するモデルの種類をモデル一覧から選んで入力する．これを人数分行うと物語データが txt ファイルとして出力される．この txt ファイルの文字コードと改行コードは Unity で正しく読み取れるよう UTF-8 と LF とした．

具体的な処理内容としては，まず python 内でデータベースを扱いやすくするためにデータベース内のテーブルをすべて 2 次元のリストに格納した．次にリストを上から順に見ていき，各テーブルの ID が同じ間だけ見ているテーブルから txt ファイルに出力する内容に追加していく．更に，ID が違うものになったらテーブルを Scene，Character，Cut，Scene…の順に変更し，テーブル内のデータを出力した．こうすることで，テーブルに格納されたデータを一定の形式で物語データを出力できる．

また，物語データの先頭にモデル名と登場人物の名前を表示するために，python の標準入力で設定できるようにした．具体的には，標準入力で入力した内容を txt ファイルの先頭にそのまま出力した．そして，入力した順番で 1 から 1 ずつ大きくなるように数字を割り当てた．ここで数字を割り当てることでモデル名とモデルの任意の名前を結びつけることが出来る．

（＊文責：寺島啓悟）

3.2.2　Unity 班

Unity 班では中間発表までの期間で Python 班の用意した物語データを読み込み，キャラクターモデルの表示，背景の表示，カメラの配置，テキストの表示を行うシステムを開発した．この内，カメラを配置するためのスクリプトは視聴覚班が作成したものである．このシステムは text ファイル形式の物語データを実行時に読み込み，画面上に物語の内容を表示する．また，ユーザがマウスをクリックするごとに物語は次へ進められる．Unity 上での実際の実行画面は図 3.2.2-1 のようになっている．

図 3.2.2-1 Unity 上での実行画面

(＊文責：津沢慎吾)

3.2.2.1 物語データの読み込み

Python 班が出力した text ファイルである物語データを読み込み，シーンとカット上の各属性として分ける SenarioLoader クラスを作成した. 物語データは行に格納されている情報によってシーンとカットが分けられており，各行にはその場面で必要な属性がカンマで区切られて格納されている．SenarioLoader クラスでは Unity 上の Resource フォルダに格納された物語データを行とその中の属性をもつ二重配列に格納し，それを他のクラスから読めるように各属性を適切なデータ型に変更し，階層的に実装されたシーン配列とカット配列の中に格納した．シーンは新しい場所の属性が宣言されるたび，カットはその中に登場する登場人物たちの属性が呼ばれるたびに作成される．

（＊文責：高橋翔太）

3.2.2.2 UI の作成

UI を作成するにあたって，まず初めに昨年度の作品を参考にして作成した．また，今回は２D の作品であるため，動きがわかりやすいようにウィンドウを消す必要がない．なので，ウィンドウを常時配置するように実装した．中間発表まででは，セリフを言う人の名前とそのセリフを表示できるクラスを作成した．

背景を多少見えるようにウィンドウを半透明にして作成したが，字が読みにくいという課題を発見した．今後は，課題や目標を改善していく必要がある．

(＊文責：南部太雅)

3.2.2.3 モデルの表示

読み込んだ物語データをもとに，視聴覚班が製作した 3D モデルを Unity 上に適切に配置するための機能を C#スクリプトによって View.cs として実装した．中間発表でのシステムでは，まず予め 3D モデルデータを Unity に読み込み，それらにモーションを付与したものを Prefab と呼ばれる Unity 独自の形式でオブジェクトとしてまとめておき，適当な場所に配置しておく．次に物語データから 1 カットにおける登場人物それぞれの名前，位置，方向，モーションを読み込む．最後に読み込まれた登場人物の名前から Prefab を参照，位置，方向，モーションを書き換える，といった処理を行っている．物語データでは位置や方向は座標や四元数としては格納されておらず，そのまま

では Unity 上で扱うことが出来ないため，それらを変換する関数もそれぞれ実装した．

実際のモデルの表示にあたっては，中間発表用に視聴覚班が用意したモデル，モーションは MMD と呼ばれる独自の形式を用いた 3DCG アニメーション作成ソフトウェア専用のものであり，互換性が無いためそのままの形式では Unity 上で読み込むことが出来ない．今回のシステムでは MMD4Mechanim と呼ばれる Unity 用プラグインを用いることでモデルとモーションの変換を行い，Unity で読み込めるようにした．また，モーションの再生にあたっては今回のシステムにおいてはモーションのブレンドを行う必要が無く，一人のキャラクターに大量のモーションが付与されるという状況を踏まえ Animation と呼ばれる Unity の機能を用いた古典的な制御によって行った．

(＊文責：佐々木奨之)

3.2.2.4 マネージャークラスの実装

中間発表までの Unity 全体の設計は，シナリオデータ読み込み，UI，モデル表示，カメラアルゴリズムそれぞれの機能を SenarioPresenter と名付けたマネージャークラスが統括的に制御する仕様になっている．この仕様に決定した理由はそれぞれの機能の依存関係を極力無くし，プログラム結合時にマネージャークラスを編集するだけで良いようにするためである．しかし現状の問題点として，マネージャークラスのソースコードの可読性の低さが挙げられる．なぜなら，複数の機能を一つのクラスで管理している関係上，それぞれの機能を利用するための様々な処理が一つのソースファイルに書かれているからだ．よって，マネージャークラスそのものを複数のクラスに分割し，今後このソースコードを利用する人間が理解できるようにリファクタリングを行うことが課題の一つである．

次に，マネージャークラスの具体的な処理内容について記述する．まず，Unity による Start イベント実行時に，シナリオ読み込み機能の実行とキャラクターモデル表示の初期化が行われ，最後にシナリオ再生の開始が宣言される．その後，Unity による定期的な Update イベント実行時にユーザのマウス左クリック入力が発生するまで待機を行い，入力が確認されたら現在実行されているシナリオを 1 カット分進めるという流れになっている．シナリオを 1 カット進めるにあたって一番始めに，登場するモデルの表示とその更新を行っている．これはカメラの座標の計算がモデルの座標，向き，ポーズに依存しているため，必ずカメラの更新よりも先にモデルの更新を行わなければならないからである．また現在，Unity の同一イベント関数上でモデルのポーズの更新とカメラの更新を行ってしまうと，カメラの位置制御が正しく行われないバグが発見されている．これは推測であるが，Unity ではプログラムによるポーズの更新の宣言が行われたタイミングから一定時間経過した別のタイミングで実際に 3D モデルの更新が行われるではないかと考えられる．よって現在のプログラムでは，Unity の Update イベントよりも必ず後に呼び出される LateUpdate イベント上でカメラの更新を行うことによってこのバグを回避している．シナリオの最初の呼び出し時，またはカットの更新を進めていくとシーンが切り替わるタイミングがある．ここでマネージャークラスは，現在までの背景オブジェクトの削除と新しい背景オブジェクトの読み込み，配置を行う．構想段階では Unity の Scene 機能を用いて背景ごとに個別の Scene を作成しそれらを呼び出すことも検討したが，Scene データ作成の手間を考慮して全て 1 つの Scene 内で行う実装になっている．シーン遷移にあたっての暗転処理等の挿入は今後開発する予定である．

ここまで説明したように，中間発表までのマネージャークラスの設計ではこのクラスがその他のすべての機能にアクセスするようになっている．そのため，機能同士のデータの受け渡しは必ずこのクラスを介して行われる．従って，このクラスの設計にあたっては，その他の機能に要求するデータ形式を慎重に決定する必要がある．正しいデータの受け渡しが行われなければシステム全体が機能しなくなるからだ．実際に現段階でも，設計段階の情報共有不足で，同じ対象を示すデータで

あるのにデータ形式が不揃いになってしまっている箇所が存在し，マネージャークラス上で強引にデータを整える処理を行っている．後期以降の開発ではより設計部分での情報共有を大切にしていく必要がある．

（＊文責：津沢慎吾）

3.3 視聴覚班

3.3.1 カメラ班

前年度のプロジェクトではカメラワークのパターンは固定であったため，視覚的な表現にやや違和感を感じるという意見がプロジェクト内で多く挙がった．それらの意見からカメラ班は良いカメラワークを生成していくことが良い作品を自動生成することに繋がると考えた．そのためカメラ班はカメラワーク分析とカメラワークの Unity 上での実装を中間発表までに行った．

（＊文責：城田晃希）

3.3.1.1 分析

3.3.1.1.1 分析の概要

我々視聴覚班はカメラワークごとに固有の意味合いがあるのではないかと考え，既存の映画作品のカメラワークを分析した．分析する作品には映画『呪怨』を用いた．理由としては，ホラー映画として知名度が高く，世間からの高い評価を得ていることが挙げられる．映像分析の手順は，カメラが切り変わる「カット」に映像を分け，どのようなカメラワークをしているかを抽出することで行った．呪怨のほかに羊たちの沈黙とインターステラーの予告編の映像を分析し，約 500 カットの分析をした．

この作業により既存のカメラワークをおよそ 8 つに分類した．ロングショット，フルショット，ミディアムロングショット，バストアップショット，ミディアムクローズアップショット，クローズアップショット，ズームアウトショットそして肩越しショットである．各ショットの意味合いとして，ロングショットは場面全体を映したショットであり，場面の状態や雰囲気を伝えることができるが人物の動作や表情は伝えることができない．フルショットは主体となる人物の頭から足先をとらえたショットであり，人物がどのような動きをしているかや場面の雰囲気を伝えることができるが，人物の表情は伝えることができない．ミディアムロングショットは主体となる人物の頭から膝までをとらえたショットであり，人物の動きをフルショットよりもダイナミックに映すことができる．バストアップショットは主体となる人物の頭から胸までをとらえたショットであり，登場人物の表情や人物同士の親しさを伝えることができるが，人物がどこにいるかを特定することはできない．ミディアムクローズアップショットはバストアップショットよりも主体となる人物に寄るためさらに詳細な表情を伝えることができる．クローズアップショットは主体となる人物の頭から胸の上をとらえたショットであり，最も詳細に主体の表情を印象的に伝えることができるが，場面は雰囲気を伝えることもできない．ズームアウトショットは主体となる人物と向かい合うもう一人の人物を横からとらえたショットであり，対象の位置関係や場面の雰囲気を伝えることができるが，人物の表情は伝えることができない．肩越しショットは主体となる人物をカメラ手前側の左右どちらかに映し動作の対象をカメラ奥の反対側に映すショットであり，会話をしている場面で多く使用される．以上のことから，カットごとに伝えたい事柄に合わせたカメラワークを生成することで，見る人に物語への没入を促すと考える．

（＊文責：三浦隆太郎）

3.3.1.1.2 分析方法

分析する作品には映画『呪怨』を用いた．理由としては，ホラー映画として知名度が高く，世間からの高い評価を得ていることが挙げられる．

分析の手順は，同じ場面を意味する「シーン」ごとに絵コンテを作成し，カメラが切り替わるまでの「カット」ごとに主体となる人物の映り方を体のどこからどこまでが映るかに種類分けをした．

呪怨のほかに羊たちの沈黙とインターステラーの予告編の映像の分析も行い，約 500 カットの分析をした．これらの作業は公に存在するロングショット，フルショット，ミディアムロングショット，ミディアムショット，バストアップショット，ミディアムクローズアップショット，クローズアップショット，エクストリームクローズアップショット，マクロショット，ズームアウトショット，肩越しショットなどといった数多のショットから一作品の中でどのショットがどの程度使われているのかを調べるために行った．ホラー作品やバトル作品でよく使われる，有用性の高いショットの種類を選定することでカメラワークを通じて場面内でのそのシーンの意味合いを表すために良いと判断がなされたからだ．

（＊文責：三浦隆太郎）

3.3.1.1.3 分析結果

中間発表までに行った分析により種類分けされたカメラワークをおよそ 8 つに選定した．ロングショット，フルショット，ミディアムロングショット，バストアップショット，ミディアムクローズアップショット，クローズアップショット，ズームアウトショットそして肩越しショットである．

各ショットの意味合いとして，ロングショットは場面全体を映したショットであり，場面の状態や雰囲気を伝えることができるが人物の動作や表情は伝えることができない．フルショットは主体となる人物の頭から足先をとらえたショットであり，人物がどのような動きをしているかや場面の雰囲気を伝えることができるが，人物の表情は伝えることができない．ミディアムロングショットは主体となる人物の頭から膝までをとらえたショットであり，人物の動きをフルショットよりもダイナミックに映すことができる．バストアップショットは主体となる人物の頭から胸までをとらえたショットであり，登場人物の表情や人物同士の親しさを伝えることができるが，人物がどこにいるかを特定することはできない．ミディアムクローズアップショットはバストアップショットよりも主体となる人物に寄るためさらに詳細な表情を伝えることができる．クローズアップショットは主体となる人物の頭から胸の上をとらえたショットであり，最も詳細に主体の表情を印象的に伝えることができるが，場面は雰囲気を伝えることもできない．ズームアウトショットは主体となる人物と向かい合うもう一人の人物を横からとらえたショットであり，対象の位置関係や場面の雰囲気を伝えることができるが，人物の表情は伝えることができない．肩越しショットは主体となる人物をカメラ手前側の左右どちらかに映し動作の対象をカメラ奥の反対側に映すショットであり，会話をしている場面で多く使用される．

このようにカメラワークには情景描写の得意不得意や使用されるタイミングといった，固有の意味合いがありカットごとの情景に合わせたカメラワークを生成することで，見る人へ物語への没入を促すと考えられる．

（＊文責：三浦隆太郎）

3.3.3.1.2 実装の概要

視聴覚班のカメラチームはキャラクターの動作や位置から場面の意味合いを汲み取り，場面にマッチするようなカメラワークが自動生成できるようにすることを目標としてきた．理由としては，

場面に合わせた適切なカメラワークを設定する事によって，見ている人により物語の没入感を与えるのではないかと考えたからである．

今回の中間発表では，Unity 上でランダムにカメラワークを生成する事が出来た．

最終目標としている場面にマッチするようなカメラワークの生成は出来なかったが，分析をした結果から，カメラワークにはそれぞれにその一場面を表現するような象徴的な意味合いが存在することが分かった．更に分析を進めることにより場面にマッチするようなカメラワークの生成は可能であると考える．

今後のカメラチームの方針としては，更に映像からカメラワークの分析を進める事である．そしてその結果から得られたカメラワークの意味合い，パターンなどをいかにして自動生成に落とし込むかを検討し，実装をしていく事を考えている．

場面の意味合いを汲み取る事として特に，重要な点として前後の場面の繋がりが挙げられる．その前後関係を整理し，記号処理的にカメラワークに落とし込む事が今後のカメラチームの第一目標である．

そのためにはまず落とし込みの手法を選定する必要がある．中間発表前に記号処理をする手法としては機械学習が挙げられた．しかしそれを用いて自動生成するためには，現状のシステムでは有用であるのかなどを吟味する必要があると考える．

（＊文責：城田晃希）

3.3.2.2.1 カメラワークアルゴリズム

Unity 上でカメラワークを生成するアルゴリズムについて記述する．まず場面の中に存在するキャラクター達の中には，カメラに中心人物として映りこんだり，セリフを喋ったりしている主体というものが存在する．その主体の情報を元に，カメラの位置と向きを相対的に計算している．カメラワークのパターンは複数存在しその中から登場するキャラクターの位置動作を元に場面に合ったパターン選択した．中間発表ではカメラワークパターンは映像分析した中から六種類をピックアップしその中からランダムで選択した．以上の事をカメラワークアルゴリズムする．

問題点として現状ではキャラクターの動作や位置関係などを一切無視してランダムでカメラワークパターンを設定しているので，場面にあったカメラワークが設定できていない点が挙げられる．今後の展望としては適切なパターンを選べるようなシステムを作りたいと考えている．

（＊文責：城田晃希）

3.3.2.2.2 カメラワークアルゴリズムのプロトタイプ制作

中間発表のために制作したプロトタイプのカメラワークアルゴリズムについて具体的に記述する．カメラワークアルゴリズム自体は Unity 内のプロジェクトに CameraManager クラスとして実装した．このクラスはカメラワークパターンの設定と計算を行うクラスとなっており，一場面内に登場するキャラクターがコンストラクタの引数となっている．CameraManager は変数に Vector3 クラスの cameraPos 配列と int 型の fov を持つ cameraPos 配列の 1 番目の要素をカメラの三次元上の座標，二番目をカメラの向きとし，fov は視野角を表している．インスタンスが生成された時点でカメラワークの導出と計算を行いそれぞれの変数の値のゲッターが存在する．

カメラワークパターンの導出と計算は関数 CalcCameraPosition で行われる．CalcCameraPosition ではまず一場面に登場するキャラクターを Unity 内でのゲームオブジェクトとして抽出しその情報を元に関数 CalcCameraPtn でカメラワークパターンを決定する．カメラワークパターンは zoomOut，longshot，midiumlongShot，bustupShot，closeupShot，oversholderShot の六種類ありプロトタイプではランダムに選択をしている．最後に CalcCameraPtn で決定したパターンを元にカメラの位置

と向き，視野角を計算する．

カメラがポーズやモデルのサイズによって想定をしない挙動をしてしまう問題が起こった．ゲームオブジェクトのモデル内の HipMaster と呼ばれる腰の部位の座標を探索で取得し，計算に踏まえることで解決を図り，子供のサイズにカメラワークを合わせる事には成功をした．しかし特定のポーズの場合は干渉することもあり完全な解決には至らなかった．

後期からはまずコードのリファクタリングをしていき処々のバグの改善，そして目標としている適切なカメラワークの生成を実現していきたい．

（＊文責：城田晃希）

3.3.2 モデル班

Unity 上で物語を表現するためには，様々なものが必要となる．そこでモデル班は，おおまかに分けて『人物モデル』『ステージ』『ポーズ』を制作した． 以下にはその制作プロセスを記述する．

（＊文責：田中瑞穂）

3.3.2.1 人物モデル

人物モデルは物語分析班からの要望により成人男性，成人女性，少年，少女，さらにそれぞれの敵バージョン，幽霊バージョンとして 2 種類追加し，通常時を含めた合計 3 種類という全部で 12 種類制作することになった．しかしモデルそのものを 1 から制作するのは極めて時間がかかり，他の制作物が疎かになってしまう可能性がある．そこでモデルは改変可能な既存のフリーモデルを使用し，それらのテクスチャを改変することで 12 種類のモデルを用意した． テクスチャとは，モデルの表面に張るイラストのことである．

テクスチャの変更は，CLIP STUDIO PAINT を利用して行った．まず，モデルのテクスチャ画像を前述のソフトで開き，色調を変更した．変更したテクスチャを別名保存し，元になったモデルのコピーしたデータのテクスチャ画像と置き換える．この変更作業を 4 体のモデル，2 種類分行うことでテクスチャの変更を行った．

（＊文責：松原千里）

3.3.2.2 ステージ

ステージとは，物語の舞台，背景のことである．我々は中間発表までの間に，『森』と『白い部屋』の２つのステージを制作した．制作に使用したツールは，blender と呼ばれるフリーの 3DCG ソフトウェアである．

まず，blender で側面と底のみの箱を制作した．森のステージは，その箱の全ての面に森の画像をテクスチャとして貼った．白い部屋は，箱のマテリアルを白に設定した後で窓のような簡単な小物を制作して追加した．このようにして，２種類のステージは制作された．

なお，ステージのデータは obj ファイルで出力している．

（＊文責：田中瑞穂）

3.3.2.3 ポーズ

ポーズとは，人物モデルに取らせる体制のことである．我々は中間発表までの間に，『立つ１(棒立ち)』『立つ２(腰に手を当てて立つ)』『驚く(両手を顔に当てる)』『怖がる(腕を組んで縮こまる)』『歩く』『走る』『拝む(両手を合わせて礼をする)』『話す(腕を組む)』『構え(ファイティングポーズ)』『手前(右手を前に出す)』『膝をつく１(片膝をついてしゃがみ込み)』『膝をつく２(両膝をついてうずくまる)』『頬杖(腕を組んで左手を顔に当てる)』の全 13 種類のポーズを制作した．制作に使用し

たツールは，MikuMikuDance(通称 MMD)と呼ばれるフリーの 3DCG ソフトウェアである．

まず，MMD に人物モデルのデータを読み込む．続いて人物モデルに入っている『ボーン』という骨組みを曲げたり移動することで，思い通りの姿勢を取らせる．この工程を繰り返すことで，全てのポーズを制作した．なお中間発表に使用した人物モデルは 4 種類あるが，この 4 種類のモデルは体格が異なるためボーンの形状も異なる．そのため，大半のポーズは個別に制作している．『立つ２』『驚く』『怖がる』『拝む』『話す』は，成人男性，成人女性，子供男性，子供女性の 4 種類分．『歩く』『走る』は，成人，子供男性，子供女性の 3 種類分．そして棒立ちである『立つ１』は，モデルごとの区別がないため 1 種類となっている．なお『構え』『手前』『膝をつく１』『膝をつく２』は，成人男性と成人女性，『頬杖』は，成人女性のみのポーズであり，どちらもそれぞれのボーンに合わせて制作されている．

なお，ポーズのデータは vpd ファイルで出力している．

（＊文責：田中瑞穂）

3.4　中間発表

2018 年 7 月 13 日に公立はこだて未来大学内にて中間発表が行われた．中間発表を行うにあたって，プロジェクト内容が理解できるような A3 サイズのポスターとスライドを制作し，20 分のプレゼンを 6 回行った.また，プレゼンの視聴者に向けてアンケートを取り，評価を受けた．

（＊文責：高橋翔太）

中間発表で 67 人から受けた評価は表 3.4.1 に示す結果となった．

表 3.4.1　中間発表での評価

評価	発表技術
1	0
2	0
3	0
4	2
5	3
6	10
7	18
8	18
9	3
10	8
無回答	5
平均点	7.42

発表技術と発表内容についてアンケートを取った．まず，発表技術についての一例を示す．

- 発表者によって発表のレベルが違う．
- スライドはもう少し大きめにしたほうが良い．

- 物語の法則が分かったといっていたが，スライドは図だけで文字がなかったため，文字を載せてほしい.
- プレゼンの時は PC を見ないほうがいい.
- 物語の生成に法則があることがよくわかったが，図がとにかく見づらかった.
- デモの位置づけがよくわからなかった.
- 基本的に見やすいが，フォントを変更したほうが良い場面があった.
- 図が多く，わかりやすいスライドでしたが，レーザーポインタなどを使うとよりわかりやすくなると思います.
- 小難しく理解できないところがあった.
- デモやアルゴリズムのフローがあったからわかりやすかった.
- 結局のところ，概要，背景，期待される成果がよくわからないので，やってることの何がすごいのか，努力したのかがうまく伝わってない気がします.
- 物語分析班の使用していた図や分析手法の説明が足りてない.
- 初めに現状の成果を出すことで，コンセプトをわかりやすくしていてよかった.
- 因子分析の結果は，グラフをカラーにし，属性のカテゴリで色分けするとわかりやすくなると思う.
- データ量としては良かった，映像について既存の物との差別化をしてほしい.
- 具体的に得られたプロットネットワークを例示してくれるとよい.
- 未来大生はある程度理解できる内容だが，一般の方には難しいかもしれない.
- 各班がバラバラのことをやっているが，どう結合していくのか知りたかった.

（＊文責：南部太雅）

次に，発表内容についての一例を示す.

- 大事なこと（どこが AI なのか）がよくわからなかった.
- テクニカルな説明が主でしたが，プロジェクトの社会的位置づけやこういうプロジェクトを求める社会的背景などについても考える必要があると思います.
- 分析以外にユニークなアプローチが必要では？分析って窮屈な感じがします.
- 物語が出来るとしてありがちで微妙な物語しかできないんじゃないかと思った.
- プロトタイプの動作手順について詳細な情報をスライドで提示してもらえるとわかりやすいと思う.
- 物語をテキストだけでなく，視覚的に楽しめるのは面白いと思った.
- BGM や効果音なども取り込むとより臨場感が出るかなと思いました.
- 因子分析の結果が面白いと思った．ただ，その結果に信ぴょう性があるのか気になった.
- 出来上がったストーリーの良し悪しをどう評価するのか，楽しみにしております.
- 自動生成も良いと思うが，もう少し割合を減らして自分達の手を加える範囲の拡大を考えても良いのではと思った.
- "適切な"カメラワークにおける適切か否かはどうやって決めるのでしょうか？
- 既存の小説やゲームは，限られた「成功例」であって，様々な可能性(可能な限りの網羅的な)のデータは入手しづらい．となると，限られた成功事例のセットだけで，システム実装に役立つデータを作れるのか？

（＊文責：高橋翔太）

発表技術については，アンケートの結果から全体的に声が小さいという意見が多く見られた．発表時は場所も原因のひとつであるが，聴講者との距離が離れていたために起こったと思われる．ま

た，発表者によって発表のレベルが違うとも言われている．今回，発表者を交代する形式をとっていたため，個人の差が顕著に出たと感じた．また，スライドの図が多く，その説明が少ないとの意見があった．しかし，図がわかりやすいという意見もあった．このことから，図に関してはこのままでよいが,万人に理解してもらうには図の一つ一つに説明を付けることが,課題であると考える．

（＊文責：南部太雅）

次に，発表内容については，AI をどこで使っているのか，また因子分析やカメラワークの分析が分かりにくいという意見が多く，理解しづらい範囲の細かい説明が必要だと感じた．また，プロジェクトの評価をどうするか疑問に思う意見も多く，期末に向けて具体的な方針を決めることが必要とされる．

（＊文責：高橋翔太）

第 4 章 成果発表までの開発

4.1 物語分析班

4.1.1 ホラー班

4.1.1.1 物語分割とプロット分類

物語を生成するためには，物語の構造を理解している必要があると考えた．そこで，作品を分析し，物語の流れを整理した．分析作品の傾向の章でも軽く触れたが，今回分析したホラー作品は導入，予告，出現，解決行動，解説，締めの流れで進むことがほとんどであった．この流れで進まない作品があっても，分析対象とした．本項ではその分類の手順とそれぞれの説明をする．

はじめに物語分類の手順について述べる．まず分析する作品を数話分熟読した．そして，物語の展開が変わっている部分で区切った．区切った部分を一グループとし，そのグループに見出しを付けた．例えば「昔，とある村におじいさんが暮らしていました．ある夜，扉をたたく音が聞こえました．扉を開けてみると幽霊が立っていました.」という文章があったとする．この場合，最初の一文と次の一文，最後の一文で文章を区切る．そして最初の一文に導入，次の一文に予告，最後の一文に出現という見出しを付ける．このような手順で物語を分類した．今回分析した作品のなかに，予告と出現が何回か起こるものがあったが，そうなるデータが少数なことととループをさせると自動生成が複雑になることを考慮し一回だけ起こるような流れにした．

次に分類したそれぞれの詳細を説明する．先ほども記述したが，物語分類は導入，予告，出現，解決行動，解説，締めの全部で 6 つある．

- 導入…登場人物の紹介や生い立ちを説明し，主人公に何らかの目的や謎が提示される部分
- 予告…登場人物の周辺に何らかの現象が起こり，霊の出現を予感させる部分
- 出現…霊が出現し，登場人物に何らかの行動を起こす部分
- 解決行動…登場人物が霊の行動に応じてとる行動を示す部分
- 解説…出現した霊に対する生い立ちや目的などを説明する部分
- 締め…物語を締めくくる部分

以上が物語分類の 6 つである．

次はこの物語分類を細分化したプロット分類について説明する．

システムに物語を一から自動生成させるのは，破綻が起きたり文章として成り立っていなかったりと問題が生じる可能性があり非常に難しいと考えた．そこで我々が物語のプロットを作成し，それを組み合わせることによって多様な物語をシステムに生成させることにした．そこで前項で述べた物語分類を細かく分類しプロットを作成した．本項ではその手順とそれぞれのプロットの細かい説明をする．

その前になぜこのプロット分類という作業を行ったかを軽く説明する．前項で物語分類を行い，物語の構造を明らかにした．しかし，この通りに文章を作っても骨組みだけで面白味がない物語ができてしまう．骨組みに肉付けをして深みを持たせることで面白味のある物語が作れると考えた．そのため，物語分類をさらに細かく分類し，物語の構造をさらによく知る必要があると考えた．また，プロットの組み合わせで物語を生成するとなった場合，プロットの種類が少ないと生成される物語がワンパターンとなってしまう．これらの理由を踏まえて，このプロット分類という作業を行った．

では，プロット分類の手順について説明する．作品をいくつか読みながら，物語分類ごとに文章を要約して付箋に書き出した．その作業が終わったら，KJ 法を用いて物語分類ごとに似通ったもの

をいくつかのグループにまとめた．KJ 法とはデータをまとめるための手法の一つである．例えば，ある作品で霊が出現するときの目的が登場人物を脅かすことで，もう一方の作品での霊が出現するときの目的が登場人物を殺害することだったとする．この 2 つは霊が自力で自らの目的を達成しようとしていることから，似通っていると判断し，まとめることができる．最後に，それぞれのグループに見出しを付けた．先ほどの例でいえば，出現部分の「自力」という見出しを付ける．ここまでの作業を終えたら，作品を読み進めて分類にあてはまるものがあれば前期同様にエクセルにデータを入力していく，という作業を行った．前期分も合わせ，約 100 作品を分析した．

次に，プロットの細かい説明をする．

導入部分は，「語り手」，「行動」，「現象」の 3 グループある．物語に登場する人物以外が導入部を説明するのが「語り手」である．例えば，「昔，とある村におじいさんが暮らしていた．」など，第三者が物語を話している場合はこれに分類される．登場人物が人物の紹介や生い立ちを説明し，目的や謎のために行動をおこしていれば「行動」に分類される．例えば，「私が小学生のころ，家の近くに幽霊屋敷と噂されている建物があった．興味本位で学校の帰りに友達と寄ることにした．」という文章がある場合はこれに分類される．「現象」は，登場人物が人物の紹介や生い立ちを説明し，目的や謎が提示される部分を指す．例えば「私の母が亡くなってから，私の身の回りで奇妙なことが起こるようになった．」などという文章がある場合はこれに分類される．

予告部分は，「直接」と「間接」の 2 グループある．「直接」は，霊が登場人物に姿を現すことである．例えば，人間と思って接していたら最後の最後で幽霊だと分かった場合などはこれに分類される．「間接」は，霊が姿を現さず，その場所の現象で出現を予感させることである．例えば霊が出現する前に，寒気がする，奇妙な気配がする，不審な音が聞こえるなどのことが書かれていた場合はこれに分類される．

