

公立はこだて未来大学 2023 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2023 Systems Information Science Practice
Group Report

プロジェクト名

未来へつなぐ新聞ビッグデータ

Project Name

Newspaper Big Data for the Future

プロジェクト番号/Project No.

19

プロジェクトリーダー/Project Leader

永井清盛 Kiyomori Nagai

グループメンバ/Group Member

木村星凱 Seiga Kimura

佐久間駿 Shun Sakuma

鈴木稜司 Ryoji Suzuki

嵯峨京介 Kyousuke Saga

中村智椰 Tomoya Nakamura

杉山孝太 Kouta Sugiyama

村澤雄斗 Yuto Murasawa

指導教員

寺沢憲吾 美馬のゆり 坂井田瑠衣 角康之

Advisor

Kengo Terasawa Noyuri Mima Rui Sakaida Yasuyuki Sumi

提出日

2024 年 1 月 17 日

Date of Submission

January 17, 2024

概要

本プロジェクトは、北海道新聞社から提供していただいた新聞記事のデータを活用してモノをつくることを目標に活動した。本年度は新聞記事を活用して、プレイヤーが函館市の市長となり、人口減少を食い止めるゲームを作成した。近年、選挙における投票率の低さが問題となっている。特に、若者の投票率は他の世代に比べて低い傾向にある。これは若者の政治への関心の低さによるものだと考えた。同時に、日本では地方の人口減少が問題になっている。これらの問題の解決に向けて、若者の政治参加を促す手段として政治への関心を高められるモノを作ることにした。政治への関心を高める方法として、対象者が自ら政治について考えられるほうが良いと考えた。そこで、政治を擬似的に体験できる、政治シミュレーションゲーム「はこじゃな」を考案した。政治シミュレーションゲームの作成に新聞記事のデータを利用する利点として、過去に現実で起こった政治に関わる出来事をゲーム内に組み込み可能であることが挙げられる。作成した政治シミュレーションゲームでは、新聞記事を要約したデータや、記事から頻出語を抜き出して使用した。これにより、リアリティの高い政治シミュレーションゲームを作ることができ、プレイヤーが現実世界の政治について考える機会になる。政治シミュレーションゲームに、ミニゲームとしてタワーディフェンスゲームを実装した。タワーディフェンスゲームでは、新聞記事の頻出語を取り出したデータを利用した。完成したゲームは、プレイヤーが函館市の市長となり、人口減少を食い止める政治シミュレーションゲームと、次々にやってくる敵から防衛対象を守るタワーディフェンスゲームの2つである。政治シミュレーションゲームは、プレイヤーが街の状態に合わせてコマンドを使用し、パラメータを変動させる。これを繰り返すことで、人口減少を食い止める。タワーディフェンスゲームは、攻めてくる敵から防衛対象を守るゲームである。敵として攻めてくるのは、1年間の新聞記事の頻出語である。これらのゲーム作成を通してプロジェクトメンバの政治への意識が向上した。

キーワード ビッグデータ, 自然言語処理, 政治ゲーム, 若者の投票率の向上

(※文責: 村澤雄斗)

Abstract

This project aims to create something using data from newspaper articles provided by The Hokkaido Shimbun Press. This year, using newspaper articles, we created a game in which players became the mayor of Hakodate to stop the population decline. In recent years, low voter turnout in elections has become an issue. In particular, youth voter turnout tends to be lower than that of other generations. The reason for this is thought to be that young people have little interest in politics. At the same time, the declining population of rural areas in Japan is also a problem. In order to solve these problems, we decided to create something that can increase interest in politics as a means of encouraging young people to participate in politics. As a way to increase interest in politics, we thought it would be good if target people could think about politics on their own. Therefore, we devised a political simulation game “HakoJurna” that allows the players to experience politics. One of the advantages of creating a political simulation game using newspaper article data is that it is possible to incorporate political events that occurred in the past in the real world into the game. In the political simulation game we created this time, we used summary data of newspaper articles and words that frequently appear in the articles. This allows us to create a highly realistic political simulation game and gives players an opportunity to think about politics in the real world. A tower defense game was implemented as a mini-game for the political simulation game. The tower defense game utilized data on words that frequently appear in newspaper articles. The two completed games are a political simulation game in which players take the role of the mayor of Hakodate to stop population decline, and a tower defense game in which players defend defensive targets against onrushing enemies. In the political simulation game, players take on the role of the mayor of Hakodate to stop the population decline. The player executes commands to change parameters according to the city’s situation. By repeating this process, the player can stop the population decline. In tower defense games, players defend defensive targets against invading enemies. The enemies are strings of frequent words taken from a year’s worth of newspaper articles. Through the creation of these games, the project members’ political awareness was heightened.

Keyword Big Data, Natural Language Processing, Political Games, Increasing youth voter turnout

(※文責: Yuto Murasawa)

目次

第 1 章	はじめに	1
1.1	プロジェクトの概要	1
1.2	社会的背景	1
第 2 章	活動内容	4
2.1	前期の活動内容	4
2.1.1	プロジェクト活動における目標設定の過程	4
2.1.2	目標設定のためのグループワークと調査	5
2.1.3	活動目標の決定	5
2.1.4	ゲームデザイン班の活動	5
2.1.5	ゲームプログラミング班の活動	6
2.1.6	自然言語処理・自動生成班の活動	7
2.1.7	中間発表会	8
2.2	後期の活動内容	10
2.2.1	ゲームデザイン班の活動	11
2.2.2	政治シミュレーションゲーム班の活動	12
2.2.3	タワーディフェンスゲーム班の活動	14
2.2.4	自然言語処理・自動生成班の活動	15
2.2.5	成果発表会とその準備	18
第 3 章	成果物	24
3.1	成果物の概要	24
3.2	新聞ビックデータを用いたゲームの設計	24
3.2.1	ゲームデザインとゲームバランスの調整	24
3.2.2	基盤システムとユーザインタフェースの実装	26
3.2.3	新聞データの活用	26
3.2.4	ゲームで使用したライブラリの説明	29
3.2.5	タワーディフェンスゲームの基盤設計	30
3.2.6	新聞データ活用を始めとしたゲームの機能拡張	36
3.2.7	ミニゲームであるタワーディフェンスゲームとの統合	40
3.3	「はこじゃな」の概要	41
第 4 章	技術的挑戦・背景技術	42
4.1	新聞記事からの頻出語の抽出技術	42
4.2	Word2Vec を用いた演算の活用	44
4.3	TF-IDF を活用した新聞記事の重要語判定	46
4.4	自動文章要約技術の利用	49
4.5	データの読み込み・書き出し・整形	49

第 5 章	まとめ・振り返り	51
5.1	まとめ	51
5.2	個人の振り返り	51
5.2.1	永井清盛の振り返り	51
5.2.2	木村星凱の振り返り	52
5.2.3	佐久間駿の振り返り	54
5.2.4	鈴木稜司の振り返り	56
5.2.5	嵯峨京介の振り返り	57
5.2.6	中村智椰の振り返り	59
5.2.7	杉山孝太の振り返り	62
5.2.8	村澤雄斗の振り返り	64
5.3	全体の振り返り	66
付録 A	謝辞	67
	参考文献	68

第1章 はじめに

1.1 プロジェクトの概要

本プロジェクトは北海道新聞函館支社から、ご協力の下、新聞のビッグデータをお借りして実現した。このビッグデータは、北海道新聞の1988年～2022年まで34年分のテキストデータおよび30個の号外記事画像データにより構成されている。これらのデータを活用して、新しい「何か」を生み出すことが本プロジェクトの大きな目的である。今年度は、政治への関心を高めることが出来るシリアスゲームを作成することに決定した。シリアスゲームとは、ゲームの目的とは別に、社会問題の解決を目的としたゲームのことである。

(※文責: 佐久間駿)

1.2 社会的背景

昨今、若者の政治離れが問題となっている。総務省 [1] によると、衆議院議員総選挙における投票率は、昭和時代までは70%付近を保っていたものの、それ以降の投票率は右肩下がりであり、令和3年の投票率は55%となっており、以前と比較するとかなり低い数字となっている(図1.1)。その理由の一つとして、若者の投票率が低いことが挙げられる。内閣府 [2] によると、若者の政治への関心は、“関心がない”、“どちらかと言えば関心がない”の割合が高く、諸外国と比較しても高い割合となっている(図1.2)。さらに、総務省 [3] によると、日本人の若者の投票率は、昭和時代では50%以上だったが、近年の20歳代の投票率は約35%にとどまっており、かなり低い数字となっている(図1.3)。

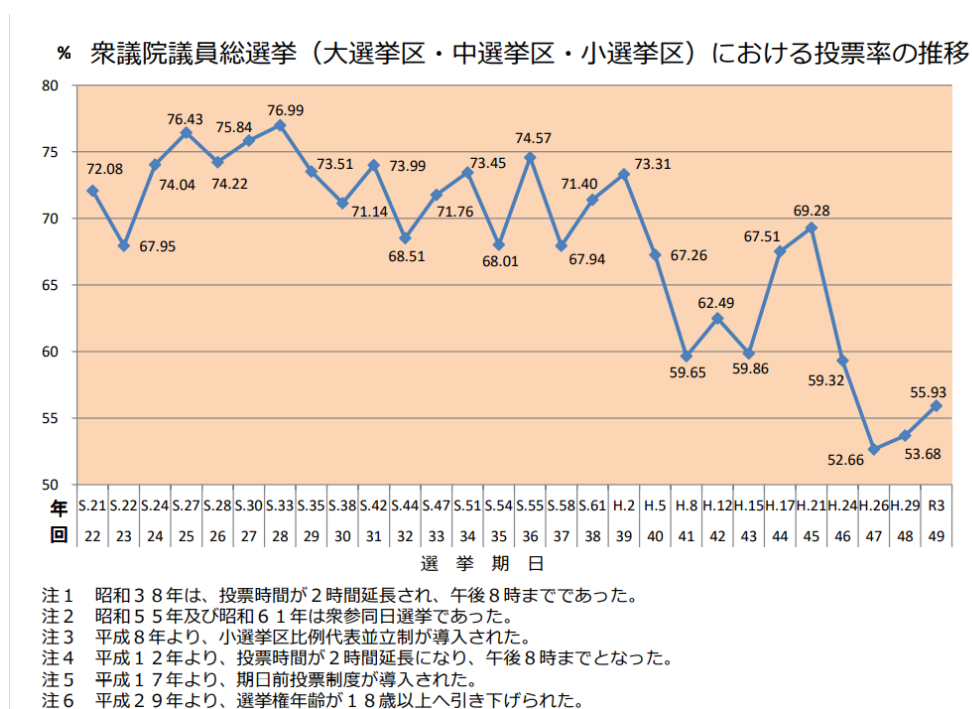


図1.1 衆議院議員総選挙の投票率の推移

図表 13 今の自国の政治にどのくらい関心があるか

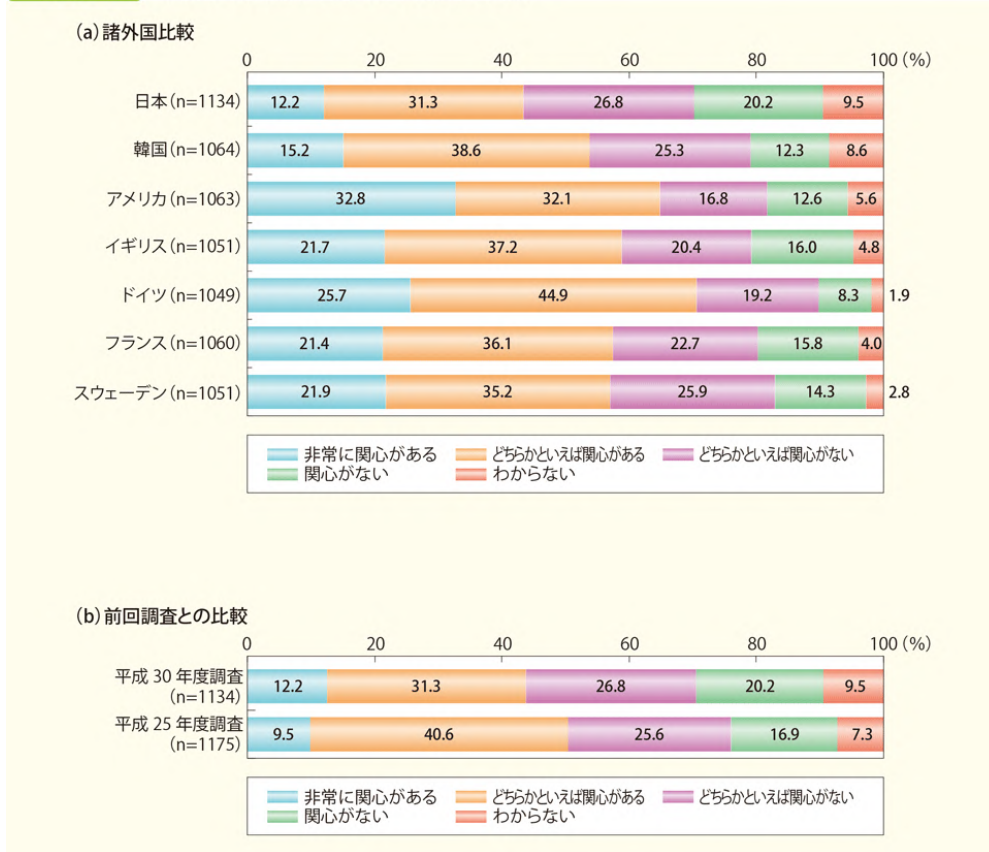


図 1.2 今の自国の政治にどのくらい関心があるか

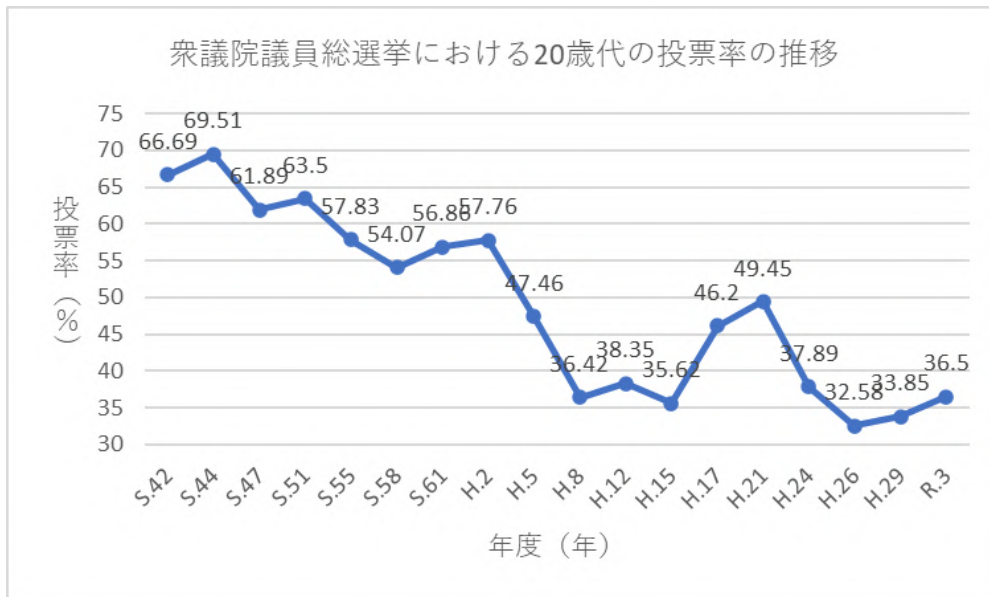


図 1.3 衆議院議員総選挙における 20 代の投票率の推移

本プロジェクトでは、新聞ビッグデータを利用して、若者の投票率が低い現状を打破するためのアプローチをしようと考えた。若者の投票率が低い理由として、「政治に興味がない」「政治の仕組みが分からない」などの理由が挙げられたため、まずはそれを解決するための施策案を出すことにした。プロジェクト内で挙げた施策案として、SNS を用いた広報システムや Web システムの活用などが提案された。しかし、町長の立場で実際に政治をおこなうことで政治の仕組みを理解して

もらえれば、政治に対する興味の獲得に繋がるのではないかと考えた。そこで、新聞ビッグデータには実際の政治のニュースも蓄積されているため、その特徴を活かし、実際の出来事を基にしたイベントが発生する現実味のある政治シミュレーションゲームを作成することに決定した。それに加え、ゲームの一部として新聞ビッグデータを利用したタワーディフェンスゲームを導入することで、プレイヤーに親しみやすい形で政治をおこなう疑似体験ができるゲームにすることにした。本プロジェクトでは、2023 年内にこのゲームをリリースすることを目標として活動した。

(※文責: 佐久間駿)

第2章 活動内容

この章では、プロジェクト学習での私たちの活動について記述する。まず私たちの前期の活動内容を記述した後、後期の活動内容を記述する。また、夏期休暇中の活動に関しては、後期の活動と同じ節に記述する。

(※文責: 永井清盛)

2.1 前期の活動内容

まず始めに、メンバ全員で成果物を決定するためにブレインストーミングをおこなった。ブレインストーミングを通して、若者の政治への関心が低いという社会課題があることがわかった。それを解決するために、若者の政治への関心を高めることができるゲームの作成をすることを目的とした。その後、成果物を作成するためにプロジェクトメンバを「ゲームデザイン班」、「ゲームプログラミング班」、「自然言語処理・自動生成班」の3つに分けた。各班でそれぞれ成果物の作成のため、基礎学習をおこない、中間発表に臨んだ。

(※文責: 永井清盛)

2.1.1 プロジェクト活動における目標設定の過程

まず、私たちは成果物を決定するために1人1人がどのようなモノを作りたいのか、このプロジェクトで何をするかを決めるためのブレインストーミングをおこなった。しかし、そこで挙がった案について、担当教員に「着地点が見えるものが多く、面白みに欠ける」と指摘された。そのため、まだまだ知らないことが多すぎると感じたので、このプロジェクトに関連する先行研究や、先行技術について調査した。この調査では、キーワード抽出や自然言語処理などの技術について調査した。この調査をもとに、再度ブレインストーミングをおこなった。その結果、新聞を用いたゲームを作りたいという意見が多かったため、成果物としてゲームを作ることにした。各自が考えたゲーム例を以下に示す。

1. 国家シミュレーションゲーム
2. 推理ゲーム
3. タワーディフェンスゲーム
4. 裁判ゲーム
5. 記事を活用したクイズゲーム
6. 記事を活用したすごろく
7. 記事からカードを自動生成したカードゲーム

(※文責: 鈴木稜司)

2.1.2 目標設定のためのグループワークと調査

2.1.1 で提案されたゲーム案をよりよいものにするために、グループワークをおこなった。各自が考えたゲーム案の特徴や、尖っている部分を話し合ったところ、新聞記事データを用いて何らかの自動生成をおこない、ゲームと組み合わせるのが面白いのではないかという意見が出た。そこで、ゲームの企画案を考え、どのような開発ツールがあるかを調べるゲーム班と、自動生成について調べる自然言語処理・自動生成班に別れた。ゲーム班は、今あるゲーム案を面白くするにはどうすればよいかや、調べた開発ツールである Unity や Unreal Engine などのメリット、デメリットを理解することができた。自然言語処理・自動生成班は、Python での形態素解析の成功、画像の自動生成などの成果を得ることができた。

(※文責: 鈴木稜司)

2.1.3 活動目標の決定

新聞データとゲームを組み合わせたものを作るという案を、担当教員に提案し話し合った結果、政治シミュレーションゲームとタワーディフェンスゲームを組み合わせたゲームを制作することに決定した。それに伴い、ゲーム班をゲームデザイン班とゲームプログラミング班の2つに分けた。政治シミュレーションゲームは、プレイヤーが町長として政治をおこなえる。また、新聞データを活用してイベントを作成し、ゲームを進行していくことで、現実味のある体験ができることが面白みである。そして、タワーディフェンスゲームは新聞データを活用して政治を脅かす概念を自動生成し、敵として登場させ、その概念への対策を考えながら敵と戦うことで、それぞれの理想の町を守れることが面白みである。最終的には、ゲームを通して政治に興味を持ってもらうことで、若者の政治への関心の向上を目的としている。

(※文責: 鈴木稜司)

2.1.4 ゲームデザイン班の活動

ゲームデザイン班は、制作する政治シミュレーションゲームと、タワーディフェンスゲームの仕様を検討し、設計した。

政治シミュレーションゲームは、プレイヤーがある町の町長となってその町の問題に対処しながら町の政治をするゲームである。すでにリリースされている政治シミュレーションゲームや、類似しているゲームの仕様を参考にしながら、政治シミュレーションゲームの仕様を検討した。具体的には、町の状態を表す人口や資金、支持率、産業のレベルなどのパラメータを決めた。さらに、プレイヤーがゲームに対して入力をおこなうために、コマンドを利用することにした。コマンドはゲーム内では条例として扱うことにした。また、新聞データから生成したイベントをゲーム内で発生させることにした。これらの仕様で絵コンテを作成し、ゲームプログラミング班や自然言語処理・自動生成班とゲームのイメージを共有した(図 2.1)。

タワーディフェンスゲームは、プレイヤーが自陣に向かって攻撃してくる敵に対して、防衛ユニットを配置して自陣を防衛するゲームである。制作するタワーディフェンスゲームは政治シミュレーションゲーム内のミニゲームとした。攻撃してくる敵は、ゲーム内時間の年を象徴するような単語の文字列とした。また、ゲーム内時間と現実時間をリンクさせて敵を発生させることとした。例えば、ゲーム内時間が2019年であれば、「パンデミック」や「新型コロナウイルス」といった文

字列が攻撃してくる。これらの敵は新聞データをもとに作成することにした。

また、中間発表で使用する政治シミュレーションゲームのデモを作成するために、その仕様を検討した。デモはゲームがどのようなものであるのかを体験できるようにするためのものであった。そのため、最低限のパラメータとコマンドのみを実装した。実装したコマンドは、増税と減税、パラメータは資金と支持率などである。

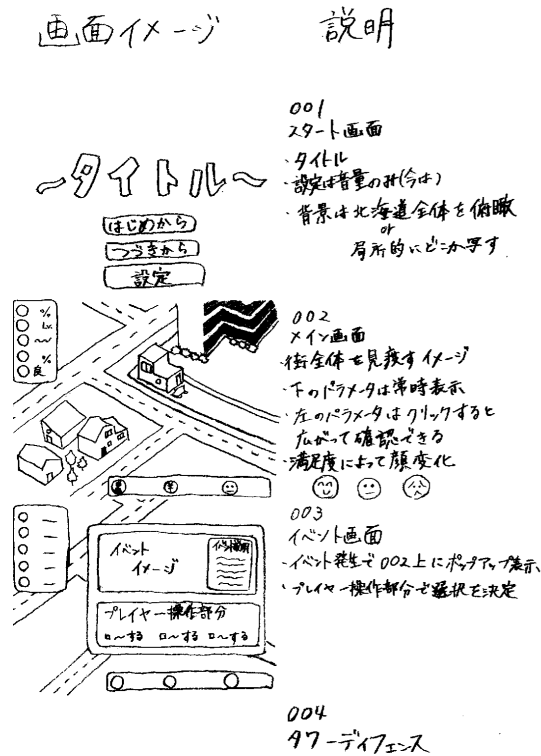


図 2.1 シミュレーションゲーム初期案の絵コンテ

(※文責: 村澤雄斗)