出現部分は，「自力」と「依頼」の 2 グループある．「自力」は，霊が登場人物に行動を起こし，自らの目標を達成しようとすることで，例は先程も軽くあげたが，登場人物を脅かそうとしたり殺害しようとしたり，また救出しようとする場合もこれに分類される．「依頼」は，霊が登場人物に依頼をし，自らの目的を達成しようとすることである．例えば，霊が自分の親に届けたいものがあって，登場人物に頼る場面があったらこれに分類される．特に霊の出現に目的や目標がなく，どちらにも分類しがたいと判断した場合は，無理に分類せず無視することとする．

解決行動は「受容」，「反抗」，「パニック」の 3 グループとその結果「成功」と「失敗」の 2 グループがある．「受容」は霊の行動・要求を受け入れることで，襲われて抵抗しなかったり，依頼されたことを遂行したりした場合はこれに分類される．「反抗」は，霊の行動・要求を受け入れずに反抗することで，霊と対峙して戦ったり，依頼を断ったりした場合はこれに分類される．「パニック」は，霊の行動・要求に対してパニックを起こすことである．「反抗」は意識的に霊に反抗するが，「パニック」は逃げたり叫んだり意識を失ったりなど，無意識的に霊に反抗，または受容する場合を指す．これらの解決行動が成功すれば「成功」，失敗すれば「失敗」に分類される．

解説部分は「あり」と「なし」の 2 グループある．出現した霊に対して，生い立ちや目的などが明かされる文章がある場合は「あり」に，明かされる文章がない場合は「なし」に分類される．必ずしも解説は解決行動をとったあとに書かれてあるとは限らず，導入と予告の間に書かれてあることもあるため，見逃さないように気を付ける必要がある．

締め部分は「後日談」と「増長」の 2 グループある．一連の出来事の後，どのようになったかについての説明がある場合，例えば，「数日後，幽霊屋敷は取り壊され，二度と幽霊の噂が立つことはなかった」など，締めの最初が「数日後」や「それから」などで始まる文章がある場合は「後日談」に分類される．一連の出来事を，より強調して伝えようとする場合は「増長」に分類される．例えば，「これは本当にあった話なんだよ」や「友人と海に出かけ，写真を撮った．それから何日か経っ

た後，友人が亡くなった．後になって写真を見返すと，海から友人に向かって無数の手が伸びていた.」など，物語の最後に怖さを増す文章がある場合はこれに分類される．

（＊文責：松浦史佳）

4.1.1.2 プロット分類と登場人物の関係性

各プロット分類項目間と登場人物の関係を分析するため，因子分析を行った．因子分析は，多変量解析の手法の一つで，ある観測された変数がどのような潜在的な変数から影響を受けているか探る手法のことである．この手法は，複数の変数の関係性を基にした構造を探る際に頻繁に用いられる． 例として，因子分析はアンケート結果やテストの成績などのデータに用いることが多い．

因子分析の対象として，プロット分類項目と登場人物の属性を用いた．登場人物の属性は，主人公が男女どちらかという”性別”，主人公が成人か未成年かという”世代”，登場する霊が人型であることを表す”人”，鬼などのように化け物であることを表す”化け物”，地蔵や建物など場所や物であることを表す”場所・物”がある．

上記の属性を用いた因子分析の結果について記述する．因子分析は，R 言語を用いて行った．最尤法を用いて因子を抽出し，回転はプロマックス回転を用いた．また，因子数の決定には固有値 1 以上の基準を採用した．表 4.1.1.2 は，因子分析の結果である．数値は，因子分析によって求められた各属性の因子に対する因子負荷量で 0.1 以上の物を表している．通常は，0.4 以上のものを相関があるとして扱うが，今回は，物語の多様性などの要因によって因子負荷量の値が大きくならなかったため，0.2 以上の値をしようした．下記に各因子の解釈を記す．

表 4.1.1.2　因子分析の結果

		因子							
		1	2	3	4	5	6	7	8
主人公	性別	0.2						-0.2	0.2
	世代					0.2			
霊の形態	人	-0.2	0.1		-0.7	0.1			
	化け物	0.2			1				
	場所・物		-0.2		-0.2	-0.3			
導入	語り手			-1			-0.1		-0.4
	行動			1		0.1			-0.4
	現象					-0.1	0.1		1
予告	直接		1		-0.1				
	間接		-1						
出現	自力				0.3	-0.1			-0.1
	依頼				-0.2	0.6	0.2		
解決行動	受容	-0.2				1		0.3	
	反抗		0.2	0.1	0.2	-0.3	0.4	0.7	0.2
	パニック		0.1			-0.1		-0.6	
	成功		0.2			0.2	-0.4	0.8	0.1
	失敗					0.3	1	-0.2	0.1
解説	有無	0.1			-0.2				-0.1
締め	後日談	1			0.2				
	増長	-1			-0.2				

因子 1 は，性別が女性で霊の形態が化け物の時，締めが後日談になりやすいことを示す因子である． 逆に霊の形態が人であることと解決行動が受容であることは同時に起こりにくい．

因子 2 は，予告が直接であったときに，解決行動として反抗をとって成功しやすいことを示している． 逆に予告が間接であるということは，同時に起こりにくいことを示している．

因子 3 は，導入が行動であることと，導入が語り手であるということは，同時に起こらないということを示す因子であることを示している．

因子 4 は，霊の形態が化け物であったときに，出現が自力で解説がないことが多いことを示している． 逆に，霊が人または場所・物であるということと，出現が依頼であるということは，同時に起こりにくいことを示している．

因子 5 は，出現が依頼であったときに，解決行動が受容，反抗であると成功しやすいことを示している． 逆に，霊が場所・物であるということと，解決行動が失敗であることは同時に起こりにくいことを示している．

因子 6 は,出現が依頼であったときに,解決行動が反抗であると失敗しやすいことを示している．逆に，解決行動が成功であるということは同時に起こりにくいということを示している．

因子 7 は，解決行動が反抗，受容であったときに成功しやすいことを示している．逆に，解決行動が失敗であるということと，解決行動がパニックであるということは，同時に起こりにくいことを示している．

因子 8 は，性別が女性であったときに予告が直接になりやすいことを示している．逆に，導入が語り手であることと，導入が行動であるということは，同時に起こりにくいことを示している．

これらのことから分析した作品には，その物語らしい構造というものがあり，因子分析を用いることによってそれらの構造を明らかにすることができているということが分かる．

（＊文責：渡邊広基）

4.1.1.3 遷移確率モデルの基盤

遷移確率モデルを作成する前に，モデルの基盤を作るためにプロット分類のそれぞれの遷移数を手動で数え上げた．作品の中にはプロット分類のいずれにも分類しがたいようなあいまいな表現があったり，明記されていなくて分類できないものがあったりしたが，それらに関しては無視することとした．

それでは，まず初めに導入から予告への遷移の結果を示す.「語り手」から「直接」に遷移する作品は 21,「間接」に遷移する作品は 13 あった．また,「行動」から「直接」に遷移する作品は 20,「間接」に遷移する作品は 20 あった．また,「現象」から「直接」に遷移する作品は 8,「間接」に遷移する作品は 5 あった．

次に予告から出現への遷移の結果を示す.「直接」から「自力」に遷移する作品は 26,「依頼」に遷移する作品は 12 あった．また,「間接」から「自力」に遷移する作品は 16,「依頼」に遷移する作品は 7 あった．

次に出現から解決行動への遷移の結果を示す．解決行動は，その種類（受容，反抗，パニック）に遷移する数ではなく，その結果（成功，失敗）に遷移する数の数え上げを行った.「自力」から「成功」に遷移する作品は 14,「失敗」に遷移する作品は 12 あった．また,「依頼」から「成功」に遷移する作品は 9,「失敗」に遷移する作品は 7 あった．

次に解決行動から解説に遷移する結果を示す.「成功」から「解説あり」に遷移する作品は 17,「解説なし」に遷移する作品は 12 あった．また,「失敗」から「解説あり」に遷移する作品は 14,「解説なし」に遷移する作品は 7 あった．

最後に，解説から締めに遷移する結果を示す.「解説あり」から「後日談」に遷移する作品は 29,「増長」に遷移する作品は 22 あった．また,「解説なし」から「後日談」に遷移する作品は 16,「増長」に遷移する作品は 19 あった．

（＊文責：松浦史佳）

4.1.1.4 遷移確率モデル

分析作品に類似した展開となる物語を生成するために，プロット分類結果の遷移確率モデルとして，マルコフ連鎖モデルをベースにアレンジしたものを用いた．

初めに，分析作品においてプロット分類の項目がどのように連続して出現するかを分析した．また，その結果から分析作品に基づく遷移確率モデルを作成した．さらに，分析作品の遷移確率モデルを主人公の性別，世代，霊の形態などの初期値によって変化させるために，因子分析の結果を用いた．因子負荷量の大きさによって，対応する値を初期値から増減させた．これにより与えられた

初期値に対応する因子に含まれるプロット分類項目が出現しやすいように遷移確率を変化させ，分析対象における物語構造の特徴を反映した遷移を実現した．

図 4.1.1.4-1 は，分析作品の遷移確率モデルを主人公の性別，世代，霊の形態などの初期値によって変化させたものの一例である．この図では，主人公が男性で出現する霊が人の場合の各プロット分類からの遷移確率を表している．例えば 4 行目では，シーン 3 から 4 に遷移する確率は 0.32，5 に遷移する確率は 0.68 であることを表している．

```
1:0.43,2:0.53,3:0.04
1-4:0.32,1-5:0.68
2-4:0.2,2-5:0.8
3-4:0.32,3-5:0.68
4-6:0.48,4-7:0.52
5-6:0.5,5-7:0.5
6-8:0.64,6-9:0.36,6-10:0.26,6-11:0.74,6-12:0.26,6:13,0.74
7-8:0.84,7-9:0.16,7-10:0.86,7-11:0.14,7-12:0.26,6:13,0.74
8,10,12-14:0.69,8,10,12-17:0.31
9,11,13-14:0.77,9,11,13-17:0.23
14-18:0,14-19:1
17-18:0,17:19:1
```

図 4.1.1.4　主人公が男性で出現する霊が人の場合の遷移確率モデル

（＊文責：渡邊広基）

4.1.1.5　データ形式

上記の遷移確率モデルに合わせて，各プロット分類項目の内容を生成するための物語構造データベースを作成した．システムが読み込める形式にするため，登場人物の行動を一単位とし，各行動単位で 9 つの情報を表 3 に示した形式で付与した．

表 4.1.1.5　各行動単位でのデータの形式

情報	説明
場所	行動が起こっている場所
主体	行動の主体となっている人物
セリフ	行動で発したセリフ
役割	行動の動作主
ポーズ	行った動作
方向	動作主が向く方向
位置	動作主の位置
表情	動作主の表情
効果	動作主にかかる効果

（＊文責：松浦史佳）

4.1.2　バトル班

4.1.2.1　バトルジャンルの定義

物語を分析するにあたり，主人公や敵などの登場人物が戦闘を行うことで目的を達成していく物語をバトルジャンル物語と位置付けた．また，物語の展開を繋ぎ合わせて一つの物語を作ることを目標とし，そのために物語の展開をカテゴリ分けし，一般的な物語の展開を再構成する方法を用いた．

（＊文責：袴田翔）

4.1.2.2　バトルジャンルの採用理由

バトルジャンル物語を用いた理由として 2 点挙げられる．1 つ目はバトルジャンル及び対人戦闘を含む物語は，古代から数多くあり，かつ 1 つのジャンルとして評価されている点である．2 つ目はこれから先もこのバトルジャンル要素を含む物語は一定以上の評価を得られる可能性が高いと推測した点である．以上の 2 点からバトルジャンル物語を採用した．

（＊文責：袴田翔）

4.1.2.3　分析作品

バトルジャンル物語を自動生成するために『Fate/stay night』と『ジョジョの奇妙な冒険』の 2 作品を分析データとして用いた.『Fate/stay night』はバトルロイヤルを題材にしたノベルゲームであり，『ジョジョの奇妙な冒険』はアクション・アドベンチャー系統の少年漫画である．採用理由として，『Fate/stay night』は一つの物語の中に複数の展開が存在する点,『ジョジョの奇妙な冒険』は物語の展開が一つ一つ細かくまとめられている点，そして 2 作品とも映像化や発行部数の多さ，そして派生作品が複数販売されていることから一定以上の評価を得られている点が挙げられる．

物語の展開が複数存在することで，様々な戦闘パターンや登場人物の位置関係が抽出することができる．また物語の展開が細かくまとめられていることで，戦闘パターンを抽出しやすい．

具体的に示すと，『Fate/stay night』は主人公とヒロイン３人から構成されており，各ヒロインそれぞれで物語の展開が分岐し変化していく．それに応じて登場人物の役回り，主人公と登場人物の関係性が変化していく．また各ヒロインの物語の展開の終わり方も複数存在する．物語の終わり方が複数存在することにより，幅広い展開のパターンを得ることができる．

『ジョジョの奇妙な冒険』では物語が部の単位で区切られている．そのため，各部における物語の流れがつかみやすく，バトルジャンル物語の構造をモデル化しやすい．また登場人物が数多く存在し，各登場人物における戦闘のスタイルが異なるため，戦闘のパターンを多数抽出できる．

映像化や発行部数が多いこと，派生作品が複数販売されていることで一定の評価を得られていることは，分析作品の質が高いことを示している可能性が高い．

以上の点から 2 作品を分析データに用いた．

また，その他のバトルジャンル物語の定義を満たす漫画やゲーム作品からも物語の展開を少しずつ拝借した．

（＊文責：袴田翔）

4.1.2.4　データ形式

バトルジャンル物語の特徴として，物語を大きく分けて"戦闘シーン"と"それ以外の物語シーン"に分割できることが挙げられる．これら 2 種類のシーンには相互作用があり，戦闘シーンの展開によってその後の物語が変化する等の関連性がある．そこでデータ作成にあたっては戦闘シーンにおける物語の展開と戦闘以外の幕間のシーンにおける物語の展開を別個に分析し，ホラージャンルと同様のデータ形式での記述を行った．また幕間のシーンに関しては,戦闘前，勝利後，敗北後などに分けてプロットの構造を分析し，ホラージャンルの場合と同様の遷移確率モデルを作成した．

遷移確率モデルを採用した理由は，物語の流れは図２に示す通りだが，必ずしも【日常】から【戦闘】までの間は全てのカテゴリを通過するわけではないからである．そのため,【日常】から【戦闘】までの間を確率で遷移するカテゴリを変化させる．それにより物語の展開に幅を持たせることができた．

また物語のデータは Excel シートにまとめ，対戦相手や物語終了時の勝敗ごとにシートを分けた．シートの物語の内容は，主人公は必ず３回戦闘を行うことを条件として各シート【日常】から【勝利】もしくは【敗北】までのカテゴリを 1 セットとして 3 セット用意した．また 1 つのカテゴリにはそれぞれ最低でも 3 つの展開を記述してある．対戦相手の登場人数はシートによって異なり，対戦相手に数字を割り当てた. 対戦相手が 1 人しか存在しない物語の場合は”1_1_1”のように名前を付け，3 回目の戦闘で【勝利】へ遷移するシートと【敗北】へ遷移するシートがあり，計 2 種類用意した． 対戦相手が 2 人存在する物語の場合は 1 回目の戦闘で 1 人目の対戦相手に勝ち，2 回目の戦闘で 2 人目の対戦相手に負け，3 回目の戦闘で 2 人目の対戦相手に勝つもしくは負けるかを記述した”1_2_2”のシートと 1 回目の戦闘で 1 人目の対戦相手に負け，2 回目の戦闘で 1 人目の対戦相手に勝ち，3 回目の戦闘で 2 人目の対戦相手に勝つ又は負けるかを記述した”1_1_2”のシートがあり，計 4 種類用意した．対戦相手が 3 人存在する物語の場合は 1 回目の戦闘で 1 人目の対戦相手に勝ち，2 回目の戦闘で 2 人目の対戦相手に勝ち，3 回目の戦闘で 3 人目の対戦相手に勝つ又は負けるかを記述した”1_2_3”のシートがあり，計 2 種類用意した．

以上のシート 8 種類用意した．物語の展開数としては約 2500 億通りの物語が生成可能である．

シートを勝ち負けや対戦相手で分けた理由として 2 つある．1 つ目は 1 戦闘ごとの【勝利】と【敗北】を同じシートに記述すると，物語を自動生成する際にループが起きてしまう点である．2 つ目

は倒した敵が次の戦闘で再び現れるといったような物語に矛盾が起きる可能性，そして物語の内容が不自然になり，読み手からすると意味の分からない物語になる可能性がある点である．

以上の 2 点から事前にシートごとに勝敗を分け，その勝敗に合う物語の展開を用意した．

（＊文責：袴田翔）

4.1.2.5 分析

物語分析バトル班が行った分析方法について説明する．私たちは以下の手順で物語分析を行った．

1)分析作品の物語展開を始まりから終わりまで記述を行う．

2)物語上の役割に着目し，各展開をカテゴリに分類する．

3)カテゴリからカテゴリの遷移より，物語展開の遷移確率モデルを作成する．

1)の作業は，自動生成する際の主軸となる展開を抽出することが目的である．今回は物語上で戦闘を行うまでの展開・戦闘後から物語の終わりまでの展開に重点を置き記述を行った．分析の際，主体者・行動・対象・目的・行動結果の形式で記述した．

2)の作業は，分析作品群から共通の物語展開を見つけ出すために各展開をグループ化することが目的である．各カテゴリとその定義は表 4.1.2.5-1 の通りである．分析した作品群に含まれる各カテゴリの数は表 4.1.2.5-2 の通りである．表 4.1.2.5-2 から，【移動】・【人数】・【感情】・【強さ】・【姿】の含数が他のカテゴリに比べて少ないためこれらのカテゴリを【変化】に統合した．【変化】の定義は表 4.1.2.5-3 の通りである．統合後の分析作品群に含まれる各カテゴリの数は表 4.1.2.5-4 の通りである．

3)の作業は，分析作品群のカテゴリの遷移からバトルジャンル物語の展開をモデル化することが目的である．各カテゴリが遷移するカテゴリは表 4.1.2.5-5 の通りである．表 4.1.2.5-5 より各カテゴリで最も遷移数の多いカテゴリから, バトルジャンル物語の展開の基本遷移を, 次の 2 つに定めた．

- 戦闘の結果が勝利の場合

【始まり】→【日常】→【変化】→【問題】→【被害】→【戦闘】→【勝利】→【解決】→【終わり】

- 戦闘の結果が敗北の場合

【始まり】→【日常】→【変化】→【問題】→【被害】→【戦闘】→【敗北】→【終わり】

なお今回は簡単化のために同カテゴリへの遷移をひとつのシーンとしてまとめ，遷移は【始まり】から【終わり】に向かう一方向のみとした．また，分析作品群の一部では，

【始まり】→【日常】→【被害】→【戦闘】→【敗北】→【終わり】

のように，基本遷移とは異なる物語展開が存在していた．物語の自動生成を行う際，この物語展開を実現するために，遷移するカテゴリを確率で選択する方法をとった．基本遷移を基に，遷移方向は基本遷移に従い，以下の条件を満たすカテゴリを遷移可能なカテゴリとする．

1. 遷移先のカテゴリは遷移元のカテゴリとは異なる
2. 戦闘】前のカテゴリ(【始まり】，【日常】，【変化】，【問題】，【被害】)は必ず【戦闘】に遷移する．
3. 【戦闘】後のカテゴリ(【勝利】，【解決】，【敗北】，【終わり】)は必ず【戦闘】から遷移する．

各確率は次のように求めた．

(遷移先のカテゴリの遷移数)／(すべての遷移可能なカテゴリの遷移数の合計)

計算結果は表 4.1.2.5-6 の通りである．

この結果から求めたバトルジャンル物語の遷移カテゴリモデルが図 4.1.2.5-7 である．図 4.1.2.5-7 は現在のカテゴリが【日常】であった場合，0.6 の確率で【変化】，0.15 の確率で【問題】，0.2 の確率で【被害】，0.05 の確率で【戦闘】へ遷移することを表している．

表 4.1.2.5-1　各カテゴリの定義

カテゴリ	定義
【始まり】	世界観や主人公に関する説明がされる展開.
【日常】	物語開始時の状態が維持されている展開.
【移動】	【日常】の状態から場所が変化する展開.
【人数】	【日常】の状態から人数が変化する展開.
【感情】	【日常】の状態から感情が変化する展開.
【強さ】	【日常】の状態から登場人物の強さが変化する展開.
【姿】	【日常】の状態から登場人物の姿が変化する展開.
【問題】	主人公の行動を妨害する障害，または物語の目的が登場する展開.
【被害】	主人公，及びその所属集団に対して，負の影響をもたらす展開.
【戦闘】	主人公が戦闘行為を行う展開.
【勝利】	主人公が【戦闘】で勝利する，もしくは戦闘によって正の影響があった展開.
【敗北】	主人公が【戦闘】で敗北する，もしくは戦闘によって負の影響があった展開.
【解決】	【問題】で登場した障害を乗り越える，もしくは目的を達成する展開.
【終わり】	物語の終幕展開.

表 4.1.2.5-2　分析作品群での各カテゴリの含数

カテゴリ	含数
【始まり】	180
【日常】	112
【移動】	44
【人数】	40
【感情】	33
【強さ】	20
【姿】	13
【問題】	94
【被害】	159
【戦闘】	179
【勝利】	92
【敗北】	88
【解決】	72
【終わり】	180

表 4.1.2.5-3　【変化】の定義

カテゴリ	定義
【変化】	【日常】の状態が変化する展開.

表 4.1.2.5-4　【変化】に統合後の分析作品群での各カテゴリの含数

カテゴリ	含数
【始まり】	180
【日常】	112
【変化】	150
【問題】	94
【被害】	159
【戦闘】	179
【勝利】	92
【敗北】	88
【解決】	72
【終わり】	180

表 4.1.2.5-5　各カテゴリが遷移するカテゴリ先

遷移先 カテゴリ	【始まり】	【日常】	【変化】	【問題】	【被害】	【戦闘】	【勝利】	【敗北】	【解決】	【終わり】
【始まり】	0	83	0	0	0	19	0	0	0	0
【日常】	0	-10	37	6	11	4	2	0	1	0
【変化】	0	7	-62	32	31	32	9	17	10	3
【問題】	0	3	10	-4	63	9	1	3	2	0
【被害】	0	3	12	22	-21	56	1	23	4	4
【戦闘】	0	3	11	9	13	-44	59	28	0	0
【勝利】	0	0	5	4	2	5	-16	2	47	12
【敗北】	0	0	12	13	16	10	2	-14	5	38
【解決】	0	3	1	4	2	0	2	1	-3	112
【終わり】	0	0	0	0	0	0	0	0	0	-11

表 4.1.2.5-6　各カテゴリが各遷移可能カテゴリに遷移する確率

遷移先 カテゴリ	【始まり】	【日常】	【変化】	【問題】	【被害】	【戦闘】	【勝利】	【敗北】	【解決】	【終わり】
【始まり】	-	0.81372549	0	0	0	0.18627451	-	-	-	-
【日常】	-	-	0.63793103	0.10344828	0.18965517	0.06896552	-	-	-	-
【変化】	-	-	-	0.33684211	0.32631579	0.33684211	-	-	-	-
【問題】	-	-	-	-	0.875	0.125	-	-	-	-
【被害】	-	-	-	-	-	1	-	-	-	-
【戦闘】	-	-	-	-	-	-	0.67816092	0.32183908	0	0
【勝利】	-	-	-	-	-	-	-	-	0.79661017	0.20338983
【敗北】	-	-	-	-	-	-	-	-	-	1
【解決】	-	-	-	-	-	-	-	-	-	1
【終わり】	-	-	-	-	-	-	-	-	-	-

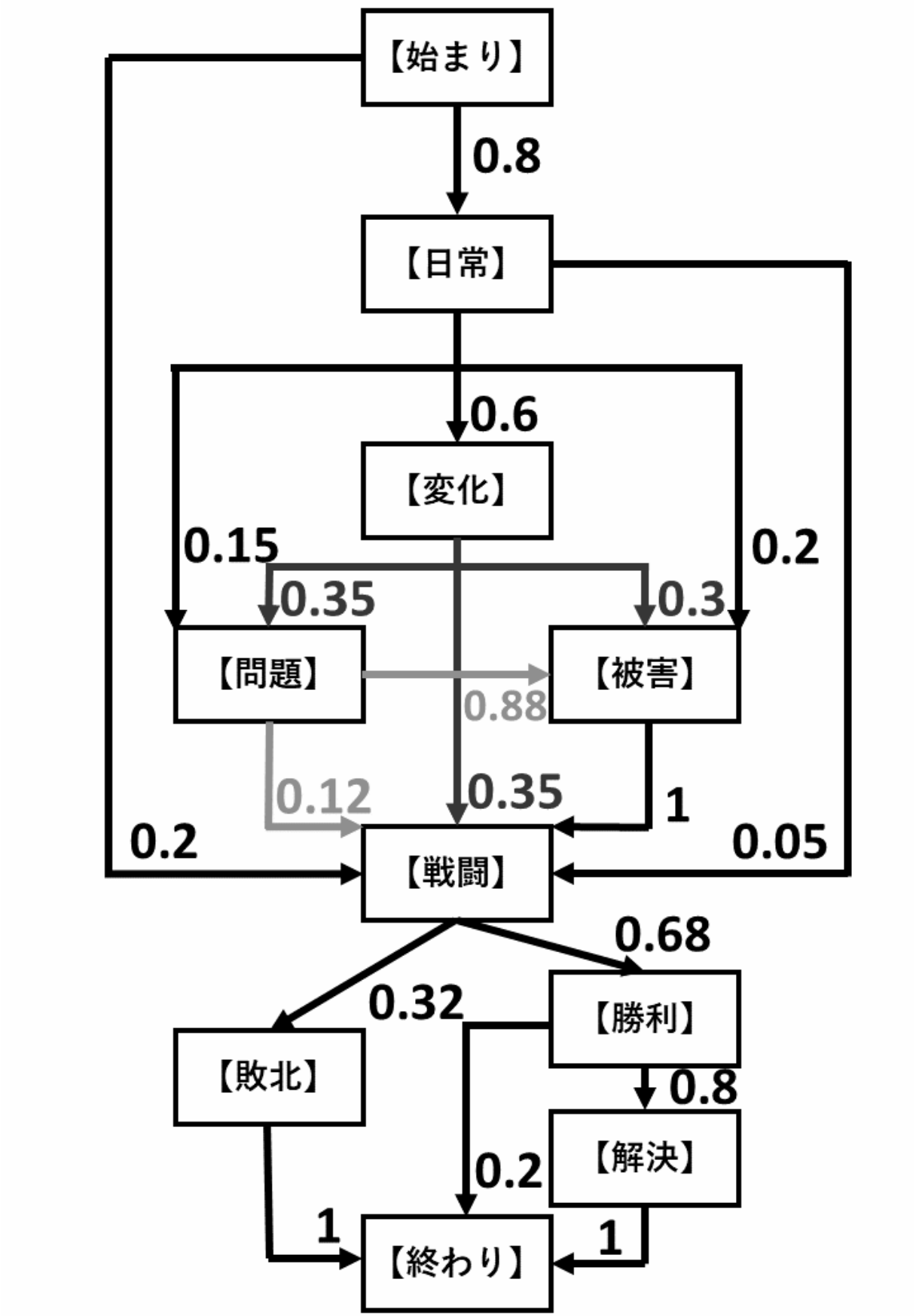

図 4.1.2.5-7　バトルジャンル物語の遷移カテゴリモデル

（＊文責：吉田拓海）

4.2 システム班

4.2.1 Python 班

4.2.1.1 ホラーシステム

ホラージャンル物語プロット生成では，ホラー物語の分析結果である物語構造データベースと遷移確立モデルを基に分類されたプロット同士を結びつけ，また分析対象の既存ホラー物語における展開の特徴的なパターンを反映するための機能を実現することでホラージャンルらしいプロットを自動で生成するシステムの開発を目的とした． ホラージャンルの物語の特徴として，主人公の属性と霊の属性，また，特定の役割を持ったプロット内で主人公の行動に応じてその後の物語の展開に特徴的なパターンが見られることが分析によって明らかになった．この物語展開のパターンをシステム内で実現することでプロットにホラーらしさを持たせることが出来ると考えられる．具体的な機能としては，登場人物によって物語の展開の遷移確立が変化する機能の実装，登場人物をユーザーがインタラクティブに選択するための GUI による設定画面の作成し，また，物語表示中にユーザーに次の展開の選択肢を与える機能の実装を行った．理論上 248,832 種類の物語パターンを生成可能である．以下では,作成したシステムをデータの入力から物語プロットの出力に分けて解説する．

（＊文責：寺島啓悟）

4.2.1.1.1 データ入力

分析班が用意したエクセルデータをデータベースに格納するまでを担当する．データベースのテーブル型には，中間までの物にいくつか要素を追加したもの使用している．また，エクセルデータには，データベースのテーブル型に対応したものを使用している．データベースのテーブル型の定義 SQL 文を以下に示す.

- CREATE TABLE Scene(sceneID,場所,選択肢);
- CREATE TABLE Cut(sceneID,cutID,主体,音楽,セリフ);
- CREATE TABLE Character(cutID,役割,動作,方向,位置,表情,効果);
- CREATE TABLE Markov(sceneID,A,B,C,D,E,F);

Scene テーブル，Cut テーブル，Character テーブル 3 つのテーブルがある．まず，Scene テーブルには，sceneID，場所，選択肢の 3 つのカルムがある．
sceneID では，そのデータの sceneID を示してある．場所では，そのシーンで使用する場所モデルを示してある．選択肢は，物語再生中での選択肢表示に使用する予定であったが現在はこのカルムは使用されていない.

次に，Cut テーブルには，sceneID,cutID,主体,音楽,セリフの 3 つのカルムがある．sceneID では，そのデータの sceneID を示しており，これは Scene テーブル sceneID と対応している．cutID では，そのデータの cutID を示してある．主体では，そのカットにおける主体となる登場人物が示してある．音楽では，そのカットで流れる音楽データを示す予定であったが，現在は使用されていない．セリフでは，そのカットで表示する文字列が示されている．

次に，Character テーブルでは，cutID,役割,動作,方向,位置,表情,効果の 7 つのカルムがある．cutID では，そのデータの cutID を示しており，Cut テーブルの cutID と対応している．役割では，表示する登場人物を示している．動作では，表示している登場人物のポーズを示している．方向では，表示している登場人物の向いている方向を示している．位置では，表示している登場人物の位置を示している．表情では，表示している登場人物の顔の表情を示している．効果では，表示している登場人物に対して掛かるエフェクトを示している．

最後に，Markov テーブルには sceneID,A,B,C,D,E,F の 7 つのカルムがある．sceneID では，そのデ

ータの sceneID を示しており，これは Scene テーブルの sceneID と対応している．A,B,C,D,E,F の 6 つのカルムではそのデータのプロットの展開の次の展開の役割に属するプロットに対しての遷移確立を示している．例として，現在のデータが物語の予告に属している場合，予告に属しているシーンの遷移先シーンは出現に属しているシーンとなっているので，A に格納されるデータが，物語の出現に属している 1 つ目のシーンへの遷移確立，B に格納されるデータが，物語の出現に属している 2 つ目のシーンへの遷移確立となる，出現に属しているシーンは 2 種類に分類されているため残りの C，D，E，F の 4 つのカルムが空欄となる．エクセルデータからデータベースに格納する方法として，Scene テーブル，Cut テーブル，Character テーブルでは Python を用いて中間発表までに作成されたプログラムを使用している．また，Markov テーブルへのデータの格納は分析班の分析結果であるシーンの遷移確立に基づいて SQL 文で直接行った．

（＊文責：寺島啓悟）

4.2.1.1.2　物語プロット出力

データベースに格納されているデータと，システムの実行者が GUI を用いて設定した情報から Unity で読み取りを行うための物語プロットを自動生成するところまでを担当する．システムの流れとして，まず，システムの実行者が GUI の指示に従って登場人物 4 人の性別，世代と形状を選択し，それぞれに任意の名前を与える．次に，その情報を考慮し，分析班があらかじめ用意した物語のシーンデータを分析結果から得られた遷移確立のデータを基に物語の始まりから終わりまで順番に結び付けて，物語プロットとして使用するシーンデータを選出する．最後に，選出されたシーンデータを実際にテキストファイルに Unity 班と合同で定めた形式で書きだす．以下では，GUI の作成，シーンの選出，テキストの書き出しに分けて詳細に解説する．

（＊文責：寺島啓悟）

4.2.1.1.2.1　GUI

システムユーザーが物語の初期設定を行うために用いる GUI の作成を行った．ホラーシステムの GUI を実装するにあたっては，Python に標準搭載されている Tkinter を用いた．Tkinter とは，簡単に GUI を組むことができるツールキットである．GUI は物語を生成するための情報をユーザーが入力するための登録画面のみを作成した．登場人物ごとに登録画面が用意され，ユーザーはモデルをプルダウンで選択し，そのモデルに名前をつけるために好きな名前を入力する．なお，モデルは表示ボタンを押すことで外見を確認でき，消すボタンを押すことで消せる．最終的に決定ボタンを押すと，その登場人物についての設定がシステムに入力される仕組みである．

（＊文責：鈴木諒輔）

4.2.1.1.2.1　シーンの選出

ここでは，物語プロットとして使用するシーンデータの選出を行う．ホラージャンル物語では，物語を 6 つのシーンに分類されており，それぞれ別の役割を持っている．シーンデータにはそれぞれ SceneID が与えられているので SceneID を 6 種類の役割に分類した．例として，導入の役割を持ったシーンを 3 つ，予告の役割を持ったシーンを 2 つ用意し，順番に SceneID を 1 から 5 まで与えた場合，SceneID の 1 から 3 までが導入のシーン，4 から 5 までが予告のシーンと表すようになる．このことによって，1 つの役割に対してのシーンの数を増やすことによって物語のパターン数を際限なく増やすことが可能となる．例として，SceneID の 6 のシーンを導入と指定することによって導入のシーンが 1 つ増えたことになる．このように，シーンに役割を持たせた後，6 種類の役割の持ったシーンからそれぞれ 1 つずつシーンを選出していく．選出する方法として，分析結果から得

られたシーンの遷移確立を使用した．遷移確立はそれぞれのシーンに対応したデータを Markov テーブルに格納した．さらに，ホラージャンル物語では主人公の性別と世代，霊の形態，特定の役割を持ったシーンでの主人公の行動によって物語の展開が変化する．登場人物の属性によって物語を変化させるために主人公の性別と世代と霊の形態の関係に応じた遷移確立をあらかじめ Markov テーブルに格納しておき，主人公の性別と世代と霊の形態の関係に応じて使うデータを変更するようにした．また，主人公の行動によって物語の展開を変化させるために，特定の役割のシーンに遷移する確率を主人公の行動によってそれぞれ与え，行動の種類分シーンを選出し，そのシーンから次のシーンをそれぞれ選出することで行動の種類分の展開が得られた．以上のことより，物語プロットで使用する SceneID が選出される．

（＊文責：寺島啓悟）

4.2.1.1.2.1 テキスト書き出し

ここでは，選出された SceneID に対応しているデータをデータベースから引き出し，Unity で読み込むためのプロットデータとしてテキストファイルを作成するまでを行う．

使用する SceneID が選出されたら，次に SceneID に対応したプロットを Scene テーブル，Character テーブル，Cat テーブルから選出する．これを選出した SceneID の数だけ繰り返すことで 1 つのプロットデータで使用する文章を決定することが出来る．具体的な方法として，SceneID の 1 がプロットデータとして使うと選出されたら，まず，Scene テーブルには SceneID のカルムに 1 が格納されているデータが有るのでこのデータを使用する．次に Cut テーブルにも SceneID のカルムに 1 が格納されているデータが有るのでこのデータを使用する．最後に，Cut テーブルには CutID というカルムが有るので Cut テーブルから使用するデータの CutID の値と同じ値を持っている Character テーブルのデータを使用する．これより，SceneID の 1 として使うデータをデータベースから引き出すデータを決定することが出来る．さらに，テキストデータとして順番に書き出すために，使用するデータの ID を順番にテーブルごとのリストに格納する．リストに ID を格納する際には，Cut テーブル用のリストと Character テーブル用のリストにはシーンの区切りを判別できるように区切りごとに c を格納する．

データベースから引き出すデータが決定され，ID が順番にリストに格納されたので，定めた形式でテキストファイルとして書き出す．定めた形式は，中間のまでの形式から変更点が有る．テキストデータの例を下記の図に示す．

```
太郎,男性 A
貞子,化物
p 洋館,1
r1,a 立つ,dE,f3,e 苦しみ,k
r2,a 立つ,dW,f5,e 余裕,k オーラ
g1,s 太郎は貞子と遭遇した.
c どうする?,振り向く. :22,振り向かずに進む. :34
p 洋館,22
j17
p 洋館,34
j19
p 洋館,17
z
end
```

図 4.2.1.1.2.1　ホラージャンル物語プロットデータ

この図から中間以降に追加された仕様を説明する．

まず，4 行目では e 記号と k 記号が追加された．e 記号ではキャラクターの表情を指定できる．k 記号ではキャラクターにかかるエフェクトを指定できる．

次に，7 行目の c 記号が追加した記号である．この行では，物語の分岐の指定が出来る．まず，「どうする?」が選択肢への質問文である．次に，「振り向く．」が SceneID の 22 へ進む選択肢で，「振り向かずに進む．」が SceneID の 34 へ進む選択肢となっている．

次に，9 行目の j 記号が追加した記号である．この行では，指定したシーンへシーンを飛ばして読むことが出来る．ここでは SceneID の 17 へ進む．

次に，13 行目の z 記号は中間以降で機能が変更している．この行では，1 つの分岐した物語の終わりを示している．この記号が出現したら再生している物語が終了する．

最後に，14 行目の end 記号ではテキストデータの最後の行を示す．テキストファイルに書き出す方法は物語の自動生成に伴い中間以降では変更してある．

まず，GUI を用いて得た情報を先頭の行に登場人物の数だけ出力する．

次に，使用する SceneID のリストの先頭の ID のデータを Scene テーブルから出力する．

次に，そのシーンに対応する Character データを出力するために，使用する Character の行番号のリストの値に c が出るまで順番に対応するデータを Character テーブルから出力する．

最後に，使用するシーンの Cut データを出力するために，使用する CutID のリストの値の 1 行だけ出力する．そのリスト値の次の値が c でなければ，再び使用する Character の行番号のリストの値に c が出るまで順番に対応するデータを Character テーブルから出力する．その後，Cut 情報の出力に進み，使用する CutID のリストの次の値が c であれば Scene データの出力に戻る

以上のアルゴリズムでテキストデータの出力を行う．また，このアルゴリズムの途中で選択肢の出力や，シーンのジャンプ情報の出力が有るので，適宜特定の役割を持ったシーンの後に出力する．ホラージャンルの物語では霊の出現シーンの後の主人公の行動によって物語の展開が分岐するため，そのシーンの後に選択肢情報を出力する．

（＊文責：寺島啓悟）

4.2.1.2　バトルシステム

バトルジャンル物語は，前述のとおり物語が"戦闘シーン"と"それ以外の物語シーン"(以下，通常シーン)に分割できる．これら 2 種類のシーンには相互作用があり，戦闘シーンの展開によってその後の物語が変化する等の関連性がある．そのため，バトルジャンルのプロットを生成するにあたって，この 2 種類のシーンを表現しつつ，お互いに関連性を持たせる必要がある．例えば，戦闘で主人公が敗北をしてしまった場合，主人公が敗北したシーンを次に挿入される通常シーンとして呼び出す等の処理が必要になる．

今回実装するシステムでは，中間報告までに開発したシステムを拡張し，戦闘シーンと通常シーンが互いに連動する物語の生成を目標とした．今回実装したシステムを実行して生成される物語の構造を図 4.2.1.2 に示す．今回生成される物語は，通常シーンと戦闘シーンが繰り返されることで 1 つの大きな物語を表現している．具体的には戦闘シーンが 3 回用意され，その前後の幕間として通常シーンが挿入されるという形式になっている．各戦闘シーンの勝敗結果等を元に，次に挿入される通常シーンを変化させることで,戦闘シーンと通常シーンの相互作用を表現するようにしている．

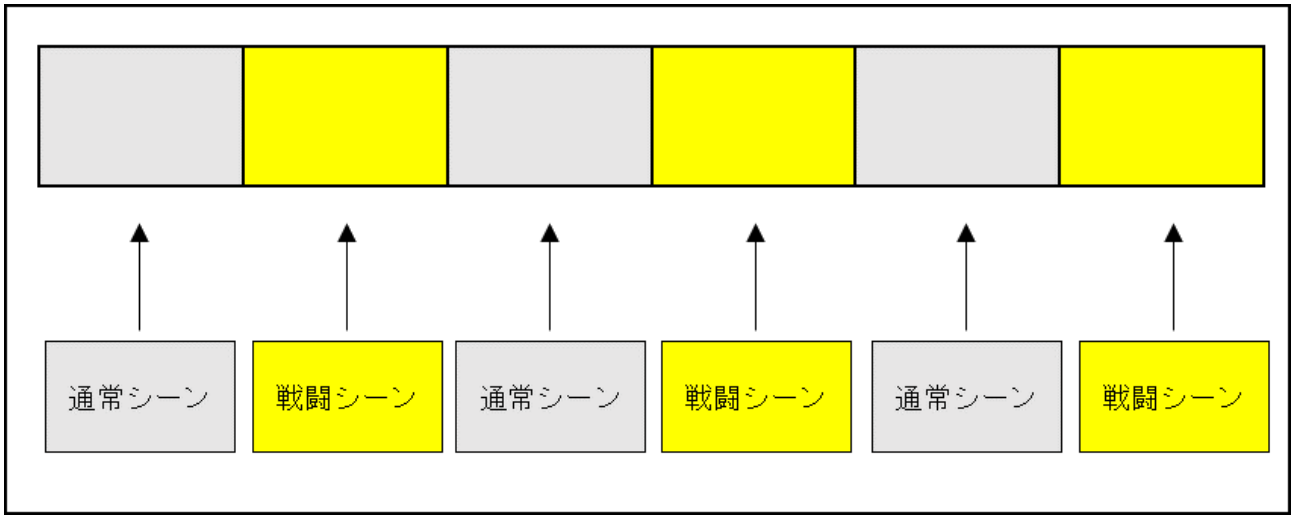

図 4.2.1.2　生成されるバトルジャンル物語の構造

上記の形式の物語生成を行うために，今回は「通常シーン生成システム」と「戦闘シミュレーション」の 2 つに大きくシステムを分割し，それぞれ開発を行った．

（＊文責：山田康貴）

4.2.1.2.1　通常シーン生成システム

4.2.1.2.1.1　通常シーン生成システムの概要

通常シーンの生成システムでは，戦闘シーンと戦闘シーンの幕間に挿入されるシーンの生成を行う．通常シーンは直前の戦闘結果によって呼び出す場面を変え，物語全体としての矛盾を無くす必要がある．その為に，物語分析班から提案されたアルゴリズムを元にシステムの開発を行った．

今回実装したシステムでは，物語分析班が用意した物語のシーンが格納されたファイル（以下，物語シート）と各シーンに移動する確率が記入された遷移確率行列シートの 2 つのファイルを用いて生成を行う．

遷移確率行列に記入されているデータの例を表 1 に示す．遷移確率行列シートには，シーンからシーンに遷移する確率が記入されている．この遷移確率行列を元に通常シーンを呼び出すことにより，物語間での矛盾が起きないようにしている．

表 4.2.1.2.1　遷移確率行列の一例

	日常1	変化1	問題1	被害1	バトル1
始まり	0.6	0.3	0	0.1	0

物語シートの内部には場面を構成するデータが記述されている．場面は 1 シート内に複数記述されており，物語シートを参照し場面を 1 つ抽出することで通常シーンが生成される．場面の抽出はランダムに行われる．

物語シートには場面の意味がネームされている．表 1 に記入されている「日常 1」や「変化 1」のように，その場面がどのような意味に属するのかを表している．遷移確率行列シートからどの場面に遷移するかを決め，物語シートから場面の抽出を行い通常シーンを生成する．この工程を繰り返すことで物語に通常シーンが生成されていくアルゴリズムになっている．

（＊文責：山田康貴）

4.2.1.2.1.2　通常シーン生成システムにおけるクラスの定義

クラス名:storyDB

メソッド名:readExcel2
機能:物語シートを読み取り，バッファとしてデータベースに書き込むメソッド
引数:読み取る Excel ファイル名,読み取る物語シート名
返り値:なし

メソッド名:readPro2
機能:遷移確率行列シートを読み取り，リストに格納するメソッド
引数:読み取る Excel ファイル名,読み取る遷移確率行列シート名
返り値:リストとして格納された遷移確率行列

メソッド名:selectScene2
機能:readExcel2 で得られたデータベースから，シーンを 1 つランダムで抽出するメソッド
引数:使用済みのシーン ID,使用済みのカット ID,1 つ前のシーンで描写された場所,主人公の名前
返り値:使用したシーン ID,使用したカット ID,描写で使用した場所

メソッド名:selectNextSheet
機能:readPro2 で得られた確率を元に，次に遷移する物語シートを選定するメソッド
引数:readPro2 で得られた遷移確率
返り値:次に遷移する物語シート名

メソッド名:insertBattle
機能:戦闘シミュレーションで得られた戦闘シーンを挿入するメソッド
引数: 使用済みのシーン ID,使用済みのカット ID,1 つ前のシーンで描写された場所,戦闘の回数
返り値:使用したシーン ID,使用したカット ID

（＊文責：山田康貴）

4.2.1.2.1.2 通常シーン生成システムの詳細説明

通常シーン生成システムのフローチャートを図 4.2.1.2-2 に示す．通常シーンの生成は物語シートと遷移確率行列シートの 2 つの呼び出しを繰り返すことで生成される．物語シートには物語出力に必要なデータが集積されており，遷移確率行列シートにはどの物語シートに遷移するかの情報が格納されているため,「遷移確率行列シートでシートを決定→決定した物語シートを読み取る」という流れが基本動作となっている．

プログラムが開始すると，まず物語シート内の[始まり]と呼ばれるシートを読み取る．この中からシーンを 1 つ抽出し，次に[始まり]からどの物語シートに遷移するかを決定する．その為に遷移確率行列シートを読み取り，その確率によって次の物語シートを選定する．ここで，選定されたシート名が[終わり]だった場合はプログラムを終了する．そうでなかった場合は，そのシートが戦闘直前のシート(以下,バトルシート)だった場合の判定を行う．

バトルシートには[バトル 1],[バトル 2]のように”バトル”+戦闘回数のようなシート名が付けられている．読み取られたシーンがバトルシーンだった場合は，そのシートを読み取った後に戦闘シーンを挿入する.ここで挿入する戦闘シーンは,戦闘シミュレーションから得られた物になっている．その後，戦闘結果に応じて次に遷移する物語シートを指定する．その後は図 2 の①に戻り，処理を繰り返し実行する．

読み取られたシートがバトルシートではなかった場合は，そのまま何もせずに①に戻り，こちらも繰り返し処理が実行される．これらの処理を[終わり]シートが読み取られるまで繰り返し，プロットデータの生成を行う．

[終わり]シートが読み取られた後は，生成されたプロットデータの記号化を行い，最終的にはテキストデータで出力する．

以上が通常シーン生成システムの全体の流れである．

（＊文責：山田康貴）

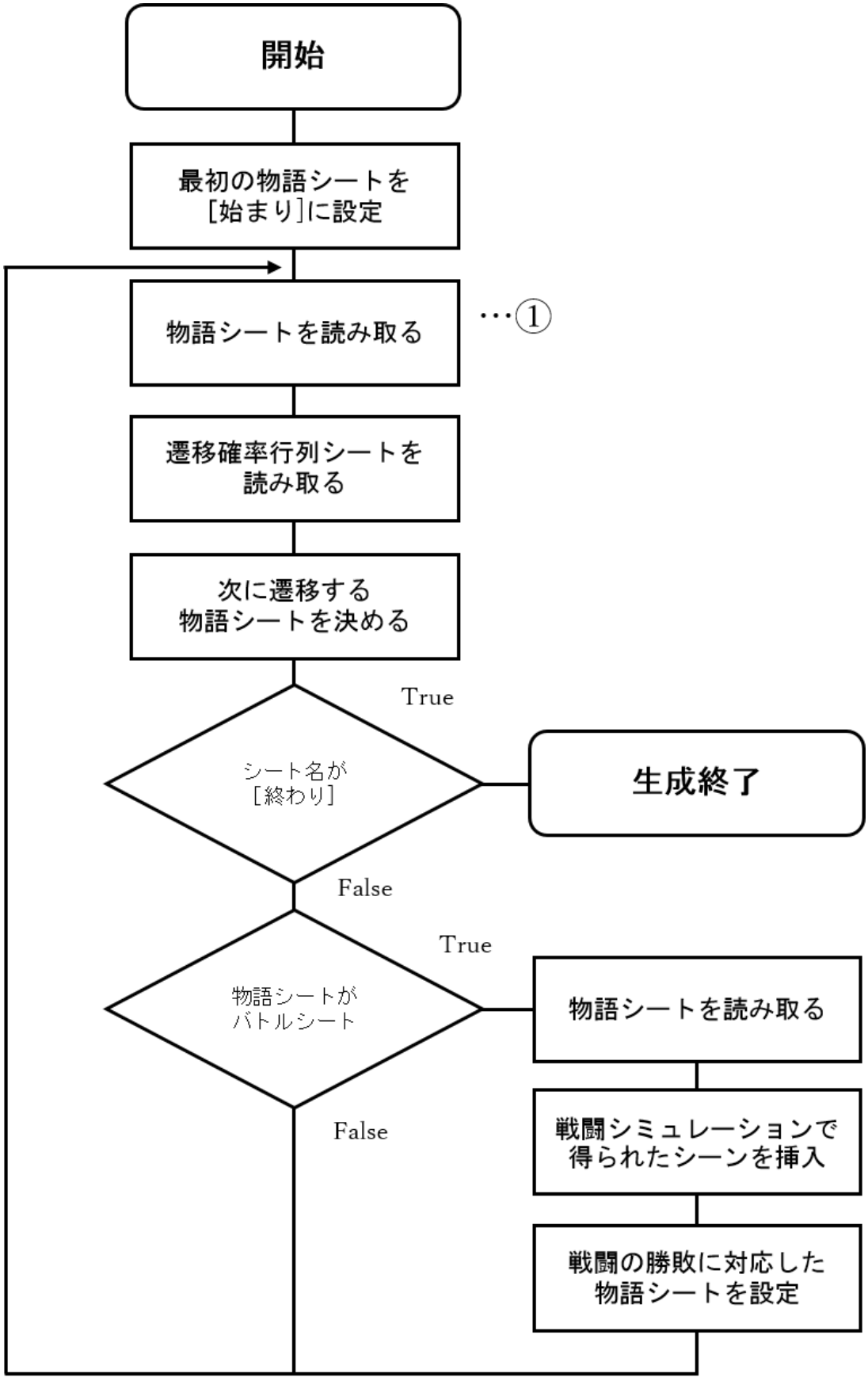

図 4.2.1.2.1.2　通常シーン生成システムのフローチャート

（＊文責：山田康貴）

4.2.1.2.2　戦闘シーンの生成システム

4.2.1.2.2.1　戦闘シミュレーションの概要

戦闘シミュレーションでは，通常シーンと通常シーンを大きく区切り，物語を分岐させるための戦闘シーンの生成を行う．

図 4.2.1.2.2.1 に今回実装を行った戦闘シミュレーションの全体像を示す．戦闘シミュレーションには，「主人公と敵」のような 2 人の情報を入力する．その後，その 2 人を実際に戦わせた場合のシミュレーションを行い，戦闘の過程と勝敗結果を出力する．その後，得られた結果を Unity 班に渡す物語データに直接組み込む．これにより，戦闘の表現が可能になる．

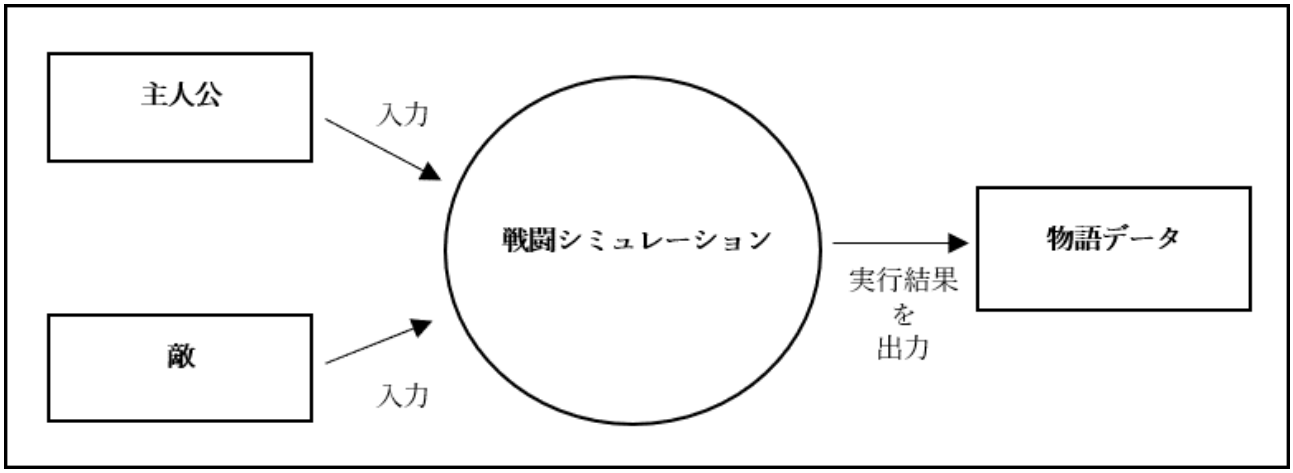

図 4.2.1.2.2.1　戦闘シミュレーションの全体像

この手法を取る事により，主人公と敵の状態によって戦闘の展開を大きく変えることができる．また，結果に対応して通常シーンの分岐も可能であるため，戦闘と物語の関連性を高めることもできる．それに加え，今後追加したい要素が出現した場合であっても容易に改変が可能であるため，この手法は非常に拡張性も高い．

詳しい内容は後述するが，戦闘形式については物語分析班から得られた形式を採用している．

（＊文責：山田康貴）

4.2.1.2.2.2　戦闘シュミレーションにおけるクラスの定義

クラス名:database

メソッド名:setup

機能:データベース(Status,Skill)に各キャラクターの初期値を入力するメソッド

引数:なし

返り値:なし

メソッド名:inlist

機能:データベースに格納した値をリストに組み込むメソッド

引数:なし

返り値:なし

メソッド名:stUpdate
機能:各戦闘が終了する毎に主人公と敵のステータスを更新するメソッド
　　戦闘前の状態に戻すことと，次の敵の情報に更新することが可能
引数:なし
返り値:なし

クラス名:parts
メソッド名:getHName
機能:主人公の名前のゲッターメソッド
引数:なし
返り値:主人公の名前

メソッド名:getEName
機能:敵の名前のゲッターメソッド
引数:なし
返り値:敵の名前

メソッド名:getHHP
機能:主人公の体力のゲッターメソッド
引数:なし
返り値:主人公の体力

メソッド名:getHDP
機能:主人公の防御力のゲッターメソッド
引数:なし
返り値:主人公の防御力

メソッド名:getHEvaP
機能:主人公の回避率のゲッターメソッド
引数:なし
返り値:主人公の回避率

メソッド名:getEHP
機能:敵の体力のゲッターメソッド
引数:なし
返り値:敵の体力

メソッド名:getEDP
機能:敵の防御力のゲッターメソッド
引数:なし
返り値:敵の防御力

メソッド名:getEEvaP
機能:敵の回避率のゲッターメソッド
引数:なし
返り値:敵の回避率

メソッド名:getListH
機能:主人公の状態リストのゲッターメソッド
引数:なし
返り値:主人公の状態リスト

メソッド名:getListE
機能:敵の状態リストのゲッターメソッド
引数:なし
返り値:敵の状態リスト

メソッド名:setHHP
機能:主人公の体力のセッターメソッド
引数:変更後の体力値
返り値:なし

メソッド名:setEHP
機能:敵の体力のセッターメソッド
引数:変更後の体力値
返り値:なし

メソッド名:getHmodel
機能:主人公のモデル名のゲッターメソッド
引数:なし
返り値:主人公のモデル名

メソッド名:getEmodel
機能:敵のモデル名のゲッターメソッド
引数:なし
返り値:敵のモデル名

メソッド名:getHWepName
機能:主人公の所持している武器の名称のゲッターメソッド
引数:なし
返り値:主人公の武器名

メソッド名 getEWepName:
機能:敵の所持している武器の名称のゲッターメソッド
引数:なし
返り値:敵の武器名

メソッド名:getHWepKind
機能:主人公の所持している武器の種類のゲッターメソッド
引数:なし
返り値:主人公の武器の種類

メソッド名:getEWepKind
機能:敵の所持している武器の種類のゲッターメソッド
引数:なし
返り値:敵の武器の種類

クラス名:turn
メソッド名:selectActHero
機能:各ターンにおける主人公の行動を決定するメソッド
引数:なし
返り値:主人公の行動(0:攻撃,1:防御,2:回避)

メソッド名:selectAceEne
機能:各ターンにおける敵の行動を決定するメソッド
引数:なし
返り値:敵の行動(0:攻撃,1:防御,2:回避)

メソッド名:judge
機能:戦闘の勝敗を判定するメソッド
引数:なし
返り値:戦闘の勝敗(1:敵の勝利,2:主人公の勝利)

メソッド名:Action
機能:シミュレーションにおける各ターンの処理を行い，その記号化を行うメソッド
引数:主人公の行動,敵の行動,使用済みの sceneID,使用済みの cutID
返り値:使用した sceneID,使用した cutID

クラス名:simulation
メソッド名:battle
機能:turn クラス内のメソッドを使用し，戦闘シミュレーションを行うメソッド
引数:使用済みの sceneID,使用した cutID,直前に描写された場所,戦闘の勝利回数
返り値:戦闘の勝利回数,使用した sceneID,使用した cutID,戦闘結果

（＊文責：山田康貴）

4.2.1.2.2.3　戦闘シミュレーションの詳細説明

今回実装を行った戦闘シミュレーションのフローチャート図 4.2.1.2.2.3-3 に示す．戦闘シミュレーションでは，戦闘シーンの描画を行うために，主人公と敵の情報を繰り返し参照と書き込みを行う必要がある．その為に parts クラスのゲッターとセッターを利用する．parts クラスを元にターン毎の処理を turn クラスに構築し，simulation クラスで実際に戦闘シミュレーションとその結果と出力を行っている．

戦闘シミュレーションの内部では，最初に各キャラクターの初期値を設定する．キャラクターに与える情報を表 4.2.1.2.2.3-1 及び表 4.2.1.2.2.3-2 に示す．主人公と敵には,勝敗の判定に必要な体力,相手が繰り出してくる技に対しての抵抗となる防御力,相手の技をどれだけ回避できるかの回避率の 3 つの値を与える．また，それとは別に各キャラクターには使用することができる技の情報を与える．技の情報としては,技の識別に必要な技名，映像化する際に必要になってくるモーション名，その技が相手にどれだけのダメージを与えることができるかの攻撃力，その技の当たりやすさの度合いを示した命中率がある．これらの情報が今回の戦闘シミュレーションでは必要になってくる．

表 4.2.1.2.2.3-1　シミュレーションの初期値(キャラクター)

与える情報	説明
体力	キャラクターの体力を表す値
防御力	相手の攻撃力に対しての抵抗を表す値
回避率	攻撃の回避しやすさの値

表 4.2.1.2.2.3-2　シミュレーションの初期値(技)

与える情報	説明
技名	技の識別に使用
モーション名	動きの映像化に必要
攻撃力	与えるダメージの大きさの値
命中率	攻撃の当たりやすさを示す値

初期値にセット後はシミュレーションが実行される．シミュレーションでは，まず各キャラクターは行動を選択する．各キャラクターが選択できる行動は攻撃行動，回避行動，防御行動の 3 種類がある．攻撃行動を選択した場合は，キャラクターは自分の所持している技を 1 つ選択する．防御行動の場合は攻撃をすることができないが，そのターン中に限り防御力が上昇する．回避行動の場合はこちらも同じく攻撃することができないが，そのターン中に限り回避率が上昇する．各キャラクターが行動の選択を終えた後，ターンが実行される．ターンが実行されると，攻撃された側は相手が繰り出してきた技の攻撃力分のダメージを体力から減らされることになる．ただし，防御行動を選択していた場合はダメージを軽減することができる．また，技が繰り出されるときは命中するかの判定が行われるため，そこで回避することができればダメージは無しになる．回避行動を選んでいた場合はこの回避の判定が成功する確率を上げることができる．