2.1.5 ゲームプログラミング班の活動

ゲームプログラミング班では、まず成果物の要件に適したゲーム開発環境を調査した。Unity や Cocos2d-x など、さまざまなゲームエンジンが候補に挙がったが、学習用の情報が豊富である Unity を採用することにした。その後、Unity の学習を個人でおこなうこととし、ミニゲームの作成やチュートリアルを通じて基本操作を学び、得た知識をグループ内で共有した。個人の学習期間を経て、現時点でのゲームイメージを元に制作に取り組んだ。ゲーム機能の実装を各メンバーで役割分担するため、GitHub を利用してプロダクトを管理しながら共同開発を試みた。しかし、情報の共有やコミュニケーションが不十分であり、共同開発がうまく進まなかったため、GitHub 上での共有は中断し、手動でプロダクトの変更や更新をおこなった。中間発表で実際にゲームプレイを体験してもらうためにデモを作成することになった。デモに向けて、ゲームデザイン班が作成したデモ用の最低限の仕様を参考に、各メンバーにタスクを振り分けた。増税・減税コマンド、演説コマンド、投資コマンド、『次へ』ボタンの有効化、シーン遷移とスタート画面、ゲームオーバー画面の UI 作成をおこない、それらを組み合わせることでプロトタイプを完成させた (図 2.2)。

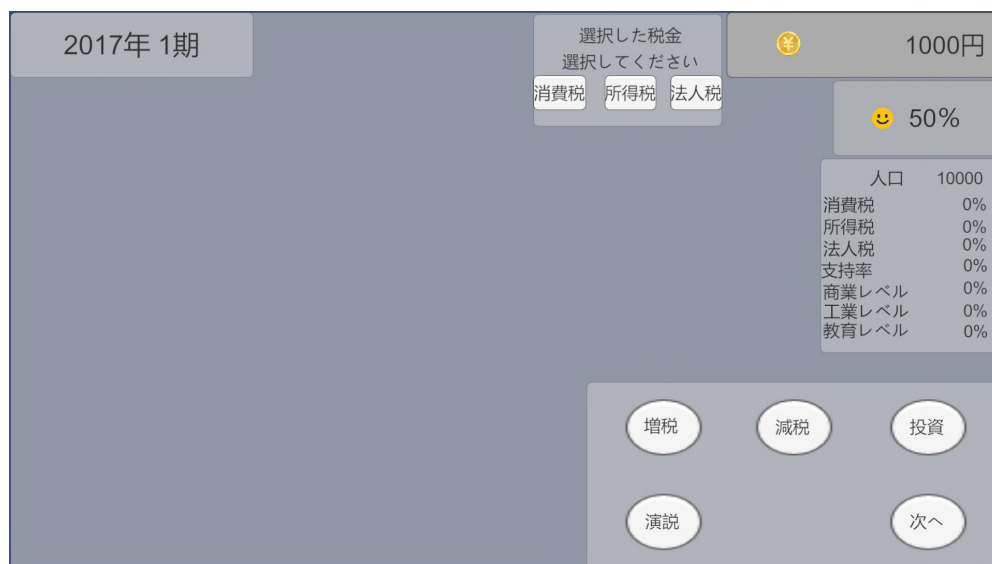


図 2.2 中間発表で用いたデモゲームの画面

(※文責: 木村星凱)

2.1.6 自然言語処理・自動生成班の活動

自然言語処理・自動生成班では、プロジェクト内でおこなった調査報告から、自動生成や自然言語処理を実行するために用いる言語を Python に決定した。開発言語が決まったことで、Python における文字列処理の学習をするために、「言語処理 100 本ノック」*1というサイトの問題にしたがい、Google Colaboratory にて各々の考えたコードを実行し、その結果、正確に結果が得られた。具体的には、簡単なものでは文字列を逆順にして出力するもの、複雑な内容では、単語の出現頻度のヒストグラムを出力するものや単語の出現頻度順位を横軸、その出現頻度を縦軸として、両対数グラフをプロットするものがあった。図 2.3 に単語の出現頻度順位を横軸、その出現頻度を縦軸として、両対数グラフをプロットした例を示す。

「言語処理 100 本ノック」というサイトの 4 章までの 39 問の言語処理に関する問題が終わった時点で、新聞の csv ファイルを丸々形態素解析することに成功した。しかし、csv ファイルを丸々形態素解析するために、環境によっては 5 分かかっても終わらず、このままでは新聞のデータをすべて形態素解析させるのに非常に長い時間がかかってしまうと考えた。それを解決するためにはどうすれば良いのかを話し合った結果、先輩方が作ったコードの一部を用いることに決まった。これにより、1 年分の新聞データを 5 分ほどで形態素解析することができた。その後、ゲームデザイン班と合同で、新聞のデータから何を使い、どのようなイベントをつくるのか考えた。

*1 言語処理 100 本ノック 2020 (Rev 2) .<https://nlp100.github.io/ja/ch01.html>

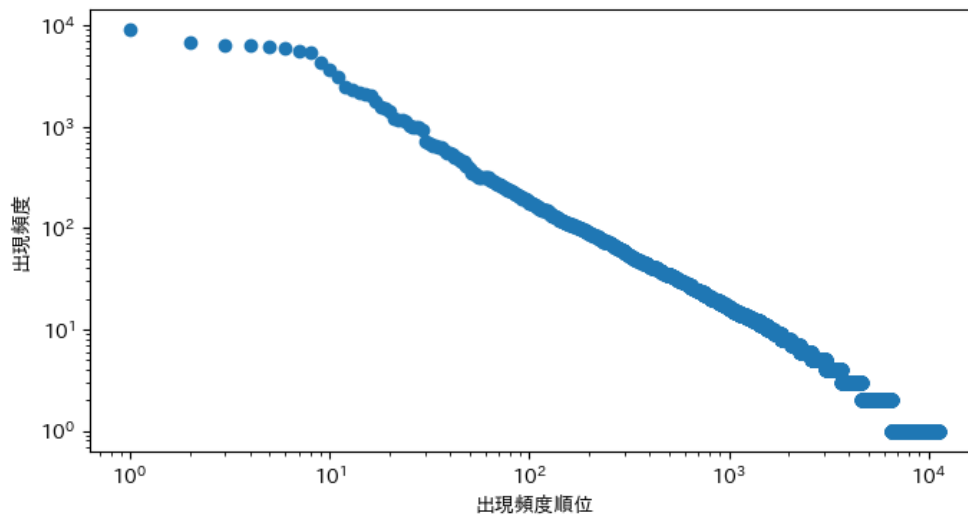


図 2.3 出現頻度順位を横軸，その出現頻度を縦軸として，両対数グラフをプロットした例

(※文責: 中村智椰)

2.1.7 中間発表会

2.1.7.1 ポスター制作の過程

中間発表に向けて、我々はポスターを作成した。ポスターの作成では、各班に前期の活動では何をおこなったのか、どのような問題があり、どのように工夫したのかなどを書いてもらった。そして、それと並行して、ポスターのデザインを考え、新聞に関するプロジェクトであることから、サブポスターを新聞調のポスターにすることに決めた。サブポスターには大量の文字を書くため、メインポスターは読みやすいように文字だけでなく、見た目でも目を引くようにした。そして、各班から提出された活動内容をポスターにまとめた。その後、担当教員に添削していただき、指摘箇所を修正していくことでポスターを完成させた（図 2.4，図 2.5）。

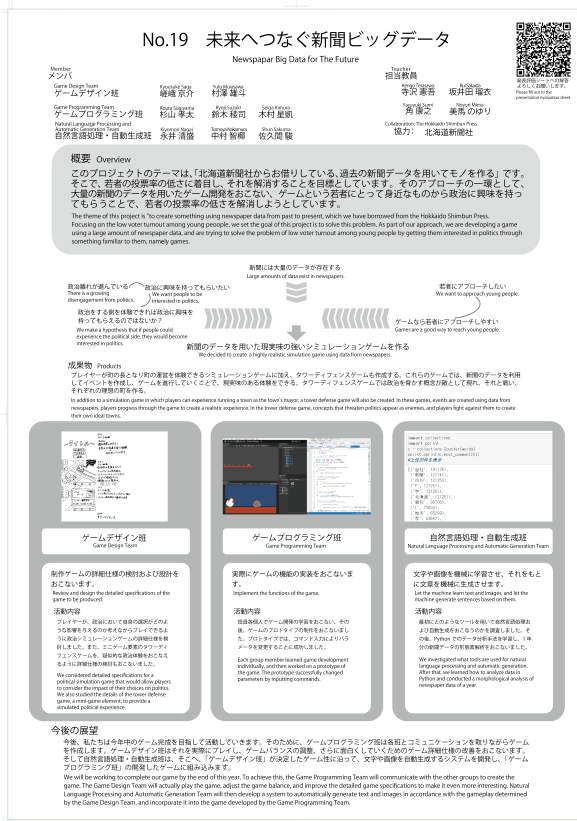


図 2.4 中間発表会の発表で用いたポスター



図 2.5 中間発表会で用いた新聞調のポスター

(※文責: 中村智椰)

2.1.7.2 発表練習の様子

中間発表に向けておこなったことは、原稿を丸暗記せず自分の言葉でしゃべれるようにすることである。「暗記して読んでしまうと、聴衆に内容が伝わらない」と担当教員に指摘されたので、自分の言葉で前期の活動内容を発表できるように練習を重ねた。また、プロジェクト学習の時間だけでなくメンバ全員が集まれる休み時間を使って発表練習をした。

(※文責: 鈴木稜司)

2.1.7.3 中間発表会の様子

ここでは、中間発表会の様子について述べる。中間発表会は各プロジェクトの進捗内容を全体で共有することで、活動に役立てることを目的に開催された。私たちのプロジェクトでは、準備したポスターを用いてポスターセッションのような形式で発表をおこなった。事前の準備のおかげもあり、発表は順調に進行することができた。オーディエンスからの発表評価フィードバックでは、暗記をせず、自分の言葉で発表をしていることに対して好意的な意見が多くあった。同時に、多くのオーディエンスと発表や質疑応答を通じて交流することで、いくつかの課題を見つけることも出来た。ポスターを中心にした発表形式は、どの部分を説明しているのかがわかりにくいという意見を受けて、後期におこなわれた成果発表会ではその反省を活かし、スライドを中心とした発表をするなどの工夫に繋がった。

(※文責: 永井清盛)

2.2 後期の活動内容

夏期休暇中の活動では、開発の効率化のためゲームプログラミング班を「政治シミュレーションゲーム班」、「タワーディフェンスゲーム班」の2つに分け、「ゲームデザイン班」、「政治シミュレーションゲーム班」、「タワーディフェンスゲーム班」、「自然言語処理・自動生成班」の4つの班それぞれで活動した。その後、後期では「ゲームデザイン班」が考えたゲームの詳細決定を政治シミュレーションゲームおよびタワーディフェンスゲームで実現していくために、「ゲームデザイン班」は「政治シミュレーションゲーム班」、「タワーディフェンスゲーム班」のそれぞれに合流した。これにより、後期には「政治シミュレーションゲーム班」、「タワーディフェンスゲーム班」、「自然言語処理・自動生成班」の3班体制で制作物の作成にあたった。

夏期休暇中、「ゲームデザイン班」がゲームの内容、バランスなどを決定した。また、「政治シミュレーションゲーム班」、「タワーディフェンスゲーム班」のそれぞれがゲームの基盤となるものを設計した。後期の活動では、作成したゲームの基盤に自然言語処理・自動生成班が作成したデータを読み込ませていくことでゲームを完成させた。最後に、成果発表会に向けてポスター、スライド、デモ動画、ゲームのデモプレイ機を作成し発表会に臨んだ。

この節では「ゲームデザイン班」の夏期休暇中の活動内容を記述した後、「政治シミュレーションゲーム班」、「タワーディフェンスゲーム班」、「自然言語処理・自動生成班」の3つの夏期休暇中の活動、後期の活動の内容を記述する。その後、プロジェクト全体で取り組んだ成果発表に向けて製作したスライドやポスターなどの活動内容を記述する。

(※文責: 永井清盛)

2.2.1 ゲームデザイン班の活動

夏季休暇中のゲームデザイン班は、ゲームの根幹となる部分の仕様を決定した。

シミュレーションゲームについては、画面構成とUIの基盤となる部分を決定した。決定した画面はメイン画面、イベント画面、コマンド画面、パラメータ画面の4つである。メイン画面は、シミュレーションゲームの中心となる画面である。プレイヤーが町の現在の状況を把握できるように作成した。ゲーム内時間や所持金、人口など基本的かつ重要なパラメータを確認できるようにした。また、次のターンへ進むボタンやコマンド画面を呼び出すボタン、パラメータ画面を表示するボタンなど、ゲームを進めるにあたって重要な機能をメイン画面に配置した。イベント画面は、定期的に発生するゲーム内イベントを表示する画面である。このゲーム内イベントは新聞データを利用して生成されたものである。プレイヤーがコマンドの選択や、パラメータの詳細の確認など、ゲームに対して何かしらの操作をしている状態でも、イベントが起きたことを知らせることができるようポップアップ形式にした。この画面は既存のゲームではなく、WindowsのようなコンピュータのUIを参考にして作成した。このようにすることで、プレイヤーが直感的に操作できるようにした。コマンド画面は、プレイヤーが実行するコマンドを選択する画面である。この画面では、このゲームで使えるコマンドを一覧で表示するようにした。このようにすることで、プレイヤーはコマンドの比較を、画面を切り替えることなくすることができ、すべてのコマンドの中から最適なコマンドを選び出すことが容易になると考えた。図 2.6 はパラメータ画面である。この画面は街の状態を表すパラメータを表示する画面である。ここで表示されるパラメータは犯罪率や出生率、税率など、コマンドを実行することで即座に数値が変わるパラメータである。パラメータ画面もコマンド画面と同じく、すべてのパラメータを一覧で表示するようにした。このようにすることで、街の状態が一目でわかるようになった。また、パラメータの数値には枠がついており、パラメータの数値の状態によって緑色か赤色に変化する。緑色であればデフォルト値よりも良い状態であり、赤色であればデフォルト値よりも悪い状態である。これにより、プレイヤーがデフォルト値を記憶しなくてもよい。また、細かな数値を見なくても街の状態を把握することができる。パラメータ画面とコマンド画面は、横に分割して並べることができるように設計した。このようにすることで、一覧表示された街の状態を見ながら実行するコマンドを吟味することができると考えた。これらの画面はUIデザインツールのFigmaを利用して作成した。次のターンへ進むボタンや資金と人口の表示などのUIについては、既存の政治シミュレーションゲームや、まちづくりシミュレーションゲームなどのシミュレーションゲームを参考にして作成した。また、ゲーム内にプレイヤーが被選挙人となって選挙をおこなうシステムを実装する予定であったが、ゲームの目的や開発時間などを鑑みて、選挙システムは実装しないことに決定した。

タワーディフェンスゲームについては、ゲームの根幹となる部分の仕様を決定した。まず、敵として新聞記事から抜き出した頻出語を使うことにした。また、敵の見た目は頻出語がそのまま文字として出てくるようにした。このようにすることで、新聞ビッグデータを使用していることを明確にすることができる。また、新聞記事の頻出語はニュースのキーワードであると捉える事ができる。つまり、このゲームをプレイすると、世間での関心が高い事柄を知ることができる。敵から防衛対象を守るために使う武器として、弓を使うことにした。フィールドに弓を配置し、次々とやってくる敵を倒していくことにした。また、敵として出てくる単語を防衛するために、それに関連する言葉を使うことを検討した。

夏季休暇明けの後期の活動では、ゲームデザイン班は解散し、代わりに政治シミュレーションゲーム班とタワーディフェンスゲーム班に合流した。



図 2.6 パラメータ画面の構成案

2.2.2 政治シミュレーションゲーム班の活動

2.2.2.1 夏期休暇中の活動内容

前期最後の活動では、後期でスムーズに進めるために、夏期休暇中におこなうことを話し合った。話し合いの結果、政治シミュレーション班では、新聞ビッグデータの具体的な使い方を考えること、ゲームの基盤を作ることを宿題として夏期休暇を迎えた。ゲームの基盤作りとしては、ゲームデザイン班が考えたゲームの仕様をもとに、ゲームの開発をおこなうことにした。夏期休暇中におこなったこととして、ゲームデザイン班によって共有されたドキュメントを見ながら、UIや機能を作成した。UIの作成では、ドキュメント内のシーンごとのイメージ画像を参考に、ゲームで用いるボタンやパラメータの配置を決めた。また、ゲームUIデザインの画像を参考に、素材を集めた。ほかにも、仕様に基づき、ゲーム画面の状態遷移を考え、必要なシーンを加えた。機能の作成では、コマンド、パラメータの設定、イベントや選挙機能を作成した。コマンドについては、一つのパネル上にすべてのコマンドを配置し、このパネルを表示・非表示と切り替える機能を持ったボタンを新たに追加した。このボタンによって、コマンドを一覧表示できるようにした。また、それぞれのコマンドの処理については、様々なパラメータを変化させるように、仮に設定した。次に、パラメータについては、コマンド同様にパネル上にパラメータをまとめて、ボタンによって一覧表示できるように変更した。また、パラメータの値が良い傾向か悪い傾向かわかるように、パラメータを枠で囲った。パラメータごとに基準を設け、基準を超えると枠の色が変化するように実装した。標準の枠の色を緑、基準値超えの枠の色を赤とした。イベント機能では、ファイルの読み取りを行いファイル内に書かれた日付に応じて、パネルオブジェクトが表示されるような機能を実装した。選挙機能では、16ターン経過ごとに選挙シーンに移動するような処理をターン進行ボタンに追加した。選挙シーンではドロップダウンオブジェクトを配置し、政策分野や具体的な政策内容を選択できるようにした。また選択したものによって遷移するシーンが変化するようにした。遷移先では、街頭演説をイメージした市民によるいくつかの質問がされて、解答を3択から選び答えると支持率が変化するようなゲームシステムを作った。この段階では、理想のゲーム動作を実現するために設定するであろうパラメータやコマンドについて、これまでに検討されたものがあつたが、まだ具体的な実装は行わなかった。今回の作業ではおおまかなフレームワークの構築に焦点を当てた。プレイヤーの基本的な操作、ゲーム内の基本的な要素の配置などをおこない、ゲームの基盤を

築くことを目指して活動した。

図 2.7、図 2.8、図 2.9 はそれぞれ、パラメータ、コマンド一覧表示画面、政策決定画面、街頭演説画面である。

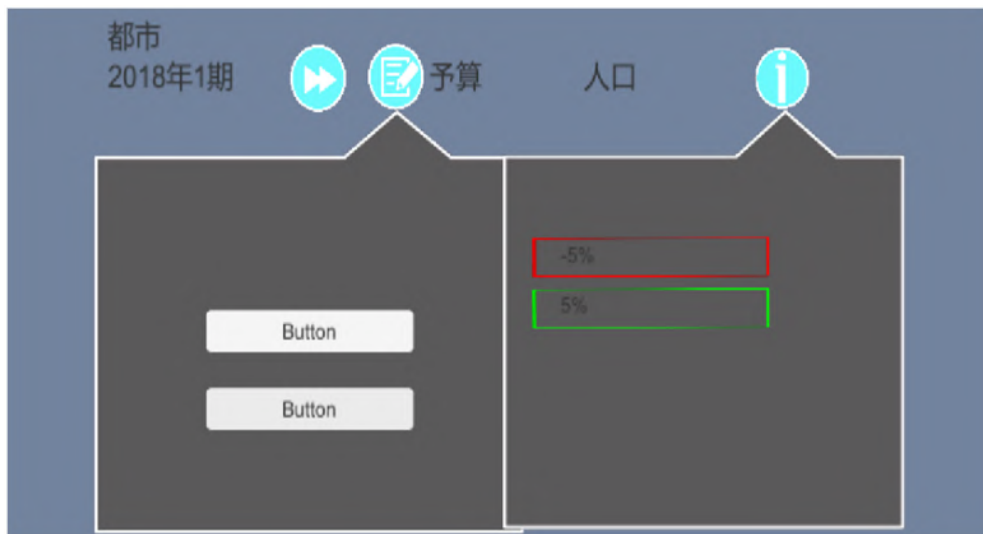


図 2.7 パラメータ、コマンド一覧表示画面

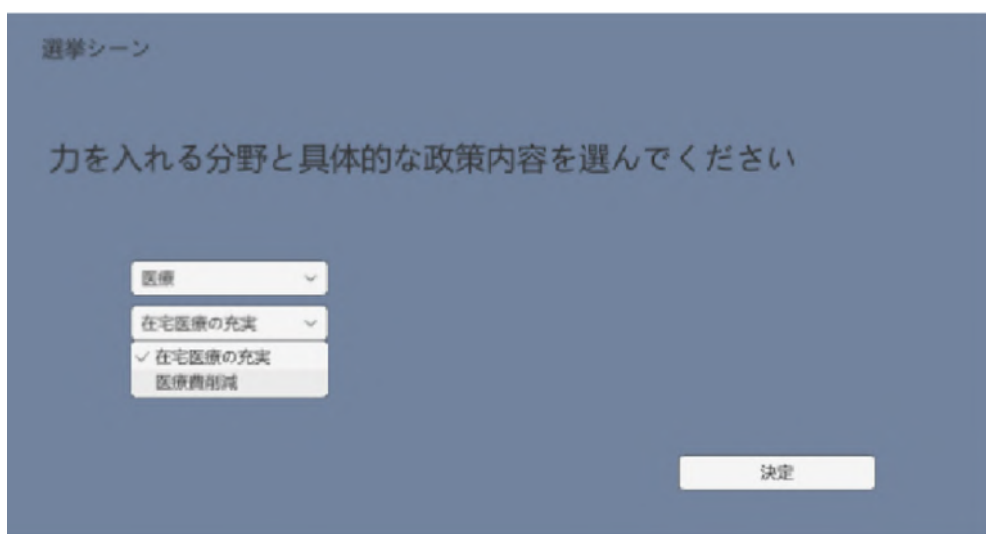


図 2.8 政策決定画面

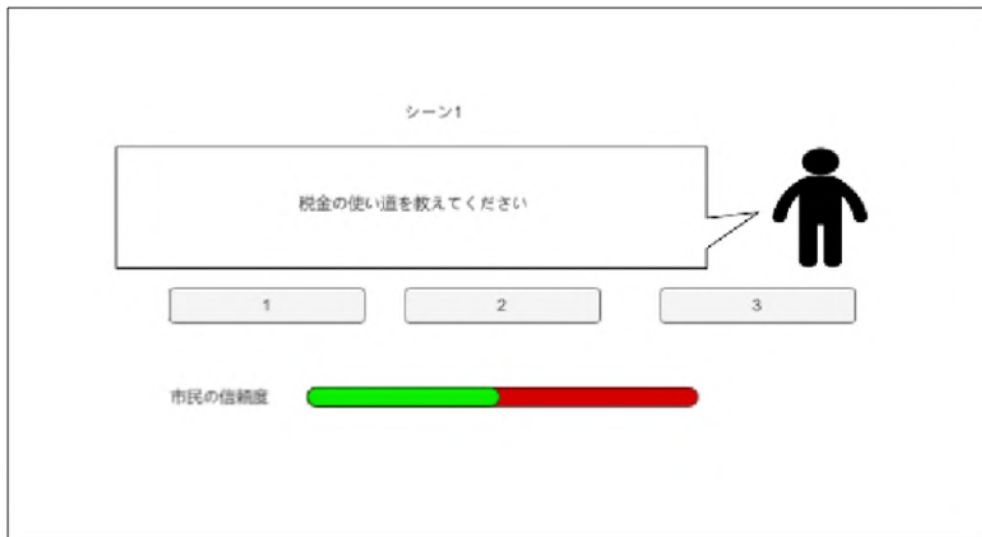


図 2.9 街頭演説画面

(※文責: 木村星凱)