この「行動の選択→ターンの実行」が物語分析班から提案を受けた戦闘の形式になっている．この形式が繰り返され，各キャラクターのどちらかの体力が 0 になるまでシミュレーションが実行される．シミュレーションが終了すると，実行中に得られた戦闘の過程と戦闘の勝敗結果を出力し，終了する．以上が戦闘シミュレーションである．

実際にシステムを動かすときは，この戦闘シミュレーションをあらかじめ 3 回実行する．ここで

得られた戦闘結果と過程をあらかじめデータベースに集積しておく．通常シーン生成システムでは 3 回の戦闘結果に対応した物語シートを参照するようにしている．通常シーン生成システムの戦闘シーンの挿入では，3 回の戦闘シミュレーションで得られた戦闘結果を呼び出し，挿入することで戦闘シーンの表現を行っている．

（＊文責：山田康貴）

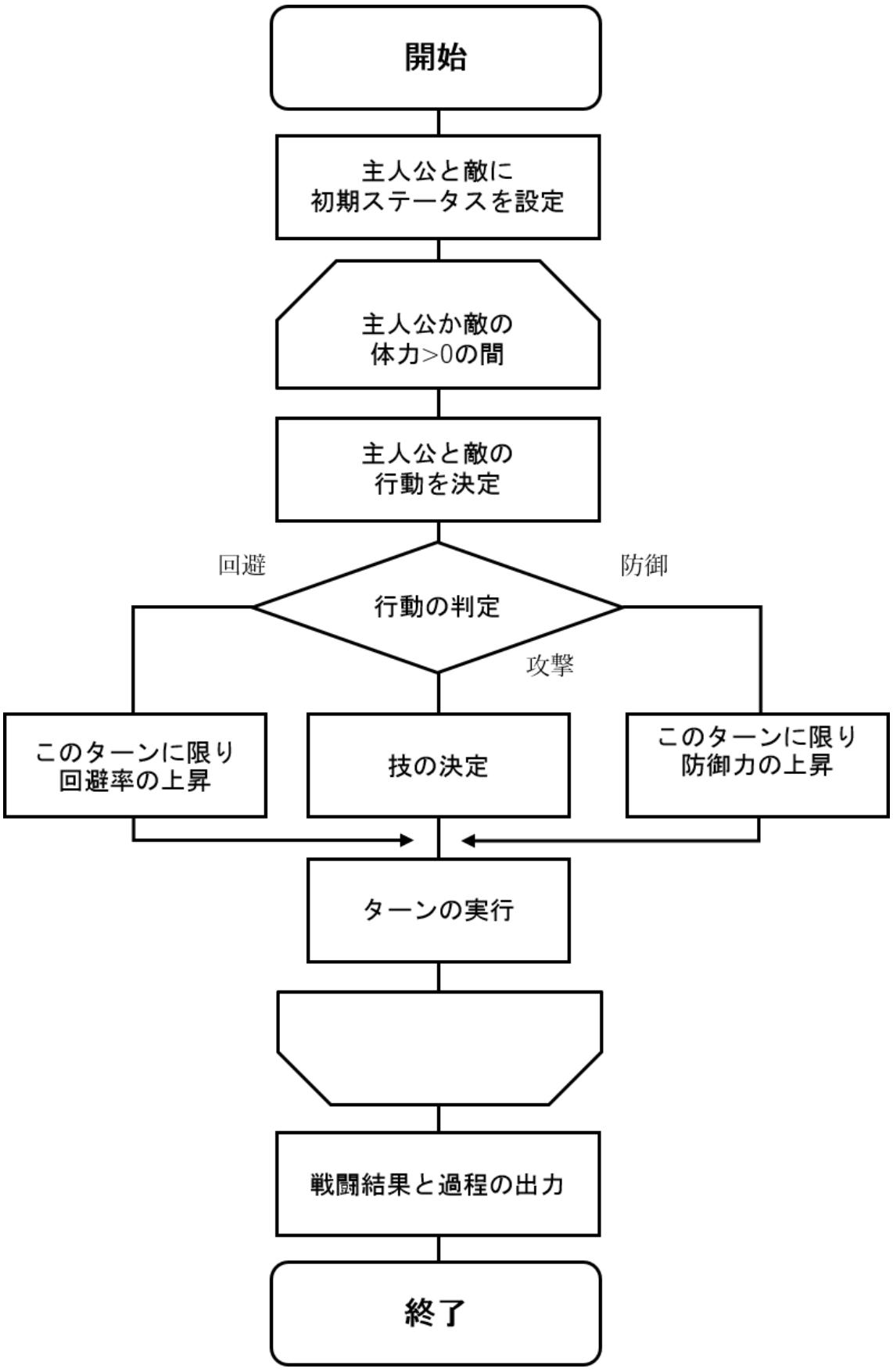

図 4.2.1.2.2.3-3　戦闘シミュレーションのフローチャート

（＊文責：山田康貴）

4.2.1.2.3　GUI

ホラーシステムと同じく，システムユーザーが物語の初期設定を行うために用いる GUI の作成を行った．バトルシステムの GUI を実装するにあたっては，Python に標準搭載されている Tkinter を用いた．Tkinter とは，簡単に GUI を組むことができるツールキットである．GUI は物語を生成するために必要な情報をユーザーが入力するための登録画面のみを作成した．登場人物は主人公と敵 3 人に決まっているので，主人公と敵 3 人の名前をそれぞれユーザーに入力してもらう．すべて入力したあと決定ボタンを押すと，その登場人物についての設定がシステムに入力される仕組みである．

（＊文責：鈴木諒輔）

4.2.2　Unity 班

4.2.2.1　物語データの読み込み

中間発表までに完成させた ScenarioLoader クラスを基に，更に豊かな物語データを読み込めるように改良を施した．中間発表までの ScenarioLoader クラスは最初に全て物語データの情報を階層的に配列に格納して他のクラスから呼び出していたが，中間発表以降はコードの可読性や拡張性を高めるために C#の List 型を用いてデータを格納した．中間発表までの物語データはテキストを上から下へ順番に読み込むだけでも読めるものだったが，自動生成され，分岐がある物語を読み込む都合上，中間発表以降の物語データは必ずしも順番に書かれておらず，シーンに数字のタグをつけてそのタグのシーンにジャンプする処理が必要となった．次に，物語を分岐させるための選択肢を表示させる仕組みが必要となったため，新たに物語データのルールを定め，これらを実現できるシステムに改良した．また，中間発表以降の活動で新たに追加する属性を大幅に増やし，セリフの発話者とカットの主体が違う場合がある，主体となるキャラクターがいない場合でもナレーションのようにセリフを言わせたい場合があるなど，セリフの発話者とカットの主体を分けるなどの仕様変更があり，属性にも階層構造が生まれた．そのため，中間発表以前は ScenarioLoader クラスに属性を追加させる際には各属性を表すアルファベット一文字が割り振られていたが，中間発表以降は「s2@セリフ（この場合は主体が 2 番目のキャラクターでないがセリフを 2 番目のキャラクターが言ったことにする)」など，特殊な記号で区切って一つのアルファベットで表している属性に複数の属性を含めるなどの処理が必要となった．

物語データの情報を Scene と Cut の階層構造に分けるために，各行の先頭に配置されている属性から次の行が今のカットか，次のカットか，次のシーンか，次はない（終了）かを取得して格納した．中間発表以降に選択肢などの属性が増え，現在の行と次の行の関係に注目する現在の物語データの書き方ではコードが複雑になるため，今後の改修が必要である．

以下が ScenarioLoader クラスのフローチャートである．

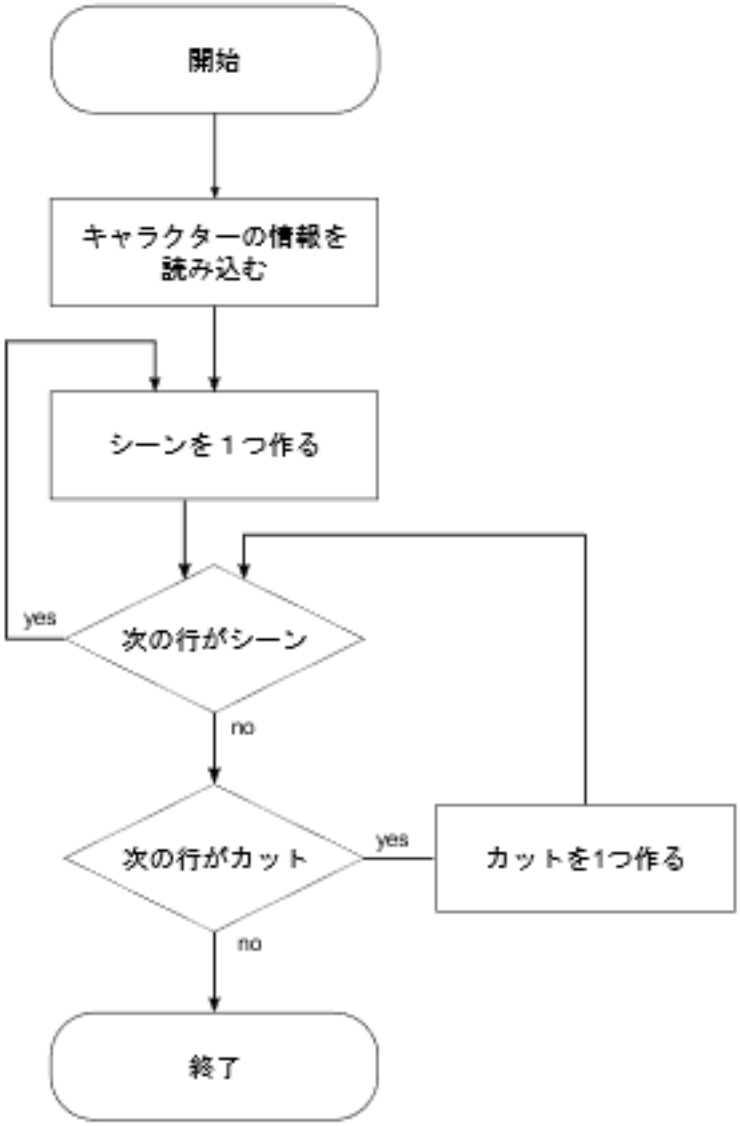

図 4.2.2.1-1　ScenarioLoader クラス

次に物語データが読み込まれたときの ScenarioLoader クラスの動きについて説明する．以下の図が物語データの例である．

```
A さん, 女性_通常
B さん, 男性_幽霊
p 森, 1
r1, a 立つ, dW, f4, e 無表情, k
r2, a 話す, dE, f6, e 真剣, k
g1, m, s こんにちは
r1, a 話す, dW, f2, e 無表情, k
r2, a 考える, dE, f6, e 恐怖, k
g2, m, s はい
p 洋館, 2
r1, a 走る, dE, f1, e 無表情, k
r2, a 怖がる, dE, f6, e 恐怖, k
g1, m, s さようなら
z
```

図 4.2.2.1-1　物語データの例

この物語データを ScenarioLoader クラスに通すと，まず，A さんと B さんのキャラクターデータを読み込む. p で始まっている行はシーンの始まりを意味するのでシーンを一つ作る. p で始まっている次の行から g の行までが 1 つのカットの情報なのでカットを作り，格納する．g の行の次が r で始まっているためもう一つカットを作り，次の g の行までを格納する．g の行の次が p で始まっている，つまり次のシーンのため，シーンを作る．p で始まっている次の行から g の行までが 1 つのカットの情報なのでカットを作り, 格納する. g の次の行が z(終了記号)のためロードを終了する.

以上の処理により，この例の場合はカットを 2 つ持つシーンとカットを 1 つ持つシーンが生成される.

（＊文責：高橋翔太）

4.2.2.2　UI の作成

中間発表までの課題として，字が読みにくいというのがあった．テキストを表示させるウィンドウの濃さを調節し，物語を読み進めていく際のストレスや読みやすさの改善を行った．最終成果発表までの開発でウィンドウは半透明でなくてもいいことが判明したので，濃くすることにより文字を見やすくすることにした. また, ウィンドウの色を変えることで読みやすさの改善を試みた結果，白色か黒色の二つの案が出た．白色の案がウィンドウの色を白くして，文字を黒くするといった一般的な色彩にしようとする案である．黒色の案が白色の案とは真逆のウィンドウの色を黒にし，文字の色を白にする案である．黒色のウィンドウをメインにするとホラーとしての不気味な感じを演出することが可能となった．しかし，ホラーとバトルのジャンルの物語を作るうえでウィンドウには統一感があったほうが良いという意見を取り入れ，白色で開発を進めることとなった．ホラーと相性の良い黒色のウィンドウをメインに開発を進めになかった理由は，どちらの物語にとっても読み手に同じ印象を与えるようにすることが目的だからである．名前を表示するウィンドウは，6 文字を上限とした．事前の話し合いでキャラクターの氏名はフルネームではなく，名前とのことだったのでそのようにした.

過去のセリフを見るためのログ機能を実装した．機能の詳細は，過去 20 個のセリフの表示，スクロールを可能にする，通常の画面に切り替える，3 つである．過去 20 個のセリフの表示をするため

にキューを実装した．キューというのは，コンピュータの基本的な構造の一つで，データを先入れ先出しして保持する．セリフを表示する関数が呼ばれるたびにそのセリフの文字列をキューに格納していく．また，格納するときのルールとして，名前と文章があるときと文章のみ，名前と文章がない時の 3 つのパターンで格納時の内容を変更した．基本的にはキャラクターの名前やセリフに記号を含めてキューに格納する．名前やセリフがない場合には付随する記号を消してキューに格納した．21 個目のセリフを格納する際は，キューの中で一番古い文字列を削除し格納していく．また，格納する際，ログを見やすくするためにハイフンを付けて区切った．区切る前は，ただの文字列の集合で見づらかったが，区切ることにより，セリフ間が見やすくなり，一目でわかるようにした．ログの機能の流れとしては，セリフを表示する関数が呼ばれるとキューにセリフを格納し，各ウィンドウに名前とセリフを表示するという流れである．

中間発表までの課題の文字の見やすさに関しては，おおよそ解決したといえる．しかし，文字の見やすさを重視する反面,各ステージとウィンドウの色があっていないことが見受けられた.また，今回はウィンドウを白色にしたことも原因の一つであると考える．これにより，とても個人的な見解ではあるが，少なからず読み手に不快感を与え，物語の内容を理解することを阻害する可能性がある．以上より，今後の課題として，ウィンドウを各ステージの色に寄せるか，もしくは，白色以外の色をメインに考えていく必要がある．

まとめると，最終成果発表までての開発では，中間発表までての開発でわかっていた課題の修正と UI の向上を目的としたログの機能を実装した．課題の修正では，字の読みやすさを重視した開発を行った．ログの機能を実装する目的は，ログがあることで読み手が物語を読み直すことを可能にし，読み手に与える負担を減らすことが目的であった．

しかし，課題の修正およびログ機能の実装をすることで新たな課題を発見することができた．その課題は，見やすさを重視するあまりに画面全体の色彩が好ましくないことが挙げられる．この課題の解決策としては，ウィンドウの色をあらゆる色に対して調和のとれた色にした開発をしていくことが望ましい．

(＊文責：南部太雅)

4.2.2.3　モデルの表示

中間に引き続き読み込んだ物語データをもとに 3D モデルを Unity 上に配置する View.cs という C#スクリプトを主に制作した．また，モデル班が制作した MMD 用 3DCG モデルや MMD 用モーションの変換も並行して行った．

MMD とは,3DCG キャラクターモデルを操作しアニメーションを制作するソフトウェアであり，専用のデータ形式の 3D モデルデータやモーションを使用するため, Unity 上で MMD 用の 3DCG キャラクターデータを直接利用することはできない．モデルの変換にあたっては MMD4Mechanim というライブラリを使用した．

MMD4Mechanim とは，MMD 専用の 3DCG モデル形式である pmx ファイルやモーションファイルである vmd ファイルを，Unity で利用可能なモーションを内包した fbx ファイルに変換するものである．また，モーションデータやモーフ情報を操作するいくつかのスクリプトも含まれている．

具体的な処理としては，まず pmx ファイルと vmd ファイルを Unity にインポート可能な fbx ファイルに変換し，Unity の Mechanim というシステムに適合するようにスケルトンを調整する．スケルトンとは，キャラクターをアニメーションさせる際に用いられる機能のことで，キャラクターに合わせて構築し，メッシュとスキニング(関連付け)することでスケルトンの操作によりメッシュをアニメーションさせることができる．

スケルトンはジョイントと呼ばれる関節部分を制御する機能とくさび形のボーンとが組み合わさ

った構造となっており，ジョイントの回転によって動きを操作できる．今回のシステムではモデル班によって地蔵を除く全てのキャラクターにスケルトンが実装されており，それらの変換も行う必要性があった．また，Bullet Physics というシステムによって物理演算をモーションに焼き込む作業も同時に行っている．

次に, マテリアルとシェーダーを MMD4Mechanim 専用のものに設定する. シェーダーとは, 3DCG 上で陰影や表面の質感，表示色などを行うプログラムで，マテリアルとは光学的な特性や材質感を処理するプログラムのことであり，Unity 上ではシェーダーとマテリアルは一対一関係となっている．MMD で使用されているトゥーンシェーダーは MMD 独自のものであり，Unity のデフォルトシェーダーに置き換えると MMD 上でのデザインにならず，モデル班との連携が困難になる．そこで MMD4Mechanim に付属のマテリアルとシェーダーに置き換えることで, MMD 上での見た目に可能な限り近づけることができるようになった．また，ここで同時に表情モーフを Unity で使用可能な形式に変換し, MMD4Mechanim Anim Morph Helper での操作が可能になるようにした. 最後に, Unity の Mechanim としての利用が可能になるよう調整し，変換作業は終了となる．実際の作業としては Unity 上に pmx ファイルと vmd ファイルを読み込み，各種設定しスクリプトを呼び出すことで自動的に変換が行われるようになっている．これらの作業はシステム制作時に全て行い，システムの動作中に変換が行われることはない.

MMD4Mechanim Anim Morph Helper とは，MMD4Mechanim に付属のコンポーネントであり，MMD4Mechanim により変換された 3D モデルのモーフを操作することができるというものである．モーフの操作は他のモーションファイルを利用する方法と数値で調整する 2 つの方法があるが，今回のシステム制作では MMD を用いてモーションを制作するモデル班との連携を円滑にするため前者の方法を採用した.

システムの前回からの変更点としては，まず Animation という古典的なモーション制御から，Animator と呼ばれる比較的新しい手法に切り替えた．前回のシステムではモーション数が少なかったことや，モデルデータが既存のものであったため自作モーションによる破綻が起きにくいこと，また単純な機能のためシンプルに実装できるという理由から Animation を採用したが，今回のシステムでは後述のモーフや武器を利用することや，モデルデータを一新した都合上モーションの修正を頻繁に行う必要性があり，より柔軟性の高いシステムを構築する必要があったためである．この変更によりモーションデータの追加や変更をスムーズに行うことが出来るようになり，映像の品質向上に貢献することができた.

また，中間までのシステムでは全モーションをそれぞれ個別に扱っていたためキャラクターの男女, 大人や子供といったモーションの違いを物語データ側で記述しておかなければならなかったが, 今回の手法によってキャラクター側にそれぞれ固有のモーションを持たせておくことで，大人や子供などの体格の差や，人間と非人間キャラクターなど様々なキャラクターに対しモーションの違いを吸収しつつ同一のモーション名で簡単にアクセス出来るようになった．既存モデルからモデル班による自作モデルになったことで，バトルとホラーという物語ジャンルの文脈に合ったキャラクターを使用することが可能になる反面, ポーズの記述による物語データの複雑化が懸念されていたが, この変更により物語データの可読性が良くなり，デバッグが効率よく行えるようになった.

次に，モデルの表情の変化に対応させた．今回も 3D モデルは MMD 形式のものが使われているため，MMD 形式の 3D モデルを Unity で使用可能な形式に変換するライブラリである MMD4Mechanim に付属している MMD4Mechanim Anim Morph Helper というスクリプトを使用し，表情変更を実装した．最初は 1 つの表情に対応するモーフの値をリスト化しテキストデータとしてモデル班と連携することを考えたが，今回のシステムでは表情の数が膨大になり，表情データの管理が非常に難しくなった．そこで MMD4Mechanim の機能を利用し，予めモーフのデータのみを記

録したアニメーションデータをキャラクターに読み込んでおき，MMD4Mechanim Anim Morph Helper に登録し任意のタイミングで読み込めるようなシステムを構築した．

具体的には，MMD4Mechanim Anim Morph Helper のコンポーネントフィールドにモーフの変更のみを記録したモーションデータの全表情 99 種類を登録しておき,ポーズと同様にスクリプトによってスクリプトのコンポーネントフィールドを書き換えるという仕組みで実装している．また，静止画を出力する都合上表情変化スピードを最低値に設定，再生後モーションを停止させるようにするといった変更も行っている．　この変更によって，モデル班とのモーションデータの連携がスムーズに行えるようになり，表情モーフデータの変更が用意に行えるようになった上に，管理が容易になった．

次に，武器の使用に対応させた．視覚化班が新たに作成した剣と槍のモデルを，キャラクターのポーズに合わせて Unity 上の適切な位置に配置するようにスクリプトに処理を追加した．最初はキャラクターと武器の位置調整を Unity 上で行うことを検討していたが，3DCG にはそれぞれ固有の重心という回転の基準となるパラメータがあり，それらの位置を合わせることが困難なため，スクリプトで一様に設定することができなかった．また，キャラクターの体格や武器の持ち手部分にもバラつきがあり，それらを同じ値で位置を設定するとどうしても位置にズレが生じ，破綻した映像となってしまう．そこで，モデル班にキャラクターの体格に合わせた武器専用のモーションデータを制作してもらい，キャラクターのモーションと対応付けることで武器それぞれに合った位置を武器オブジェクト自身に行わせるという方針に変更した．

具体的な実装としては，まず全キャラクターに子オブジェクトとしてそれぞれ”道具”という空のゲームオブジェクトを作成し，非表示設定にした全武器モデルを上でその道具オブジェクトの子オブジェクトとした．次に武器モデルに Animator コンポーネントを作成し，視覚化班が制作した武器専用モーションデータを設定した．次に，EquipAnchor.cs という新しい C#スクリプトを作成し，道具オブジェクトにアタッチした．これは，View.cs が全てのキャラクターの武器モデルにアクセス出来るようにするためにキャラクターそれぞれの武器オブジェクトの場所を指定するスクリプトである．このスクリプトによってボーンの構成などが異なる各キャラクターのオブジェクト構造に依存せず，一律な方法で武器モデルを操作することを可能にした．最後に武器使用モーションが読み込まれると同時に，物語データに記載されている武器名から適切な武器モデルをゲームオブジェクトの子オブジェクトから探しだし，非表示設定を解除した上で武器の持っているモーションからキャラクターのモーションに対応したモーションを読み込む，という流れになっている．このように武器自体がモーションデータを持つように実装にすることによって，武器による持ち方，構え方の違いを吸収することで，映像の表現力をより広げることに成功した．

（＊文責：佐々木奨之）

4.2.2.4　ステージの配置

物語再生時のステージの配置について説明する．ステージとは 3D 上に物語のシーンを描画する際にキャラクターたちの背景に表示される 3D オブジェクト群のことである．中間発表時点まではステージを Unity 上で扱えるデータ形式の一つである Prefab という形式で保存していた．Prefab は作成したゲームオブジェクトの情報をテンプレートとして保存できるもので，これを実行時に読み込むことで予め必要な設定のなされたオブジェクトを生成することができる．Prefab からオブジェクトを生成する方法は Unity での開発においてリソースを実行空間に持ち込みたいときにほとんどの場合で有効的だが，今回のステージ読み込みには適していなかった．なぜならこの方法では 3D の背景を作成する上で重要なライティングの設定を行えないからだ．というのも，Unity では Scene というオブジェクト環境とメニューを定義するファイルを編集することによってアプリを構築して

いくのだが，ライティングの設定はこの Scene 単位でしか行えない．つまり，1Scene に 1 ライティング設定しか用意できないのである．Prefab を使用する方法では物語再生用オブジェトが置かれたメインの Scene しか利用しないため，ステージごとのライティングを反映することができない．これはよりクオリティの高いステージを作成する上で非常に厄介な制約となるため，この方法は最終的な成果物には使用しなかった．

替わりに使用した方法はメインの Scene とは別にそれぞれのステージ用の Scene を作成し，ステージの配置時に追加読み込みを行うものである．追加読み込みというのは Unity が標準でサポートしている機能であり，通常の Scene 遷移処理とは異なり現在実行中の Scene に上乗せする形で Scene を読み込むことである．この方法ならばそれぞれのステージのライティング設定をそれぞれの Scene の編集画面上で行うことができ，Prefab 使用時の課題を解決できることが予想される．しかしこの方法には問題があった．それは追加読み込みで読み込まれた Scene のライティング設定よりも先に読み込まれていた Scene のライティング設定が優先されることである．これではステージごとのライティングの設定を反映させることができない．これを解決するためにはステージ Scene それぞれのライティング設定を何らかの形で保存し，追加読み込み時にメインの Scene に適用する必要がある．そこで PlaceSettingHolder というスクリプトを作成し，ステージ Scene に配置した．このスクリプトは編集時にはライティングの設定が変更されるたびにその全てを保存し，実行中に読み込まれた際には現在のライティング設定を保存したもので上書きするという処理を行う．これを導入したことにより，ステージの編集時には自由にライティングの設定が行え，ステージの読み込み時には正しくその設定が反映されるような仕組みを作成することができた．実際にステージ作成の際にはこの仕組みのおかげで柔軟かつ簡単にステージの作成とその配置を行うことができた．

（＊文責：津沢慎吾）

4.2.2.4　マネージャークラスの実装

再生システムのマネージャークラスについて説明する．システム全体を管理するマネージャーは適切な物語データを読み込み，他の要素に対して必要なデータの受け渡しを行う処理を担っている．Unity 上では SenarioPresenter と名付けたクラスが主にこのマネージャーとしての処理を行っている．一部の機能を除いてシステムの進行状態はこのクラスが管理している．

次に Unity 上での実行の流れを追ってマネージャークラスの処理を説明する．まず，Unity を実行すると図 4.2.2.4-1 のような物語選択ウィンドウが表示される．これは SenarioSelector というクラスが管理している処理で，物語再生システム内で唯一 SenarioPresenter から独立して起動する．

図 4.2.2.4-1　シナリオ選択画面

この表示は Unity の UI 機能を利用して実現されており,ドロップダウンリストによって再生するシナリオの選択,ボタンによってその確定を行うことができる.再生するシナリオが決定されると，SenarioSelector はシナリオ設定を保存するためのオブジェクトへその内容を書きこんだ後 SenarioPresenter の初期化関数を呼び出し，物語の再生を要求する．物語の再生が要求されると，SenarioPresenter は最初に物語データの読み込みを行う．データの読み込みが正常に終了すると，物語冒頭のシーンが表示される．その後はユーザーがマウスをクリックするごとに物語が順番に表示されていく．物語の表示がすべて終了すると，画面上に終了を示す文字列を表示する．その状態でもう一度マウスをクリックするとシナリオ選択画面に戻るようになっている．

SenarioPresenter が物語を表示する工程は主に４つに分かれている．１つ目はステージの配置である．ステージは物語データの１シーンに１つ設定されているため，シーンが遷移するごとに一度ステージの再配置が行われる．２つ目はキャラクターモデルの配置である．キャラクターはすべてのカットごとに位置，動作等が指定されているため，毎カットキャラクターの再配置が行われる．3 つ目はテキストの表示である．テキストもキャラクター同様すべてのカットで文章が指定されているため，基本的には毎カット更新が行われる．しかし，テキストの内容によってはテキストウィンドウの中に収まり切らない文量である場合がある．その場合はテキストを表示し終えるまで，テキストウィンドウのみをユーザーの入力によって更新する処理を行う．4 つ目はカメラの配置処理である．カメラの座標と回転の計算はキャラクターの配置，動作に依存するため，必ずキャラクターの更新が行われたあとにカメラの更新処理を行うようになっている．以上の物語再生の工程を図 4.2.2.5.2 に表した.