2.2.2.2 後期の活動内容

政治シミュレーションゲーム班では、主にゲームデザインを担当するゲームデザイナー 1 人と、Unity でのゲーム開発をするプログラマー 2 人に役割分担をした。前期で問題となっていた共同開発の手法は、Google Drive を活用して開発データを共有することで、安定した共同開発の体制を整えたことで解決した。プログラマーは開発を進めながら途中経過をデモや動画として共有し、教員やプロジェクトメンバからのフィードバックを受けた。フィードバックをもとにゲームデザイナーを中心としてゲームの改善点や新たなアイデアを考え、ゲームを修正した。この共有と修正のサイクルを繰り返すことによってゲームのクオリティを高めることができた。具体的には、プレイヤーが社会問題について考えることを促すシリアスゲームとしての側面を強めたり、社会の実態が伝わるように出来るだけリアルな社会を表現したりするなどの方向でクオリティを高めていった。また、後期中盤でそれまで曖昧だったゲーム内の目的や、ゲームのシリアスゲームとしての目的を明確にした。以降はその目的に向かって開発を続け、ゲームを完成させることができた。

(※文責: 杉山孝太)

2.2.3 タワーディフェンスゲーム班の活動

2.2.3.1 夏期休暇中の活動内容

タワーディフェンスゲーム班は、Unity 入門の森を参考にし、ゲームエンジンである Unity を用いてタワーディフェンスゲームを作った。Unity 入門の森とは、Unity 入門のための情報・ゲームの作り方・Unity と C# の使い方・Unity オリジナルゲーム開発チュートリアル講座を多数公開しているサイトのことである。メンバに、Unity 入門の森を利用した人がおり、そのメンバによると Unity 入門の森での学習が初学者にとってわかりやすいということでこのサイトを選んだ。後期の活動に向けて、タワーディフェンスゲームがどのようなゲームなのか、どのように作られているのかを理解するためにゲームを 1 つ作ってみるということが目的であった。

(※文責: 鈴木稜司)

2.2.3.2 後期の活動内容

ここでは、タワーディフェンスゲーム班の後期の活動について述べていく。タワーディフェンスゲーム班は、夏期休暇中に作成したゲーム基盤をもとに、引き続き詳細仕様の検討をしつつ、タワーディフェンスゲームの完成を目指した。制作には、ゲームエンジン Unity を用い、政治シミュレーションゲーム班と相互にコミュニケーションを取りながらゲーム制作を進めた。詳細仕様については、政治シミュレーションゲーム内でのタワーディフェンスゲームの立ち位置や発生頻度、防衛アイテムの試行錯誤など前期に大まかに決まっていた部分を何度も試行錯誤し、自分たちの技術で可能な範囲で実装した。また、ゲーム内で使用するデータに関しては、自然言語処理・自動生成班から提供して貰い、ゲーム内に活用した。

(※文責: 嵯峨京介)

2.2.4 自然言語処理・自動生成班の活動

2.2.4.1 夏期休暇中の活動内容

自然言語処理・自動生成班は夏期休暇中に、前年度以前、同プロジェクトに所属していた過去のメンバが、言語処理に使用していたプログラムのソースコードの解析をおこなった。25種類以上あるソースコードを班のメンバ内で分担し、ソースコードを解析したり、実際にプログラムを動かしたりなどをおこない、プログラムの内容を調査した。そして、ゲームで使用するデータを作成するために何ができるのかを各々がまとめ、報告した。以下にその調査結果を述べる。

関連語

このプログラムは、単語の関連語を json 形式のファイルとして出力する。例えば、入力を「北海道」とした時、「北海道」に対して類似度の高い単語(岩手県、富山県など)を5つ出力する。しかし、このプログラムは、形態素解析の際に用いる辞書のインストールに非常に時間が掛かり、実際に動かすことは難しいことが調査の段階で分かった。

頻出単語

このプログラムは、単語の出現回数をカウントし、その単語を出現回数が多い順に並べ替え、json ファイルとして書き出すプログラムだった。

TF-IDF

このプログラムは、文章中に含まれる単語の重要度を評価し、TF-IDF 値として数値化するプログラムだった。TF-IDF 値が 0.2 以上かつ文字の長さが 3 以上の単語を抜き出し、小数第三位を四捨五入したものを出力する流れであったが、元のプログラムでは TF-IDF 値を出力する段階でエラーが発生したため、出力することが出来なかった。試行錯誤の結果、エラーを出さずに TF-IDF 値の出力をすることが出来たものの、「名詞」「普通名詞」などの形態素解析際に出た品詞名の TF-IDF 値も出力されてしまったので、改善の余地があることが分かった。

YearsNounsAnalys.ipynb etc.

このプログラムは、各単語がどの程度の頻度でいつ出現しているのかを分析し、その度合いを倍率として表現し、グラフによる可視化をおこなっていた。一部動かないコードがあったものの、プロットの部分であったため利用することは容易にできそうであることが分かった。また、YearsNounsAnalys.ipynb は年ごと、MounthsNounsAnalys.ipynb は月ごと、EveryYeaeNouns.ipynb は年ごとと月ごとのどちらにも共通しているプログラムであるこ

とが分かった。

hintnoshusei.ipynb

このプログラムは、漢字クイズに準ずるもののヒントのデータを整形しているプログラムだった。ある単語がいつ頻出しているのかをまとめ、クイズの答えのために読み仮名とセットにして保存し、その記事を同じ CSV ファイルに出力するプログラムであったため、政治シミュレーションゲームやタワーディフェンスゲームを作成するにあたっては活用することが難しいことが分かった。

DataAnalysis1、DataAnalysis2

DataAnalysis1 では、指定した範囲のデータを読み込んで形態素解析を行い、その後、特に名詞に注目して頻出単語を出力していた。DataAnalysis2 は、DataAnalysis1 で作成したテキストファイルを読み込み、DataAnalysis1 と同じように出力することが出来るかを試しているプログラムだった。新聞のデータを扱いやすい形に整形するためのプログラムであったため、整形されたデータを準備する際に利用することが出来そうであることが分かった。

CSV.ipynb

ドライブ上にある CSV ファイルの文字コードのエンコードを変えるためのプログラムであった。

Word2Vec.ipynb

このプログラムは、単語をベクトルとして表現し、数値に変換する Word2Vec を用いて、単語の数値化や類似度の高い単語の表示、2つの単語の類似度の算出などを行っているプログラムであった。単語を多次元ベクトル化させることで演算をおこなうことができ、簡単に扱うこともできるため、活用することが容易であることが分かった。

Kazuma DataAnalysis1

このプログラムでは、データを前処理した後、見出しと本文の抽出を行い、json ファイルへの書き出しをおこなっていた。また、1年ごとの記事をリストとして格納し、それをテキストファイルとしてドライブ上に保存をおこなっているプログラムだと分かった。

地域性

このプログラムでは、新聞のデータの前処理をおこない、指定年の CSV ファイルを読み込み、データフレームを作成した後、年代ごとに区分けをおこなっていた。その後、すべての記事情報を年月のみに変更し、月ごとのデータフレームを作成していた。また、形態素解析をおこなっており、リストとして1年ごとの記事を格納したものをテキストファイルとしてドライブに保存し、このテキストファイルを MeCab を用いて取り込み、リスト化と品詞のパーツ分けをおこなっていることが分かった。

hoge

このプログラムでは、指定したテキストデータの形態素解析をおこない、文章の要約をおこなっていた。その際、カウントディレクトリの移動の際にコードを書き換えなければ移動することが出来なかったり、引数を名前付き引数に直さなければエラーが出ることが分かった。

DataAnalysisMonthDict.ipynb

このプログラムは、テキストデータに含まれる単語の表層系、基本形、品詞、フリガナを取り込み、その後各月ごとの名詞のみを抽出し、各月で名詞の出現回数を数え、出現回数の多い順番にソートして出力しているプログラムだと分かった。地域性にて作成したデータでプログラムを実行したところ、途中に見えないデータが取り込まれ、それによって起因するエ

ラーが表示されることが分かった。

tanaka_dict DataAnalysisMonthDict.ipynb

このプログラムは、新聞データのすべてのテキストファイルから名詞を抽出し、その名詞の漢字と振り仮名(カタカナとひらがな)を一行ずつテキストファイルに書き込みをしているプログラムだった。名詞を抽出した後、その名詞をカタカナに変換し、そこからまたひらがなに変換するという流れでおこなわれていた。

tanaka_dict midasi.ipynb

このプログラムは、新聞のビッグデータに対して前処理をおこなうプログラムであった。tanak.dict の DataAnalysisMonthDict.ipynb などで作成したテキストファイルの単語に対して、文字数ごとの単語数をカウントしていた。

テキストを全抽出.ipynb

このプログラムでは、記事を改行ごとに分割してリストに追加し、その後、1988 年から 2022 年までの新聞データの最初の一文のみを追加していた。

DataAnalysis1.ipynb

このプログラムは、新聞記事の本文を形態素解析し、単語の形態素の情報が一行に書かれているので、それを改行ごとに分けた後、入力したキーワードと同じ単語を CSV ファイルに書き込むプログラムだった。

hint midashi.ipynb

このプログラムでは、データの前処理をおこなっていた。年代を設定し、その年代の記事をリストに追加して、記事を昇順に並べ替えていた。

hint midashi_全文.ipynb

このプログラムでは、CSV ファイルを前処理したものをテキストファイルとしてドライブに保存していた。半年分のテキストファイルを出力するために約 5 分かかり、34 年分のファイルを生成するために 5 時間以上費やすため時間はかかるが、正常に作動することが分かった。

DataAnalysisMonthsNEologd.ipynb

このプログラムでは、CSV ファイルの前処理をおこない、形態素解析をした後、名詞のみを抽出しテキストファイルとして保存していた。辞書を使用するところでエラーが発生したため形態素解析が出来なかったり、辞書を使わずに形態素解析をしたとしても名詞を抽出する段階でパソコンがクラッシュするなど、改善すべき所が多いことが分かった。

(※文責: 佐久間駿)

2.2.4.2 後期の活動内容

後期の自然言語処理・自動生成班は夏期休暇中の活動内容を踏まえ、政治シミュレーションゲーム班とタワーディフェンスゲーム班から指定されたデータを形態素解析を用いて新聞ビッグデータから作成していった。その過程で技術的挑戦や背景技術として「頻出語の抽出」、「Word2Vec」、「TF-IDF」、「文章要約」、「読み込み・書き出し・整形」などに取り組んだ。これら技術的挑戦や背景技術は 4 章で詳しく述べる。政治シミュレーションゲーム班とタワーディフェンスゲーム班から指定されたデータを作成するにあたり、過去のプロジェクト学習にて作成されたソースコードを参考に改良を加えることでソースコードを作成していった。また、単純に指定されたデータを作成するだけでなく、どのようにデータを使いたいのかなどのヒアリングをおこなうことで、政治シミュ

レーションゲーム班やタワーディフェンスゲーム班が使用しやすいように整形などもおこなった。それに加えて、些細な解釈の違いなどが起きないように1度政治シミュレーションゲーム班やタワーディフェンスゲーム班にデータを渡した後に、改善点などのヒアリングもおこなった。これにより、政治シミュレーションゲーム班やタワーディフェンスゲーム班にとって使いやすいデータを制作することができた。

(※文責: 中村智椰)

2.2.5 成果発表会とその準備

ここでは、成果発表会に向けて我々が何をおこなったについて説明していく。成果発表会に向けて、我々はスライド、ポスター、タワーディフェンスのデモ、政治シミュレーションゲームのデモ動画を作成した。スライドは、中間発表を踏まえてスライドがあった方がより効果的な発表をおこなえと考へ作成した。ポスターは、見るだけでプロジェクトの概要や学習内容を理解できるよう、詳細な情報を充実させた。タワーディフェンスゲームのデモは、オーディエンスにゲームを触ってもらおうという目的で制作した。タワーディフェンスゲーム自体、ウェブがあり短い時間でも遊べる、理解してもらえとすることで制作した。政治シミュレーションゲームのデモ動画は、成果発表会の時間中ずっと流し、近くを通る人の興味を引くことを目的として制作した。また、政治シミュレーションゲーム自体が長いゲームなのでデモ動画を流すことで簡潔にわかりやすくオーディエンスに伝えるために制作した。

(※文責: 鈴木稜司)

2.2.5.1 スライド制作の過程

初めに前期の発表を振り返り、ポスターのほかにスライドも作成することに決めた。最終発表が近づき、発表スライド、ポスターの制作を始めた。これからの活動方針を担当教員と共有した際に、ポスターとスライドで用いる言葉を統一するため、ポスターの構成を決めるためにはスライドは先に作るべきだという助言を担当教員からもらい、スライドの制作を優先しておこなうことにした。まず初めに、メンバ全員で、スライドのアウトラインをつくった。挙がった意見をまとめると、アウトラインは、成り立ち、プロダクト紹介、背景、課題設定、使用データ、ゲーム内容(シミュレーション・タワーディフェンス)、新聞データの扱い、期待される効果、今後の展望となった。その後、メンバで分担し、文章、使用する画像を考え、スライドの内容を深めた。スライドの中身も定まったところで、担当教員から助言をもらうことにした。助言により、大きく2つを変更した。1つ目は、今後の展望を提言に変更した。今後の展望の中身が思うように定まらなかったため、代替案として出た提言に変更することにした。提言の内容についても案をもらい、提言スライドの中身を書き換えた。2つ目は、背景に用いるデータを変更した。年代別投票率の推移データから若者の投票率のデータのみを抽出し、20歳代の投票率の推移のグラフに変更した。また、政治に対する関心度の国際比較グラフも用いて、より日本の現状が伝わるようにした。ほかにもスライドに最適な文字サイズ、文章量の具体的な基準を教えてもらい、文章にそのルールを反映させた。大まかにスライドが完成したところで、次にスライドのデザインの検討をおこなった。担当教員からデザイン四原則を守るべきという助言から、図の配置や用いる記号をそろえた。またスライドの認知負荷が高くなるような色・レイアウトにし、バランスをとった。そのほかにも、スライド内にガイド線を挿入し、文章間の余白を合わせたり、スライド間のタイトルと中身の位置を揃えたり

した。すべてのスライドを編集し、一貫性のあるデザインになったところで、他のプロジェクトメンバーや担当教員からフィードバックをもらった。フィードバックをもとに、要点を的確に伝えるための強調や前後のスライドを覚えてもらう工夫を取り入れた。発表原稿をもとに、効果的なプレゼンができるか各スライドにおける話の進行を確認した。これらの過程を踏み、最終的なスライドに仕上げた。最終的なスライドでは、新聞ビッグデータプロジェクトの概要、今年度のプロジェクトの概要、背景、成果物の紹介、技術的挑戦、背景技術、使用データ、まとめ、期待される効果、活動を通して、提言というアウトラインになった。技術的挑戦スライドでは、成果物の特徴と使用したデータの種類を並べて明示している。このスライドを設けることで、成果物の特徴や優れた点が強調されたと考える。また、データがどのように活用されたのかがわかりやすく示すことができた。背景技術スライドでは、処理前と処理後のデータの中身を並べて明示している。また、スライド内で専門用語や技術的な言葉を避け、シンプルで簡潔な表現にしている。このスライドを設けることで、データから導き出された結果やデータ処理による変化が効果的に伝えることができたと考える。デザイン面では、重要な言葉には水色の四角で囲み、強調させた。また、見ている人の見やすさを考慮して水色とした。グラフを用いたスライドでは、重要なデータ点を強調するために丸い枠で囲んだ。これらの工夫をすることで、プレゼンテーションの効果的な伝達に寄与したと考える。

(※文責: 木村星凱)

2.2.5.2 ポスター制作の過程

最終発表に向けて、我々は政治シミュレーションゲームとタワーディフェンスゲームの開発、スライド作成と同時進行でポスターの制作をおこなった。ポスターの制作では、まずグループメンバー全員で何を書くか検討した。そこで、決定した内容は題名、プロダクトの概要、背景、目的、使用データ、政治シミュレーションゲーム班、タワーディフェンスゲーム班、自然言語処理・自動生成班、効果、展望である。また、配置は図 2.10 のようになった。その後、Adobe 社が提供している Illustrator の使い方を、グループ内で使ったことがある人に教わった。主に、テキストの使い方、レイヤーの使い方などである。次に、図 2.10 とスライドを参考に Illustrator で制作した。しかし、シミュレーションゲームとタワーディフェンスゲームと自然言語処理・自動生成の 3 つのプロダクトがあるように見えるという指摘を坂井田瑠衣准教授から受け、自然言語処理・自動生成班を政治シミュレーションゲーム班とタワーディフェンスゲーム班の下に書き背景技術に変更した。また、展望はこの活動を通じて新たに見えた課題などを示す部分である、という角康之教授のアドバイスを受け活動を通してどのようにメンバーが変化したかに変更した。ポスター制作で意識したことは、ポスターを見ただけでプロジェクト全体が理解できるように情報量を多くすること、デザイン 4 原則、英訳は何回も繰り返すことである。最初、スライドを参考にしていたため情報量が少なくなっていた。ポスターは、ポスターだけで完結するように書かなければならないとメンバー同士で話し合い、プロジェクト全体が理解できるように、ポスターに詳細な情報を盛り込んだ。背景・目的では、図を用いて日本では現状どのようなことが起きているのか、何のために成果物を作ったのかを書いた。成果物は、各班にどのようなモノを作ったのか、ゲームをプレイした人がどのような効果を得られるであろうかを詳しく書いてもらった。デザイン 4 原則は、美馬のゆり教授からアドバイスをもらった。近接は関係する情報どうしを近づける、整列は要素に一体性を持たせて組織化する、強弱は情報に優先度をつけ違いを明確にする、反復は特徴的な要素を繰り返すことで一貫性を持たせる。ポスターを見る人にわかりやすく情報を伝えるために、ポスター制作をする上で以上のことを意識した。英訳は、日本語を英語に翻訳し、翻訳した英語の文章を再度日本語に

Newspaper Big Data for the Future

翻訳し、意味がしっかりわかるようになっているか確認した。また、必要に応じて DeepL 翻訳を使った後に、Google 翻訳も使い、意味がわかるようになっているか確認した。以上のことを意識し、制作したポスターが図 2.11 である。

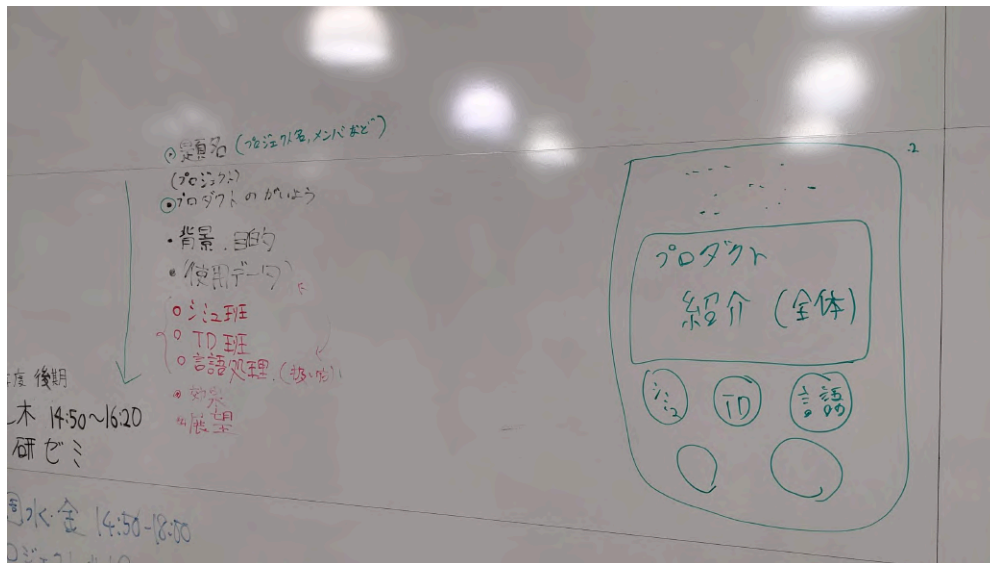


図 2.10 ポスターレイアウトの案

No.19 未来へつなぐ新聞ビッグデータ

Newspaper Big Data for The Future



Member: 木村 聖樹, 藤嶋 真介, 佐久間 龍, 石山 孝太, 鈴木 雅司, 伊藤 智輝, 永井 清盛, 村澤 達斗
 担当教員: 寺沢 善吾, 美馬 のぶり, 角藤 之, 坂井田 瑞衣
 協力: 北海道新聞社

概要 Abstract

本プロジェクトでは「新聞データを用いてモノをつくる」というテーマのもとにシリアスゲームを作成しました。現在の日本では投票率や政治への関心の低下が懸念されています。投票率が特に低いのは若年層であり、私たちはこの現状を打破するために若者になじみ深い、ゲームという形で政治への関心を高める方法を考えました。このゲームは北海道新聞のデータを利用しています。これによりリアリティのある政治体験が可能となっています。このゲームの目的は、プレイヤーの政治や社会問題への関心を高めるきっかけとなることです。

In this project, we created a serious game based on the theme of "Using Newspaper Data to Create Things". In Japan today, there is concern about declining voter turnout and interest in politics. In order to break through this situation, we came up with a way to raise interest in politics in the form of a game that is familiar to young people. This game uses data from The Hokkaido Shimbun Press. This allows players to experience realistic politics. The goal of the game is to trigger players' interest in politics and social issues.

背景・目的 Background and Goal

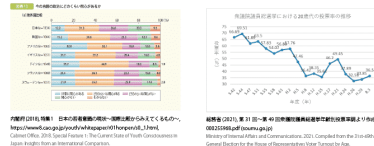
日本の若者は、諸外国と比べて自国の政治に関心が無い傾向にある。20歳代の投票率はおよそ35%程度と、低い水準となっている。

新聞のデータを活用した、政治への関心を高めることができるシリアスゲームの作成
 Our goal is to create a serious game that can players' interest in politics using data from newspaper.

シリアスゲーム: ゲームの目的とは別に、社会問題の解決を目的としたゲームのこと
 Serious game: game that aim to solve social problems apart from the purpose of the game.

日本の若者の政治への関心

Low Interest in Politics among Japanese Youth



成果物 Deliverable

Unityを用いて政治への関心を高めることができるシリアスゲームを作成しました。このゲームは、主となる政治シミュレーションゲームと、ミニゲームのタワーディフェンスゲームで構成されています。これら2つのゲームは、新聞データを活用して作成しています。

We used Unity to create a serious game to raise interest in politics. The game consists of a main political simulation game and a mini-game tower defense game. These two games were created using newspaper data.

政治シミュレーションゲーム Political Simulation Game

街を発展させ人口減少を食い止めるゲームです。The game is to develop the city and stop population decline.
 政治シミュレーションゲームは、国政の市長となって街の人口減少を食い止めるゲームです。ゲーム内の時間に合わせて、その年を象徴するような追加コマンドが登場します。この追加コマンドは自然言語処理を用いて新聞データから作成されています。この追加コマンドから当時の価値観を知ることができます。また、よりリアルなゲームにするために、厚生労働省が公開している死亡や生存率などのデータを用い、ゲーム内処理にも利用しました。
 The political simulation game is a game in which the player becomes the mayor of Hokkaido to stop the population decline of the town. Depending on the time in the game, additional commands that symbolize the year will appear. Those additional commands are created from newspaper data using natural language processing. Players can learn about the values of the time from this additional command. To make the game more realistic, we also used data published by the Ministry of Health, Labor and Welfare on mortality and survival rates to create the game.



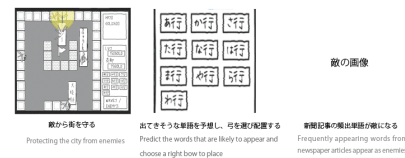
過去の出来事を体験しながら現実的な政治体験が可能!
 Players can have a realistic political experience as they experience past!

このゲームはプレイヤーに地方が抱える人口減少問題について考えることを促し、政治や社会問題への関心を高めるきっかけを与えます。また、新聞のデータを利用したことにより、プレイヤーは過去と現在の価値観の違いを知ることができます。
 The game encourages players to think about the demographic problem facing rural areas, and provides an opportunity to raise interest in politics and social issues. The use of newspaper data also allows players to see the difference between past and present values.

タワーディフェンスゲーム Tower Defense Game

新聞でよく目にする単語から適切な武器を用いて街を守るゲームです。
 It's a game where you have to defend your city from words you often see in the newspaper with the right weapons.

このゲームでは、弓を配置することによって、攻めてくる敵から街を守ります。敵は新聞記事の頻出単語を元に作り、単語の頻出文字に合わせて弓を配置することによって敵に与えるダメージを増やすことができます。また、タワーディフェンスの構築によって、シミュレーションゲームに影響があります。
 In this game, you have to defend your city from words that appear in newspaper articles. Enemies are created based on frequently appearing words in newspaper articles output by language processing. By placing bows appropriate to the first letter of the word, the player can increase the damage inflicted on the enemy. The results of the tower defense will influence the simulation game.



新聞の頻出単語は、その新聞が作られたときの社会の関心を反映しています。このタワーディフェンスをプレイする過程で、プレイヤーは世間で関心が高い事柄を知ることができます。
 Frequently appearing words reflect the interests of the society at the time the newspaper was produced. In the process of playing this tower defense game, players can learn about public concerns.

背景技術 Background Technology

新聞ビッグデータを利用し、ゲームで使用するデータをPythonで作成
 Creating data for games in Python from newspaper big data.

- 頻出単語の抽出** Frequently appearing Words Extraction
 Morphological analysis was used to extract frequently appearing words from newspaper articles by day, and sorted in order of frequency of occurrence.
- ベクトル計算** Vector Computation
 単語を数値ベクトルに変換し、頻出単語と指定単語の意味的な近さを数値化した。
 We converted the text using a vectorization algorithm that extracts important sentences from the entire text.
- 本文要約** Text Summarization
 文章全体から重要な文を抽出する要約アルゴリズムを用いて本文を要約した。
 We summarized the text using a summarization algorithm that extracts important sentences from the entire text.
- 重要度の算出** Importance Calculation
 ある単語の文中での出現頻度と文書間の相対度から、その単語が文中でどの程度重要なかを算出した。
 Based on the frequency of occurrence of a word in a sentence and its rarity between sentences, we calculated how important the word is in the sentence.

活動を通して Through Project-based-Learning



このプロジェクトを通して、メンバーの考えも変化しました。はじめは政治への関心が低く、投票に行ったことのないメンバーもいました。しかし、今では毎週のミーティング時に政治についての話題が出るようになりました。また、新聞を読まないメンバーが多かったのですが、今では新聞に触れる機会が増え、新聞を読む回数が増えました。
 Through this project, our members' views have also changed. At first, there was little interest in politics, and some members had never voted. Now, however, politics has become a topic of discussion at our weekly meetings. Also, many members did not read newspapers, but now they have more access to newspapers and read them more often.

図 2.11 成果発表会で用いたポスター

(※文責: 鈴木稜司)

2.2.5.3 タワーディフェンスゲームデモの作成

最終成果発表へ向けて、制作ゲームのデモを制作した。成果物2つのうち、政治シミュレーションゲームに関しては、短い発表時間の中であればデモ動画を作成した方が良く判断し、デモは作成しないこととした。タワーディフェンスゲームのデモ作成は、主にどの年のデモを見せるかの決定、ゲームバランスの調整の2つをおこなった。

まず、どの年のデモを見せるかの決定に関しては、出現単語を予測することによって時事に関して考える機会となることを体験してもらうため、比較的出現する単語を予測しやすい年として、歴史的に大きな出来事があった年のものを2つ用意することにした。1つ目は、新型コロナウイルス感染

症の流行が本格化し、新聞に取り上げられ始めた 2020 年ものとした。2 つ目は、東日本大震災が起こった 2011 年のものとした。

次に、ゲームバランスの調整に関しては、短い発表時間の中でより多くの人に体験してもらえるよう、難易度を低めにし、クリアまでの時間が短くなるように設定した。主に出現する敵の数と出現間隔、体力、速度を適切に設定することで、可能な限り 1 人あたりの体験時間が短くなるようにした。まず敵の数と出現間隔に関しては、通常であれば敵の数は数十体と多めに設定し、出現間隔は 1 プレイが長くなりすぎないようにしたうえで、防衛アイテムの配置を考える時間が少しでも多くなるように 7~8 秒と長めに設定した。しかし、今回のデモでは時間を短くすることが優先であったため、敵の数は 6 体まで減らし、出現間隔は単語が重ならないように配慮したうえで、5 秒毎に出現するように設定した。次に、敵の体力に関しては、今回実装している「特定の敵に対し与ダメージが上がる弓」の効果を体験しやすいように、全ての敵の体力を 10 とした。そして、敵の速度に関しては、難易度が上がりすぎないように気を付けながら、出現間隔と合わせて可能な範囲で速くなるように設定した。また、クリアができなかった場合にどんな単語が出てくるのかを見ることができなくなる可能性も考えられたため、敵の攻撃でプレイヤーの体力が 1 削られるのに対し、プレイヤーの体力を 100 と多めに設定することでゲーム失敗が起きないようにした。

(※文責: 嵯峨京介)

2.2.5.4 デモ動画の作成

成果発表会ではオーディエンスに制作した政治シミュレーションゲームのイメージが伝わりやすいように、発表時にデモを提供することになった。政治シミュレーションゲームは成果発表会の発表時間ではプレイしてイメージをつかんでもらうのは難しいと考えたため、プレイの様子を写したデモ動画を発表中に再生するという事に決定した。デモ動画の形式については、同じスペースで他のプロジェクトも発表しているという環境から、音声での説明は適さないと判断しゲームのプレイ動画にテロップでの説明を加えたシンプルな動画にした。動画制作は 2 人でおこない、目的別に 2 つの動画を制作して最終的に 1 つの動画にした。1 つ目の動画はゲームプレイの様子を流しながら、テロップでゲームのシステムや特徴を説明するものであり、ゲーム自体のことを知ってもらうことを目的とした。2 つ目の動画は、2 人のキャラクターが話しながらゲームをプレイする寸劇方式の動画である。キャラクターが楽しくゲームをしている様子を見せ、ゲームの楽しみ方を説明することで、オーディエンスのゲームに対する解像度をより高めることを目的とした。それぞれ、動画編集ソフトは Microsoft Clipchamp と AviUtl を用いた。ソフトに対する慣れの都合上、別々の環境で制作することになったが、問題なく結合ができた。発表用のデモ動画を制作したことにより、成果発表でオーディエンスのゲームに対する理解に寄与することができた。

(※文責: 杉山孝太)

2.2.5.5 成果発表会当日の様子と振り返り

ここでは、成果発表会の様子やオーディエンスの評価、評価を受けての我々の反省について述べる。成果発表会では質疑応答を含め 15 分の発表を 6 回おこなった。その際、プロジェクトメンバ全員が発表に参加するように担当を割り振った。発表形式としては、モニターに発表スライドを映しながら説明し、発表中はオーディエンスがゲームのイメージをしやすいようにプロジェクターでゲームプレイの様子を映した動画を繰り返し流した。また、発表の横でオーディエンスがタワーディフェンスを体験できるように、タワーディフェンスのみプレイ可能なデモを用意した。発表は

スムーズに進み、問題なく成果発表会を終えることができた。

本プロジェクトの成果発表に対しての成果シートは 41 件提出され、発表技術に対しての平均評価点は約 8.1 点、プロジェクトの内容に対しての平均評価点は約 8.7 点だった。発表技術に対してのコメントでは、説明内容やスライドを高く評価している人が多かった。しかし、声が聞こえない、または聞き取りづらいというコメントが 15 件と最も多かった。1つの部屋で3つのプロジェクトが発表していたため、発表場所の都合上仕方がないというコメントもあったが、一部オーディエンスが聞き取れない発表をしてしまったのは事実であり、声量が足りなかったことは、成果発表会で一番の反省点であった。プロジェクトの内容に関しては、プロジェクトの目的からアプローチの方法、開発したゲームの内容まで、ほとんどのコメントで好評だった。まとめると、声量が十分でないという反省点はあったが、プロジェクトの内容や発表方法については高い評価を受けることができた。そのため本プロジェクトでの発表に対する自己評価は 5 段階中 4 という評価をつけた。

(※文責: 杉山孝太)

第3章 成果物

ここでは、成果物の概要、新聞ビッグデータを活用したゲームの設計、そしてゲームの概要について述べる。成果物の概要では、シリアスゲームの説明と、成果物内でのシリアスゲーム要素に焦点を当てて、説明している。新聞ビッグデータを活用したゲームの設計では、成果物が完成するまでの活動について説明している。最後に、「はこじゃな」の概要では、ゲームの特徴や進行について具体的に説明している。

(※文責: 木村星凱)

3.1 成果物の概要

私たちは、若者の政治への関心が低いという課題を解決するために2つのシリアスゲームを作成した。その2つとは、「政治シミュレーションゲーム」と「タワーディフェンスゲーム」である。これら2つのゲームはいずれも、新聞の記事データが活用されている。政治シミュレーションゲームではコマンド生成、イベント生成、パラメータ調整に使用されており、タワーディフェンスゲームでは敵の生成に役立っている。この2つのゲームは1つのゲームとしてプレイすることが出来る。シミュレーションゲームをメインとし、タワーディフェンスゲームをシミュレーションゲームにおけるミニゲームとして実装した。

シリアスゲームとは、ゲームの目的とは別に、社会問題の解決を目的としたゲームのことを指す。例えば、九州大学シリアスゲームプロジェクト [4] により作成されたゲーム「あいにくけん」などがシリアスゲームの例としてあげられる。同ゲームは、デスクトップアプリケーションゲームであり、キャラクターを操作しながらゲーム内で起こる街のゴミ問題を解決し、目的のキャラクターに会いに行くというゲームである。ゲームの目的とは別に、プレイヤーはこのゲームをプレイすることで、福岡市のゴミルールブックに基づいた、身近ですぐに役に立つゴミ問題についての知識が身につく。実際、財津ら [5] によると、「あいにくけん」をプレイしたことにより、環境問題に対して何らかの行動を取っていない人達への、行動意識を変化させることに効果があることが示されている。

以降この章では、政治シミュレーションゲームの詳細な説明をした後、タワーディフェンスゲームについての詳細な説明をおこなう。

(※文責: 永井清盛)

3.2 新聞ビッグデータを用いたゲームの設計

3.2.1 ゲームデザインとゲームバランスの調整

シミュレーションゲームの制作において、ゲームデザインの設計と微調整した。最初に、ゲームデザインの設計を進めた。まず、このゲームのプレイ時間は5~15分程度になるように設計した。その理由は2つある。第1に、多くの人に楽しんでもらうためである。シミュレーションゲームのプレイ時間は、最低でも数時間かかるものが多い。このゲームはシミュレーションゲームである

が、シミュレーションゲームに馴染みの薄い人にも楽しんでプレイしてもらおうべきであると考えた。そこで、このゲームは5~15分程度の短時間でも楽しめるように設計した。第2に、成果発表会でのデモプレイを考慮した。成果発表会の1回の発表に使える時間は15分である。この制約内で、制作過程や成果物についてオーディエンスに伝える必要がある。短い時間内で実際にゲームを体験してもらうため、プレイ時間を5~15分程度に設定した。実際にプレイしてみると5~15分でクリアすることはほぼ不可能であった。しかし、成果発表会ではゲームをクリアすることが目的ではなく、ゲームの特徴を体験してもらうことが目的としているため問題はないと考えた。

政治シミュレーションゲームの舞台は函館市である。このプロジェクトで使用する新聞ビッグデータは、北海道新聞社から提供していただいている。このデータが他の新聞のデータと違う点として、北海道の政治やニュースについてのデータが多く含まれている点が挙げられる。この特徴を活かすために、政治シミュレーションゲームの舞台に北海道を選択した。しかし、北海道を舞台にするにはあまりにも広く、ゲームの目的が曖昧になってしまうため、北海道の函館市を舞台にすることにした。また、函館市は、公立はこだて未来大学が位置する町であり、我々にとって身近であることも採用した理由である。

政治シミュレーションゲームのゲームの目的は、「人口減少を食い止める」に決定した。ゲームの目的がこれになった理由はいくつかある。1つ目に、函館市が抱える問題として人口減少問題があるからである。函館市 [6] によると、函館市の人口は1980年の34.5万人がピークであり、その後は減少傾向にある。2015年は26.6万人であり、2040年には18.7万人、2060年には13.4万人になると推計されている。この問題について考える機会をプレイヤーに与えることが狙いである。2つ目の理由として、ゲームの目的がわかりやすいという点が挙げられる。多くのシミュレーションゲームは、ゲームの明確なクリア条件や明確な目的がなく、どこまでプレイするかはプレイヤーの裁量に任されている。しかし、このゲームはシミュレーションゲームに慣れ親しんでいる人のみならず、そうでない人にもプレイしてもらおうべきである。そこで、ゲームの明確な目的として、人口減少を食い止めることとした。

ゲームに実装するパラメータの選定に際しては、既存の政治シミュレーションゲームやまちづくりシミュレーションゲームを参考にした。これらのゲームでは人口や資金、税率などのパラメータが利用されていることが多かった。そこで、このゲームでもこれらのパラメータを採用することにした。パラメータの初期値に関しては、函館市の人口統計や予算案を参考にしながら、実際の市の状況を反映させた。ゲーム内の人口や予算、転入数、転出数の初期値は現在の函館市とほぼ同様の数値となっている。これにより、ゲームをプレイしながら、現在の函館市の状況を少しでもプレイヤーに伝えることを目的としている。

また、コマンドに関しては適切な資料が見当たらなかったため、自分たちで作成した。まず、ゲーム内で利用される全てのパラメータをリストアップし、それらのパラメータを増減させるにはどのような政策が必要なのかを考え、リストアップした。その後、これらの政策を実行すると、他のパラメータにどのような影響を与えるか考え、コマンドを実行したときのパラメータの変動量を決定した。例として、警察強化コマンドを作成した場合を挙げて説明する。まず、ゲーム内のパラメータをすべてリストアップした。次に、犯罪率のパラメータを下げるには、警察を強化する政策が最適だと考えた。その次に、警察を強化した場合、他のパラメータにどのような影響があるかを検討した。この場合、犯罪率のパラメータの値が減少するだけでなく、支持率や支出のパラメータも変動すると考え、これらのパラメータの値の変動量を決定した。そして、警察強化コマンドが完成した。

こうしてゲームバランスを考慮してパラメータやコマンドを構築し、実装した。その後、実際の

プレイを通してゲームバランスの微調整を行い、難易度や数値の調整に関する討議を重ねてゲームを完成させた。

(※文責: 村澤雄斗)

3.2.2 基盤システムとユーザインタフェースの実装

本ゲームは統合開発環境 (IDE) を内蔵したゲームエンジンである Unity を用いて開発がおこなわれた。政治シミュレーションゲームの基礎となるシステムである、ターン進行やパラメータ、フラグなどを変数で管理した。加えて、クリックするとパラメータが変わるコマンドやゲームオーバーの条件設定などプログラムにより実装した。これらのシステムを実装したことにより、ターン進行型のシミュレーションゲームとしての最低限の基盤を完成させることができた。また、このゲームにはチュートリアルを実装しており、プレイヤーが操作方法、ゲームの遊び方などの面で困ることがないようにしている。この政治シミュレーションゲームの特徴としては、人口が年齢ごとに分けて管理されており、少子高齢化などの社会問題が表現できることが挙げられる。よってプレイヤーは年齢の要素も考慮して、戦略を考えることになる。加えて追加コマンドやイベントなどのゲームの目玉となる機能を実装している。追加コマンドとイベントは決められたタイミングでターン進行の際に出現するように設計した。追加コマンドの出現時は、コマンドが追加されたことが分かりやすいようにポップアップを表示し、コマンドの文字は赤色となっている。また、イベント発生時にはテレビのニュースのような画面が表示され、新聞から生成された見出しと、新聞から生成した要約を表示する。ニュースのように表示することによりゲーム内のイベントに対する臨場感を演出した。そして、市のパラメータである人口、予算、支持率が閾値を下回るとゲームオーバーになる設計をしている。ゲームオーバー時には、開始から数えたターン数と原因を表示している。この要素は市政をどれだけ続けることができたか明確に示すことで、記録を伸ばす楽しさや他プレイヤーと競えるようなゲーム性を実現している。

UI の面では、見やすくシンプルなものになるようこころがけた。ボタンや文字などの基本的な素材は Unity 標準なものを使い、ウィンドウや文字、ボタンは主に黒、白、灰色など素朴で見やすい色を用いてプレイヤーが認識しやすいものを表示している。また、追加コマンドの文字は赤色で表示され、追加コマンドの出現時にはポップアップが表示されるため、プレイヤーが追加コマンドに気づきやすい設計になっている。チュートリアルやイベント発生時のポップアップでは外部の素材を使っており、この素地も落ち着いたモノトーン調のものを用いてデザインに統一感を持たせている。コマンド実行時にはどのパラメータが変化したかをプレイヤーに分かりやすくフィードバックするために、ポップアップで変化したパラメータを表示している。このポップアップが表示されている時間もデモをプレイしながら調整し、プレイヤーのストレスにならないよう適切な時間になっている。そして、ゲーム中の背景には函館市の写真を使用し、プレイヤーの没入感を高めている。これらの使用素材の選択や調整により、このゲームの UI はプレイヤーがゲームの状態を分かりやすく認知することができ、直感的にストレスなく操作できるように設計されている。

(※文責: 杉山孝太)

3.2.3 新聞データの活用

自然言語処理班が作成したデータをゲームで活用するため、Unity へデータの受け渡しをする機能を追加した。はじめに、自然言語処理によって得られたデータを Unity に適した形式に整形す