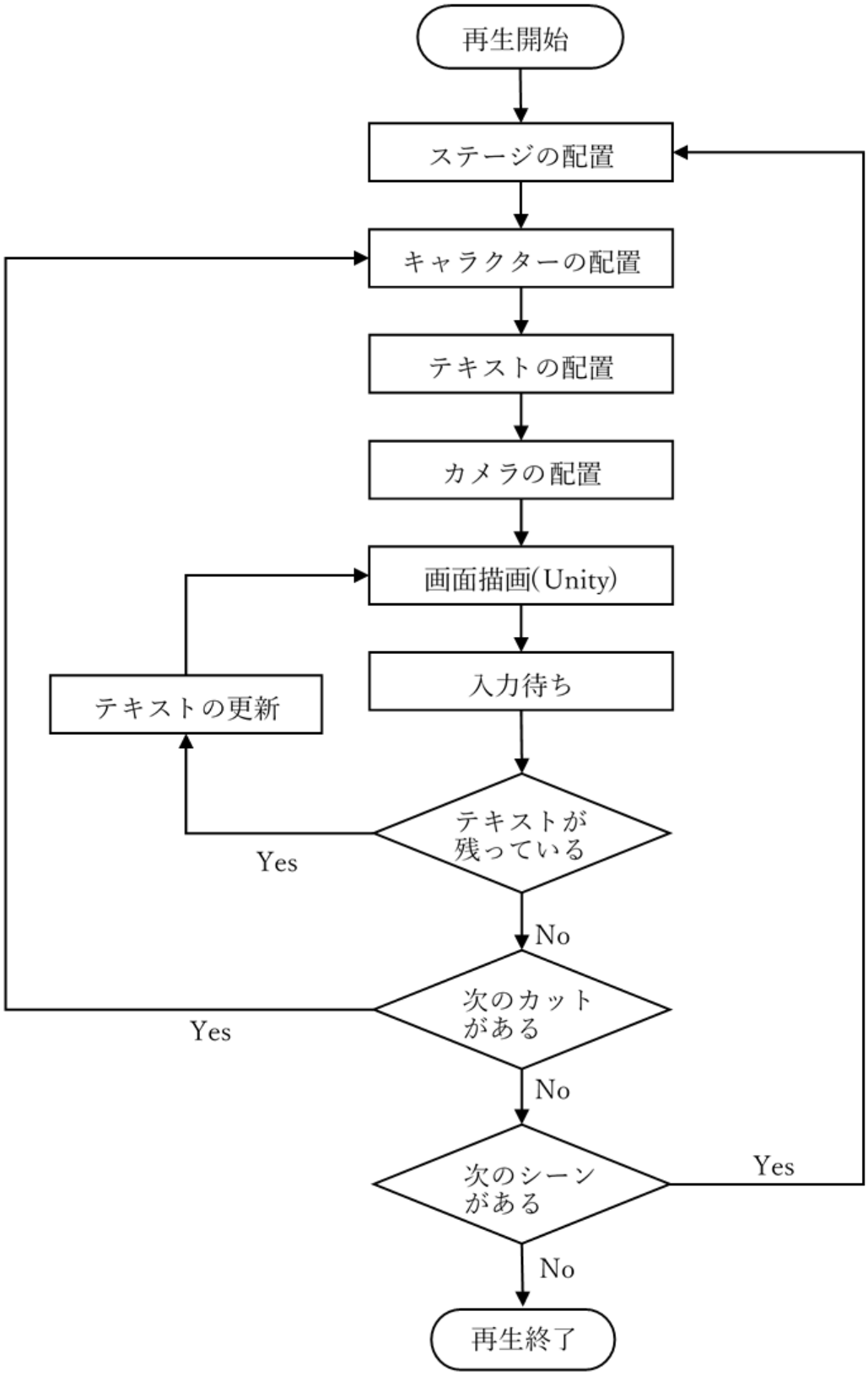

図 4.2.2.4-2　物語再生フローチャート

図 4.2.2.4-2 にあるように，再生開始後最初のシーン読み込みが行われ，ステージの配置が行われる．その後のキャラクターの配置，テキストの配置，カメラの配置が物語データ中の 1 カットを表示する処理に該当する．1 カットの中ではテキストを最後まで表示するループが存在し，入力ごとにテキストの更新と画面描画が行われる．テキスト表示が最後まで行われた後再生中のシーンに次カットが存在するか確認する．存在する場合は次のカットへ進み，再びカットの内容を配置する処理が行われる．次のカットが存在しなかった場合はシーン遷移の判定が行われる．物語再生中は常にこのシーケンスで処理が行われており，基本的に他の処理がこの中に割り込むことはない．例外として，シーン遷移時の処理では一定時間ユーザーの入力を受け付けないことや，ユーザーに特殊な入力を要求する場合が存在する．

上述したように必ずキャラクターの更新処理後にカメラの更新処理を行わなければならないが，キャラクターの更新が反映されるタイミングには注意する必要がある．カメラの配置情報を計算するために用いている値はUnityのシーン上に存在するキャラクターの3Dモデルから直接取得されている．その情報の中には 3D モデルの正確な頭の位置なども含まれているため，キャラクターのアニメーションが全て終了したあとでなければ正しくカメラの計算を行うことができない．Unity では毎フレームの処理を行いたい場合基本的に Update というイベント関数用いる. 従ってキャラクターの更新，UI の更新等の更新処理は通常 Update 関数内で記述する．しかし，Unity が実際に 3D モデルのアニメーションの更新処理を行うのは全ての Update 関数が呼ばれた後のことなのだ. つまり，カメラの更新処理を Update 関数内, キャラクターの更新処理の直後に書いてしまうことはカメラがアニメーションの更新が行われる前のキャラクターの情報を取得して計算を行ってしまうことを示す．これを回避するためにはカメラの更新処理をアニメーションの更新処理後に実行される Unity のイベント関数である LateUpdate 関数の中に書かなければならない．このような理由から，図 4.2.2.4-2 上では連続的に処理しているように見えるカメラの更新は，プログラム中では実行タイミングに配慮した記述を行わなければ正しい動作が得られないことに留意する必要がある．上記の説明に関わる Unity 上でのイベント更新順を表したのが図 4.2.2.4-3 である．

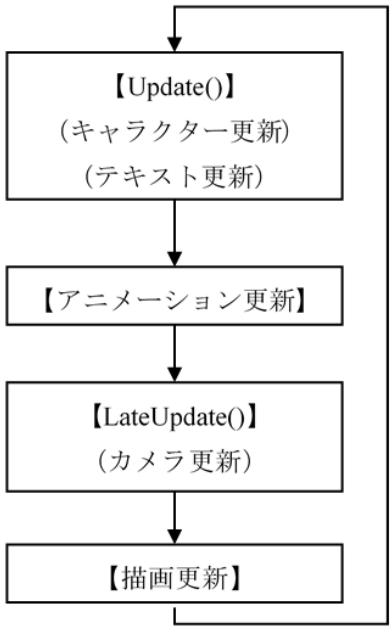

図 4.2.2.4-3　Unity によるイベント更新の順番

テキストの更新処理では最初にテキストウィンドウに表示しきれるようにテキストを分割する．分割方法は単純で，1 度に表示できる最大文字数を設定しその長さを超えないように区切っている．そして分割した文字列を入力のたびに順番にテキストウィンドウに渡している．ただしこの方法は文章の区切りを無視して分割してしまい，ユーザー違和感をもたせてしまうことがあったので優秀な方法では無かったと思われる．日本語に限れば句読点が文章の意味の区切りとなるので，それを検出して適切に文字列を区切るようにすればより良いテキスト表示が実装できたのではないかと予想される．

表示中のシーンの全てのカットの表示が終了すると，新たなシーンに遷移する．図 4.2.2.4-2 中では省略しているがシーンが遷移するタイミングで画面が暗転，明転する処理が行われる．画面の暗転中はユーザーの入力を受け付けず，また，次シーンの 1 カット目を表示する処理は暗転により画面が隠されている状態で行われる．つまり，実際の処理では図 4.2.2.4-2 中の次のシーンが存在することを判定する分岐が正の値を返した時点から暗転が始まり，暗転中にステージの配置からカメラの配置までが行われ，一定時間後画面が明転するまで待機した後に入力待ちを行う流れになっている．シーンの遷移には 3 つのパターンが存在する．1 つ目は単純に次のシーンに進むパターンである．物語データは基本的にテキストデータの先頭から末尾に向かって物語の進行順に記述されているため，特殊な処理がない場合は直後に記述されているシーンに遷移する．2 つ目は指定したシーン番号にジャンプするパターンである．それぞれのシーンには番号が割り振られており，ジャンプタグにより指定された番号のシーンに遷移する処理が行われる．3 つ目は選択肢分岐を行うパターンである．シーンに選択肢分岐が指定されていた場合はシーンの再生完了時に図 4.2.2.4-4 のような選択肢の表示を行う．選択肢はマウスクリックによって選ぶ事ができ，ユーザーが選んだ選択肢の内容によって次に遷移するシーンが決定される．選択肢を表示している間マネージャークラスは完全に停止しており，選択肢の決定コールバックを受けるまで待機する．いずれのパターンでもシーンの再生中に他のシーンに遷移することはなく，シーン終了時にのみシーン遷移の確認が行われる．シーン遷移確認時に現在表示しているシーンに物語終了フラグが設定されているかも合わせて確認する．物語終了フラグが設定されていた場合はそこで物語が終了となり，暗転を行ったあと，次のシーンを表示する代わりに物語終了を示す文字列が表示される．その後入力を行うと，Unity 上のアクティブな Scene がリロードされ，起動時と同様の状態になる．

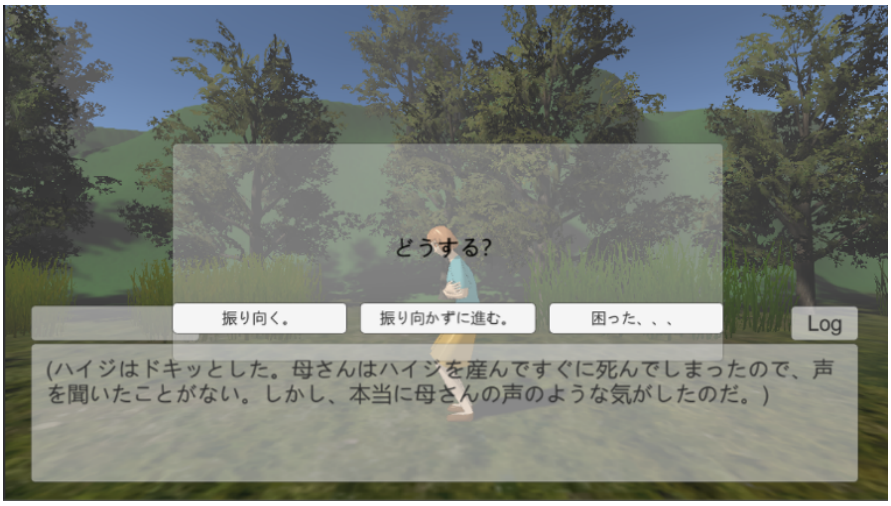

図 4.2.2.4-4　選択肢の表示

本プロジェクトではホラーとバトルの２つのルールの物語が存在し，それぞれには選択肢分岐の有無や登場人数などの違いがあった．このマネージャークラスはそれらの違いを吸収し，一つのシステムで物語を再生することに成功している．このシステムは Unity 上で物語再生を行う上での最低限の仕様を満たしていると私は考える．一方で，マネージャークラスである SenarioPresenter は再生システムのほとんどの要素を一元的に管理している．これは開発管理が行いやすいという点では優れているが，機能拡張を行いづらいという欠点がある．なぜなら新しい機能を追加しようと考えるたびにクラスの内容が大きくなってしまうからだ．また，本プロジェクトで作成した物語生成システムは紙芝居形式であったため，マネージャーでは 1 カットの時間の管理を行わなかった．しかし，今後の活動次第ではユーザーによりよい体験をさせるためにリアルタイムのキャラクターアニメーションを実装することがあるかもしれない．その場合はシーケンス制御の機能をより拡張する必要があるだろう．また，このシステムは物語データを物語再生の開始時にテキストファイルから一度に読み出し，その後は読み込んだデータに従って物語を再生するだけのものである．もしこの先，物語を動的に変化させるような実装を行おうとするならばマネージャーの設計を新たに考え直す必要がある．

（＊文責：津沢慎吾）

4.3 視聴覚班

4.3.1 カメラ班

4.3.1.1 分析

以下カメラワークの分析について説明を行う．

(＊文責：三浦隆太郎)

4.3.1.1.1 分析の概要

我々カメラ班は中間発表までに得られた 8 種類のカメラワークを基準として，生成された物語を効果的に描写するためのカメラワークアルゴリズムを実現するために，必要となる要素を選定しその場面での最適なカメラワークを実現することを目的に活動を行ってきた．中間発表での結果を踏まえて,場面の状況に関わらずランダムであったカメラワークを場面に適したものにするためには,何がカメラワークを選択するための要因となっているかを知らなくてはならない．そのため我々カメラ班はこれまでの分析から手法を変えた映像分析を行った．分析の対象としたのは,『Fate/stay night』『Fate/Zero』の 2 作品である．この 2 作品を選んだ理由としては多くの武器種が作品内に登場し戦闘に関わってくるキャラクターが多いバトル作品だからである．バトル作品を 2 つ選んだ理由としては中間発表までに分析をした作品が『呪怨』や『羊たちの沈黙』などいずれもホラー色の強い作品であったため，バランスの良いカメラワークを生成するためである．

まず 8 種類のカメラワークがそれぞれどの条件で使われているかを分析するために「場面の人数」,「主体となる人物のアクション」,「前カットと比較した際の主体の変化の有無」,「主体の感情の度合い」を 1 カットごとにどのショットが選ばれているかと共にまとめた．この時の分析したカット数はおよそ 400 カットであった．しかしこの分析方法には問題があり，カメラワークの自動生成をするには分析項目が見合っていなかった．よって我々カメラ班は成果発表までの残り時間が約一か月であったため，分析結果の利用方法をデータベースの作成からルールベースの作成へと変更することで，約一か月の作業時間で分析とカメラワークアルゴリズムの作成を行うこととした．今回作られるカメラワークアルゴリズムが場面から「シーンの種類」,「場面の人数」,「主体となる人物のアクション」からカメラワークを選定するものであったためこれら 3 つの条件を分析の項目とした．

この条件で分析に『呪怨』『新耳袋』『鬼談百景』『Fate/Zero』『Fate/Apocrypha』『ジョジョの奇妙な冒険第一部』『ジョジョの奇妙な冒険第二部』の 7 作品を使用して約 1700 カットの分析を行い，ルールベースを作成した．

分析を行った結果として各条件につきおよそ 3 つのカメラワークが該当した．シーンの種類によるカメラワークの違いとしては，ホラー作品とバトル作品では大きな差が存在せず場面の人数や主体となる人物のアクションの種類がカメラワーク選定の多くを占めていることが分かった．

（＊文責：三浦隆太郎）

4.3.1.1.2 分析方法

中間発表での結果を踏まえて，場面の状況に関わらずランダムであったカメラワークを場面に適したものにするためには，何がカメラワークを選択するための要因となっているかを知らなくてはならない．そのため我々カメラ班はこれまでの分析から手法を変えた映像分析を行った．分析の対象としたのは，『Fate/stay night』『Fate/Zero』の 2 作品である．この 2 作品を選んだ理由としては多くの武器種が作品内に登場し戦闘に関わってくるキャラクターが多いバトル作品だからである．バトル作品を 2 つ選んだ理由としては中間発表までに分析をした作品が『呪怨』や『羊たちの沈黙』などいずれもホラー色の強い作品であったため，ホラーに偏ったカメラワークを生成することなくバランスの良いカメラワークを生成するためである. まず 8 種類のカメラワーク, ロングショット，フルショット，ミディアムロングショット，バストアップショット，ミディアムクローズアップショット，クローズアップショット，ズームアウトショットそして肩越しショットがそれぞれどの条件で使われているかを分析するために「場面の人数」,「主体となる人物のアクション」,「前カットと比較した際の主体の変化の有無」,「主体の感情の度合い」を表計算ソフトで 1 カットごとにどのショットが選ばれているかと共にまとめた．この時の分析したカット数はおよそ 400 カットであった．

しかしこの分析方法には問題があり，カメラワークの自動生成をするには分析項目が見合っていなかった．よって残りの作業時間で行える分析方法およびカメラワークアルゴリズムで使用可能なデータを作るために分析方法の見直しと簡略化を行った．見直した結果としてはこれまでの分析で使われてきた分析項目 5 つ「場面の人数」,「主体となる人物のアクション」,「前カットと比較した際の主体の変化の有無」,「主体の感情の度合いをシーンの種類」,「場面の人数」,「主体となる人物のアクション」から「シーンの種類」,「場面の人数」,「主体となる人物のアクション」の 3 つに変更した．簡略化としてはそれぞれの分析項目にあった選択肢を「場面の人数」が種類分けされておらず場面にいる人数を直接入力していたものから場面の人数を 0 人と 1 人と 2 人とそれ以上の 4 種類に分けた,「主体となる人物の動作」が歩く，話す，走る，もがく，座る，死んでる，謝る，拝む，のぞく，構え，ステップ，殴る，武器攻撃，投げる，防御とモデルに用意されているモーションすべてに対応した 15 種類素材していたものを戦う，移動，話す，静止，に分け人物がいない場面用の計 5 種類に選定した．そしてシーンの種類は共通のシーンとホラーシーン，バトルシーンの 3 種類を採用した．60 パターンの表現が可能なルールベースの作成を行った．分析に使用した作品は『呪怨』『新耳袋』『鬼談百景』『Fate/Zero』『Fate/Apocrypha』『ジョジョの奇妙な冒険第一部』『ジョジョの奇妙な冒険第二部』の 7 作品である．この 7 つが選ばれた理由としては『呪怨』はホラー映画として知名度が高く，世間からの高い評価を得ているため.『新耳袋』はオムニバス形式で多くの分析を行うことができるため.『鬼談百景』はこちらもオムニバス形式で多くの分析を行うことができるため.『Fate/Zero』は多くの武器種が作中に登場し戦闘に関わるキャラクターが多いバトル作品だからである.『Fate/Apocrypha』は『Fate/Zero』よりさらに登場キャラクター及び戦闘回数が多い作品であるため.『ジョジョの奇妙な冒険第一部』及び『ジョジョの奇妙な冒険第二部』は戦闘の描写が

多く，知名度や評価の高い作品であるからである．これら 7 つの作品を用いて 1693 カットの分析を行った．

（＊文責：三浦隆太郎）

4.3.1.1.3 分析結果

ルールベースを作成することで得られた結果として，シーンの種類によるカメラワークへの影響は少なく場面の人数と主体となる人物のアクションの影響が大きいことが分かった．またルールベースを作成して発見された法則として，場面の人数が 0 人の場合はほとんどのカメラワークがロングショットであり場面の人数が 1 人の場合はロングショットやミディアムロングショット，ミディアムクローズショットが多く，場面の人数が 2 人の場合はミディアムクローズショットやズームアウトショット，肩越しショットが多い．場面の人数が 3 人以上の場合はロングショットやバストアップショット，肩越しショットが多い．主体となる人物のアクションが戦うものであるときはズームアウトショットと肩越しショットが多く，ミディアムロングショットやバストアップショットも時折存在した．主体となる人物のアクションが移動である場合はフルショットやミディアムロングショット，バストアップショットが多い．主体となる人物のアクションが話す場合，バストアップショットやミディアムクローズショット，クローズアップショットが選ばれることが多い．主体となる人物のアクションが静止にあたる場合，フルショットやバストアップショット，肩越しショットが多い．

今回はルールベースを用いてカメラワークの選択をすることになったが，シーンの種類よりも場面の人数や主体のアクションがカメラワークの選択において重要視されることは，ホラーやバトル特有のカメラワークが存在するという初期の考えと全く異なるものであった．

（＊文責：三浦隆太郎）

4.3.1.2 実装

以下カメラワーク生成システムの開発においての実装について説明を行っていく．使用したツールは Python，Unity であるが最終的な成果物としては Python を使う事は無かった．

（＊文責：城田晃希）

4.3.1.2.1 実装の概要

中間発表までのカメラワークアルゴリズムではカメラワークをランダムで選択していた．そして後期からは物語に伴った適切なカメラワークの生成を実現することを目標としていた.「適切なカメラワークの生成を行うシステム」の開発を行うにはいくつか問題点が存在していた．まずカメラワークにおいて適切とは何かという事である．カメラワークは映像作品であれば監督によって全く映し方が変わってくる，そしてそのカメラワーク，映し方によって全く作品の完成度や，視聴者が作品を見た時の感じ方が変わってくる．いわばカメラワークも見る人，撮る人によって感じ方が全く変わってくる芸術の一部であると一般的には考えられる．人によって感じ方が全く変わってくるものは，分析をするにしても，開発成果を評価するにも，点数をつけ数値化する事が非常に困難であると考えられる．

それに加えて先行研究の少なさも問題点の一つであると考えられる．先行研究が殆ど見つからないので何を指標にしたら良いのかという事から議論を始めなければならなかった．

作成する物語はバトル物とホラー物の２種類あった. 当然ホラー物とバトル物では物語の構成も，登場するキャラクターや道具，セリフや視聴者に与える印象も全く変わってくるものである．よってそれぞれに適切なカメラワークを当てはめなければならないと思われる

以上の事柄を解決するために我々は時間をかけて議論をしたかったが，最終的には取り合えず考えたことで開発を行っていき徐々にフィードバックをすることで解決していくという方針を取ることになった．

（＊文責：城田晃希）

4.3.1.2.2　カメラワークアルゴリズム

カメラワークアルゴリズムとは中間までの内容で記したしたように，物語の場面や内容によって適切なカメラワークを生成できるようなアルゴリズムの事である．分析結果でも記したようにカメラワークには様々な種類があり，それぞれに効果的な見せ方や意味合いがある事が分かっている．したがって上記に示したようなアルゴリズムの実装は物語を視覚化する上でより分かりやすく，より完成度を上げるものになるであろうと考えた．後期の開発では最初に作ったアルゴリズムは納得できるような結果には至らなかったため，破棄してしまい一からシステムを作り直した．結果としては最初に作ったアルゴリズムよりは良いものができたと思われる．次の項で説明することは破棄した最初に作成したアルゴリズムについてである．

（＊文責：城田晃希）

4.3.1.2.2.1　没アルゴリズム

中間の段階では，カメラ班としては機械学習を用いてカメラワークアルゴリズムの実装を行うという方針を取っていた．理由としては実装案としてのカメラワークアルゴリズムは数種類のカメラワークの中から適切なカメラワークを選択するというものであったためである．つまりはカメラワークアルゴリズムとは何らかのパラメータを用いてカメラワークをグルーピングし，実際に物語のパラメータと照らし合わせ適切なカメラワークを選択することであるため，機械学習を用いることが効率的なのではないかと考えたためである．

そこで初期案として考えたカメラワークアルゴリズムは次のようになった．

まず映像をカット割りし分析を行う．分析するパラメータは人数，動作，発話の有無，前のカットと主体は同一であるかどうか，感情の段階，カメラワークのパターンの６種類である．そしてその分析をしたデータを CSV に変換し，Python とライブラリの sci-kitlearn を使い決定木学習を行わせる．そして物語のプロットをパラメータに変換し生成した木に読み込ませ，それを用いてカメラワークの選択を行うというものであった．

しかしアルゴリズムを実装している段階で実際に収集したデータとカメラワークとの間の相関係数が低すぎるという問題があることが分かった．

この問題について議論と調査を行った結果，原因としては２つある事が分かった．一つ目はパラメータの項目の量と分析に必要なデータの数が釣り合っていない事である．上記に記した通り分析したパラメータの項目数としては６種類であったが，実際に学習を行わせるためのデータフォーマットに変換を行うとパラメータの数は全部で３０種類ほどあり，決定木学習を行わせるためのデータ数を集めるとするとおおよそ短期間では不可能であることが分かった．二つ目はカメラワークの決定と殆ど関係ないようなパラメータが含まれていたため，学習を行う際にノイズになっていたためである．この問題はパラメータ事に実際のカメラワークとの相関を出力することによって分かった．

以上の二つの問題点を解決する方法としては３つ挙げられた．１つ目は必要なパラメータを今一度分析し選定し直す事である．これを行い，関係ないパラメータの排除，あるいは更に重要なパラメータの追加をする事で，データと実際のカメラワークの相関を上げ，不必要なデータによるノイズの削減を図った．２つ目は機械学習という手法の再検討である．機械学習という方法を使うメリ

ットとして，大量のデータをパラメータを用いて自分達が望むようなグルーピングを行う事が容易に行える事が挙げられる．しかし今回の結果から自分達が予想していた以上にデータのフォーマットによって必要とするデータが膨大になってしまう事が分かった．したがってカメラ班の本来目標と照らし合わせて本当に機械学習を使う必要があるのかの再検討を行った．３つめはアルゴリズムによって何を表現したいかのフォーカスを行う事である．上記で記したアルゴリズムはカメラワークを統括的に定義するパラメータセットを必要とした．よって短期間での実装を行うには時間的な無理が生じると考えた．そのため表現をしたいこと，例えば時間的なカメラワークの推移などをフォーカスすることによって完成度を保ちつつ実現可能な範囲に規模を縮小しようとした．

（＊文責：城田晃希）

4.3.1.2.2.2　最終的なアルゴリズム

続いて最終発表までに作成したカメラワーク選択アルゴリズムについて説明を行う．作成したアルゴリズムは新たにシーンの意味合いのパラメータを導入して，より物語の場面に合ったようなカメラワークを生成することを目指した．機械学習を用いたカメラワーク生成は一旦取りやめ，ルールベースを用いて確率的と傾向を元にカメラワークを生成する方針に切り替えた．カメラワーク選択アルゴリズムの動作について説明する．分析したカメラワークのデータは CSV 形式であり，このデータの読み込みと整理を最初に行う．具体的には物語の情報からカメラワークを選択できるように専用のクラスに格納してゆく．そして物語が再生されるとともに物語のデータを受け取り，それを元にカメラワークを生成する．カメラワークを生成するにはそのシーンで誰を映してほしいか，場面に何人いるか，そのシーンの意味合いは何であるか，映したいキャラクターの動作は何であるか，の情報が必要である．カメラワークは全部で８種類あり，概要についてはカメラワーク分析の項に記してある．実際に生成する際の動作としては，最初に格納されたクラスには物語のデータからなるカメラワークの傾向が格納されており，物語の場面が進むたびに物語のデータから確率的に八種類のカメラワークから選択するようになっている．そして選択したカメラワークからキャラクターの位置情報，向きなどを元にメインカメラの座標と回転の計算を行いメインカメラの位置を決定する．以上が作成したアルゴリズムの動作の概要となっている．

作成したアルゴリズムにはいくつかの問題点が存在する．このアルゴリズムだとあくまでもカメラワークを人間が決める際に何の情報を元に考えているかを議論によって推定したパラメータ群を元としているので，実際にパラメータの情報が確実にまとめ上げられているかと考えればそうとも言い切れない．また例えばキャラクターが武器を振った際や，キャラクターが何かしらの物，あるいは部位に注目するような動作をするならば，その武器や物にカメラが注目するべきであると思われるが，このアルゴリズムだとそのような事ができない．

（＊文責：城田晃希）

4.3.1.2.2.3　プログラムの説明

作成したプログラムについて説明を行う．カメラワークを生成するアルゴリズムの核となっているのは CameraManager クラスとなっている．CameraManager のインスタンスを生成した時点でカメラワークのデータの整理と選択を行う CameraSelecter クラスが生成され，物語のデータからカメラワークが生成できる状態になる．CameraManager の CalcCameraPosition 関数は CameraActor クラスを元にカメラワークの選択と計算を行う関数となっている．CameraActor クラスは詳しい内容は後述するが物語のデータと，キャラクターのデータをシステム班との兼ね合いで作業を行いやすいように情報を整理したクラスとなっている．CameraManager クラスの SetCameraPosition 関数は選択したカメラワークパターンと CameraActor クラスからカメラの座標と回転を計算し，カメラワークの

生成を行う関数となっている．カメラワークの計算は写したいキャラクターの位置から相対的に計算をしている．CameraManager クラスの RotateCamera 関数は，カメラの座標と中心とする座標，回転させたい角度を引数としてカメラの回転を行う関数である．肩越しのショットを生成する際に使用する．CameraManager クラスの CalcCenter 関数は複数のキャラクターの座標からその中心の座標を計算するクラスである．ズームアウトショットを生成する際に使用する．

続いて CameraSelecter クラスについての説明を行う．このクラスは分析したデータの整理とカメラワークの選択を行うクラスとなっている．CameraSelecter クラスのインスタンスを生成し初期化を行う事で CameraDataSet クラスの配列と CameraProbabilitySet クラスの配列の初期化を行うが，詳細については後述する．CameraSelecter クラスは引数を分析した CSV データのファイルと全シーンの数を引数として初期化を行い，分析したデータをプログラムで扱いやすいように整理を行う．CameraSelecter クラスの Select 関数はシーン番号（意味合い），場面に存在する人数，写したいキャラクターの行っている動作を元にカメラワークを選択し，String で返す関数である．String の中にはカメラワークパターンが代入され"Long","Full","MidiumLong","BustUp","MidiumClose", "ClosedUp","ZoomOut","OverShoulder"の８つでありそれぞれが選定したカメラワークパターンに対応付けている．

続いて CameraDataSet クラスについての説明を行う．このクラスは CameraSelecter で配列として宣言され，シーン番号（シーンの意味合い）ごとに分析したデータを扱いやすいように整理する役割を持つ．CameraDaraSet クラスの InputData 関数は指定されたファイル名の csv ファイルを読み取り，ReadAllLineCSV 関数を用いてデータの読み込みを行う関数である．ReadAllLineCSV クラスでは指定されたシーン番号のデータのみのパラメータの読み込みを行う関数である．データの格納にはリストクラスの dataArray を用いている．

CameraProbabilitySet クラスの説明を行う．このクラスは CameraSelecter で配列として宣言され，シーン番号（シーンの意味合い）ごとに分析したデータを扱いやすいように整理された CameraDataSet クラスから確率的に傾向を計算できるように，データを変換する役割を持つクラスである．このクラスは CameraDataSet クラスを引数として初期化され，CameraProbability クラスに CameraDataSet クラスのデータを物語のデータごとに分類し，格納する．CameraProbability クラスの Probability 関数は CameraSelecter クラス内で呼び出され，整理されたデータからシーンに登場する人物と，写したい人物を元にカメラワークの選択を行う関数である．

続いて CameraProbability クラスの説明を行う．このクラスは CameraProbabilitySet クラスで二重配列として宣言され，その場に登場する人数と写したい人物の動作に対応する番号がインデックスとなって宣言される．このクラスの役割はその場に登場する人数と写したい人物の動作に対応する番号ごとに何のカメラワークパターンが，いくつあるかをセットし，確率的に計算できるようにする事である．CameraProbability クラスの SetProbab 関数は CameraDataSet のデータからその場に登場する人数と写したい人物の動作に対応する番号のみに対応するカメラワークを探索し，List クラスの probab 変数に格納を行う関数である．CameraProbability クラスの CalcProbab 関数は格納されたその場に登場する人数と写したい人物の動作に対応する番号のみに対応するカメラワークリストの中から，ランダムにカメラワークを選択し，その番号を返す関数である．上記の処理により，確率的にカメラワークを選択することを可能にすると考えた．

続いて CameraActor クラスの説明を行う．このクラスはシステム班により定義された Actor クラスの格納，シーン番号の整理，キャラクターの動作番号の整理，キャラクターの数の格納，写したいキャラクターの頭，右腕，左腕の三次元座標を格納するためのクラスである．シーン番号の整理はとキャラクターの動作番号の整理は Tables クラスを用いて行う．Tables クラスにはキャラクターの動作の String と分析に使用した動作番号を対応付けた Dictionary と物語分析にしようしたシーン

番号と，カメラワークアルゴリズムに使用するシーン番号を対応付けた Dictionary が定義されている．この Tables クラスの必要性としては，動作番号の整理とシーン番号の定義を行う話プロジェクト終盤にたち上がったため，急遽実装を行う必要があったためシステムの開発を簡易化するためである．