ることとした。新聞記事の要約データでは、1988年から2020年までの32年分の新聞記事の要約データが1年ごとにまとめられているため、それぞれのファイルから10要素程度取り出し、1つのファイルにまとめた。その後、元々あったtitle, summary, dateというカラムのほかに、status, nというカラムを作り、データを追加した。ファイルのフォーマットとしてはjsonを用いた。パラメータと頻出語間のベクトル距離データでは、それぞれのファイルからコマンド名になりそうなものを選び、それぞれのパラメータの中で頻出語間のベクトル距離の値が大きいものを抜き出して1つのファイルにまとめた。新聞記事の要約データ同様、ファイルのフォーマットとしてはjsonを用いた。つぎにUnity上で新聞記事の要約データとパラメータと頻出語間のベクトル距離データの2つのデータを読み込むため、json形式で保存したい変数をもったクラスを2つ作成した。1つ目のクラスでは、title, summary, date, status, nというKeyを設定した。title, summary, dateというKeyにはそれぞれ新聞記事の要約データの新聞記事の見出し、新聞記事の要約、新聞の日付が格納されている。status, nというKeyにはそれぞれパラメータを変化させる処理の入ったメソッド名、0or1が入っている。statusには、新聞記事の内容を見て、妥当なパラメータを変化させているメソッドを選択している。nにはイベントとして表示するものには1、表示しないものには0を設定している。このnの値は、イベントを定期的に出現させるための要素である。これらの機能を実装することで、新聞記事の日付とゲーム内時間が一致した新聞記事をゲーム内に表示することができた。新聞記事の日付ごとに複数のデータを抽出しており、その中からその日付の記事を一つ選ぶため、ゲームをプレイするごとに表示される新聞記事をランダムにできた。また、2つ目のクラスでは、date, status, word, parameterというKeyを設定した。date, status, word, parameterというKeyにはパラメータと頻出語間のベクトル距離データ内の日付、単語とパラメータ間のベクトル距離が最も近いパラメータの単語、単語、パラメータが格納されている。クラスができて、ファイル内のデータを扱えるようになると、次にイベント機能、追加コマンド機能にそれらのデータを反映させた。イベント機能では、ゲーム内時間とdateが一致したときに、新聞記事の見出し、新聞記事の要約を表示させ、statusに応じてメソッドを呼び出してパラメータを変化させた。追加コマンド機能では、イベント機能同様、ゲーム内時間とdateが一致したときに、wordをコマンド名とするコマンドを追加するようにした。また、追加コマンドを押したときの処理として、statusに応じてパラメータを変化させるようにした。パラメータの変化量についてはパラメータごとに基準となる変化量を決めて、その変化量とstatusを掛けた値を追加コマンドによるパラメータの変化量と定めている。各パラメータの基準となる変化量はそれぞれ支持率:5、犯罪率:0.5、市内総生産:200、教育:3、環境:4である。これらの機能を実装することで、その年代に関心のあったトピックを追加コマンドとしてコマンドを追加、変化させることができるようになった。また、コマンドを押したときの処理も、データを用いてパラメータの変化量を設定できた。このようにして、Unity上で自然言語処理班が生成したデータを活用し、ゲーム内でリアリティのあるイベントやコマンドの実現に成功した。図3.1、図3.2は自然言語処理によって得られたデータをUnityに適した形式に整形したものである。

```

{
  "title": "TestTitle",
  "notes": [
    {
      "title": "ホタテのマヒ性貝毒、バイオ物質使い実験で解毒に成功。小樽水産高校 境一郎教諭",
      "content": "【小樽】日本水産学会会員で小樽水産高校栽培漁業科の境一郎教諭（５９）が、これまで有効な除去方法のなかったホタテ貝のマヒ性貝毒を、バイオ物質を使って解毒させる実験に成功、研究内容を来春の同学会春季大会で発表する。本道では噴火湾を中心に毎年大量のマヒ性貝毒が発生しており、実験が実用化されれば生産者には大きな朗報となる。",
      "date": "1988/10/28",
      "status": "4",
      "n": "1"
    },
    {
      "title": "和光研が８９年３月期の決算予想－利益は過去最高に",
      "content": "和光経済研究所は一日、九月中間決算の実績を基に、一九八九年三月期決算の予想をまとめた。それによると、好調な内需景気はさらに持続、売上高は前期比五・七％増にとどまるものの、経常利益は同二〇・二％と大きく伸び、八五年三月期以来四年ぶりに過去最高益を更新すると予想している。",
      "date": "1988/10/28",
      "status": "1",
      "n": "1"
    },
    {
      "title": "大蔵・自治省がふるさと財団の設立を認可、地域づくり事業を正式決定",
      "content": "竹下政権の税制改革後の重要課題であるふるさと創生構想の具体化のため二つの事業が二十一日、正式決定した。",
      "date": "1988/10/28",
      "status": "3",
      "n": "1"
    }
  ],
}

```

図 3.1 Unity 用の整形した新聞記事の要約データ

```

{
  "title": "TestTitle",
  "notes2": [
    {
      "date": "1988/07/12",
      "parameter": "市内総生産",
      "word": "行財政",
      "status": "0.39992973"
    },
    {
      "date": "1994/01/06",
      "parameter": "出生率",
      "word": "出稼ぎ",
      "status": "0.44704112"
    },
    {
      "date": "1997/01/06",
      "parameter": "教育",
      "word": "統廃合",
      "status": "0.39405096"
    },
    {
      "date": "1998/07/12",
      "parameter": "支持率",
      "word": "利下げ",
      "status": "0.60186714"
    }
  ]
}

```

図 3.2 Unity 用の整形したパラメータと頻出語間のベクトル距離データ

(※文責: 木村星凱)

3.2.4 ゲームで使用したライブラリの説明

本ゲームでは、1年の終わりにこれまでの人口推移を表す、横軸が時間、縦軸が人口の折れ線グラフが生成され表示される(図 3.3)。これにより、プレイヤーは市の人口がどのように変化しているか理解でき、それまでのプレイの良し悪しを判断し、それからの戦略を決定するための材料を得ることができる。

グラフを生成する専用の機能は Unity には実装されていないが、グラフを生成するのみならラインを描画する Unity 標準の機能を利用して作ることができる。しかし、この実装方法は Unity 標準で搭載されている UI 描画システムである uGUI(Unity Graphical User Interface) との相性

が良くない。具体的には、描画されるラインの座標はキャラクターなどゲーム内に存在するものの位置情報を設定するワールド座標で管理されるのに対し、uGUI で実装された UI はプレイヤーが見ている画面上の位置情報を設定するスクリーン座標で管理されている。この違いにより、Unity のライン描画によりグラフを表現してしまうと、グラフとその他 UI で座標の基準が統一されていないので、グラフが扱いづらくなってしまふ。特にサイズの変更などの修正をする際は、座標の基準が統一されていないと UI とは異なる変化をしてしまふ。よって、グラフの生成は外部のライブラリである、unity-ugui-XCharts を用いた*¹。unity-ugui-XCharts は Unity 用のグラフ作成およびデータ視覚化ライブラリである。このライブラリは純粋なコードでレンダリングされ、外部から追加のテクスチャなどを使う必要がなく、このライブラリのみで完全なグラフの生成、描画が可能になっている。また、折れ線グラフや棒グラフなどの 10 種類以上の豊富なグラフを扱うことができ、他にも多彩なラインの種類、アニメーションの利用など、高いカスタマイズ性を持っている。そして、これらの豊富な機能をコードのみで実現できることが特徴的である。加えて、unity-ugui-XCharts で作成したグラフは、UI と同じく位置情報はスクリーン座標で管理されるため、UI と同じようにサイズの調整や位置の移動ができる。以上の理由により、このゲームでは unity-ugui-XCharts を採用した。ゲーム内での人口推移グラフの実装は、ゲーム開始からターン進行ごとに人口をリストに記録し、年が変わるタイミングでリストから折れ線グラフを生成して表示した。グラフの表示は最初のターンから順に表示されていくアニメーションを用いることで、それまでの人口の推移を強調している。また、グラフ自体も背景を白、文字を黒、グラフを青色に設定することで、シンプルな UI を実現している。

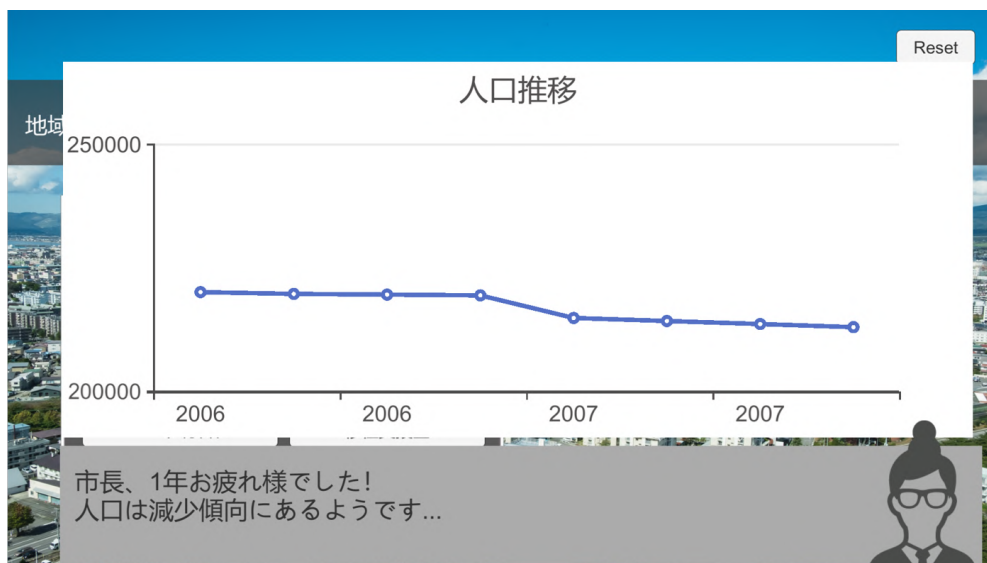


図 3.3 人口推移グラフの表示

(※文責: 杉山孝太)

3.2.5 タワーディフェンスゲームの基盤設計

タワーディフェンスゲーム班は、Unity 入門の森を参考にし、ゲームエンジンである Unity を用いてタワーディフェンスゲームを作った。

*¹ XCharts-Team, <https://github.com/XCharts-Team/XCharts/blob/master/README-en.md>

(※文責: 鈴木稜司)

3.2.5.1 ゲーム画面の概要

ゲーム画面は、右側には敵が攻めてきたり敵から防衛対象を武器で守ったりするプレイ画面と、HP やゴールド、ウェーブ、敵の数の表示・弓矢のレベルアップなどがおこなえる画面でできている。

(※文責: 鈴木稜司)

3.2.5.2 プロジェクト作成とタワーディフェンスゲームの素材のインポート

これを作るにあたり最初におこなったのは、プロジェクト作成とタワーディフェンスゲームの素材のインポートをおこなった。プロジェクトを作成後、「Images」、「Prefabs」、「Scripts」、「Fonts」、以上4つのフォルダを作成した。「Images」には、ゲームで使用する敵の画像や武器の画像などを置いておく場所である。「Prefabs」には、「Prefab化」したモノを置いておく場所である。「Prefab化」とは、作成済みのオブジェクトを複製する仕組みのことである。例えば、「大量の同じ敵」や「同じ仕掛けを持ったトラップ」などを、簡単に複数作ることが可能になる。「Scripts」には、ゲームを動かす上で制作したプログラムなど置いておく場所である。「Fonts」は、ゲームで使用する文字などを置いておく場所である。

これらのファイルを作る理由は、ゲームの複雑さが増すごとに関連するファイルが増えてきてしまうので、どこになにがあるかわからなくならないようにするためである。また、今回使った画像と文字は以下のサイトから zip ファイルをダウンロードしている。^{*2}

(※文責: 鈴木稜司)

3.2.5.3 敵オブジェクトの作成・パラメータ設定・敵処理のプロトタイプ作成

ここでは、敵オブジェクトの作成・パラメータ設定・敵処理のプロトタイプ作成について説明する。敵オブジェクトは、空の親オブジェクトと子要素となる画像で構成されている。階層構造にしておくことで、敵の位置の移動と、演出としての位置の移動を分離するためである。敵のパラメータは、HP、進行速度、倒したときに手に入るお金と設定した。これらは、スクリプトでプログラムした。しかし、これだけでは移動しないのでスクリプト内で Update 関数を使い、オブジェクトの位置を表す `transform.position` を徐々に変化させた。また、敵は1体だけでは寂しいので、画像を変えて複製し Prefab 化した。

(※文責: 鈴木稜司)

3.2.5.4 移動経路の構築と OnDrawGizmos による移動経路の可視化

ここでは、移動経路「Route」の構築について説明する。まず、通過点を置くために親オブジェクト「Points」を作る。次に、通過点を示す空オブジェクト「Point」を子要素として作る。しかし、このままだと Point がどこにあるかわかりづらいので、アイコンをつけた。その後、Point スクリプトを追加する。スクリプト内に、OnDrawGizmos という関数を追加することで矢印やカメラアイコンなどが Scene ビューだけで表示されるようになる。今回は、矢印が見えると「Route」がわかりやすくなるので追加した。また、UnityEditor.Handles.Label(表示位置、表示する文字

^{*2} (1)<https://www.kenney.nl/assets/scrabble-platformer> (2)<http://font.sumomo.ne.jp/shigoto.html>

列); という UnityEditor 用に用意されている、指定した場所に文字列を表示するメソッドを使用している。表示位置には `transform.position` を指定しているなのでそのオブジェクトの位置そのものを指定し、表示する文字列には `name` を指定した。これは、オブジェクトの名前が入っているメンバ変数である。今回の例だと、「Point」という名前にしたので、Scene ビューに「Point」と表示されるようになっている。次に、Point と Point を繋いでいく Route オブジェクトを作る。まず、Route を置くための親オブジェクトとして Routes オブジェクトを空オブジェクトで作成する。さらに、その子要素に Route オブジェクトを空オブジェクトで作る。その後、Route スクリプトを追加しメンバ変数として Point の配列 `points` を追加する。Route (経路) は複数の Point (通過点) から作られる。スクリプトで、複数のオブジェクトを管理するには、配列や、List (リスト) を使うが、今回は途中で増えたり減ったりしないので、配列 (`Point[]`) にした。以上のことをおこなうことで、UnityEditor の Inspector ビューで Point を指定することができるようになる。Start から Goal まで通過する Point を指定して、1 つの Route が出来上がる。今回は、通過点を変更し 4 つの Route を作成した。また、Route スクリプトにも `OnDrawGizmosSelected` という関数を追加することで、選択した Route 以外は見えないようにした。

(※文責: 鈴木稜司)

3.2.5.5 目的地設定とベクトルを用いた経路設計

ここでは、敵が Route に沿って移動する仕組みについて説明する。スタート位置 `route.points[0].transform.position(From)` から、次のポイント `route.points[1].transform.position(To)` へと向かうベクトル (`Vector3`) を計算するように Enemy スクリプトを修正した。しかし、問題が 2 つ残った。1 つは、移動速度が速くなってしまっていることである。これは、ベクトルの正規化をしていなかったためである。この正規化 (Normalize) というのは、ベクトルの方向はそのままに、距離だけを 1 にするという意味になり、正規化をしないと、Point から Point の距離が長ければ長いほど、移動速度が速くなってしまふ。そこで、移動ベクトル (`v`) をそのまま使うのではなく、`v.normalized` (正規化) したもを使用した。もう 1 つは、曲がるべきところで曲がっていないということである。この Enemy スクリプトでは、ずっとスタート地点 (`route.points[0]`) から、次のポイント (`route.points[1]`) への移動ベクトル (`v`) だけで移動している。Enemy が移動をして、次のポイント (`route.points[1]`) まで辿り着いたら、今度はその次のポイント (`route.points[2]`) への移動ベクトルを計算。また次のポイント (`route.points[2]`) まで辿り着いたらさらにその次のポイント (`route.points[3]`) を・・・といったように参照する `route.points` を状況 (今、どの Point からどの Point への移動中なのか) に応じて変化させていく必要がある。この、「どの Point から移動中なのか」を示すメンバ変数として `private int pointIndex;` を追加した。移動ベクトル (`v`) も `var v = route.points[pointIndex + 1].transform.position - route.points[pointIndex].transform.position;` のように変更した。そして、次のポイント (`route.points[pointIndex + 1]`) まで辿り着いたかどうかの判断をする。現在参照している `route.points` の 2 点間 (`route.points[pointIndex]` から `route.points[pointIndex+1]`) の距離と今の Enemy の現在の Point からの距離 (`route.points[pointIndex]` から Enemy の位置) を比較して、判断することにする。まず、現在の Point の起点である `route.points[pointIndex]` (From) から Enemy の位置 (To) のベクトルを変数 `pv` に求めている。次にベクトルの長さを調べるには `pv.magnitude >= v` というプロパティを使う。そして `pv.magnitude >= v`、すなわち、Enemy が次のポイントに付いた (または通り過ぎてしまった) 場合は、`pointIndex`

を1増やして、参照する route.points を1つ先に進めるようにした。ゴール判定は、pointIndex を1増やして、参照する route.points を1つ先に進めた時点で、pointIndex がゴールである route.points の最後の要素を差している場合は、ゴールに到着したと考えた。最後まで到達したら、Destroy(gameObject) して、Enemy オブジェクトそのものを消すという処理をした。

(※文責: 鈴木稜司)

3.2.5.6 敵の生成処理の作り方 ウェーブの作成

ここでは、敵の生成処理の作り方とウェーブの作成を説明する。敵オブジェクトを複数出現(生成)させていくにあたり、その生成タイミング等を管理するオブジェクト(EnemyManager)を作った。EnemyManager スクリプトには、今が第何 wave か管理(wave)、wave がスタートしてからの経過時間を管理(time)、生成する敵(Enemy)を指定、生成する時間(Time)を指定、Enemy が通る経路(Route)を指定、を実装した。wave は、EnemyManager オブジェクトの Inspector で、wave 内の Size(ウェーブ数)、Patterns の Size(第何ウェーブで敵を何体出現させるか)、time(敵を出現させる時間)、Enemy(どの敵を出現させるか)、Route(どの Route を通過するか)を指定して作成した。

(※文責: 鈴木稜司)

3.2.5.7 敵当たり判定とコルーチンを用いた弓矢の生成&射出処理の作成

ここでは、敵の当たり判定と弓矢の生成、射出処理の作成を説明する。敵の当たり判定は、Enemy をプレハブ編集モードで調整した。Enemy に、Inspector ビューで BoxCollider2D を追加した。これを追加することで、プレハブ編集モードを使い、当たり判定を調整できるようになる。以上のことを使い、当たり判定を Enemy の大きさに合わせた。これで、Enemy を攻撃する準備が整った。次に弓の生成である。弓の空オブジェクト(Bow)に、ダウンロードした画像の item.bow と item.arrow の2つを子要素として入れた。次に、Bow スクリプトを作成した。これに、必要な要件は「一定間隔で周囲の敵を探す」と「矢(Arrow)を撃つ」であった。「一定間隔で周囲の敵を探す」を実装する上で、今回は「コルーチン」という機能を使った。普通の関数は、処理を開始すると終了まで(return まで)処理は止められないが、コルーチンは、処理の中断が出来、また中断した場所から再開が出来るというのが普通の関数とコルーチンの違いである。また、「一定範囲の敵を探して当たり判定をおこなう」という機能も実装した。2Dでの円の当たり判定結果を返却してくれる関数 Physics2D.OverlapCircle 関数を使い、結果をローカル変数 collider に格納した。今回は、Physics2D.OverlapCircle(円の中心位置, 半径, 対象レイヤー)を使用しており、円の中心位置は Bow オブジェクトの位置そのもの(transform.position)、円の半径は 2.0f、対象レイヤーは Enemy レイヤー → LayerMask.GetMask("Enemy")を指定した。そして、もしどの敵オブジェクトも範囲内ない場合は collider が null になるため、逆に「null じゃなければ処理を行う」という if 文を書いた。

次に矢(Arrow)を作成した。Arrow オブジェクトに Arrow スクリプトを追加した。スクリプト内で、public Enemy targetEnemy; でどの敵オブジェクトに向かって飛ぶかが外(Bow)から指定できるようにし、var v = targetEnemy.transform.position - transform.position; transform.position += v.normalized * speed * Time.deltaTime; で指定敵オブジェクトの位置へと向かうベクトルを求めて、移動させた。そして、敵との距離(当たり判定と)がある程度小さく(ここでは 0.7f) になったら、敵に当たったとみなした。敵に当たったら、当然敵にダメージ

を与え、敵の HP が 0 になったら敵を削除 (`Destroy(targetEnemy.gameObject)`) し、矢も削除 (`Destroy(gameObject)`) した。作成した Arrow オブジェクトは Prefab 化した。

次に、Arrow Prefab を Bow から生成する仕組みを作成した。生成する Arrow プレハブを Inspector ビューで設定できるように、メンバ変数を追加した。そして、今は範囲内に入った敵オブジェクトを Destroy して削除してしまっているが、これを敵の方向に向く、arrowPrefab を生成 (複製) する、Arrow にどの敵に向かって飛ぶかセット、に変更した。発見した敵の方向に向く処理は、`Quaternion.FromToRotation` (もともと向いている方向ベクトル, 向かせたい方向ベクトル) 関数でおこなった。Arrow プレハブから、実際に Arrow オブジェクトを生成する処理は、`var arrow = Instantiate(arrowPrefab, transform.position, transform.rotation);`、`arrow.targetEnemy = collider.GetComponent<Enemy>();`でおこなった。この時、`transform.position` と `transform.rotation` を渡しているため、生成されたタイミングで場所は Bow と同じ場所、向き (回転) も Bow と同じ向き=敵の方を向いた状態となる。Arrow オブジェクトは「targetEnemy = 目標とする敵」をセットすることで、その方向へ飛ぶようにスクリプトを組んだため、collider (範囲内の当たったオブジェクト) から `GetComponent<Enemy>()` で、Enemy オブジェクトを抽出して、渡している。以上をおこない、Bow を選択し、Inspector で Arrow Prefab をセットすることで生成することができるようになった。しかし、このままだと敵にどれくらいダメージを与えたかわからないので、今回は「矢が刺さったまま」にし、与えたダメージをわかりやすくした。Arrow スクリプトの、`Destroy(gameObject);`で矢そのものを削除している箇所を `transform.SetParent` で、親オブジェクトを対象の Enemy にして、当たった Arrow オブジェクトを Enemy オブジェクトの子オブジェクトにする。次に Arrow オブジェクトの Update 関数が呼ばれ続けると、既に当たった敵に向かって移動し続けてしまうので `enabled = false;`として、スクリプトそのものを無効化する。これにより、Update 関数は呼ばれなくなるため移動も止まり、Enemy の子オブジェクトになっているため、まるで刺さったようになる。また、大量の敵が攻めてくるため Bow オブジェクトは複数配置することになる。そのために、Bow も Prefab 化した。

(※文責: 鈴木稜司)

3.2.5.8 プレイヤーの管理処理の作成、HP やゴールドの表示・弓矢のレベルアップ処理の作成

ここでは、プレイヤーの管理処理、HP やゴールドの表示・弓矢のレベルアップ処理について説明する。まず、空の Player オブジェクトを作った。Player に必要なのは、弓 Prefab、HP(拠点の体力)、GOLD(所持金)、選択中の弓、指定の場所に弓の建設ができる、選択中の弓のレベルアップができる、選択中の弓の売却ができる、である。指定の場所に弓の建設ができるは、所定の金額以上お金を持っているかを調べ、持っているなら所持金を減らし bowPrefab を Instantiate するだけとした。なお、Instantiate の第二引数に Transform を渡している。

これにより親 Transform を指定しながら生成することが出来る。選択中の弓のレベルアップができるは、選択中の Bow が無い場合は何もしないので if 文で `selectBow == null` の場合と、所持金がレベルアップの金額に足りない場合、何もせずに関数を抜けている。if 文で return されない場合、すなわち選択されている Bow があり、所持金が足りている場合は、レベルアップ処理をする。選択中の弓が売却できる場合は、同じように選択中の Bow が無い場合は return している。選択されている Bow がある場合は、Bow のレベルに応じて Gold を増やした後に Bow を削除 (`Destroy`) する必要がある。Update 関数では、左クリック (`Input.GetMouseButtonDown(0)`): `CreateBow`、右クリック (`Input.GetMouseButtonDown(1)`): `SellBow` を呼ぶようにした。しか

し、左クリックを連続で何回も押すと同じ場所に Bow が何個も置かれてしまし、右クリックしても消えなかった。消えるのは最後に作られた Bow だけだった。

これでは、ゲームにならないので、クリックした特定の場所 (今回はタイルというブロック) に弓が置けるようにした。Player スクリプトで、左クリックでただ CreateBow を呼んでいたところを、クリックした位置がタイルなのかどうかを当り判定で調べ、当たっている場合はその場所に Bow を作るように修正した。タイルが無いところをクリックしても何も起こらず、tile_border をクリックすることで、弓オブジェクトが生成されるようになった。ただ、まだ問題があった。既に弓が生成されている tile_border をもう一度クリックすることで同じ場所に何個でも弓が置けてしまうということである。既に子要素に弓 (Bow) がある場合は、新規で弓は作らず、既にある弓を選択する。この「既に子要素に弓 (Bow) があるか」を取得するために GetComponentInChildren<Bow>(); を呼んでいる。

これは読んで字のごとく、子要素から指定 (<>の中で指定) のオブジェクトを取得する関数である。それを childBow に格納している。もし、子要素に弓 (Bow) が無い場合は null なので、その場合は今まで通り CreateBow 関数を呼んでいる。null ではない場合は、既に子要素に弓 (Bow) があった。という事なので selectBow に格納して選択状態にする。これで後に、選択中の弓のレベルアップや売却の処理もできるようになった。

次に、画面左側の UI を作成していく。表示するのは、Player 情報 (HP や GOLD)、ゲーム情報 (現在の Wave、敵の数)、選択中の弓への操作 (レベルアップ・売却等) である。それぞれに枠を用意し、見やすくした。Player 情報は、PlayerStatusUI スクリプト内で HP や GOLD の変化をプログラムした。また、Inspector ビューで初期の HP、GOLD を決められるようにしている。

(※文責: 鈴木稜司)

3.2.5.9 防衛アイテムの売買・攻撃範囲強化機能

ここでは、タワーディフェンスゲームの防衛アイテムの売買・攻撃範囲強化機能について述べていく。まずは弓のアップグレードが出来るように、弓にレベルや、アップグレードにかかるコストを設定した。具体的には、レベル、攻撃範囲・攻撃速度・コスト・売却額を追加し、レベルに応じて変化するように適当な式を適用した。まず攻撃範囲は、「元の攻撃範囲+ = レベル * 0.5f (f は直前の数値を float 型に指定するためのリテラルと呼ばれるもの)」という式を適用し、レベルが上がれば上がるほど攻撃範囲が広がるように設定した。攻撃速度は、攻撃間隔で管理しており、「元の攻撃間隔 * = 0.9f のレベル乗」としてレベルが上がれば上がるほど攻撃間隔が短くなるように設定した。コストは、「元のコスト * = 1.5 のレベル乗」という式を適用し、レベルアップさせるたびに高くなるように設定した。売却額は、「コスト / 2」という式でコストの半分になるように設定した。次に選択中の弓 UI を実装した。既に配置をした弓をクリックすることで、内部的には選択状態になっているが、画面上ではそれが分からなかった。そこで、選択中の弓とその攻撃範囲がわかるような UI を実装した。Update 関数で、選択中の弓がある場合は、攻撃範囲画像を攻撃範囲分広げ (ただし、弓が持っている攻撃範囲情報は『半径』なので、2 倍する)、選択中の弓が中心になるよう移動させて表示し、逆に選択中の弓が無い場合は、攻撃範囲画像は非表示なるように設定した。最後にレベルアップと売却用のボタンを実装した。まずは Unity 側で UI オブジェクトを用意した。その後、レベルアップボタンに関しては、スクリプト側でボタン上にコスト、そのボタンの上側に選択中の弓レベルが表示されるよう設定した。同様にして売却用ボタンに関しては、ボタン上に売却額が表示されるように設定した。また、ボタンが押された際に先に書いた式が適用され

るように実装した。

(※文責: 嵯峨京介)

3.2.5.10 ゲームループ処理の作成とレベルデザイン

ここでは、タワーディフェンスゲームのループ処理作成とレベルデザインについて述べていく。まずおこなったのは、敵の情報に関する UI 部分の作成である。Unity 上で UI オブジェクトを作成し、「WAVE:1/3」のような形で現在の状況と「ENEMY:20」のような形で敵の数が画面上に表示されるようにした。次におこなったのは、ゴールドの獲得とダメージ処理である。画面上に出てきた敵に gold という変数を持たせ、その敵が倒された際に、プレイヤーの持ち金はその gold 分増えるようにした。また、敵が倒されず、防衛対象に到達した場合にプレイヤーの hp を減らすようにした。続いておこなったのは、ゲームループの作成である。今回作成したタワーディフェンスゲームは 6 つの状態を条件によって切り替えている。「TITLE」、「WAVE_CHANGE」、「GAME_PLAY」、「WAVE_CLEAR」、「GAME_CLEAR」、「GAME_OVER」の 6 つである。「TITLE」では、タイトルが表示され、スペースキーが押されると「WAVE_CHANGE」へ移動する。「WAVE_CHANGE」は、現在何ウェーブ目であるかが 2 秒ほど表示され、「GAME_PLAY」へ移動する。「GAME_PLAY」は、ゲームが開始され、敵が攻めてくる。ここで、クリアできた場合には「WAVE_CLEAR」へと移動し、全てのウェーブがクリアされると、「GAME_CLEAR」へ移動する。「WAVE_CLEAR」は、そのウェーブで獲得したゴールドが表示され、「GAME_CLEAR」は、ゲームクリアと表示される。「GAME_PLAY」で失敗した場合には、「GAME_OVER」へと移動し、ゲームオーバーと表示される。これらをプログラムによって実装している。

(※文責: 嵯峨京介)

3.2.6 新聞データ活用を始めとしたゲームの機能拡張

ここでは、夏期休暇中に設計したゲーム基盤をもとにした、ゲームの機能拡張について詳しく述べていく。機能拡張は大きく分けて「新聞データを用いた敵実装」、「頻出語データの受け取り」、「政治シミュレーションゲーム内時間に同期した敵実装」、「防衛アイテム実装」の 4 つをおこなった。

(※文責: 嵯峨京介)

3.2.6.1 新聞データを用いた敵実装

まずは「新聞データを用いた敵実装」について述べていく。今回作成したタワーディフェンスゲームでは、新聞データの頻出語が敵として出現するように実装した。敵実装については、まず初めに敵ビジュアルを決定した。単語がそのまま画面上に現れるようにすると背景と同化してしまう可能性があることと当たり判定を均一にしたいという 2 つの理由から、単語をそのまま生成するのではなく、(図 3.4) のような黒縁で中が白塗されている長方形の上に黒い文字で単語を乗せるような形で敵のビジュアルとすることにした。次に基盤をもとにして実際に Unity 内に Prefab という形で敵を作成した。Prefab というのは、何度もゲーム内に配置できるゲームオブジェクトのテンプレートのことである。単語の部分のみを変更するだけで、背景の部分は同じであるため、Prefab を使うことで、効率的に敵の実装を試みた。しかし、Unity 内で単語は Text オブジェクトとして扱われ、Prefab 化することが出来なかった。そこで、図 3.4 のように敵の画像を大量に用意し、それを Prefab 化することで解決した。画像を用いた Prefab 化の具体的な手順としては、長方形の

ゲームオブジェクトを用意し、そこに画像を貼り付ける。そして長方形に合わせて当たり判定を合わせ、Enemy スクリプトを適用するといった流れである。最後に単語に頭文字の概念を実装した。初めは、Text オブジェクトの Prefab から単語情報を取得し、一文字目で判別することを試みた。しかし、先に述べたように、Text オブジェクトの Prefab 化ができなかったため、頭文字がある単語であれば 0 を変数として持たせ、か行であれば 1 を変数として持たせる、といったように頭文字を数字で判別することで頭文字の概念の実装に成功した。

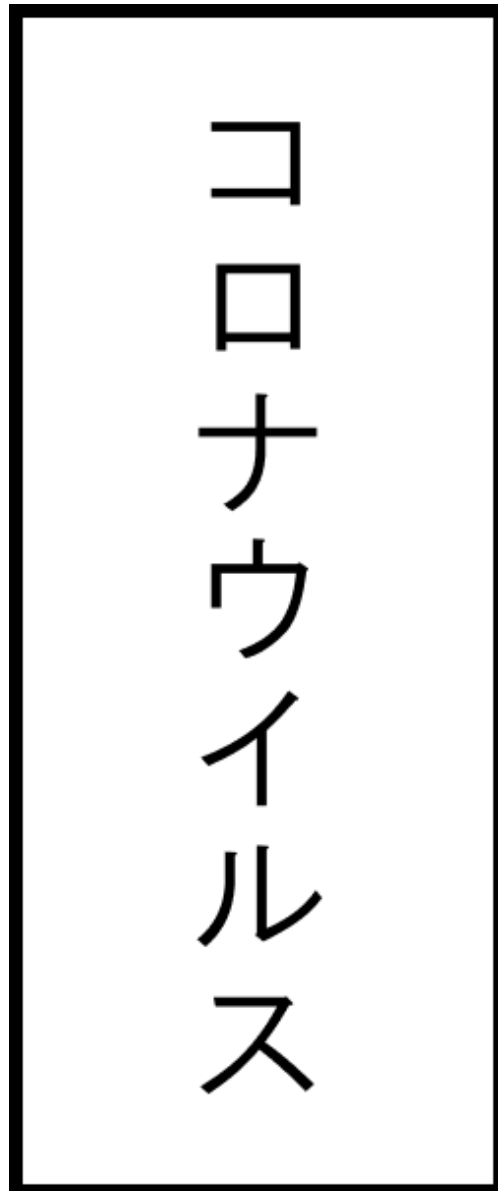


図 3.4 タワーディフェンスゲームに出現する敵

(※文責: 嵯峨京介)

3.2.6.2 頻出語データの受け取り

次に「頻出語データの受け取り」について述べていく。敵に使う頻出語について、自然言語処理・自動生成班が 1988 年 7 月から 2020 年 12 月までの新聞データから抽出した json 形式のデータの受け取りをおこない、Unity 側で読み込みできるよう、スクリプトで実装した。ここで json について少し説明をする。json とは、JavaScript Object Notation のことであり、データ記述言語の 1

つである。ゲーム等で使うデータを保存するのに広く利用されており、軽量のデータ交換用フォーマットで、プログラミング言語を問わずに利用できるのが特徴である。説明を加えたところで実装の流れを述べていく。具体的にはまず json ファイル読み込みのためのライブラリをスクリプトによって導入した。ライブラリの導入が終わった後、実際に読み込むためのスクリプトを Unity 内で作成した。流れとしてはまず json ファイルをテキストとして読み込んだ。読み込んだ後、toString メソッドを用いて文字列へ変換した。変換後、JsonNode.Parse メソッドを用いて、ただの文字列でなく、json ファイルであることを Unity 側に教えてあげることで、読み込みの実装に成功した。

(※文責: 嵯峨京介)

3.2.6.3 メインゲーム内時間に同期した敵実装

続いて「メインゲーム内時間に同期した敵実装」について述べていく。今回のタワーディフェンスゲームはメインゲームである政治シミュレーションゲームのミニゲームのような役割で実装した。政治シミュレーションゲームは3カ月毎に時間が進むようになっており、タワーディフェンスゲームは4期進んだ1年の最後にその年に対応したものが発生することとした。具体的におこなった内容は各年に応じた頻出語群の作成である。1988年から2022年まで各年分の頻出語上位30個が入った配列を用意した。その後、用意した配列を各年分の個数ある配列に格納し、2重配列の形で頻出語群を作成した。次に配列からランダムで単語が取り出される関数を実装した。流れとしては、出現させる敵の数と同じ回数 for 文でループをおこない、Random メソッドを用いて0~29までの乱数を取得する。返ってきた乱数を添え字として反映させ、配列から単語を取り出してくることで単語がランダムに出現するよう実装した。

(※文責: 嵯峨京介)

3.2.6.4 防衛アイテム実装

最後に「防衛アイテム実装」について述べていく。まず、今回作成したタワーディフェンスゲームの防衛アイテムについて簡単に説明する。今回は防衛アイテムとして「特定の敵に対し与ダメージが上がる弓」を実装した。あ行からわ行までの10種類の弓があり、攻めてくる単語の頭文字が弓の対応する行であれば与ダメージが3倍になる。これについて例を出して具体的に説明する。例えばコロナウイルスという単語が攻めてきた場合には、頭文字がか行であるため、か行の弓を配置していればダメージが上がり、効率よく敵を倒すことができる。各弓については図3.5のようなボタンを押すことで切り替えることができる。また、基盤設計時に実装されていたレベルアップや売却の機能はそのまま引き継いで使用することができるようになっている。このように、配置場所以外で弓の配置に戦略性を持たせることで、プレイヤーがより楽しめるように工夫した。

続けて Unity 内・スクリプト側の実装について詳しく説明していく。まず各弓の Prefab 作成をするため、各弓と矢のビジュアルを決定した。基盤ゲームの引継ぎとわかりやすさを加味し、図3.6のような防衛アイテムで作成していくことに決定した。Unity 内でゲームオブジェクトを用意し、そこに弓の画像を貼り付け、Bow スクリプトを適用し、Prefab 化した。同様に各弓の矢についても Prefab を作成した。その後、単語の頭文字に応じてダメージが上がるように実装した。弓に適用されている Bow スクリプトに自分が狙っている敵の情報を取得する関数があり、そこから敵の頭文字に対応している数字を取得する。その数字が弓の行と一致している場合にダメージが3倍になるように if 文で条件付けをおこなうことで特定の敵へのダメージアップを実装した。そして最後に各弓の切り替えボタンを実装した。まず、Player スクリプトに切り替えようの int 型の

変数を一つ持たせた。その変数が0であればあ行の弓の操作を、1であればか行の弓を操作できるように各行の弓に対応した関数を実装した。その後、各ボタンを押した際に、各変数が実行されるようにUnity側で設定し、切り替え機能の実装に成功した。



図 3.5 防衛アイテムを切り替えるボタン



図 3.6 タワーディフェンスゲームの防衛アイテム

(※文責: 嵯峨京介)

3.2.7 ミニゲームであるタワーディフェンスゲームとの統合

政治シミュレーションゲームでは、ゲーム中のミニゲームとしてタワーディフェンスを導入する予定であり、プロジェクト終盤でタワーディフェンスゲーム班が作成したタワーディフェンスとの統合作業をおこなった。しかし、初期統合の際はタワーディフェンスがうまく動作しない問題があったため、タワーディフェンスのコードと設定を調査し統合時の問題を特定した。

問題となっていたのはゲーム制作時の設定で、政治シミュレーションゲームとタワーディフェンスを別々に作っていたので、Unity のプロジェクト設定に違いが生じていた。設定の違いは主にレイヤーと画面の縦横比の設定である。Unity ではゲームのオブジェクト毎にレイヤーが設定でき、オブジェクトを複数のレイヤーに割り当てることで、オブジェクトの一部のみを描画したり、レイヤー毎に当たり判定や描画の順番を設定したりすることができる。他にもレイヤー毎にオブジェクトを管理することで、ゲームの編集を快適にする目的で使用されることもある。タワーディフェンス班によって制作されたタワーディフェンスでは、UI と他オブジェクトの区別や編集の利便性などのため独自のレイヤーが設定されていた。そのため、ただタワーディフェンスを政治シミュレーションゲームの中にインポートすると、政治シミュレーションゲームでは設定されていないレイヤーを用いていたことにより統合時に問題が発生することになる。また、タワーディフェンス制作時にゲームプレイ時の画面サイズをデフォルトの設定から変更していたため、異なる画面サイズ

で作っているゲームと統合した際にはゲームオブジェクトの縦横比が制作時の設定とは変わってしまう。以上の2つの理由から、2つのゲームを統合する際に問題が起こってしまった。これに対して、統合において問題となったレイヤーの設定をゲーム内で適切に動作するように、政治シミュレーションゲーム内での設定を変更し、オブジェクトのサイズが望ましいサイズになるよう手動で調整することで対処した。政治シミュレーションゲーム内でのタワーディフェンスの扱いは、一定時間毎に発生するイベントとして実装し、タワーディフェンスの結果によって政治シミュレーションゲームの予算が追加されるという仕様となった。

(※文責: 杉山孝太)

3.3 「はこじゃな」の概要

本プロジェクトで制作した政治シミュレーションゲームである「はこじゃな」について説明する。このゲームでは、プレイヤーが函館市の市長となり、人口減少を食い止めることを目的とする。また、このゲームはただの娯楽のためのゲームではなく、プレイヤーに地方が抱える人口減少問題について考えることを促し、政治や社会問題への関心を高めるきっかけを与えるシリアスゲームとして作られた。人口、予算などの函館市の状態はパラメータとして値で表現される。プレイヤーはコマンドを選択し政策を実行することで市のパラメータを変化させ、市の発展とそれによる人口減少の解決を目指していく。そうしてプレイヤーに簡易的な政治体験をさせることが、このゲームのシリアスゲームとしての目的に寄与している。加えて、主に3つの要素がシリアスゲームとしての側面を強めている。1つ目は追加コマンドである。ゲーム内の時間に合わせてその年を象徴する追加コマンドが出現することがあり、追加コマンドは新聞の内容から自然言語処理によって生成され、プレイヤーはそのコマンドから当時の価値観を知ることができる。2つ目はゲーム内で発生するイベントである。イベントはランダムなタイミングで発生し、そのときのゲーム内の年に実際に起こった出来事が起きるイベントを実装している。イベントでは新聞データから自然言語処理により抽出した本文 Group Report of 2023 SISP-24 Group Number 19 Newspaper Big Data for the Future と、本文から生成された要約が表示される。これによりプレイヤーは歴史的な出来事や市民の声を知ることができる。3つ目はリアリティのある政治体験が可能なことである。例えば、人口変動の計算式は厚生労働省^{*3}が公開している、簡易生命表から参照した年齢別死亡率を利用している。そのため、人口変動にある程度の現実味を持たせている。また、コマンドで変動するパラメータやその変動値なども、リアルな変動を目指し設定されている。4つ目の要素は、ミニゲームとして実装されているタワーディフェンスである。タワーディフェンスはシミュレーションゲーム内でのイベントとして実装されている。しかし、タワーディフェンス自体が1つのシリアスゲームとして作られており、プレイすることで歴史的な出来事や、そのときの価値観を知ることができるゲームとなっている。よって、タワーディフェンスは「はこじゃな」において、ゲームが単調にならないためのミニゲームとしての役割と、シミュレーションゲームとは違う形で学びを得ることができるゲームとしての役割を果たしている。以上の要素によりこのゲームは、プレイヤーが過去と現在の価値観の違いを理解し、歴史的な変遷を感じることができるようなシリアスゲームとなった。

(※文責: 杉山孝太)

^{*3} 令和3年簡易生命表の概況. <https://www.mhlw.go.jp/toukei/saikin/hw/life/life21/index.html>

第 4 章 技術的挑戦・背景技術

ここでは自然言語処理・自動生成班がおこなった技術的挑戦や背景技術について説明していく。そもそも自然言語処理・自動生成班がおこなった自然言語処理とは、人間と同じようにテキストや音声言語を理解できる機能をコンピューターに与えることに関連したコンピューター・サイエンスの一分野（さらに具体的に言うなら、人工知能つまり AI の一分野）のことを指す [7]。IBM によると [7]、自然言語処理はコンピューター言語を統計モデル、機械学習モデル、およびディープ・ラーニング・モデルと組み合わせ、これらの技術をまとめて使うことで、コンピューターはテキストや音声データなどの形式の人間の言語を処理し、その意味を発話者または執筆者の意図や感情とともに完全に「理解」する。このことから、私たちは大量の新聞のデータを処理し、ゲームで使用できる形に整形するために自然言語処理をおこなう必要があると考えた。自然言語処理・自動生成班がおこなった技術的挑戦や背景技術は「頻出語の抽出」、「Word2Vec」、「TF-IDF」、「文章要約」、「読み込み・書き出し・整形」の 5 つだ。これら 5 つの技術的挑戦や背景技術は基本的に自然言語処理の基礎技術の 1 つである形態素解析を利用している。形態素解析とは、文章を「形態素」と呼ばれる、意味を持つ最小単位に分解する手法を指す。例えば、「私は猫が好きである」という文章であれば、「私/は/猫/が/好き/で/ある/」というように分解される。この技術を用いることで単語同士の前後の意味関係が分かるようになる。また、休暇に分析した、過去のプロジェクト学習にて作られたソースコードの中には、この技術を活用して、文章の要約をおこなっていたものがあった。なお、以下で説明する「文章要約」については、過去のプロジェクト学習にて作られたソースコードと同じように、形態素解析を使って要約しているが、全く別の形態素解析エンジンを使用している。

以下では「頻出語の抽出」、「Word2Vec」、「TF-IDF」、「文章要約」、「読み込み・書き出し・整形」の 5 つについて、それぞれ述べていく。

（※文責: 中村智椰）

4.1 新聞記事からの頻出語の抽出技術

この節では自然言語処理・自動生成班がおこなった技術的挑戦や背景技術のうち、「頻出語の抽出」について述べる。頻出語の抽出で使用した技術は、プログラミング言語の 1 つである「Python」と形態素解析エンジンの 1 つである「MeCab」だ。「MeCab」は 京都大学情報学研究科 - 日本電信電話株式会社コミュニケーション科学基礎研究所 共同研究ユニットプロジェクトを通じて開発されたオープンソース 形態素解析エンジンだ。この「MeCab」を用いることで、上で挙げた「私は猫が好きである」という文章は、「私/は/猫/が/好き/で/ある/」と分解されるだけでなく、図 4.1 のように書字形、発音形、語彙素読み、語彙素、品詞、活用型、語形、書字形基本形、語種なども分かるようになる。図 4.1 の結果によれば、「私は猫が好きである」は「私」、「は」、「猫」、「が」、「好き」、「だ」、「ある」という単語により構成されていることが分かる。このように「MeCab」を用いることにより、活用が複雑な日本語でも、活用形の違う同じ単語を同じ単語として扱うことができ、日本語における形態素解析の精度を向上させることができる。

書字形	発音形	語彙素読み	語彙素	品詞	活用型	活用形	語形	書字形基本形	語種
私	ワタクシ	ワタクシ	私-代名詞	代名詞			ワタクシ	私	和
は	ワ	ハ	は	助詞-係助詞			ワ	は	和
猫	ネコ	ネコ	猫	名詞-普通名詞-一般			ネコ	猫	和
が	ガ	ガ	が	助詞-格助詞			ガ	が	和
好き	スキ	スキ	好き	形状詞-一般			スキ	好き	和
で	デ	ダ	だ	助動詞	助動詞-ダ	連用形-一般	デ	だ	和
ある	アル	アル	有る	動詞-非自立可能	五段-う行	終止形-一般	アル	ある	和