最後に JointPos クラスの説明を行う．このクラスは Unity 上でゲームオブジェクトにコンポーネントとして貼り付けて使用し，public として定義されている head,rArm.lArm のゲームオブジェクトクラスにそれぞれ，頭，右腕，左腕の子オブジェクトを格納することが役割である．中間発表ではカメラワークの計算を行う際にカメラとキャラクターとの干渉を考慮していたが，腰の座標を探索し，それを基準として計算をしていたためいくつかの特定の場面でカメラとキャラクターとの干渉が見られた．それを防ぐために頭，右腕，左腕の座標を基準として計算することで，干渉を防ごうとし，JointPos クラスを実装した．

Dictionaly クラスはキーから参照が出来なくなった時点でエラーを起こしてしまうので, 拡張クラスを実装し参照が出来なくても進行が止まらなくなるように変更を行った．

（＊文責：城田晃希）

4.3.1.2.2.4　カメラワーク生成アルゴリズムの開発における反省

開発において度重なる仕様変更や無理やりに作った資源を再利用してしまった事によりソースコードが複雑化してしまい，無意味な処理が多数残る形となってしまった．最初の要件定義や議論をしっかり行わなかった事による結果と考えている．あるいは殆ど知識が 0 に状態から学習しながら開発を行ったために機能や方法を絞り込めてなかったため開発そのものが煩雑化してしまったのではないかと思われる．

JointPos クラスを導入し，カメラワークとキャラクターとの干渉を完全に防止しようと試みたが, 結果としては上手くいかなかった．理由としては，計算上，理論上では干渉は起こりえないのだが, Unity で MMD4mechanim を用いて MMD のキャラクターを制御する仕様そのものの理解が完全には至っていなかったため，処理の順番や時間，見えている情報と実際に内部での処理の違い等から干渉を完全に防止することは出来なかった．

（＊文責：城田晃希）

4.3.2　モデル班

4.3.2.1　モデリング

モデリングとは，コンピューターグラフィックスで立体的な物体の形状を計算及び形成することである．本システムでは，視覚的な表現を 3 次元空間で行うため，オブジェクトに 3D モデルを使用している．視聴覚班ではシステムで使うモデルの規格を統一するために，人物等のモデルを一から制作した．

(＊文責：田中瑞穂)

4.3.2.1.1　モデリングの手順

モデルは，Blender と PmxEditor を使用して作成した．Blender では，モデリングやテクスチャの張り付けを行いモデルの外形の制作を行った．PmxEditor では，モデルの関節を曲げるために必要なボーンの導入を行った．以下に人物モデルの制作手順を記す．

(＊文責：田中瑞穂)

4.3.2.1.1.1　素体を制作

まず初めに既存の様々なモデルを参考にして，服や髪型の付いていない素体を制作した．モデリングに使用したソフトは，前述にあった通り blender である．本来であれば男性と女性は骨格が異なるため，モデルも男性と女性で形状を変更した方が良い．しかし，男性用と女性用で別のモーションを制作する必要が出てくる可能性がある．今回は作業量の短縮のため，大人用の素体と子供用の素体の 2 種類のみ制作し，髪型や服装等のパーツで男女の区別を付けることにした．

(＊文責：田中瑞穂)

4.3.2.1.1.2　キャラクター案を制作

次にどのような髪型や服装のキャラクターを実装するかを話し合い，それを元にキャラクター案を描き起こした．モデリングの負担を減らすため，服装は基本的にシャツとズボン，またはシャツとスカートのみである．髪型も複雑な造形を避け，髪が長いキャラクターは縛るようにしている．これはモーションを実装した時に，身体と髪が干渉するのを防ぐためである．

(＊文責：田中瑞穂)

4.3.2.1.1.3　服及び髪型のモデリング

キャラクターの下絵を元に，服及び髪型のモデリングを行った．制作したモデルのデザインは 4.3.2.1.2 にて後述する．

(＊文責：田中瑞穂)

4.3.2.1.1.4　テクスチャの張り付け

テクスチャとは，3D モデルの表面に貼りつける画像素材のことである．これを元に服や髪の色を設定していく．

まず身体のパーツごとにマテリアルと呼ばれる材質を設定していく．次に 3 次元のメッシュを 2 次元座標に対応させるために，UV 展開を行う．しかしこのまま UV 展開をすると面が重なる箇所が出てくるため，色が塗りづらくなってしまう．そこで展開されたものに色を塗りやすく分割するため，シームと呼ばれる切れ目を入れる．これはあくまで UV 展開を分割するものであり，実際のモデルは分割されない．UV 展開されたものを元に，それぞれの箇所に色を塗っていく．これでモデリングは完了である．

(＊文責：田中瑞穂)

4.3.2.1.1.5　Pmx 形式に出力

MMD や PmxEditor で読み込むために，モデルは Pmx 形式で出力する．Pmx 形式に出力するためには，MMD のファイルを読み込めるようにする為のアドオンが必要となる．手順としては，まず「mmd_tools」をダウンロードして Blender フォルダの「scripts→addons」の中に移動する．次に Blender で「ファイル→ユーザー設定」からアドオンタブを開いて，mmd_tools にチェックを入れて保存する．この作業を行うことにより，Blender 上で Pmx 及び Pmd ファイルの入出力が可能となる．

(＊文責：田中瑞穂)

4.3.2.1.1.6　ボーンの実装

モデルの関節を曲げるために必要なボーンを実装する．ここからは PmxEditer を使用して制作する. 今回は MMD にデフォルトで入っている初音ミクのモデルを参考にボーンを設定した. 膝や肘，指などの基本的な関節，黒目も動くように設定している．

(＊文責：田中瑞穂)

4.3.2.1.1.7　ウエイトを塗る

ウエイトとは，特定のボーンが移動した時にモデルのどの部分が移動するのかを指定するための物である．例えば肘のボーンを曲げても，ウエイトを塗っていないと肘は曲がらない．だが肘にウエイトを塗ると，その部分がボーンと共に曲がるようになる．ウエイトは，曲がる度合いも設定することが出来る．例えば『肘は関節の付近は良く曲がるが，その周りは少ししか曲がらない』といった設定が出来るため，それらを意識することで滑らかに関節を曲げることが出来るようになる．

(＊文責：田中瑞穂)

4.3.2.1.1.8　モーフの実装

最後に表情を実装するためにモーフを実装する．モーフとは物体を自然に変形するための設定であり，これを使用して顔を変形することで表情を実装することが出来る．詳しい説明は後述する．

(＊文責：田中瑞穂)

4.3.2.1.2　作成したモデル

4.3.2.1.2.1　作成した人物モデル

我々モデリング班は，ゲーム内でのカメラの設定との兼ね合いや，モデルに取らせるポーズを設定するときの利便性を向上させるなどの理由から，統一された規格を持つオリジナルのモデルを作成する必要があった．キャラクターの等身は現実的なサイズ感に合わせて高めに設定している．

（＊文責：松原千里）

4.3.2.1.2.1.1　大人

標準的な大人のモデルである．等身は 7 等身程度で，ManO を除いた 4 人は 20 代，ManO は 60 代を想定して作成している．

ManA

銀髪蒼眼の理知的な男性キャラクターである．髪型は 7:3 分けで，シンプルな服を着ている．バトルでは主人公として登場する．完成したモデルは図 4.3.2.1.2.1.1-1 に示す．

図 4.3.2.1.2.1.1-1　完成した ManA のモデル

ManB

茶髪黄眼の優し気な男性キャラクターである．髪型は少し長めで，落ち着いた色合いの服を着ている．完成したモデルは図 4.3.2.1.2.1.1-2 に示す．

図 4.3.2.1.2.1.1-2　完成した ManB のモデル

WomanA

茶髪橙眼の大人しい女性キャラクターである．髪型はボブヘアで，スカートをはいている．完成したモデルは図 4.3.2.1.2.1.1-3 に示す．

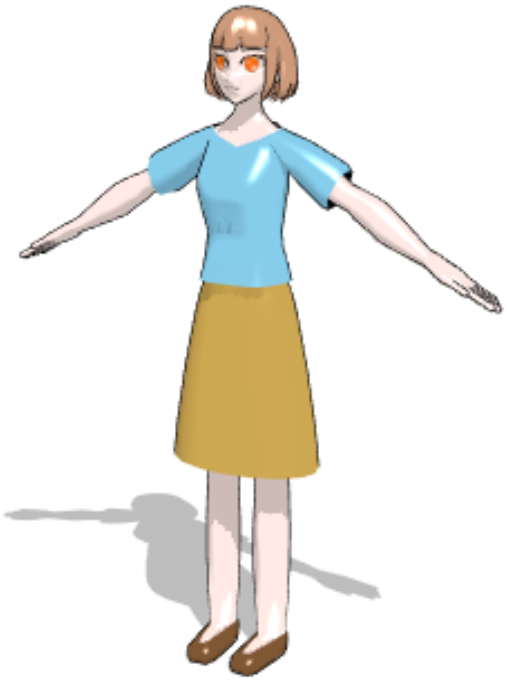

図 4.3.2.1.2.1.1-3　完成した WomanA のモデル

WomanB

金髪蒼眼の強気な女性キャラクターである．髪型はポニーテールで，ズボンをはいている．完成したモデルは図 4.3.2.1.2.1.1-4 に示す．

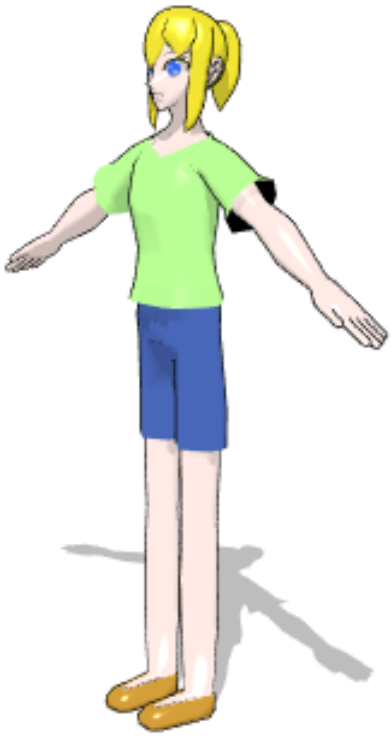

図 4.3.2.1.2.1.1-4　完成した WomanB のモデル

ManO

　老人のキャラクターである．ManB をベースに作られており，髪が無くしわがあるのが特徴である．バトルでは敵キャラクターとして登場する．完成したモデルは図 4.3.2.1.2.1.1-5 に示す．

図 4.3.2.1.2.1.1-5　完成した ManO のモデル

（＊文責：松原千里）

4.3.2.1.2.1.2　子供

大人のモデルの頭部以外を縮小して制作した子供のモデルである．等身は 5 等身程度で，小学一年生を想定して作成している．

BoyA

金髪翠眼の大人しい少年キャラクターである．髪型は長めで，半袖短パンを身に付けている．完成したモデルは図 4.3.2.1.2.1.2-1 に示す．

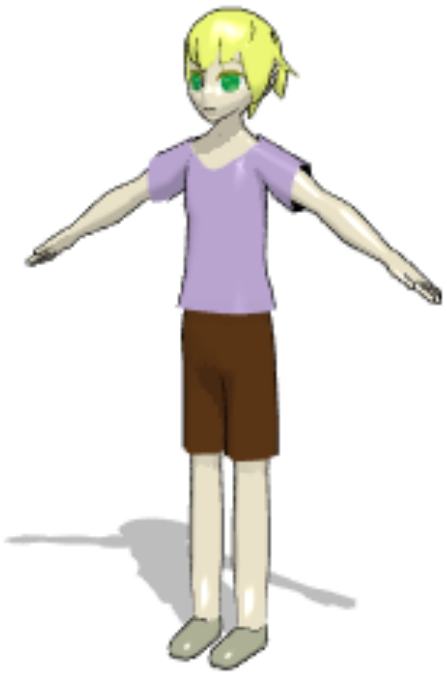

図 4.3.2.1.2.1.2-1　完成した BoyA のモデル

BoyB

黒髪赤目のやんちゃな少年キャラクターである．髪型は短めで，半袖短パンを身に付けている．完成したモデルは図 4.3.2.1.2.1.2-2 に示す．

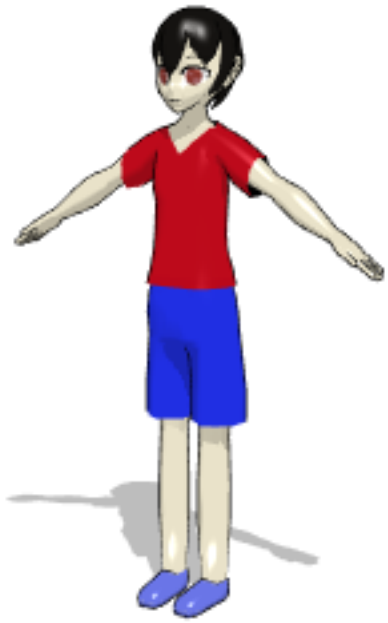

図 4.3.2.1.2.1.2-2　完成した BoyB のモデル

GirlA

　茶髪紫眼のおませな少女キャラクターである．髪型はツインテールで，スカートをはいている．完成したモデルは図 4.3.2.1.2.1.2-3 に示す．

図 4.3.2.1.2.1.2-3　完成した GirlA のモデル

GirlB

黒髪黒眼の大人しい少女キャラクターである．髪型はおさげで，サスペンダー付きのスカートをはいている．完成したモデルは図 4.3.2.1.2.1.2-4 に示す．

図 4.3.2.1.2.1.2-4　完成した GirlB のモデル

（＊文責：松原千里）

4.3.2.1.2.1.3　**人外**

主にホラーで使用するために制作したモデルである．WomanA をベースに制作しているため，ボーンの形状は大人のモデルと同じである．

ghost

幽霊のキャラクターである．WomanA をベースに作られており，肌が青白いのが特徴である．完成したモデルは図 4.3.2.1.2.1.3-1 に示す．

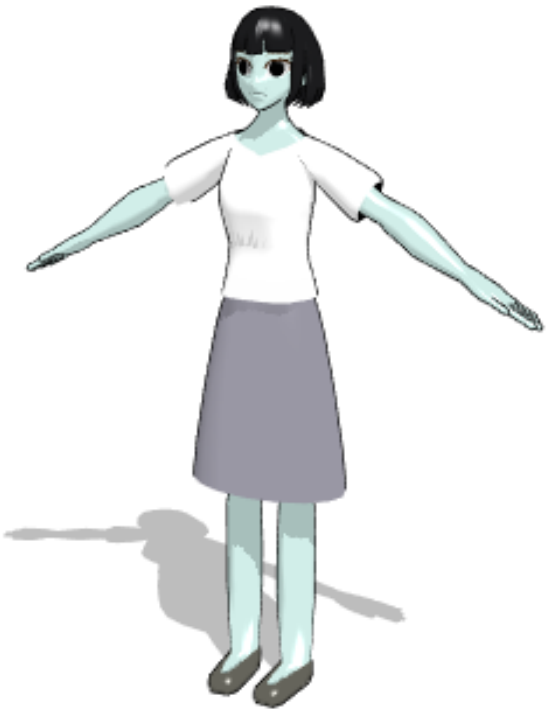

図 4.3.2.1.2.1.3-1　完成した ghost のモデル

monster

化け物のキャラクターである．WomanA をベースに作られており，肌が青色になっているのが特徴である．完成したモデルは図 4.3.2.1.2.1.3-2 に示す．

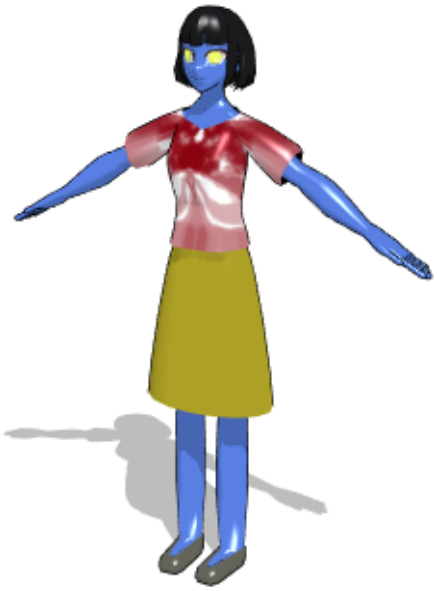

図 4.3.2.1.2.1.3-2　完成した monster のモデル

（＊文責：松原千里）

4.3.2.1.2.2　作成した武器モデル

blade1

持ち手が青色の西洋の剣のモデルである．完成したモデルは図 4.3.2.1.2.2-1 に示す．

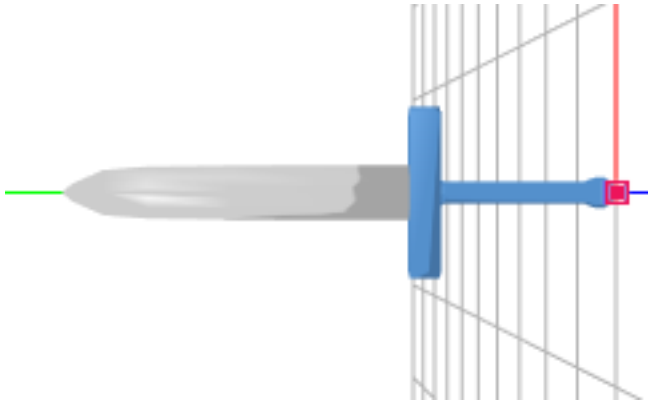

図 4.3.2.1.2.2-1　完成した blade1 のモデル

blade2

持ち手が水色と銀色の西洋の剣のモデルである．完成したモデルは図 4.3.2.1.2.2-2 に示す．

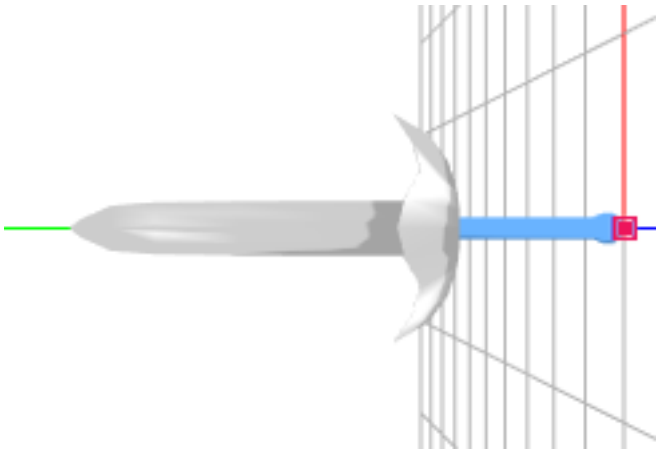

図 4.3.2.1.2.2-2　完成した blade2 のモデル

blade3

持ち手が灰色の日本刀のモデルである．完成したモデルは図 4.3.2.1.2.2-3 に示す．

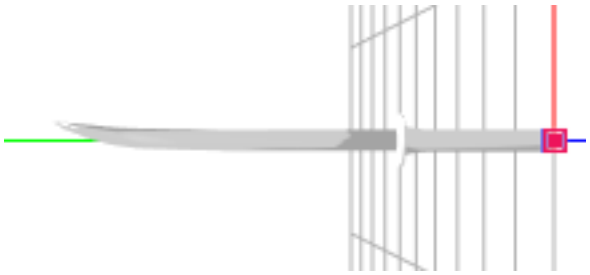

図 4.3.2.1.2.2-3　完成した blade3 のモデル

blade4

鍔部分が黄色のレイピアのモデルである．完成したモデルは図 4.3.2.1.2.2-4 に示す.

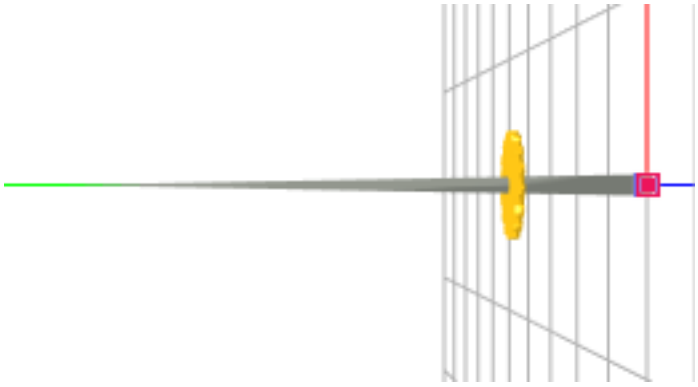

図 4.3.2.1.2.2-4　完成した blade4 のモデル

blade5

持ち手が灰色，刀身が水色の特殊な剣のモデルである．完成したモデルは図 4.3.2.1.2.2-5 に示す.

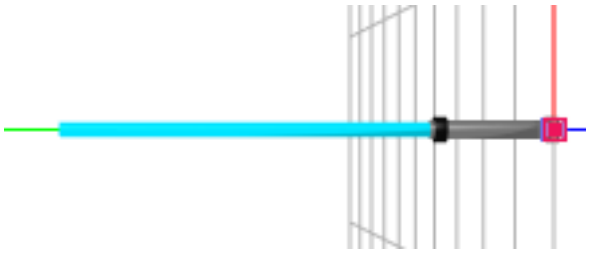

図 4.3.2.1.2.2-5　完成した blade5 のモデル

槍 1

全体が赤色の槍のモデルである．完成したモデルは図 4.3.2.1.2.2-6 に示す.

図 4.3.2.1.2.2-6　完成した槍 1 のモデル

槍 2

全体が灰色の槍のモデルである．完成したモデルは図 4.3.2.1.2.2-7 に示す.

図 4.3.2.1.2.2-7　完成した槍 2 のモデル

槍 3

先端が二股になっている赤色の槍のモデルである．完成したモデルは図 4.3.2.1.2.2-8 に示す.

図 4.3.2.1.2.2-8　完成した槍 3 のモデル

槍 4

槍 2 よりも先端部分に重量感のある槍のモデルである. 完成したモデルは図 4.3.2.1.2.2-9 に示す.

図 4.3.2.1.2.2-9　完成した槍 4 のモデル

槍 5

全体が灰色の薙刀のモデルである．完成したモデルは図 4.3.2.1.2.2-10 に示す．

図 4.3.2.1.2.2-10　完成した槍 5 のモデル

竹槍

竹槍のモデルである．完成したモデルは図 4.3.2.1.2.2-11 に示す．

図 4.3.2.1.2.2-11　完成した竹槍のモデル

棒

全体が赤色の棒のモデルである．完成したモデルは図 4.3.2.1.2.2-12 に示す．

図 4.3.2.1.2.2-12　完成した棒のモデル

（＊文責：松原千里）

4.3.2.1.3　モデリング中の問題の対処法

モデリングを行った際，様々な問題に直面した．今後同じような問題が起こった時のため，以下に実際に起こった問題点とその対処法について示す.

(＊文責：田中瑞穂)

4.3.2.1.3.1　下絵が表示されない

モデリングを行う際, 最初に直面した問題である．下絵は基本的に視点が正面, 右正面, 左正面, 背面，真上，真下に設定されている時しか表示されない．それでも表示されない場合は，平行投影ではなく透視投影になっている可能性が高い．blender の視点は透視投影がデフォルトなので，下絵を表示する場合は真っ先に投影を変更するべきである．投影は『View』の『View Persp/Ortho』から変更できる.

(＊文責：田中瑞穂)

4.3.2.1.3.2　出力できない

解決方法がわからず，一番手間取った問題である．『オブジェクトにマテリアルが設定されていないこと』が主な原因であるので, マテリアルが設定されていることをしっかりと確認すること. (テクスチャは設定されていなくても構わない．)また pmx 出力では非表示のパーツも全て出力されるので，出力前に非表示のパーツの有無も確認するべきである.

(＊文責：田中瑞穂)

4.3.2.1.3.3　Unity 上で正確に表示されない

Unity に導入した際に発生した問題である. 面の向きが統一されていないことが主な原因である．『Shading』の『Backface Culling』で裏面を非表示にすることがで，面の向きを確認することができる．また，面を指定して『w キー→Flip Normals』で面の向きを変更することができる.

(＊文責：田中瑞穂)

4.3.2.1.3.4　モデルが大きすぎる

小さくする他ない．しかし blender 上でサイズを変更して出力し直すと，ボーンやモーフを最初から設定し直さなければならないという問題が発生する．ボーンやモーフを既に実装しているモデルは，PmxEditor を用いてサイズを変更することを勧める．PmxEditor にサイズを変更したい pmx ファイルをドラッグアンドドロップすることでサイズの変更画面が表示される.

(＊文責：田中瑞穂)

4.3.2.1.3.5　モデルが破綻する

この問題は未だに解決に至っていない．改善策として，服と肌を一体化してモデルを制作することが挙げられるが，それでも肩などの関節が破綻してしまう．今後の課題である.

(＊文責：田中瑞穂)

4.3.2.2　モーフ

4.3.2.2.1　モーフィング

モーフィングとは，物体を別の物体へ自然に変形する，コンピューターグラフィックスの手法の１つである．元は 2 次元素材に用いられていた手法だが，現在は 3DCG の表情制御にも用いられている．昨年度の今後の目標に『作品の表現を豊かにするための表情の実装』が挙げられており，今

年度はこれを目標にモデリングを行っていた．その目標を達成するために使用した機能がモーフである．

(＊文責：田中瑞穂)

4.3.2.2.1 モーフの制作手順

モーフは，PmxEditor を使用して作成した．以下に新規モーフの制作手順を記す．

4.3.2.2.1.1 設定を変更する

まず頂点を移動するために PmxEditor の設定を変更する必要がある．上のメニューバーの『頂』と下のメニューバーの『頂点』と『選択頂点』にチェックを入れる．これで点が表示され，点のみを編集できるようになる．点同士のつながりが見たい場合は『Wire』にもチェックを入れると良い．

(＊文責：田中瑞穂)

4.3.2.2.1.2 モーフ編集画面で点を移動する

『編集』の『モーフ編集』をクリックすると図のようなものが表示されるので，名称やパネルを設定してから『編集開始』ボタンを押す．ここで『編集開始』ボタンを押さずに始めると，元のモデルが変形されてしまうので注意すること．ボタンを押した後は，理想の表情になるようにひたすら頂点を移動していく．

(＊文責：田中瑞穂)

4.3.2.2.1.3 SubView で動きを確認する

表情が完成したら，『反映』ボタンを押す．すると SubView でモーフが破綻していないかどうか確認できるようになる．モーフの強度は図の箇所で変更することが出来，初期位置とモーフの間の中間点の確認もすることが出来る．

(＊文責：田中瑞穂)

4.3.2.2.1.4 追加する

編集が終わったら，『追加』ボタンを押す．これでモーフ一覧に新規モーフが追加される．

(＊文責：田中瑞穂)

4.3.2.2.2 制作物について

4.3.2.2.2.1 各パーツの変形

4.3.2.2.2.1.1 眉

上

眉毛が上に移動する．形状は変化しない．

下

眉毛が下に移動する．形状は変化しない．

にこり

眉毛がアーチ状に変形する．

真面目
眉毛が逆八の字型に変形する．

怒り
眉毛が逆八の字型に移動し，眉間にしわが寄る．

困る
眉毛が八の字型に変形する．

（＊文責：松原千里）

4.3.2.2.2.1.2　目

瞳小
黒目が縮小する．

まばたき
逆アーチ状に両目を閉じる．

笑い
アーチ状に両目を閉じる．

ウインク
アーチ状に右目をのみを閉じる．

眠り
並行に横並びになるように両目を閉じる．

（＊文責：松原千里）

4.3.2.2.2.1.3　口

あ
『あ』と発音する時のように口を開く．

へ
への字のような形をとる．

歯ぎしり
歯が見える程度に口を横に開き，歯をかみしめる．

にやり
通常の状態から，両方の口角が上がる．

にこ

両方の口角を上げたまま口が開く.

にや

左側のみ口角が上がる.

（＊文責：松原千里）

4.3.2.2.2.2　表情について

上記であげたモーフの中から 3～7 種類の数値を変更，組み合わせることで，以下の 11 種類の表情を制作した.

無表情

通常時である．モーフを一切適用していない，デフォルトの状態である.

怒り

怒りの表情である．バトルで怪我をさせられた時，相手に不満がある時などに使用する．『眉：怒り(1.00)』と『眉：下(0.16)』と『目：瞳小(0.15)』と『口：あ(1.00)』を組み合わせて作成している.

驚き

驚きの表情である．突然何かが起こった時，衝撃の事実を知った時などに使用する．『眉：にこり(1.00)』と『眉：怒り(0.25)』と『眉：上(0.50)』と『目：瞳小(0.55)』と『口：あ(1.00)』を組み合わせて作成している.

余裕

余裕の表情である．弱い相手と戦う時，相手を見下す時などに使用する．『眉：にこり(0.64)』と『眉：上(0.50)』と『口：にやり(0.68)』を組み合わせて作成している.

恐怖

恐怖の表情ですある．恐ろしい物を見た時，絶望的なことが起こった時に使用する．『眉：困る(0.47)』と『眉：怒り(0.26)』と『目：瞳小(0.61)』と『口：あ(0.96)』と『口：へ(0.41)』を組み合わせて作成している.

嫌悪

嫌悪の表情である．苦手な相手に会った時，見たくない物を見た時などに使用する．『眉：怒り(0.45)』と『眉：困る(0.39)』と『眉：下(0.50)』と『目：まばたき(0.25)』と『目：ウインク(0.20)』と『口：へ(0.50)』と『口：あ(0.40)』を組み合わせて作成している.

苦しみ

苦しみの表情である．怪我をしている時，病気を患っている時などに使用する．『眉：困る(0.73)』と『眉：怒り(0.65)』と『目：ウインク(0.35)』と『目：まばたき(0.25)』と『口：歯ぎしり(1.00)』を組み合わせて作成している.