図 4.1 「私は猫が好きである」を MeCab を用いて形態素解析した結果

「MeCab」を使うにあたって使用した辞書は「mecab-unidic-NEologd」という辞書である。この辞書は UniDic に多数の Web 上の言語資源から得た新語や固有表現、絵文字などのエンTRIESを足して作られた MeCab のシステム辞書である [8]。「UniDic」とは、国立国語研究所の規定した斉一な言語単位と、階層的見出し構造に基づく電子化辞書の「設計方針および、その実装としてのリレーショナルデータベース」、「UniDic データベースと、そのデータベースからエクスポートされた短単位をエンTRIES (見出し語) とする、形態素解析器 MeCab 用の解析用辞書」、「解析用 UniDic」の総称である [8]。Toshinori Sato[9]によると、この辞書の特徴として、「UniDic には含まれていない固有表現などの語の表層 (表記) とフリガナの組を約 338 万組 (重複エンTRIESを含む) 採録している」、「辞書の更新が開発サーバ上で自動的におこなわれる」、「Web 上の言語資源を活用しているので、更新時に新しい固有表現を採録できる」ことなどが挙げられる。現在、この「mecab-unidic-NEologd」に使用されている資源は、「はてなキーワードのダンプデータ」、「郵便番号データダウンロード」、「日本全国駅名一覧のコーナー」、「人名 (姓/名) エンTRIESデータ」、「Unicode 10.0/Emoji 5.0 以下の絵文字に手作業で読み仮名を付与したデータ」、「Web からクロールした大量の文書データ」である [9]。

頻出語の抽出では、まず、新聞の記事データの csv ファイルを 1 年ごとに読み込み、日付、本文などの項目ごとにリストに保存していった。そのうち、本文部分に当たるデータを保存したリストに対して、「MeCab」を用いて形態素解析をおこなうことで新聞の記事を品詞ごとに分けた単語リストを作成した。その単語リストを日付ごとに分けたのち、新聞の記事データの単語リストから名詞のみを取り出し、それらをまとめた名詞リストを作成した。そして、名詞リストに載っている単語の出現回数を 1 日ごとに数え、1 日の中で出現回数が多い順に並べ替えた。最後に並べ替えた後の名詞リストを txt ファイルや json ファイルとして書きだした (図 4.2)。これを 30 年以上のデータに対して適用した。

```

{
  "title": "FrequentWord",
  "notes": [
    {"date": "1988_07_12-07-01", "word": "北海道", "count": 32},
    {"date": "1988_07_12-07-01", "word": "苫小牧", "count": 24},
    {"date": "1988_07_12-07-01", "word": "ところ", "count": 19},
    {"date": "1988_07_12-07-01", "word": "センター", "count": 19},
    {"date": "1988_07_12-07-01", "word": "ソウル", "count": 13},
    {"date": "1988_07_12-07-01", "word": "ながら", "count": 12},
    {"date": "1988_07_12-07-01", "word": "サービス", "count": 11},
    {"date": "1988_07_12-07-01", "word": "いずれ", "count": 10},
    {"date": "1988_07_12-07-01", "word": "ホテル", "count": 10},
    {"date": "1988_07_12-07-01", "word": "アジア", "count": 9},
    {"date": "1988_07_12-07-01", "word": "だろう", "count": 9},
    {"date": "1988_07_12-07-01", "word": "モスクワ", "count": 9},
    {"date": "1988_07_12-07-01", "word": "コンテナ", "count": 9},
    {"date": "1988_07_12-07-01", "word": "オーストラリア", "count": 8},
    {"date": "1988_07_12-07-01", "word": "カンボジア", "count": 8},
    {"date": "1988_07_12-07-01", "word": "イデオロギー", "count": 8},
    {"date": "1988_07_12-07-01", "word": "北朝鮮", "count": 8},
    {"date": "1988_07_12-07-01", "word": "見通し", "count": 7},
    {"date": "1988_07_12-07-01", "word": "それぞれ", "count": 7},
    {"date": "1988_07_12-07-01", "word": "行革審", "count": 7},
    {"date": "1988_07_12-07-01", "word": "グロムイコ", "count": 7},
    {"date": "1988_07_12-07-01", "word": "システム", "count": 7},
    {"date": "1988_07_12-07-01", "word": "スタート", "count": 6},
    {"date": "1988_07_12-07-01", "word": "シーリング", "count": 6},
    {"date": "1988_07_12-07-01", "word": "グループ", "count": 6},
    {"date": "1988_07_12-07-01", "word": "ライン", "count": 6},
    {"date": "1988_07_12-07-01", "word": "水揚げ", "count": 6},
    {"date": "1988_07_12-07-01", "word": "ローカル", "count": 6},
    {"date": "1988_07_12-07-01", "word": "われわれ", "count": 6},
    {"date": "1988_07_12-07-01", "word": "PLO", "count": 6},
    {"date": "1988_07_12-07-01", "word": "ゴルバチョフ", "count": 6},
    {"date": "1988_07_12-07-01", "word": "線引き", "count": 6},
    {"date": "1988_07_12-07-01", "word": "ルール", "count": 5},
    {"date": "1988_07_12-07-01", "word": "ばかり", "count": 5},
    {"date": "1988_07_12-07-01", "word": "レベル", "count": 5},
    {"date": "1988_07_12-07-01", "word": "ベレストロイカ", "count": 5},
    {"date": "1988_07_12-07-01", "word": "歯止め", "count": 5},
    {"date": "1988_07_12-07-01", "word": "はじめ", "count": 5},
    {"date": "1988_07_12-07-01", "word": "テレビ", "count": 5},
    {"date": "1988_07_12-07-01", "word": "振り込み", "count": 5},
  ]
}

```

図 4.2 1988 年 7 月 1 日の頻出語の一部

(※文責: 中村智椰)

4.2 Word2Vec を用いた演算の活用

この節では自然言語処理・自動生成班がおこなった技術的挑戦や背景技術のうち、「Word2Vec」について述べる。

まず、「Word2Vec」とは自然言語処理の手法の 1 つであり、単語をベクトル空間で表す手法のことである。ここで表されたベクトルは「単語ベクトル」と呼ばれる。自然言語処理の手法の 1 つである「Word2Vec」は Tomas Mikolov、Kai Chen、Greg Corrado、Jeffrey Dean ら 4 人が開発したモデル・アーキテクチャで構成されている (図 4.3)。Tomas ら [10] によると、このモデル・

アーキテクチャは文脈に基づいて現在の単語を予測する「CBOW アーキテクチャ」と、現在の単語から周囲の単語を予測する「Skip-gram アーキテクチャ」の2つのアーキテクチャから作られている。「Word2Vec」では、これら2つのシンプルなモデル・アーキテクチャを用いることで、一般的なニューラルネットワークモデル（フィードフォワード型とリカレント型の双方）に比べ高品質の単語ベクトルを学習でき、さらに、計算複雑度がはるかに低いため、はるかに大きなデータセットから非常に正確な高次元単語ベクトルを計算することが可能である [10]。ここで例を示す。単語「王様」は、単語「男」が単語「女」に似ているのと同じ意味で、単語「女王」に似ている。つまり「単語『男』が単語『女』に似ているのと同じ意味で単語『王様』と似ているものは何ですか？」という質問に対し、単語「女王」だと答えることができる。これは単純に $X = \text{vector}(\text{“王様”}) - \text{vector}(\text{“男”}) + \text{vector}(\text{“女”})$ として表すことができる。その後、 X に最も近い単語をベクトル空間で検索すれば単語「女王」という答えが出てくる。ほかにも、大量のデータに対して高次元の単語ベクトルを学習させると、その結果得られるベクトルを用いることによって、都市とその都市が属する国、例えば、パリはフランスに存在し、ベルリンはドイツに存在するというように、単語間の非常に微妙な意味的關係にあるものを答えることができる [10]。「Word2Vec」を用いれば以上のようなことをすることができる。このように「Word2Vec」を用いれば単語に関するあらゆる処理ができるようになることから、単語間の距離や単語間の関連度合いを調べる手法に「Word2Vec」を選択した。この「Word2Vec」を使うにあたり、ニューラルネットワークを使い自分たちでモデルを作るのは非常に大変だったため、「Gensim」を用いた。「Gensim」とは、ドキュメントを意味ごとのベクトルとしてできるだけコンピュータのように効率的に、かつ人間にとってわかりやすく表現するための無料のオープンソース Python ライブラリのことである [11]。

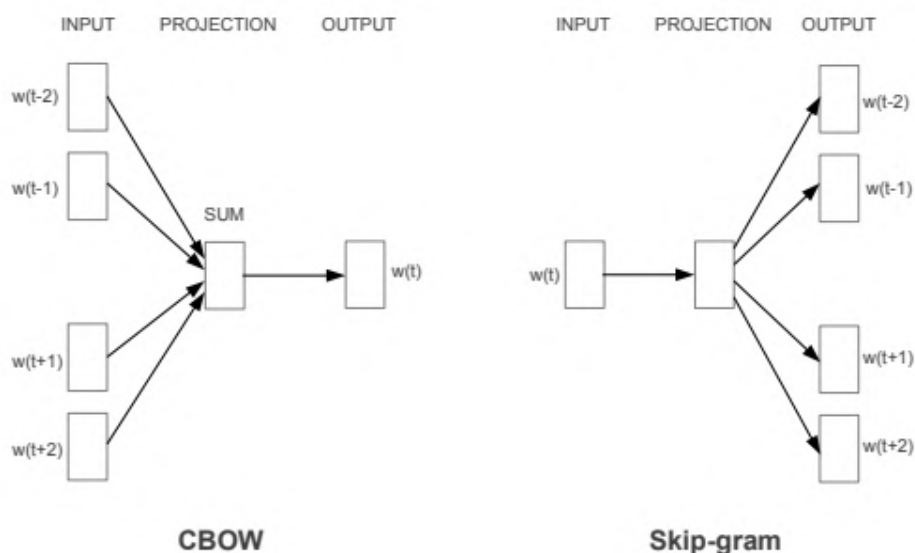


図 4.3 出典：Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean (2013) p.5

「Word2Vec」を用いた単語間の距離や単語間の関連度合いの計算のために、頻出語の抽出で取り出した txt ファイルを単語リストとして読み込んだ。その単語リストとシミュレーションゲーム班から指定されていた「支持率」、「教育」、「産業」、「犯罪率」、「出生率」、「死亡率」の6つのゲームと関係する単語との距離を「Word2Vec」を用いて調べ、調べられたものは（'頻出単語', '6つのゲームと関係する単語', 距離）という形でリストに保存した（図 4.4）。この過程で調べられなかったものが出てきたが、その要因として頻出単語と6つのゲームと関係する単語との間が離れすぎて

いたことなどが挙げられる。

```
('北海道', '支持率', 0.0017610972)
('北海道', '教育', 0.14573123)
('北海道', '産業', 0.29708874)
('北海道', '出生率', 0.04759881)
('北海道', '死亡率', 0.030247176)
('苫小牧', '支持率', -0.111645654)
('苫小牧', '教育', 0.043752585)
('苫小牧', '産業', 0.29441312)
('苫小牧', '出生率', -0.02925786)
('苫小牧', '死亡率', -0.13805802)
('ところ', '支持率', 0.19778489)
('ところ', '教育', 0.033170737)
('ところ', '産業', 0.02768145)
('ところ', '出生率', 0.028727543)
('ところ', '死亡率', 0.1816131)
('センター', '支持率', -0.020048503)
('センター', '教育', 0.2607726)
('センター', '産業', 0.37560427)
('センター', '出生率', 0.004894373)
('センター', '死亡率', -0.035036497)
('ソウル', '支持率', 0.06643376)
('ソウル', '教育', 0.15424794)
('ソウル', '産業', 0.12926851)
('ソウル', '出生率', 0.03588058)
('ソウル', '死亡率', -0.11067675)
('ながら', '支持率', 0.1465227)
('ながら', '教育', 0.054939397)
('ながら', '産業', 0.024284039)
('ながら', '出生率', 0.020864928)
('ながら', '死亡率', 0.018913807)
('サービス', '支持率', 0.06455127)
('サービス', '教育', 0.29483184)
('サービス', '産業', 0.19866544)
('サービス', '出生率', 0.13697729)
('サービス', '死亡率', 0.079835735)
('いずれ', '支持率', 0.025317857)
('いずれ', '教育', -0.1535068)
('いずれ', '産業', -0.07136785)
('いずれ', '出生率', -0.064508975)
('いずれ', '死亡率', 0.0050817938)
```

図 4.4 1988 年 7 月 1 日の頻出語と 6 つのゲームと関係する単語のベクトル距離

(※文責: 中村智椰)

4.3 TF-IDF を活用した新聞記事の重要語判定

この節では自然言語処理・自動生成班がおこなった技術的挑戦や背景技術のうち、「TF-IDF」について述べる。

TF-IDF とは、Term Frequency – Inverse Document Frequency の略である。文章中に含まれる単語の重要度を評価する手法の 1 つであり、ある文書の特徴付ける重要な単語を抽出したいとき

に有効な手法である。TF-IDF は TF と IDF の 2 つの指標を掛け合わせた値になる。TF は、ある文書における単語の出現頻度を表しており、IDF はある単語の逆文書頻度を表している。図 4.5 は TF-IDF と TF, IDF の関係性の概略を表した図である。



図 4.5 TF-IDF 値のイメージ図

TF とは Term Frequency の略である。先に述べたとおり、TF は、ある文書における単語の出現頻度を表している。ここで、複数の文書 T_1, T_2, \dots, T_n があるとする。このとき、文書 $T_i (i=1, 2, \dots, n)$ の中で全ての単語数の和を W_{Si} 、TF 値を求めたい一つの単語の出現回数を W_i 、その単語の TF 値を TF_i とすると、 TF_i は以下の計算式で計算することが出来る。

$$TF_i = \frac{W_i}{W_{Si}} \quad (4.1)$$

TF 値を用いることで、その文章内で頻出している単語が、どの程度頻出しているかということの数値で表すことが出来る。しかし、TF 値だけでその単語が文章内において重要な単語であるということには問題がある。それは、一般的な単語の TF 値が大きくなってしまいう問題である。実際に、今回使用した記事データに対して TF 値を求める処理をした際、「北海道」のようなその記事の特徴を表しているとはいえない単語が高い値となっていた。こうした問題を解決するために IDF 値も計算する必要がある。

IDF とは Inverse Document Frequency の略である。IDF はある単語の逆文書頻出度を表している。ここで、複数の文書 T_1, T_2, \dots, T_n があるとする。このとき、文書数を n 、各文書内で、IDF 値を求めたい単語が登場した回数を W_a 、その単語の IDF 値を IDF_i とすると、 IDF_i は以下の計算式で計算することが出来る。

$$IDF_i = \log \frac{W_a}{n} \quad (4.2)$$

IDF 値が高くなるのはどういったときかということ、複数の文書が存在する中で、ある一つの文書内にもみ出現する単語があった場合、その IDF 値は高くなる。例えば、100 個の文書が与えられたとき、ある 1 つの文書内でのみ「新聞」という単語が使われていたとするのならば、その文書において「新聞」という単語の IDF 値が高く計算される。

ここで、TF-IDF 値の計算における具体的な例を示す。例えば

1. 私は猫が好きだ。
2. 私は犬が好きだ。
3. あなたは猫と犬が好きで、猫を飼っている。
4. 私は馬を飼っている。

という4つの文書について考える。まずこれらの文書を全て形態素解析する。次に、名詞に注目して処理をおこなうために、文書から名詞だけを抽出すると以下ようになる。

1. 私 猫
2. 私 犬
3. あなた 猫 犬 猫
4. 私 馬

各文書に対してTF値を計算する。1つ目の文書は「私」と「猫」の2種類の単語があり、それぞれ1回ずつ出現しているためそれぞれのTF値は $1/2$ となる。3つ目の文書は「あなた」、「猫」、「犬」の3種類の単語があり、出現回数はそれぞれ1、2、1回となっている。従ってTF値はそれぞれ $1/4$ 、 $2/4=1/2$ 、 $1/4$ となる。その他の文書におけるTF値をまとめた表を以下で示す。

表 4.1 TF 値

	私	猫	犬	あなた	馬
1	$1/2$	$1/2$	0	0	0
2	$1/3$	$1/3$	$1/3$	0	0
3	0	$1/2$	$1/4$	$1/2$	0
4	$1/2$	0	0	0	$1/2$

次に各単語に対してIDF値を計算する。例えば「私」という単語は4つの文書の内、3度登場しているため、 $\log(4/3)$ 。おおよそ0.26という小さい値となる。それに対して「あなた」という単語は4つの文書の内、1度しか登場していないため、 $\log(4/1)$ 。おおよそ1.4という比較的大きな値となる。その他の単語のIDF値をまとめた表を以下に示す。ただし、それぞれの値は小数第3桁目を四捨五入した値である。

表 4.2 IDF 値

私	0.29
猫	0.29
犬	0.69
あなた	1.39
馬	1.39

こうして算出されたTF値とIDF値を掛け合わせていくと表のような結果が得られる。

表 4.3 TF-IDF 値

	私	猫	犬	あなた	馬
1	0.15	0.15	0	0	0
2	0.01	0.01	0.23	0	0
3	0	0.15	0.17	0.70	0
4	0.15	0	0	0	0.70

この表から3つ目の文書においては「あなた」という単語が、4つ目の文章においては「馬」という単語が他の単語と比べて重要度が高いと評価することが出来る。

このように TF-IDF を用いることで、ある文書において特定の単語がどの程度重要であるかを数値として表現することができた。このようにして表現した値を、政治シミュレーションゲームでの追加コマンドとして表示する単語の選定に活かす工夫ができると結論づけた。実際に、32 年分の使用可能な新聞記事データの全ての単語に対して TF-IDF 値を計算する処理をした。

(※文責: 永井清盛)

4.4 自動文章要約技術の利用

この節では、自然言語処理・自動生成班がおこなった技術的挑戦や背景のうち、「文章要約」について述べる。文章を自動で要約する流れとしては、まず新聞記事の本文を入力とした文章の形態素解析をおこない、単語単位に分割した。その後、文章中の重要な文を抜き出し、その文をつなぎ合わせることで要約文とした。

この文章要約のプログラムには、プログラミング言語の「Python」を使用した。また、文章の形態素解析には「Janome」の「Tokenizer」というライブラリを使用した。また、文章の重要文の抽出には、「Janome」の「LexRankSummarizer」というライブラリを使用した。Tokenizer による形態素解析は、同じ形態素解析ツールの「MeCab」による形態素解析よりも速度は劣るが、今回は重要文の抽出にも Janome のライブラリを使用していることから、Janome での形態素解析が手法として適切と判断した。

出力された要約文は、原文の 3 割程度になるように出力されているが、元の文章が十分に長くない場合、要約文が出力される際に何も出力されなくなる。そのため、要約文の出力がなかった場合は、要約する必要がないと判断し、本文をそのまま出力するようにした。

このようにして出力された要約文は、実際の新聞データから、本文、見出し、日付とともに紐づけし、CSV ファイルに出力した。

(※文責: 佐久間駿)

4.5 データの読み込み・書き出し・整形

この節では、新聞のビッグデータの読み込み手法、データの整形手法、データの書き出し形式について述べる。

今回の技術的挑戦および背景技術は、すべてプログラミング言語の「Python」を使用している。新聞のビッグデータの読み込みには、Python のデータ解析ライブラリである「pandas」を使用した。新聞のビッグデータが CSV 形式のファイルであったため、pandas を用いてデータフレームとして読み込みをおこなった。こうして作成されたデータフレームを活用して、各データを作成した。

データの整形、書き出しについては作成したデータごとに違った形式をとっている。頻出語のデータと、頻出語と指定単語間のベクトル距離を算出したデータは、json 形式で出力している。json 形式である理由は、Unity でゲームにデータを取り込む際に、json 形式であると容易に取り込むことが出来るためである。一方、文章要約のデータは CSV 形式で出力している。理由としては、もともとは json 形式で出力する予定であったが、実際に出力した際に、データが正常に出力されず、試行錯誤をしてもこの状況が改善されなかった。しかし、CSV 形式で出力した場合、データが正常に出力されたため、プログラムで出力する際は CSV 形式で出力し、その後 CSV 形式の

Newspaper Big Data for the Future

ファイルを json 形式のファイルに変換するツールを使用することで、文章要約データの json ファイルを生成している。

(※文責: 佐久間駿)

第5章 まとめ・振り返り

この章では前章までに述べたことのまとめを述べたあと、8人のプロジェクトメンバそれぞれが振り返りを述べていく。そして最後にプロジェクト全体の振り返りを述べる。

(※文責: 中村智椰)

5.1 まとめ

本プロジェクトでは、若者の政治への関心が低いという現状を解決するために、政治シミュレーションゲームとタワーディフェンスゲームの2つのシリアスゲームを作成した。政治シミュレーションゲームは、史実をもとに市政運営をおこない、人口減少を食い止めることを目的としたゲームとなっている。プレイヤーは函館市の市長となり、政策コマンドやイベントを通してパラメータを変動させていく。タワーディフェンスゲームは、ゲーム内の時間と同期したある1年分の新聞記事の頻出単語から防衛対象を守り切るゲームである。その年にどういったことが新聞で多く取り上げられたかを予想させることにより、社会情勢に関心を持たせることができる。これら2つのゲームは新聞記事のデータを活用している。自然言語処理による新聞記事の要約、頻出単語の取り出し、単語のベクトル化による単語同士の類似度の計算などの技術を用いて、ゲームに使えるデータを作成した。実際に、プロジェクトメンバでこのゲームをプレイした結果、メンバの一部が過去の政治への興味を持つようになった。シミュレーションゲームで提示された要約文から興味を持ち、新聞記事データを実際に見て理解を深めるといった行動変容も現れた。

(※文責: 永井清盛)

5.2 個人の振り返り

5.2.1 永井清盛の振り返り

プロジェクト全体としては、前期の段階でとても雰囲気の良いチームとして活動できた。対面でのコミュニケーションでも、全員が積極的に意見を出し合い、自分たちのアイデアをプロジェクトに反映させることができた。その点では非常に良い活動であった。ただし、チーム全体の意識を一致させることに関しては反省点があった。意見を述べるだけでなく、それをしっかりとまとめて共通の理解に導く必要があると感じていた。後期の活動においてはこの反省点を活かしながら活動することが出来た。特に、成果発表会にむけての準備においては、全てのメンバが意見交流をし、役割分担を的確におこなうことができた。

また、自然言語処理・自動生成班のメンバとしては、主にシミュレーションゲーム班とタワーディフェンスゲーム班との間に立ち、記事データからどういったデータを抽出し、どのような形式にして渡すかといった要件定義を中心におこなった。プロジェクトで使用しているマシンスペックの高いPCが1台しかないため、実質的に処理をおこなうことが出来る人間が一人しかおらず、作業効率が悪いという課題があった。この課題のボトルネックになっていた問題が、「PCが1台しか無い」ということではなく、「両ゲームで使用したいデータの形式がバラバラだったり、しばし