喜び

喜びの表情である．幸せなことがあった時，友人と話している時などに使用する．『眉：にこり(1.00)』と『眉：上(0.25)』と『目：笑い(1.00)』と『口：にこ(1.00)』を組み合わせて作成している．

喜び２

上記の『喜び』の，目を閉じていないバージョンである．より，喜びが大きい時などに使用する．

悲しみ

悲しみの表情である．相手に負けた時，誰かが亡くなった時などに使用する．『眉：困る(1.00)』と『眉：怒り(0.10)』と『目：まばたき(0.15)』と『口：へ(0.65)』を組み合わせて作成している．

真剣

真剣な表情である．集中している時，強い相手と対面した時などに使用する．『眉：真面目(0.75)』と『眉：困る(0.15)』と『目：まばたき(0.10)』と『口：へ(0.30)』を組み合わせて作成している．

（＊文責：松原千里）

4.3.2.2.2.2.3　追加要素について

直接的な顔の動きとは異なる追加要素である．それぞれの表情に追加して使用できるようになっている．血 3 種類は同時に表示することが出来るが，血や涙などの種類の違う追加要素は，他の追加要素と同時に表示することが出来ないようになっている．普段は人物モデルの頭内部に格納している．

涙

目の下に涙がたまる．悲しみなどを強調することが出来る．

汗

顔から汗を流す．焦り，疲労などを強調することが出来る．

血額

額から出血する．相手の攻撃を受けて負傷した時に利用する．

血口

口の端から出血する．

血頬

頬から出血する．

（＊文責：松原千里）

4.3.2.2.3　モーフの問題の対処法

モーフを行った際，様々な問題に直面した．今後同じような問題が起こった時のため，以下に実際に起こった問題点とその対処法について示す．

(＊文責：田中瑞穂)

4.3.2.2.3.1　間違った点を指定してしまう

編集中に良く起こり，ストレスとなる事象である．これは上のメニューバーの『絞』から編集したいパーツを絞り込むこと，下のメニューバーの『表頂』で表の点のみを表示することで表示する点を減らすことで解決出来る．

(＊文責：田中瑞穂)

4.3.2.2.3.2　モーフが勝手に消える

『編集開始』ボタンを押し忘れた時に，よく起こる問題である．恐らくモーフ編集外で頂点を移動すると，対応する頂点を使用していたモーフが消えてしまうのではないかと考える．なお消えてしまったモーフを復元する手段はないため，こまめに確認と保存をすることを勧める．

(＊文責：田中瑞穂)

4.3.2.3　モーション

モーションとは「動作」や「運動」を意味する英語であり，IT 用語では映像処理や 3DCG における動作処理を意味する．本稿では，モデルを移動するための動作処理としてこの言葉を用いる．今回のシステムでは，人物モデルの動きの表現や武器モデルの位置調整にモーションを用いた．

(＊文責：田中瑞穂)

4.3.2.3.1　モーションの制作手順

モーションは MMD を使用して作成した．以下にモーションの制作手順を記す．なお今回のモーションは特定の位置から動かないため，動かし方についての説明は省いている．

(＊文責：田中瑞穂)

4.3.2.3.1.1　MMD にモデルを読み込む

図の『モデル操作』の『読込』から pmx 形式のモデルを読み込むことが出来る．この時そのモデルにボーンが設定されていれば，ボーンも同時に表示される．

(＊文責：田中瑞穂)

4.3.2.3.1.2　ボーンを移動させる

モデルのボーンを移動及び回転させて，思い通りのポーズをとらせる．もしも移動や回転が出来ない場合はボーンそのものの設定が間違っている可能性があるので，PmxEditer の「ボーン」で設定を確認及び修正する必要がある．

(＊文責：田中瑞穂)

4.3.2.3.1.3　モーションを保存する

まず図の『登録』ボタンを押す．これを押さないで保存すると，ボーンの位置が初期から変更されないままモーションを保存することになるので注意すること．次に登録したモーションをコピーペーストして複数フレーム分用意する．この工程を飛ばすと，ボーンが途中で初期位置に戻ること

がある．最後に『ファイル』から『モーションデータ保存』を選択し，モーションの名称を決定して保存する．これでモーションは完成である．

(＊文責：田中瑞穂)

4.3.2.3.2　制作したモーション

上記の手順で制作したモーションである．主に以下の 4 種類に分類できる．

(＊文責：田中瑞穂)

バトルシーン用モーション

バトルシーン用に制作した，戦闘向けの激しい動きのあるモーションである．制作したモーションは以下の通りである．

- 移動前
- 移動後
- 素手構え
- 素手殴る
- 素手投げる
- 素手防御
- 剣構え
- 剣振り下ろし
- 剣横薙ぎ払い
- 剣防御
- 槍構え
- 槍突き
- 槍横薙ぎ払い
- 槍防御

武器用モーション

バトルシーン用モーションに対応した，武器に適用する用のモーションである．武器にこのモーションを適用することで，人物モデルの手の中に武器をちょうどよく収めることができる．

幽霊用モーション

幽霊のモデル用に制作したモーションである．同じボーンを使用している大人モデルにも適用できるが，人外モデルに実装することを前提として制作している．制作したモーションは以下の通りである．

- 立つ 1
- 立つ 2

その他通常モーション

腕を後ろに回すテストをした際に出来た『拘束』．武器を持たせるのと同じ要領で，小道具を持たせるための『持ち歩く』のモーションを制作した．また，小道具用のモーションはシステム上では実装されていないため利用されることはないものとなっている．

(＊文責：田中瑞穂)

4.3.2.3.3　モーションの問題の対処法

モーフを行った際，様々な問題に直面した．今後同じような問題が起こった時のため，以下に実際に起こった問題点とその対処法について示す．

(＊文責：田中瑞穂)

4.3.2.3.3.1　モーションが保存されない，初期位置に戻る

モーションを保存されず，ボーンが初期位置のままになっていることがある．これは『登録』ボタンを押さずにモーションを保存していることが主な原因である．モーションが完成したら，保存の前に登録を押すこと．またモーフのモーションは，ボーンと登録ボタンの箇所が異なるため注意すること．

(＊文責：田中瑞穂)

4.3.2.3.3.2　ボーンが上手く選択できない

ボーンを入れた後，PmxEditer の『表示枠』を下の図のように設定する．こうすることにより MMD の左側にボーンの一覧が表示されるようになる．ここから編集したいボーンを押すことで，任意のボーンを選択することができる．『表示枠』を用意しておくと選択箇所がわかりやすくなるので，設定することを勧める

(＊文責：田中瑞穂)

4.3.2.3.3.3　Unity で読み込めない

Unity で読み込めなくなる原因は様々だが，例としてファイルの形式があげられる．MMD で制作できるファイル形式には，vmd ファイルの「モーション」と vpd ファイルの「ポーズ」が存在する．しかし Unity では vpd ファイルは読み込むことが出来ない．もしも Unity で読み込みが出来ない場合は，ファイルの形式が vpd になっていないか確認してみると良いだろう．

(＊文責：田中瑞穂)

4.4　成果発表

2018 年 12 月 7 日に公立はこだて未来大学内にて成果発表が行われた．成果発表を行うにあたって，プロジェクト内容が理解できるような A1 サイズのポスターを 7 枚制作し，1 回 20 分のポスターセッションを 6 回行った.また，プレゼンの視聴者に向けてアンケートを取り，評価を受けた．
成果発表で 77 人から受けた評価は表に示す結果となった．

表 4.4　成果発表での評価

評価	発表技術
1	0
2	0
3	1
4	1
5	5
6	13
7	22
8	19
9	5
10	5
無回答	6
平均点	6.7

発表技術と発表内容についてアンケートを取った．まず，発表技術についての一例を示す．

- 全体の説明で少し各班の話が聞ければよかったと思う
- 動画が用意されていてわかりやすかった
- ポスターセッションなのに説明者が受動的だった
- 全体終わってから各班についての説明もなく質問形式になって伝えたいことが分からなかった．
- プレゼンターの話し方が聞きやすくてよかったです．
- 実際にデモを見ることができたので，理解しやすかった．
- クリエイター支援に大きくつながりそうでよかった．まだ実証が少ないかなと感じるのでこれからに期待しています．
- 同じ設定にしてもワンパターンにならないのもよい
- 分析だけではなく実際の映像まで作れていてよかった．カメラワークの種類がもっと増えてもいいかなと感じた．
- プロジェクトの要点が分かりやすく，はきはきした発表だった
- 各班ごとの説明が詳細にあってわかりやすいと思った
- 最初の全体の説明で，各班の活動がわかりやすかったです．
- 発表時のデモは最初は 1 つずつ流したほうが分かりやすいと思った．
- 声が聞き取りにくい場所だからこそもう少し声を大きくしてほしかった．動画ではなく実際に使用している場所を見たかった．
- 各ポスターの距離が近いのでかき消されたりする．
- ポスターの近くに人を集めるよう呼びかけたほうがいい．ポスターセッションの流れを最初に話していたのがよかった．
- 質問に対して丁寧に答えてくれたのでとてもわかりやすかった．

次に，発表内容についての一例を示す.

- パネルやモニターを使いより効果的に発表出来ていると思います．ですが，もう少し積極的に発表内容を説明したほうが良いと思います.
- 実際の映像で見せてくれるのは視覚的にもよくわかりやすい．それぞれアピールしたいポイントなどが特に聞けなかったことが残念
- 分析するのが，１つのジャンルにつき１つや２つだと傾向が似ていて，あまり代わり映えのない物語になってしまうような気がするので，分析する本はたくさんあればバラエティにとんだ作品ができると思う.
- 自身が選択した内容が反映されるのは非常にインタラクティブだと思った.
- 全体の説明が抽象的すぎてどこがすごいのかいまいち伝わって来なかった．たぶん，個別のポスターでは説明さしていたのだと思うが詳細すぎて全体像がわかりにくい
- グラフにまだ改善の余地があるように感じた．武器がどのキャラクターでも実装できるのは良いと思う.
- 特徴的なカメラワーク８種類とあったが，その他にもまだ特徴のあるカメラワークなどがあると面白いと思いました.
- 物語の自動生成→映像化の仕組みがすごいと感じた．カメラワークも自動生成の不自然さがない.
- 物語の分析や抽出などの過程を説明できていてよい.
- カメラワークでそれぞれのジャンルのカメラワークを分解しているのがこだわっていると感じた
- アクションとホラーのストーリーがかなり違うものになっており，それぞれ特徴も取れていたのですごいと思った
- 映像化だけでなくゲーム化すると用途が増えると思う.
- 物語生成だけでなく，映像生成など，幅広く取り組んでおられて驚きました
- 展開を推移確率モデルを使い分析しているのがいいと思いました.
- 場面に合わせて音楽が流れるとより面白くなると感じました.
- 同じ主人公の説明という分類でも，複数の要素などから生成して複雑にすると面白いものになると思う
- 自動的にプロットが構成される仕組みは理解できたが，”AI”である要素が見えなかったように思える．自動書記としての機能がクリエイティブな動作をしているかは疑問である．しかし自動化はある程度行えており面白かった.

発表技術については，アンケートの結果からポスターセッションなのに発表者が受動的である．また，ポスター同士の距離が近く声が重なって聞き取りにくい．など発表自体をうまく聞くことができなかったという意見が多く見られた．このことから，発表者は積極的に声掛けを行い，聴講者を近くに集めた後に自分の説明を行えるとこの問題は改善できたと感じた．また，デモ動画に関しては，発表の内容の理解促進に繋がったという様な意見も多くあった．したがって，聴講者にとって分かりやすいデモの用意は必須であると感じた.

次に，発表内容については，本プロジェクトの目的に関心のある人が多い様に感じられた．したがって，この研究課題は多くの人から期待されている内容であるため，これからも発展してくべき分野であると感じた．また，分析する物語のジャンルを増やすことで自動生成できる物語の幅を増やせるという意見があった．この意見から，この研究課題は発展させるには，人間の作った物語作

品を様々な視点から分析することが重要であると感じた.

(＊文責：寺島啓悟)

4.5 評価実験

4.5.1 目的

クリエイティブ AI の最終的な目標は，自動生成された多種多様なプロットに基づく魅力的な映像を鑑賞できるシステムを開発することである．その目標を達成しているかを確認するためにこの評価実験を行なった．目標の達成を確認するためには，調べる内容をより具体的に決める必要がある．そこで，3 つの観点を定めた．1 つ目は，自動生成された物語が物語として成立していること．2 つ目は，自動生成された物語が読み手に対して，影響を与えることができていること．3 つ目は，適切なカメラワークの選択により，シーンの意味を理解できていること．以上の 3 つを明らかにすることで目標の達成を確認できると考えた.

（＊文責：南部太雅）

4.5.2 実験方法

評価実験では，目的で述べた 3 つの観点を調べるために良いか悪いかという問いに対しての 5 段階でのアンケート調査を実施した.アンケートの質問は,「話が自然であった.」,「物語を楽しめた.」,「物語の全体を通して場面の状況が理解できる.」の 3 つである．被験者は 10 代から 20 代前後の男女 12 人に協力してもらった.

実験の手順は，周りが静かな集中できるところに被験者を待機させる．その後，被験者に同意書を書いてもらい，実験の目的とアンケートの仕方を教示した．そして，アンケート用紙を被験者に渡し，開発したシステムによって自動生成された物語を見せた．見せ終わった後にアンケートに答えてもらった．実験をする上での注意点として，被験者同士および実験者との物語に関する会話を禁止した.

分析の方法として，ホラーの物語とバトルの物語の評価の項目にある「普通」で区切り，「良い」のグループと「悪い」のグループに分け，「とても良い」と「良い」の度数を足したものを「良い」のグループの数とする．また，「悪い」のグループの数も同様に，「悪い」と「とても悪い」の度数を足したものとする．この時，「普通」の度数は考えないとする．そして，2 つの物語の「良い」グループの数と「悪い」のグループの数を比較し，数が大きい方の物語を満足度が高いとする.

（＊文責：南部太雅）

4.5.3 分析結果

分析した結果より，ホラーの物語に関しては目的で定めた 3 つの観点を十分に満たしているといえる．また，バトルに関しては，最低限 3 つの観点を満たしているといえるが，ホラーの結果のように十分に満たすことはできていないといえる．以下に実験の結果より得られたグラフを示す.

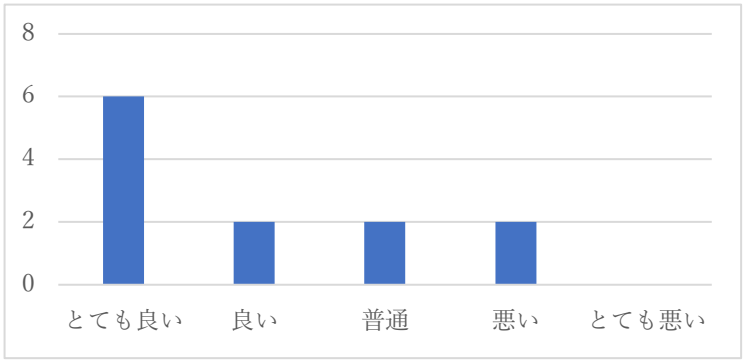

図 4.5.3-1　話が自然であった(ホラー)

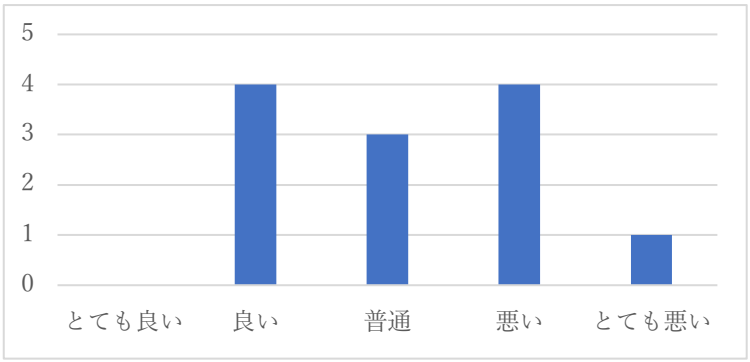

図 4.5.3-2　話が自然であった(バトル)

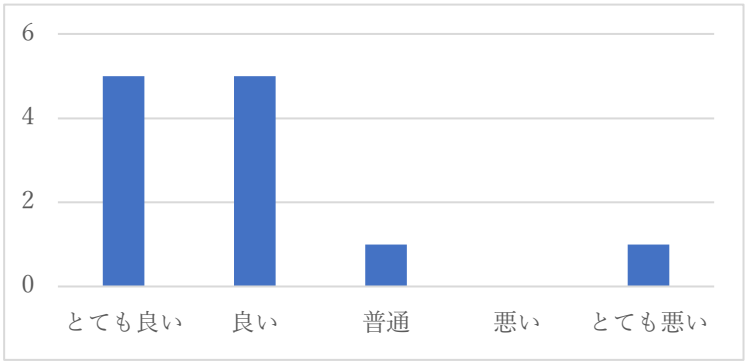

図 4.5.3-3　物語を楽しめた(ホラー)

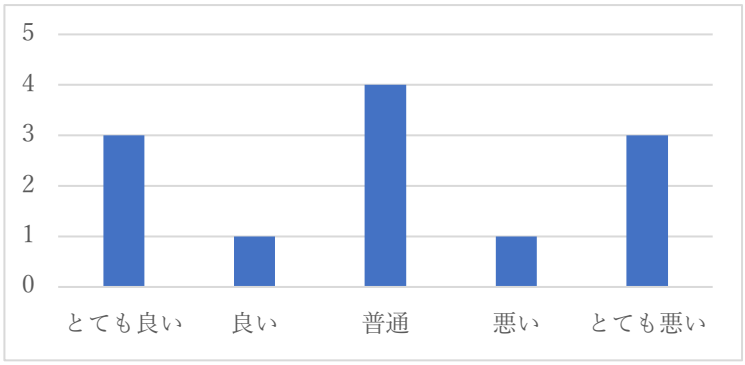

図 4.5.3-4　物語を楽しめた(バトル)

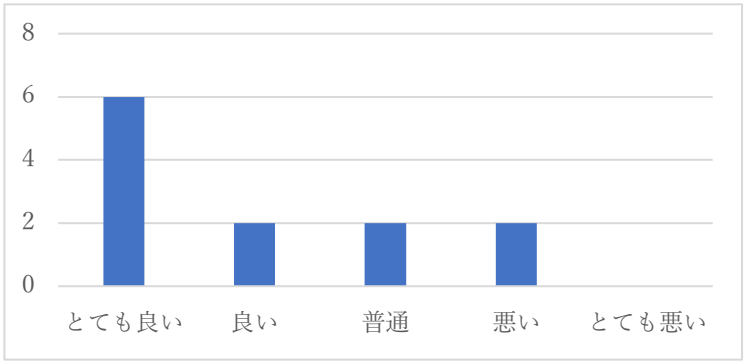

図 4.5.3-5　物語の全体を通して場面の状況が理解できる(ホラー)

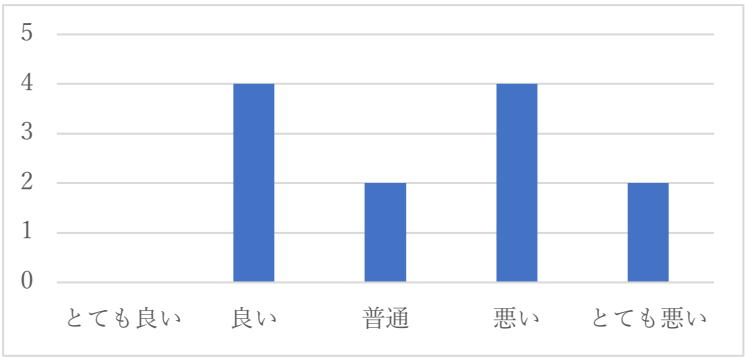

図 4.5.3-6　物語の全体を通して場面の状況が理解できる(バトル)

図 4.5.3-1 と図 4.5.3-2 のグラフに関しては，ホラーとバトルの「話が自然であった」という問いについての集計のグラフである．この 2 つのグラフを比較すると，ホラーの物語の方が圧倒的にバトルの物語より高評価であることがわかる．また，両方のグラフを「普通」の項目で区切り，「良い」のグループと「悪い」のグループに分け，それぞれの数を求めるとホラーの物語の「良い」のグループの数は 8，「悪い」のグループの数は 2 である．そして，バトルの物語の「良い」のグループの数は 4，「悪い」のグループの数は 5 である．このことから，ホラーの物語に関しては「良い」のグループがバトルの物語の「良い」のグループより，高いのでホラーの物語は満足度が高いといえる．したがって，ホラーの物語は，物語として成立している．また，バトルの物語に関してはホラーの物語ほどではないが，物語として成立しているといえる．そして，ホラーの物語の方がバトルの物語より，満足度が高いので，物語の完成度は高いといえる．

図 4.5.3-3 と図 4.5.3-4 のグラフは「物語を楽しめた」という問いについての集計のグラフとなっている．この 2 つのグラフをみると，ホラーの物語とバトルの物語は比較的に楽しめたといえる．しかし，「話が自然であった」のグラフと同じように，普通で区切った場合，ホラーの物語は，「良い」のグループの数が 10 で，「悪い」のグループ数が 1 であった．バトルの物語は，「良い」のグループが 4 で，「悪い」のグループが 4 であった．それぞれ物語の「良い」のグループの数を比較してみると，ホラーの物語の方がバトルの物語の「良い」のグループの数より多いので，満足度が高いといえる．したがって，物語が読み手に影響を与えることができるということに関しては，2 つの物語は満たしていると言える．そして，こちらの問いに関しても，ホラーの物語の方が，バトルの物語より，満足度が高いので，物語の完成度は高いと言える．

図 4.5.3-5 と図 4.5.3-6 のグラフは，「物語の全体を通して場面の状況が理解できる」という問いについての集計のグラフである．この 2 つのグラフを見ると，ホラーの物語の方が，理解できている

人が多い．しかし，バトルの物語に関してはあまり理解できているとは言えない．また，これまでと同様に，「普通」で区切った場合，ホラーの物語の「良い」のグループの数は８で，「悪い」のグループの数は２であった．バトルの物語は，「良い」のグループの数は４で，「悪い」のグループの数は 6 であった．それぞれの物語の「良い」のグループの数を比較すると，ホラーの物語の「良い」のグループの数の方が，多いので満足度が高いと言える．したがって，ホラーの物語に関しては，適切なカメラワークを選択できていると言える．しかし，バトルに関しては，適切なカメラワークは選択できていないと言える．そして，こちらもホラーの物語の方が，完成度が高いといえる．

(＊文責：南部太雅)

4.5.4 考察

評価実験の結果からホラーとバトルの物語の内容や表現の違いが評価に影響を与えたと考える．まず物語の自然さについて考えると，ホラーの物語では初めに物語の世界観の説明から入り，主人公が目的を達成していくようになっていた．評価結果からもわかるが，物語自体に不自然と感じるところはなかった．一方バトルの物語では初めに物語の最終的な戦闘部分を一部見せてから，主人公がその最終的な戦闘に至る様子を描いている．被験者がバトルの物語をあまり自然と感じなかった理由として主人公が戦闘を行う明確な理由が作中で記述されていなかったこと，主人公の存在する世界観が不明確であったことが考えられる．物語を楽しめたかについての評価も上記で示したことでバトルの物語はホラーに比べて評価が低かったように思われる．

次に物語全体の場面状況の理解度について考える．バトルの物語では所々におかしなカメラワークがあったように思われる．一例を示すと，登場人物の持つ武器にズームがされていたり，登場人物がセリフを話しているシーンでカメラが背景だけを映してあった．また，実験後に被験者から登場人物の戦闘中にセリフや状況を説明した解説文等がなかったため，戦闘時の主人公や敵の状況が理解しにくかったという意見も頂いた．戦闘中のセリフの表示は開発チームの負担が大きくなり，時間的にも厳しかったため実装には至らなかったのでその分物語のデータ作成時に登場人物の位置関係や場面の映し方に工夫を凝らす必要があると実感した．

今回の評価実験の結果を受けて物語の自然さ，わかりやすさを違和感なく表現する際に注意すべき点や考慮すべきことを改めて考える必要があると感じた．

(＊文責：袴田翔)

第 5 章 課外活動

5.1 北の四大学ビジネスプラン発表会 2018

2018 年 12 月 15 日，北海道札幌市にある北海道庁旧本庁舎にて北の四大学ビジネスプラン発表会が行われた．北の四大学ビジネスプラン発表会とは，北海道にある 4 つの大学，公立はこだて未来大学，小樽商科大学，帯広畜産大学，北見工業大学がそれぞれの持つ文系や理系の知識を合わせて将来の社会構想を描く催し物である．主に地域課題への着目とその解決策，その解決策に用いられる技術力とプランの独創性，プランが現在の社会問題や，将来的な社会へどのような貢献ができるのか，そしてプランは実用化・実践できそうなのかを 4 つの企業が審査員となってビジネス目線で審査・評価し，最も優れたプランを発表した大学に最優秀賞が贈られる．

北の四大学ビジネスプラン発表会は本来，夏と冬の 2 回行われる計画であり，夏の合宿で他大学の学生と交流しチームを組み，与えられた課題をチームで意見を出し合い解決をしていくことになっていた．しかしながら，2018 年 9 月 6 日に北海道胆振東部地震が起きてしまったため夏の合宿はやむを得ず開催されることはなかった．それにしたがい，北の四大学ビジネスプラン発表会は冬のみの開催となり，本来夏の合宿で行われるはずだった他大学の学生と顔合わせは Skype を通じて簡単にプロジェクトメンバーと研究分野を紹介することになった．

ビジネスプラン発表会に参加するにあたり，本プロジェクトの内容と成果をビジネス目線で応用できるかを考えた．しかしながら，プレゼンテーションを聞く人の殆どは人工知能を知らない可能性が高いため，私たちが講義やプロジェクト学習を通じて学んできた人工知能の知識をどのようにしてわかりやすく伝えることができるかを念頭に置いてアイデアを出し合った．ビジネスプランに関連付けた構想として，ニュース原稿やテレビコマーシャル，ゲームシナリオの自動生成への応用を考案した．実際にフランスでは物語自動販売機が存在し，短編物語をシートで出してくれる．この応用例から企業に向けたアイデアとしてテレビコマーシャルは良い観点だと思われた．

ビジネスプラン発表会の結果，プロジェクトの概要やその成果を用いたプランは参加者や審査員にはあまり理解されなかった印象であった．原因として，人工知能を知らない人への説明をしたものの伝わりにくかったことが考えられる．私たちの講義やプロジェクト学習では人工知能という分野が当たり前の存在になってしまっており，無知の人が知るべき本質を説明出来ていなかった．何よりわかりやすさとは何かを深く考えることが足りてないと言える．

また，現代の科学技術でメディアで取り上げられている人工知能の分野と言えば，会話ロボットやオセロ・将棋をするといった技術の方がよく知られており，自動生成の分野は一般的には知られていないように思われた．

以上から私たちが理解している内容をどのようにわかりやすく伝えるか，またどのように物語の自動生成という分野を社会に広めていくかを考える必要があると感じた．

ビジネスプラン発表会で私たちが考案したプランは質が高く，発表技術も申し分ない物だと思う．しかしながら，ビジネスへの応用を考える時間的余裕がなく準備が不十分であった部分もある．他大学のプランには何年もかけて成果物をアプリとしてリリースしたり，企業から出資をしてもらい実際に起業して地域でビジネスをしているものもあった．やはり長く時間をかけて課題に取り組み，様々な機関と連携して解決していくことが社会の問題をとらえる際に重要であるとわかった．審査員からは先端技術の紹介やビジネス観以外の将来的な技術の発展分野として高い評価を頂いたので私たちの成果に自信が持つことができた．

（＊文責：袴田翔）

5.2 第 50 回 EC 研究発表会

2018 年 12 月 21 日，22 日の 2 日間，公立はこだて未来大学で行われた「第 50 回 EC 研究発表会（第 50 回 情報処理学会エンタテインメントコンピューティング研究会 研究発表会)」に参加し，萌芽的発表を行った．なお，萌芽的発表とは発展途上の研究発表や研究テーマの筋の良さを確認するための研究発表である．プロジェクト学習の成果発表とフィードバックを目的に本プロジェクトの代表として，プロジェクトメンバーの鈴木諒輔と教員の村井源が参加した．萌芽的発表は発表時間 10 分，質疑応答 5 分の計 15 分で行った．発表内容は本プロジェクトの成果物である統合的な物語自動生成システムについて，その背景からシステムの概要，今後の予定や課題をわかりやすく発表した．発表に用いたスライドに関しては，少し内容を硬い表現にしてしまった部分もあったので，バラエティに富んだ見ただけでわかるようなものにすればよかったと反省した．発表や質疑応答を通じて，概ね伝えたいことが言えた点や質疑応答も返答できた点は良かったが，マイクの使い方が下手で聴講者が聞きづらいような喋り方になってしまった点は改善点であった．

（＊文責：鈴木諒輔）

5.3 秋葉原課外成果発表会

本プロジェクトは 2019 年 2 月 18 日に東京都秋葉原にある秋葉原 UDX で行われる課外成果発表会に参加する．この課外成果発表会は例年はこだて未来大学の主催で行われており，本プロジェクトからは高橋翔太と寺島啓悟の二人が参加する．この成果発表会ではプロジェクトで作り上げた物語生成システムだけでなく，システムの内部のインフォメーションフローや完成に至るまでの試行錯誤，全体を通して得た知見を発表する．また，この発表会で得られた知見を持ち帰り，来年度以降のプロジェクト活動に役立てるつもりである．

（＊文責：高橋翔太）

第 6 章 プロジェクト内のインターワーキング

- 鈴木諒輔

  私はシステム班に属し，統合的な物語を自動生成するシステムの開発を行った．具体的には，物語を自動生成するために必要な物語プロットを生成するために，初期条件などをシステムに入力するための GUI の作成などを行った．Python 班としての活動として，システムの作成をしたが，メンバーに助けられることも多く，さらなる技術力の向上を図らなければならないと行けないと痛感した．しかし，メンバー間のコミュニケーションや連携に不足はなく無事にシステムの開発を余裕を持って開発することができてよかったと感じている．

  またプロジェクトリーダーとして，プロジェクトの運営方針・開発目標・開発計画などをまとめるとともに，司会進行や作業分担，事務作業などを行った．さらに，課外活動にも積極的に参加し，プロジェクトの成果を学外へ積極的に発信することに努めた．プロジェクトにおけるコミュニケーションも円滑に取ることができたことやスケジューリングもうまくできたことは良い点として今後にも生かしていきたい．しかし，全体ミーティングなどでは言葉足らずなところも災いして，自分の伝えたいことを一発で伝えきれなかったりした場面もあったので，もっと一対多における会話力を向上させなければならないとも感じた．これらのプロジェクトで学んだことや良かった点，改善点などを踏まえながら，今後の大学での活動に活かしていきたいと思っている．

  （＊文責：鈴木諒輔）

- 高橋翔太

  システム班のグループリーダーとして活動したが，基本的に Unity 班のメンバーと活動することが多く Python 班の動向に気を配っていなかったため，システムの連結作業などでお互いの考えの食い違いが発生したことが何度かあった．活動場所の広さの関係上 Unity 班と Python 班が別々の部屋で活動することになったという理由もあるが，定期的に進捗状況などを聞きに行くことはできたため反省したい．Unity 班の中でも中心として進捗状況の発表などを行った．自分の知識不足によるエラーが幾度か発生したがメンバーのお陰もあって班としては上手く回っていた．

  個人の仕事としては ScenarioLoader クラスの作成と一部ステージの制作を行った．ScenarioLoader クラスの制作にあたっては物語データの型を決める段階から Python 班や物語分析班と協力して制作しており，相互の情報のやり取りを行う機会が多く，コミュニケーションが重要となった．

  （＊文責：高橋翔太）

- 松浦史佳

  私は物語分析班に所属し，ホラー作品の分析を行った．具体的には，ホラー作品を読んで自動生成に必要なデータを抽出したり，物語を分類してどう遷移するか数えたり，自動生成される文章を考えたりした．グループ活動では，作業で分からない点があればメンバーに質問したり相談したり自分では意欲的に作業していたつもりであったが，仕事量でみるとメンバーのほうが率先して作業を行っていたため，積極性が足りなかったと反省している．また，物語分析班のグループリーダーを務めたが，ホラー班とバトル班での共同作業は少なく，自分の班の作業に集中してバトル班の動向を把握することができなかったため，この点も反省である．しかし，たてた計画の通りに作業が進み，余裕を保ったまま終わらせることができたため，この点は今後も生かしていきたい．

  （＊文責：松浦史佳）

- 寺島啓悟

  私はシステム班に所属し，物語を自動生成するシステムの開発を行った．具体的には，物語を映像で表現するために必要な文章データである物語プロットデータの自動生成を行うシステムの開発を Python 言語や，データベースソフトを用いて行った．主にホラージャンルのシステムの開発を行っていたので，ホラージャンルらしい物語の自動生成のために，主にホラージャンルの分析班と円滑な連携を取り，システムを期間内でより良くするために何を実装したらよいか話し合うことが出来た．また，システムの開発を進める中で，他の班との連携も必要であったため，適宜コミュニケーションをとることが出来た．その結果，プログラムで実装する方法を話し合った内容を踏まえて考えることが出来た．その為，システムの開発をスムーズ進めることが出来た．

  （＊文責：寺島啓悟）

- 津沢慎吾

  私はシステム班 Unity のメンバーとして，物語再生システムの開発に参加した．私が行った作業は，物語再生システムの根幹となるプログラムの作成である．中間発表までの開発では読み込まれた物語データを順番に参照し，物語を再生していく仕組みを作成した．開発の中で BGM，SE を再生する仕組みも作成したが，今回のプロジェクトでは使用されなかった．最終発表までの開発では中間発表での内容に加えて，選択肢分岐機能の追加とステージ読み込み機能の更新，物語選択機能の追加などを行った．

  私が開発した箇所は物語再生システムの中心ともいえる部分であり，開発の中で他のメンバーと円滑にコミュニケーションを取ることが重要になった．私はクラス図を書いてメンバーと共有するなど，積極的に情報のやり取りを行った．その結果，それぞれが開発したプログラムを結合する作業では大きな問題がほとんど起こらず，順調に開発を行うことができた．

  (＊文責：津沢慎吾)

- 渡邊広基

  私は物語分析・ホラー班に所属し，主にホラージャンルの物語とプロットの遷移確率について分析を行った．具体的には，ホラージャンルの物語である“怪談レストランシリーズ“を分析し，プロット分類や登場人物の属性について分析した．また，それらの内容について因子分析という手法を用いてプロット分類と登場人物の属性の間の関係性を明らかにした．次に，それらの内容からシーン間の遷移確率モデルを作成した．

  グループ活動では，主に物語に使うモデルやモーションなどを扱うモデル班と，遷移確率モデルやデータ形式を扱う Python 班とのコミュニケーションを円滑に行うことができた．また，作業について適切にスケジュールを立て，円滑に進めていくことができた．

  （＊文責：渡邊広基）

- 山田康貴

  私はシステム班に所属し，主にバトルジャンル物語のプロット生成システムの開発に努めた．中間までの開発では，プロット生成システムの大まかなアルゴリズムの考案を行った．そこからホラー班とバトル班が共通で使用するシステムのプロトタイプとして，Excel 上のデータをデータベースに落とし込むシステムの開発を行った．成果発表までの開発としては，戦闘シーンを生成するための戦闘シミュレーションと幕間のシーンを生成するための通常シーン生成システムの大きく分けて 2 つの開発を行った．

  グループ活動では，普段のミーティングでも積極的に発言するように心がけ，自分の班の仕事内容以外でも可能な限り意見を出すように努めた．また，割り当てられた業務内容も期限に余裕を保って終わらせる事を意識して作業に取り掛かった．

  （＊文責：山田康貴）

- 袴田翔

私はバトルの物語分析班のメンバーとしてバトルジャンル物語の自動生成においての基礎となる物語データを作るために作品を分析した．まず，分析作品として挙がった『Fate/stay night』の内容を把握するためにゲームを一通りプレイした．その後，『Fate/stay night』の中にある 3 種類のルートから 1 つ選び，そのルートの戦闘部分と戦闘前後の展開を抜き出し，シーンから地の文，登場人物の攻守，セリフ，読み取れる表情，登場人物が持っている武器，登場人物のポーズと動作を抽出し Excel シートに記述した．シートを作成した後，バトルジャンル系統の物語の大まかな構造を定義するために物語を部分的に区切り，それぞれの展開が持つ意味を考察した．また物語データのパターンを増やすため，『ジョジョの奇妙な冒険』の第 3 部を読み，各敵ごとに物語を分割し，分割した展開を一文ずつで完結に表し Excel シートに記述した．その後，一文ごとに記した展開が物語の構造の中のどの部分に該当するかを定義するためにカテゴリ分けし，展開にカテゴリを当てはめた．そしてバトル班のポスターのデザインの提案と実装を行い，最終発表会のポスターを作成した．

自身の活動内容の反省点として，グループ全体会の時に積極的に意見を発言していなかったことが挙げられる．何より場の空気にのまれてしまい，プロジェクトの課題や将来的な目標を踏まえた上で自身の意見は的を得ているのか自信がなくなってしまい結果として何も発言しないことが多くなってしまった．今後また集団で 1 つの課題を解決していくような機会が多々あると思われるため，まずは積極的に意見交換に参加していくようにしたい．また，自身の作業のペースが遅いことも挙げられる．物語のデータを作る際に完成まで時間がかかってしまい，最終発表会の手前まで伸びてしまった．作業の遅さからプロジェクト全体の成果を示すまでに滞りを作ってしまったので，今後は作業効率を上げてより良いものを作ることができるようにしていきたい．

（＊文責：袴田翔）

- 南部太雅

私はシステム班に所属し，主に Unity の出力画面の UI を担当した．作業内容は，テキストと名前を出力画面に表示するためのスクリプトの作成とオブジェクトの配置を行なった．また，評価実験の責任者としてクリエイティブ AI に貢献しました．

自分の反省点は，システム班としての活動としては，この UI の作成以外役に立てることができなかったことが反省点の一つである．評価実験をする際に自分の連絡不足によって，当初の予定の枚数分のアンケートを集めることができなかったことが最も反省すべき点である．

今後の課題として，連絡を頻繁に送ることが挙げられる．また，システム班の人や他の班の人に声をかけ，手伝えることを見つけることも課題の一つであると考える．

（＊文責：南部太雅）

- 吉田拓海

  私は物語分析・バトル班に所属し，分析対象作品の選定・対象作品の分析・分析結果の考察・物語自動生成で使用する物語データの作成を行った．反省点として，計画の見通しの甘さが挙げられる．当初予定していた計画と最終的に行っていた作業が異なり，班全体の作業速度が遅れてしまった．今後の課題として，具体的な数字を用いた計画を立てることや，複数の計画立て有事の備えを用意しておくことが考えられます．

  （＊文責：吉田拓海）

- 城田晃希

  自分個人としては，視聴覚班のグループリーダーを務めてきたが後期の作業になって前半での議論と学習がいかに不足していたかを感じた．とくに開発も時期的に詰めに入らなければならないという状況で，書いてきたコードと決めた方針を一旦ゼロにしてしまった事が議論と調査の不足を表している．今回のプロジェクト学習を通じていかに機械学習という手法とカメラワークといういわば点数には表せない芸術のようなものを組み合わせることが難しいかを痛感した．最終成果発表会では「カメラワークの傾向をベースに分析するより，もっと根本的なカメラワークの作法をベースに分析と実装を行うべきでは」との指摘があり，確かにまずその方針で分析を進めるべきであると感じた．カメラワークの機械学習による自動生成というジャンルは先行研究を調べてみたところ殆ど見つからず，何を指標に指定以下ほとんど分からずに始めてしまった．機械学習の習熟度や Unity などの制作プラットフォームを扱った経験も殆ど無かったため，それを先行研究の少なさに加えて開発の遅延に拍車がかかってしまったかもしれないと個人的には考えている．結果としてその余裕のなさがまだ熟していない議論を早期に切り上げてしまい，一旦作ってしまったものを白紙に戻さざるを得ないような結果が生まれてしまった．そう言う意味合いでは「根本的なカメラワークの作法をベースに分析と実装を行う」という事を行った方がより高クオリティで見る人間がもっと納得するような出来のシステムを開発できたかもしれない．次から新しくこのようにチームで開発する際はもっと議論や学習を重点的にやっていきたいと考えた．

  （＊文責：城田晃希）

- 佐々木奨之

  私はシステム班 Unity に所属し，Unity システムにおけるキャラクターのセッティングや制御など視覚的な部分の制作を行った．具体的には，Unity 上に読み込まれた物語データのパラメータを入力として取り Unity シーン上でのキャラクターの制御を行う C#スクリプトを制作したほか，モデル班が制作した MMD モデルの変換，キャラクターモーションのセッティング，武器モデルの位置制御に関するスクリプト制作など Unity シーンの視覚的な部分に関する開発，セッティング，編集を行った．

  今回のプロジェクト活動において，GitHub の利用方法や Unity 内の Animator を始めとした様々なコンポーネントの知識を得ることができた．

  （＊文責：佐々木奨之）

- 田中瑞穂

  私は視聴覚班のモデル班に所属し，システム上に必要なモデルのモデリング全般を行った．blender や MMD など，あまり触れることのないツールを使用したため慣れるまでは大変だったが，最終的には人をはじめとする多様なモデルを制作することが出来て良かった．また，物語分析班やシステム班との連携をしっかり取り，他の班の要望も取り入れることが出来たためその点においても良かったと考える．

  (＊文責：田中瑞穂)

- 三浦隆太郎

  自分としては，視聴覚班がカメラとモデルに分かれてから視聴覚班のグループリーダーも務めている城田君への負担が大きかったのではないと感じている．開発の遅れやカメラワークアルゴリズムの構想は自分にももっと彼を手伝える箇所があったのではないかと感じずにはいられないし，後半の作業中に起きたカメラワークを生成するアルゴリズムのゼロからの作り直しは作業中にもっと意見の交流をすれば防げたのではないかと考えている．

  自分の主な仕事は既存の作品のカメラワークの分析や成果発表に向けてポスターを準備することである．自分に与えられた仕事は問題なくこなしているが，与えられておらず抱えている仕事の手伝いをすることができればもっとアルゴリズムの改善ができたと思われる．しかし成果発表で使用するポスターの作成中は，お互いがプロジェクトで行ってきた作業の内容や成果を把握するためにポスター更新をするたび，2 人で話し合いを行いながら作業を行った．この工程によりポスター作成を素早く丁寧に行うことができた．成果発表ではポスター作成を通じてお互いの情報共有が完了していたため質問に対し適切な回答ができたと考える．今後このようなグループでの開発を行う際は最初期の議論を大切に行っていきたいと考える．

  (＊文責：三浦隆太郎)

- 松原千里

  私は視聴覚班のモデリング班に所属し，物語を効果的に表現することを目標に取り組んだ．映像作品からカメラや人物の立ち位置などの表現を学部ことに取り組んだり，人物モデル制作時には下絵の準備を主に行っていた．大きな課題として挙げられていた，モデルに表情を実装する段階では，表情の種類を分析，決定したり，モーフを利用して実装できる状態まで作り上げることが出来た．Unity や Blender などの多くの新しい知識を必要とするソフトに触れていく中で，多くの学習をしなければならない場面があったが，積極的が足りずに満足に使いこなすことが出来なかったところが反省点としてあげられる．目標を立てて効率的に作業することも欠けていたので今後は改善していきたい．

  （＊文責：松原千里）

第 7 章 まとめ

7.1 プロジェクト全体の成果

人工知能はとても高度なデータ処理能力を有している反面，0 から 1 を作り出す創造性の部分にはまだまだ発展の余地がある．人工知能に創造性を与えるために，本プロジェクトではコンピュータ上でインタラクティブに動作する物語自動生成システムの開発を行った．自動生成を行う作品のジャンルとしてはプロジェクト内で議論を行った結果，バトルとホラーに決定した．

システムの開発をするためにプロジェクトメンバー15 人を既存の物語を分析しデータ化・物語生成アルゴリズムの考案を行う物語分析，Python を用いた物語自動生成の実装と Unity を用いた映像化のシステムを実装するシステム班，映像化に使用するモデルの作成とカメラワーク生成アルゴリズムの考案・実装を行う視聴覚班の計 3 つのグループに分かれて作業を行った．

現在，バトル及びホラー共に分析が進み，物語の構造が明らかになっている．また，物語の生成に必要な遷移確率行列と物語生成時に使用するシーンのデータの用意も完了している．これらのデータを元に Python を用いたシステムで物語の生成することが可能となっている．映像面では，映像化に必要とされるモデルとモーションの作成を作成し，シーンを映し出すときに必要なカメラワークの自動生成を行うアルゴリズムの構築が完了している．これらの物を利用し，生成された物語の映像化が Unity 上で可能となっている．生成される物語のパターン数としては，ホラージャンルでは約 24 万通り，バトルジャンルでは約 550 億通りとなっている．

これら 2 つのシステムに対して評価実験を行った結果，ホラージャンルにおいては概ね高い評価を得ることができたが，バトルジャンルにおいては厳しい意見が見られた．これは，物語の表現の仕方で大きく印象が変わってしまい，評価者が物語を自然に思うか思わないかに大きく影響を与えてしまったと考えられる．これにより，物語の表現方法にまだまだ工夫の余地がある等の問題点・改善案が得られた．

（＊文責：山田康貴）

7.2 今後の課題と展望

本プロジェクトの目標は，言語表現と視聴覚表現を用いた物語を自動生成できる創造的なシステムを開発することである．

現段階での成果は，プロジェクト発足当初に決定した目的を達成しているといえる．つまり，ホラーとバトルのジャンルでオリジナルの作品を生成できる人工知能システムの開発に成功した．ここで，中間発表時点では目標を達成するためのすべきこととして 3 つ挙げた．1 つ目は，「物語分析で物語の構造・要素・特徴を明らかにして体系化をすること」．2 つ目は，「自動生成方法とインタラクティブなシステムの考案と実装を行うこと」．3 つ目は，「映像表現の強化をすること」であった．1 つ目に関しては，物語の構造や要素などの体系化に成功している．2 つ目に関しては，自動生成方法として，遷移確率モデルと戦闘シミュレーションを用いての生成となっている．3 つ目の映像表現の強化では，各場面に対応したカメラワークを 8 種類の中から自動で選択している．以上のことから，目標を達成するために必要なことはすべて行われている．

中間発表までの課題としては,物語を自動生成するためのアルゴリズムの考案とシステムの向上，分析対象作品の物語のさらなる分析，適切なカメラワークの指定，オリジナルの 3D モデルなどの制作であった．最終成果発表までの開発での今後の課題は分析対象作品の物語のさらなる分析，適

切なカメラワークの指定の２つが課題となった．分析対象作品の物語のさらなる分析を必要とする理由としては，評価時の意見として，バトルの物語に対して「何がしたいのかがわからない」という意見があった．また，バトルの物語を進めていく上で前後関係が適切ではなかった．次に適切なカメラワークの指定を必要とする理由としては，システム実行時にカメラの視点が予定していた位置より下に配置され，適切な場面の状況を表すことが困難であった．

今後の展望としては，自動生成された物語をさらにクリエイティブにすることが挙げられる．最終成果発表までの開発では，自動生成された物語が破綻していないかという点を目標に開発を行ってきた．そのため，生成された物語では最低限破綻していない物語であった．したがって，生成された物語に人間が作成したような創造性を与えるには，1 つや 2 つの作品に限らず，より多くの作品の分析をすること必要がある．

（＊文責：南部太雅）

7.2.1 物語分析班

7.2.1.1 ホラー班

今回行った因子分析では,主人公の属性や行動が物語の流れや結末に影響を与えることが分かり，仮説は支持されたと言えるが完全ではない．また，ホラー作品らしい構造を明らかにするにはまだデータが不十分で解析に至っていない．さらに因子分析によって得られた結果が偏っていたり有意でなかったりと問題があった．これらの改善すべき点を踏まえ，今後は仮説を再検討し，仮説に基づいた使用属性の検討，また違う手法での分析の検討を行う必要があると考える．そしてより信憑性のあるデータの収集，物語の構造の解明を目指したい．それに加え，今回作成した物語分類やプロット分類は，我々物語分析班が勝手にこうだと判断し決めたものであり，同じ作品を読んで皆がそう分類，判断するかはわからない．そのために一致率を検証する必要があったが，検証せず成果発表を迎えてしまった．したがって今後の課題として一致率の検証を掲げる．

（＊文責：松浦史佳）

7.2.1.2 バトル班

バトル班では中間開発の段階では物語の非戦闘パートで主人公に付与したステータスから多様なパターンに分岐して戦闘パートへ展開していく流れを考案し，戦闘パートにおける物語の分岐方法や展開は考えついていないので課題としていた．しかしながら，物語分析データを用意していくにつれて必要な仕様が多く見つかりステータスに関した物語の展開の分岐は実現が難しく断念した．そして，物語の分岐方法や展開はあらかじめシートごとに結果を固定する方法を用いることで物語の不自然さや矛盾を極力回避することに成功したと言える．

評価実験の結果を踏まえるとバトルの物語はシーンごとに何が行われているのかがわかりにくかったようなので，分析の質を上げることを今後の課題にしたい．そして，他のバトルジャンル系統の作品を数多く分析して新しいバトルジャンル系統の物語の構造を見つけ出したい．

（＊文責：袴田翔）

7.2.2 システム班

7.2.2.1 Python 班

中間以降の課題と展望として，このプロジェクトの肝となるシステムに創造性を与えることを実現するために，物語の自動生成システムのアルゴリズムの考案と実装を今後の課題としていた．また，実装方法に関しては考案の余地が多々あるため考案を他の班と連携を取りながら継続することを今後の課題としていた．中間時点の案としてホラージャンルの物語では物語が進行する途中で何

か選択肢を読者に与えることで物語が遷移する．バトルジャンルの物語では物語を生成する前に主人公のパラメータを設定することができる物語を生成するというものがあった．後期の活動では，物語の分析結果を基にホラージャンルらしい物語とバトルジャンルらしい物語のプロットデータの自動生成のために，それぞれ適した自動生成アルゴリズムを考え，プログラムで実装することで，この課題を達成することが出来た．しかし，案として挙げられていたが，今回の活動期間では実装できなかった内容がいくつかあった．したがって，今後の展望として，物語の分岐数を増やす，特定の物語の進行の場合における展開の変化，登場人物の属性による語尾の変化などが挙げられていた案を現在のプログラムに実装することにより，より創造性のあるシステムにすることが挙げられる．また，システムにより創造性を与えるために，創造性について考えシステムが完全にオリジナルな物語を生成するシステムの開発が挙げられる．

（＊文責：寺島啓悟）

7.2.2.2　Unity 班

中間発表時点での課題は表情に実装を中心とした 3D モデル制御の強化，また武器やエフェクト視覚的表現の強化を挙げていた．今回のシステムの改善によって表情の実装及び武器の実装は達成できたと言える．

エフェクトの実装に関しては，適当な DCC ツールが見つからなかったことや，制作期間の不足，静止画での出力に適切なエフェクトが作りにくいなどの要因によって今回は実装には至らなかった．エフェクトは多くの物語作品のメディア，多くの物語ジャンルで活用され，物語の没入感を向上させることに成功しているため，今後は，静止画での出力に適したエフェクトを物語解析時点から模索していくか，または後述の映像の動画化によって問題を解決し実装していく必要があると考えられる．

3D モデル制御の強化に関しての今後の展望としては，モーションのブレンドによる中間モーションの生成が考えられる．Unity には，Blend Tree と呼ばれる複数のアニメーションを組み合わせることで新たなアニメーションを表現する機能がある．例えば「投げる」というモーションと「しゃがむ」というモーションを Blend Tree によって組み合わせることで，「投げながらしゃがむ」という新たなモーションを生成することができる．この機能によって，より多くの状況にあったモーションを生成することができ，表現力が上がると考えられる．その他にも，Unity のナビゲーションメッシュと呼ばれる機能を利用した経路探索やゲーム AI を利用した戦闘シミュレーションなど Unity の機能を活用したキャラクターの動きの動的な加工，生成により映像表現を豊かにすることが出来るということも考えられるため，検討する価値があると考えられる．

また，今後も映像表現の幅を広げ自動生成された物語を映像としてよりマッチしたものとするために，Unity によるシステムへのいくつかの改善を今後の課題としている．

まず，キャラクターの位置を多様化することが考えられる．今回制作したシステムでは，1 列のマス目にのみキャラクターを配置することが可能であり，例えば「横に並んで立つ」といった状態は出力出来ないようになっている．これは，制作期間の都合と物語分析のコストを考慮しての仕様だが，今後システムを拡張し，キャラクターを様々な位置に配置出来るようにすることでより多様な映像表現が可能となる可能性がある．また，キャラクターの向きに関しても今回のシステムでは 4 方向に限られていたが，今後は他キャラクターの方向を向く，カメラを見る，落ちているものを見るなど，多様な向きを可能にすることで映像の表現力を向上させることが出来るのではないかと考えられる．

次に，映像の動画化が考えられる．今回のシステムでは開発期間や映像制作にかけられる人的コストの面から静止画での映像を出力するシステムを制作したが，今後は静止画ではなく動画として

出力することも検討すべきだと考える．今回制作したシステムにより出力された静止画は，場面によってはどのキャラクターがどのように行動しているのかが分かりにくい場面があった．これは，キャラクターのポーズに連続性が無く，またカメラがどのように動いたのかが分かりにくいからではないかと考えられる．今回のシステムによる出力と同様に静止画による物語形式として漫画があるが，そちらでは背景効果やコマ割りによる視線誘導によって可読性を上げていることが多い．しかし，今回のシステムでは自動生成のため 3D 空間上のどの位置から見ても不自然でない絵作りをしなければならず，適切な画面効果を制作することは難しく，その上 Unity による出力画面は全て同じ縦横比の長方形となってしまうため，漫画と同様の手法で表現力を上げることは難しい．同じく似た形式の物語形式としてノベルゲームもあるが，こちらは 2D の CG によって場面を構成している例が多く，工数の面から自動生成することは現実的ではないと考えられる．

よって，出力の形式を動画にすることによってより物語として説得力のある出力が得られるのではないかと考えられる．具体的な実装としては，数種類のアニメーションするモーションと Blend Tree によってそれらを組み合わせたモーションによってキャラクターを動かし，カメラはカットの主体を距離を維持しながら追尾するよう設定することで動画での出力を実現できるのではないかと考えられる．

次に，DCC ツールの連携を改善することで開発効率を上げることが考えられる．今回のシステムにおけるワークフローは，まずモデル班によって MMD 形式のキャラクターや武器のモデルと，同じく MMD 形式のモーションが制作され，Unity 班に受け渡される．次に MMD モデルとモーションを組み合わせ，Unity で利用可能なモーションを内包した fbx 形式に変換する．最後にそれらのアニメーションとモーフの設定をし，システムに組み込むという流れになっている．しかし，MMD ファイルの変換においては非公式のライブラリに依存しなければならず，デバッグがし辛い，Unity エディター上で扱いにくい, 変換のためデータの追加や修正に時間がかかるという問題点があった．さらに，キャラクターには MMD 上での見た目を再現するために MMD4Mechanim の独自のシェーダーを利用することになるが，これも他アセットと見た目上の統一感が取りにくいという問題点があった．今回はモーションが制作しやすいという点から MMD が採用されたが，今後は Blender 上でモデリングからモーション制作までを完結させることによって開発効率の改善が可能になり，映像の表現力を改善する作業により時間を使えるようになるのではないかと考えられる．

次に，ステージの表現力の強化が考えられる．今回のシステムは前年度と比較してステージの数は増加したが，空の色や天候は固定であった．今後はステージの空色を変えることで時間の経過を表現することや，天候を物語に沿って変更することで物語の表現力を向上させることが出来るのではないかと考えられる．Unity による空の表現は Skybox 呼ばれる機能で設定することが可能であるため，その機能から空色をコントロールすることでステージの表現力の強化が可能であると考えられる．また，Unity には Fog による霧の生成など，ステージの表現力を向上させることが可能な機能が複数用意されているため，それらを利用してステージの構造を多様化し物語の表現力を向上させることも可能ではないかと考えられる．

次に，音楽，効果音の実装が考えられる．開発当初は機械学習による場面に合った BGM の自動選択が検討されていたが，最終的に実装には至らなかった．また，SE に関しても物語データの規格検討段階では実装を予定されていたが開発期間の不足により実装には至らなかった．そのため今回のシステムでは映像の再生中は無音となっている．BGM や SE はアニメや映画など多くの物語メディアで効果的に使われており，物語の表現力を向上させることが出来ているため，今後のシステム開発で実装するべきだと考えられる．特に BGM の自動選択に関しては，可能であればキャラクターの心情に沿った BGM を再生することでキャラクターの表現力を大幅に向上させることが出来ると考えられるため，検討すべきだと考える．SE に関しては物語データ内に既に存在するパラメータ

を利用できるため，キャラクターのアニメーションごとに SE の音楽ファイルを紐付けることで実現出来るのではないかと考えられる．

(＊文責：佐々木奨之)

7.2.3 視聴覚班

7.2.3.1 カメラ班

今後の課題としては，場面の分析においてカメラワークを決めるのに重要な要素が今回の分析である程度求まったため，さらなる条件の選定を行いたい．また，カメラワーク生成において武器等のオブジェクトに対し注目を行うショットの作成ができなかったため，人物以外に注目をしても問題が生じないようにアルゴリズムの改善を行いたい．そしてカメラワークの分析やアルゴリズムの実装に力を入れすぎたため，当初の目的の一つにあった BGM や効果音等の音響を疎かにしてしまった．今後の展望としては場面の状況理解を促すためカメラワークのほかに音響についても分析を行い，可能であればカメラワークとともに場面に適した BGM や効果音を自動で選択できるアルゴリズムの作成を行いたい．

（＊文責：三浦隆太郎）

7.2.3.2 モデル班

モデル班では，モデルの種類を増やし表情の実装を今後行う．武器などの小物を製作することに加え，人物モデルも一から製作していく．その中でも人物モデルに表情を実装することが第一目標である．映像作品から登場人物の表情を分析し，モデルで再現を行う．表情を与えることでより物語への没入感を促すことが可能であると考えるためである．Unity 上で表情の切り替えを行う方法も，システム班とともに検討し，モデリングの方法も併せて検討する．

（＊文責：松原千里）

参考文献

[1] 松尾豊. 人工知能は人間を超えるか. KADOKAWA, 2015.
[2] 松原仁, 中島秀之, 佐藤理史, 赤石美奈, 角薫, 迎山和司, 村井源, 大塚裕子, 平田圭二, 瀬名秀明. コンピュータによるショートショートの自動生成の試みについて. エンターテインメントコンピューティングシンポジウム, 2013, p. 34-35.
[3] 豊澤修平, 工藤はるか, 石田晃大, 遠藤史央里, 川瀬稜人, 菊池亮太, 工藤健太郎, 栗原将風, 櫻井健太郎, 佐藤好高, 玉置秀基, 根本裕基, 原科充快, 久野露羽, 平田郁織, 村井源, 椿本弥生, 角薫, 松原仁. 推理小説プロットを自動生成し映像化する統合的インタラクティブシステムの開発と評価. 研究報告人文科学とコンピュータ, 2018, Vol. 13, p. 1-5.
[4] The What-If Machine Project. 2013. “The What-If Machine.” 2018/11/26 参照. http://ccg.doc.gold.ac.uk/research/whim/resources/poster_UC.pdf.
[5] 松山諒平, 佐藤理史, 松崎拓也. 人狼ログからの小説の自動生成, 言語処理学会 第 23 回年次大会 発表論文集, pp. 32-35, 2017.
[6] 佐久間友子, 小方孝. “プロップの物語内容論を利用したストーリー作成支援システムとその考察”, 日本人工知能学会第 19 回全国大会, 3D3-04, 2005.
[7] 金明哲. R によるデータサイエンス. 森北出版, 2007.

（＊文責：鈴木諒輔）

付録　新規習得技術

R

R とは，オープンソース・フリーソフトウェアの統計解析向けのプログラミング言語及びその実行環境である．ライブラリが非常に豊富であり，統計に適した解析環境であることが特徴である．

（＊文責：渡邊広基）

Python

Python とはプログラミング言語の 1 つである．ライブラリが非常に豊富で，人工知能との親和性が高いのが特徴である．

（＊文責：山田康貴）

Unity

Unity は様々なプラットフォームに対応しているゲームエンジンである．モバイル，コンシューマ，ブラウザ等多くのプラットフォームで 3D，2D のゲームを開発することができる．使用言語は基本的に C#である．3D モデルの導入や描画，物理の適用等を容易に行え，DirectX 等の知識がなくともゲームを制作することができる．

（＊文責：津沢慎吾）

Blender

blender は，オープンソースの 3DCG ソフトウェアである．本プロジェクトは，このソフトを使用してモデルやステージの制作を行っている．ソフトの使用理由としては，フリーソフトであり様々な人々に使用されているため，前例が多いこと．また本プロジェクトでは昨年も blender を使用しており，blender の形式に合わせた素材が残されていたことも挙げられる．

（＊文責：田中瑞穂）

MMD

MMD は， 3D モデルによるコンピュータアニメーションを作成するための 3DCG ソフトウェアである．ソフトの使用理由は，フリーであること，操作が簡単であること，昨年も同様のソフトを使用していたことが挙げられる．

（＊文責：田中瑞穂）