ば形式に変更が起きる」という問題であることに作業途中で気づくことができた。この問題に気づいてから実際に、ほしいデータ形式をしっかりと定義して、ゲーム側でテストデータで読み込みを試すという処理をおこなうことにより、やるべきタスクを適切に把握することができた。結果として1台しか無いPCでも、大規模なデータを処理しゲーム用データに整形する処理を余裕を持っておこなうことができた。

プロジェクトリーダーとしては、メンバ全員とコミュニケーションを取り、意見を集約し、それを議論に反映させることに力を入れた。特定の人物の意見に偏りすぎないように、ディスカッションなどの途中でも、他のメンバに話を振るなどしてより多くのメンバの意見が反映されるよう心がけた。前期の活動で見つかった課題の1つでもあったタスク・スケジュール管理に関しては、Google Calendarによるタスク・スケジュール管理をおこなった。これにより、成果発表や成果物の締め切りといった大きな目標だけではなく、ゲーム用データの整形やデモプレイ用のゲームの設計といった細かいタスクのスケジュールを共有することも出来た。チームワークにおいてリーダーを務めたことはしばしばあったが、8人という規模でおよそ1年間取り組んだことはなかったため、非常に貴重な経験になった。人数が増えれば作業効率が人数分だけ上がるということがないことを改めて実感するとともに、グループをまとめ上げることの大変さを痛感した。しかし、メンバや担当教員の助けもあったおかげで1年間活動をやり遂げることが出来たと考えている。メンバと担当教員に感謝しながら、今回の経験を今後の様々な活動に活かしていきたい。

(※文責: 永井清盛)

5.2.2 木村星凱の振り返り

5.2.2.1 前期の振り返り

前期の活動として、このプロジェクトではどのようなテーマにするかを話し合い、アイデアを出し合った。自分たちの開発したいというモチベーションや成果物をゲームとすることへのおもしろさからゲームの開発を行うことになった。まず初めに、新聞の要素を取り入れるために新聞の読む技術や活用する技術に関する書籍を読んだ。また、様々なゲームのジャンルの特徴的なゲームシステムを見つけ、新聞と掛け合わせることができないか考えた。次に、それぞれが調べ上げてきたものを共有し、再度成果物を何にするかのアイデアだしをおこない、シミュレーションゲームとなった。成果物が定まったところで、私はどのようなゲームの開発環境があるか調べた。調べてきた各ゲームエンジンの長所・短所、特徴をまとめて、グループ内で共有した。その後、決定した成果物に適したゲーム開発環境はどんなものが良いか話し合った。話し合いの中で、UnityとCocos2d-xが候補に挙がっていたため、両方を実際に使用してみて、使用感を調べることにした。その結果、ゲームの開発環境をUnityに決定し、Unityの勉強をおこなった。学習内容としては、Webサイトに公開されているゲーム作成手順をみながら、Unityの基本操作を学んだ。また、GitHubを利用した共同開発の環境構築や使い方を学んだ。しかし、GitHubを利用した共同開発において理解が不十分であり、十分な活用ができなかった。それから中間発表の準備として、デモ用のゲームの開発をおこなった。私は、各メンバが作った機能が正常に動作するように結合作業をおこなった。しかし、計画が甘く、完成がぎりぎりになってしまった。原因の一つとして、一人で悩む時間が多く、無駄な時間が発生したことがあると考える。講義での活動時間だけでは、間に合いそうになかったため、複数のメンバで集まり、作業した。グループ間で、おこなっている活動も全く異なっていたため、グループごとに作業していたが、別班内で進捗状況を確認することになり、その際他グループのメンバが協力してくれアドバイスしあったことで、イメージが定まりその後の作業を進

めることができた。このような原因もあり、計画が急務となってしまった。その結果、他班の意見をゲームに取り入れること、全体でデモの準備をすることができなかった。そのため、後期では先を見通して計画を立て、余裕を持った活動をしたい。

5.2.2.2 後期の振り返り

夏季休暇中の活動について、プロジェクト内で定めた夏季休暇中の課題を解決するために Unity に関する知識を深めた。具体的には Unity を使った 2D ゲームの開発入門の書籍や Unity に関する情報の載ったさまざまな Web サイトをみながら、勉強した。Unity 上でファイルの読み取りをおこなう処理やオブジェクトを表示・非表示する処理は最終成果物にも用いられているため、夏季休暇中の活動は、とても役に立ったと思う。その後、後期の活動が始まり、政治シミュレーション班として活動することになった。グループ体制が変わり、ゲームデザイン班が加わった。ゲームバランスの調整や、作成したゲームをプレイするなど、フィードバックを行うことができるようになった。このようなグループ体制になったことで、より速い周期で修正、確認、改善をおこなうことができるようになった。ゲームを開発する作業では、主に、ゲームデザイン班が考えたパラメータの変化やコマンドの処理機能を実装した。実装した後は、ゲームデザイン班によって確認が行われ、フィードバックを行い、コマンドが与えるパラメータの変化量を調整した。自然言語処理班が作成したデータをゲームに反映できるようにデータの受け渡しができるような機能を実装した。Unity 内でそのデータが読み込まれるようにデータの整形をおこなった。新聞記事要約データでは、年度ごとの新聞記事要約データから複数の要素を取り出し、1988 年から 2020 年の新聞記事要約データが一つのファイル内に納められたデータに整形した。パラメータと頻出語間のベクトル距離データでは、コマンド名として妥当な単語をデータから取り出し、その単語に対応したパラメータと頻出語間のベクトル距離を合体させた。活動の中で反省点がいくつかある。1 つは、ゲームに、どのような新聞から得られるデータを使用するか定められていなかったため、自然言語処理班の活動が遅れてしまったこと。また、自然言語処理班に対しておおよぼな要求をしてしまい、データが出来上がった後に作り直してもらうこともあった。このような問題が起きてしまったのは他グループに対して情報共有を随時していなかったこと、タスクの優先順位を明確にしていなかったことが原因だと思う。逐一、進捗報告するように声掛けしてくれるメンバがいたのにも関わらず、前期に引き続き、あまり他班との情報共有を怠り、共通の認識を持つことができていなかった。これをきっかけに情報共有を怠ることが、異なるグループが円滑に活動をおこなえなくなる重要な要素だと気づかされたため、活動直前に進捗を共有しておいたり、活動最初の時間に、他グループとコミュニケーションをとったりするような習慣をつけるべきだと思った。その中でも、同グループ内でのコミュニケーション、情報共有は前期に比べ、よくできたと思う。後期では Unity プロジェクトの合体方法を学び、同時に作業が可能になったこともあって、各自おこなう作業を共有したり、作業の分担を話し合ったりすることが増えた。作業を行う上で、同じスクリプト、シーンを編集してしまうとコンフリクトが生じてしまうため、事前にルールを確認しながら作業をおこなう必要がある、より密接にコミュニケーションが取れた。ほかにも後期では、スライド作成、ポスター制作をグループで取り掛かれたのが良かったと思う。前期のように個人に投げやりな体制ではなく、全員が協力するような体制でおこなうことができたのは前回の反省が活きたと思う。成果発表に向けた活動では主に、スライド作成、デモ動画の作成に携わった。スライド作成では、政治シミュレーションについてのスライドを担当し、スライドで用いる画像の選定、加工をおこなった。スライドのレイアウトについて、アイデアが思い浮かばず、ほとんど作業に加わることはできなかったが、ガイド線の活用や図式化など他メンバのスライドの作成手順を見ながら、効果的なスライド作成手

法を身に付けることができた。ほかにもスライドについて教員から多くのフィードバックをもらえる機会があり、多くのことを学べた。前後のスライドの内容やキーワードを記憶してもらう工夫、文章やデータのアピール、強調方法について学ぶことができた。デモ動画の作成では、Microsoft Clipchamp という動画編集アプリケーションを用いて動画を作成した。発表準備を進めていくうちに、ゲームの進め方はわかるが、ゲームをする人の思考が思うように伝わらないと思った。そのため、動画にはそれらの要素を含めて作成しようと思った。それらの要素を伝えるためには、どのような表現方法にするかを考えたところ、解説者とプレイヤーの会話を含めたプレイ動画が適切だと考えた。伝えたいことを伝えるため、どんなシナリオにするかを考えた。このゲームの目的の一つとして社会問題に関心をもってもらうことがあり、人口問題について伝えたい思いがあったため、それらのことが伝わる要素を取り入れることを考えた。その動画を作るために、少子高齢化や社会増減に関連する医療、教育、地域経済などの新聞記事を抽出したデータを作った。次に、そのデータを読み込んだゲームをプレイし録画した。録画した動画に、プレイヤー、解説者の会話をテロップとして付けて動画を完成させた。最終的に、他メンバが作った動画にこの動画を合体させ、発表会で流すことになった。このプロジェクト学習を通して、自分自身が気づいたこと・変化したこととして、2つある。1つ目は、モチベーションの維持である。プロジェクト学習のように期間の長いプロジェクトでは、まだ時間があるという余裕が生まれてしまっていた。その結果、前期、後期どちらでも予定通り終わらず、ぎりぎりまで作業する必要があった。そのためにゴールを見据えることで、期限までに終わってなければいけないタスクが明確になり、やるべきことがわかったり、焦りや、やらなければというマインドを高めることが可能だと考えた。2つ目は、地方の人口減少問題についての意識が変わったと思う。活動する前は、人口減少の問題は少子高齢化が原因だろうと考えていたが、活動を通していくうちに、人口減少社会を取り巻く背景・現状に気づき、自然減だけの単純な問題でないことがわかった。このように、社会問題に関心を持つきっかけになった。

(※文責: 木村星凱)

5.2.3 佐久間駿の振り返り

5.2.3.1 前期の活動

前期の活動では、自然言語処理・自動生成班として、まず、自動生成系の AI についての知識の習得に注力した。そして、無料の画像生成 AI である「Stable Diffusion」を利用し、指定したキーワードに基づいた画像の自動生成をおこなうことができた。ただし、キーワードを適切に指定しなければ、本当に欲しい画像とは異なる画像が生成されることがあり、改善の余地を感じた。成果物の決定後は、自然言語処理の手法について調査することにした。まず、「言語処理 100 本ノック」の問題を解き、自然言語処理の基礎知識をつけることに注力した。そして、2018 年の 1 年分記事データの形態素解析をおこなうことができた。ただ、まだ本プロジェクトで作成するゲームのシステムに対して、どのように自然言語処理を活用していくかについては具体的な段階まで来ていない。今後の開発でさらなる知識の習得と応用に努めていきたい。中間発表の準備では、ポスターの作成や内容をメンバの 1 人に負担をかけてしまった。プロジェクトの時間外での活動を、個人の活動だけに当てるのではなく、他のメンバの状況を確認して、協力できる体制を築いていく必要があると思った。

5.2.3.2 休暇期間の活動

休暇期間の活動では、自然言語処理・自動生成班として、同プロジェクトに所属していた過去のメンバが、言語処理に使用していたプログラムのソースコードの解析や実行をおこない、シミュレーションゲームやタワーディフェンスゲームで使用するデータを作成するために活かすことが出来るかどうかを検討した。私はそれぞれ「TF-IDF」、「tanaka_dict」、「judgeNounCorrect」、「hint」、「DataAnalysisMonthsNEologd.ipynb」、「DataAnalysisMonthsNEologd【完全版】.ipynb」と名付けられた6つのファイルを担当した。

これらのソースコードの解析をおこなった結果、使われている技術のことや、前処理の仕方などを知ることが出来た。特に、TF-IDF のコードを解析した際には、まず TF-IDF がどのようなものを調べる所から調査が始まった。そして、これが「文章中に含まれる単語の重要度を評価する手法」であることが分かり、どのように実装すればよいのかを知ることができた。また、「tanaka_dict」や「hint」のソースコードの解析した際には、データの前処理をどの様におこなうと、今後データを作成する際に加工が容易になるかを知ることができた。これらの前処理の仕方は実際にデータを作成した際に活用できた。

中には実行した際にエラーが発生したり、辞書のインストールに時間が掛かり実行するには膨大な時間とメモリを消費するため難しいと判断したものもあるが、新聞のビッグデータに対してどのような技術を扱う事ができ、どのようなデータを作ることができるのかを知ることが、実際に政治シミュレーションゲームとタワーディフェンスゲームに活用できるかどうかを検討する際には大いに役に立った。休暇期間に調査をおこなったからこそ、後期の活動でデータの作成にスムーズに取り掛かることができたのだと思う。

5.2.3.3 後期の活動

後期の活動では、主に新聞記事の自動要約の実装に取り組んだ。この自動要約の実装には、自然言語処理・自動生成班の休暇の調査結果を活かすことができたので、文章の自動要約の実装にはあまり時間はかからなかった。しかし、「本文」、「日付」、「見出し」、「本文の要約文」という4つのカラムを持つデータを作成して欲しいというゲームプログラミング班からの要望があったため、前処理に苦勞する部分が多かったと思う。本文や見出しの抜き出しには休暇の調査結果を活かすことができたものの、日付の抜き出しは自分で手法を考えなくてはいけなかった。また、本文が十分な文字数でなければ文章の要約文が出力されないという問題も発生した。しかし、試行錯誤の結果、日付は正規表現を使って抜き出し、要約文が出力されない問題に関しては本文をそのまま要約文として出力することで解決し、ゲームプログラミング班が求めるデータを作成することができた。

成果発表会の準備では、主にポスター作成に携わった。前期の中間発表の準備ではポスター作成をメンバの1人に任せてしまったという反省から、成果発表会のポスターは、自分を含めてポスター作成の経験がない4人で担当した。ポスター作成未経験ということもあり illustrator を初めて使用したため、使い方やデザインについて学びながらの作成だった。私は主に自然言語処理・自動生成班の内容の部分を担当したが、制作の過程でどうすれば見やすいポスターになるのかや、限られたスペースの中でどのように成果をまとめるかを、他のポスター作成担当のメンバと相談したり、ポスター作成経験のあるメンバにフィードバックを受けながら作成をおこなった。illustrator の操作に慣れなかったり、ポスター作成のノウハウを知らなかったため、ポスター作成に時間が掛かってしまい、結果的には完成したものの、発表練習の時間が取れなかったりなどの問題が発生したため、中間発表のポスター作成の際に協力できていたらもう少し手際よくできていたのかなと思った。

5.2.3.4 まとめ

今回のプロジェクト学習を通して、グループワークを通しての開発の難しさを実感した。8人という今まで経験したグループワークの中でも多い人数での活動だったので、誰か一人に負担をかけてしまう部分があった。それぞれがプロジェクトに貢献できる機会を持ちつつ、負担も平等にするのは難しいと感じた。今後は、メンバ全員の状況やタスクを把握し、平等にするためのコミュニケーションを知る必要があると感じた。

しかし、プロジェクト学習を通して成長を実感した部分も多くある。例えば、私はプロジェクト学習が始まる前は、コミュニケーションをとることに苦手意識を持っており、自分の意見を主張することがあまりできていなかったが、プロジェクトが進むにつれて、メンバとのコミュニケーションに慣れて来たことで自分の意見を主張することの苦手意識がなくなった。そして、プロジェクト以外でのグループワークでも自分の意見を主張することが出来るようになった。他にも、技術力の向上や調べる力など成長した側面は多くあるが、一番大きいのは自分が苦手だと思っていたコミュニケーションを改善できたことだと思う。なので、プロジェクト学習は自分の殻を破る大きなきっかけとなった。

プロジェクト学習を通して得た経験や反省から、プロジェクトで成果物を作成する際に、どのようにコミュニケーションを取れば円滑に作業を進めることができるのかを理解することができた。なので、今後プロジェクトに参加する機会があった時には、コミュニケーション面でプロジェクトに貢献できるように活かしていきたいと思う。

(※文責: 佐久間駿)

5.2.4 鈴木稜司の振り返り

5.2.4.1 前期の活動

前期の活動として、最初におこなったのが成果物案決定のためのブレインストーミングである。各々、やりたいことを出してみたが指導教員たちに着地点が見えて面白くないと言われてしまった。そこで、美馬のゆり教授にアドバイスを貰い、再度ブレインストーミングをおこなった。その結果、今まで出たことがない意見がたくさん出た。このアドバイスは、今後も活かしていきたいと考える。成果物が決定した後、ゲームプログラミング班としてゲームの開発に取り組んだ。ゲーム開発は、初めてだったのでゲームを開発するツール「Unity」の勉強から始めた。Unity 公式の、チュートリアルを使って勉強をした。その後、すぐにゲームの開発に取り組んだ。これが、前期の失敗だった。もっとゲームの勉強や、ゲームデザイン共有をするべきであった。なぜなら、同じ班の人とイメージを共有できず上手く作業が進まなかったからである。その反省を活かし、再度ゲームの勉強に戻った。私は、タワーディフェンスゲームの勉強をしている最中である。以上より、ゲーム デザインの共有が後期の課題として挙げられた。

5.2.4.2 夏期休暇中の活動

私が、夏期休暇中におこなったことはタワーディフェンスゲームの基盤作成である。前期の課題として、タワーディフェンスゲームを知らないという問題があった。それを、解決するためにどのようなゲームなのか、どのような機能があれば面白いのか、どのように作られているのかをゲームを作成しながら学んだ。タワーディフェンスゲームの作成は、「Unity 入門の森」を参考にした。10 個のステップを踏みながら作成していくものだ。Unity の使い方、UI の作成の仕方、プログラムの細かい説明など、とてもわかりやすかったのでタワーディフェンスゲームについて楽しく学ぶ

ことができた。タワーディフェンスゲームの基盤を、夏期休暇中に完成することができたから後期の活動で拡張機能の制作に手をつけることができたと思う。

5.2.4.3 後期の活動

後期の活動では、タワーディフェンスゲーム班としてタワーディフェンスゲームの機能拡張と、ポスター制作に力を入れた。

タワーディフェンスゲームは、夏期休暇中に作ったモノを基盤として機能拡張をおこなった。機能拡張の案として、攻めてくる敵は新聞の頻出単語というのは決定していた。しかし、防衛アイテムの案については新聞とどう組み合わせれば面白くなるのかということで、決定が難航していた。そこで、担当教員である寺沢憲吾教授に相談し「防衛アイテムは、特定の敵に対しダメージがあがる仕様にしてはどうか」というアドバイスももらった。その後、防衛アイテムは敵の頭文字と同じ行のモノを配置することでダメージが上がる仕様に決定した。例えば、「北海道」という敵が攻めてきたら、「は行」の防衛アイテムを置くことで効率よく敵を倒すことが可能だ。これにより、プレイヤーは新聞をたくさん読むことで頻出単語を予想でき、ゲームで良い結果が出せるようになるため、プレイヤーの新聞を読む意欲を高めることができるようなゲームになったと考える。

ポスター制作では、前期ではメンバ1人だけが頑張ってしまうグループワークの意味が無くなってしまっていた。その失敗を活かし、後期では4人で作成をおこなった。ポスター作成は未経験だったので、作成経験があるメンバに Illustrator の使い方を教わりながら作成をした。私は、主にタワーディフェンスゲームの部分を担当した。限られたスペースの中で、ポスターを読む人にタワーディフェンスゲームを理解してもらい、新聞をどのように活かしているかを理解してもらうことを意識して作った。ただ、4人ともポスター制作が未経験ということで、ポスター制作にかなり時間がかかってしまい、発表練習の時間を十分に取ることができなかった。中間発表の時に、1人に任せずメンバで協力していればこのようなことにはならなかったと思うので反省するべきである。

5.2.4.4 まとめ

プロジェクト学習を通して私は、チームで協力する大切さを学んだ。ゲーム制作の時に、わからないところがあった時、メンバに相談し協力することで実装できた。ポスター制作では、経験者に意見を貰うことでより良いポスターができていたと考える。これらは、自分1人だけでできることではなかった。メンバ1人1人が、できることをやり協力し合ったからこそできたことだと考える。プロジェクト学習を通して学んだことを、これからの人生に活かしていきたい。

(※文責: 鈴木稜司)

5.2.5 嵯峨京介の振り返り

5.2.5.1 前期活動について

前期の活動では、私はゲームデザイン班として、主にタワーディフェンスゲームの詳細仕様の検討をおこなうという役割を果たすことができたと感じている。タワーディフェンスゲームの経験が無いことや政治に関する知識が足りないことがゲーム作成の壁となることがあったが、担当教員や班員からアドバイスを貰いながら、より良くしようと活動することができた。しかし、他の班との相談が不十分であったため、後期では積極的に共有するようにしていきたい。また、「このゲームのどの部分で若者に興味を持たせるのか」というゲームの面白さに関わる部分が曖昧になって

しまっており、これでは目標を達成することができない。そのため、後期の活動では、この部分について改善し、さらに良いゲームになるよう努めたい。また、中間発表に向けてのポスター作成に関しては、大部分を班員に任せきりにしてしまったため、反省している。後期はグループの一員として積極的に取り組みたい。発表準備に関しても、聴講者に十分に配慮しきれていなかった部分があった。これについては改善し、最終発表会に活かしたい。以上が前期の活動の振り返りである。

5.2.5.2 夏季休暇中の活動について

夏季休暇中の活動では、明確にノルマ等は設けていなかったものの、意識的にニュースを見たり、歴史的に大きな出来事を調べたりなど時事問題について触れる機会を増やした。また、ゲームプログラミング班が基盤設計を Unity でおこなっていたため、後期活動に備え、C #の基本的な知識の勉強に努めた。そして夏季休暇中の活動で最も力を入れたのは、ゲーム作成の基本的な考え方と作成方法の知識習得である。前期活動中に寺沢憲吾准教授からゲームは「プレイヤーの楽しさ」をどこに置くかが非常に大切であると助言をいただいた。そこで、新聞データを最大限活用するため、ゲーム作成者の動画やサイトなどを見て、ゲーム作成の根本的な部分を学んだ。

5.2.5.3 後期活動について

後期の活動では、まず、プロジェクト全体で進捗報告をおこなった。具体的には夏季休暇中に仮決定したタワーディフェンスゲームの詳細仕様についてメンバと進捗具合の共有である。進捗報告後、タワーディフェンスゲーム班として活動を開始した。タワーディフェンスゲーム班としての活動として主におこなったことは、大きく分けて2つある。

1つ目は、詳細仕様の更なる検討である。後期活動開始時点では、攻めてくる敵を単語にすることは確定していたが、防衛アイテムの案については新聞データを組み込むことが難しく、決定が難航していた。そこで、一度担当教員である寺沢憲吾准教授に助言を求めた。すると、「防衛アイテムを特定の敵に対し与ダメージがあがるものとしてみてはどうか」と助言をいただいた。その助言と攻めてくる敵が単語であることから成果物に実装している頭文字に着目した防衛アイテムの作成を決めることができた。その後、攻めてくる単語に関してもどのような単語を敵として採用するのが良いかを助言をもらいつつ、頻出語に決定することができた。このような活動を通して、一人で考えるだけでなく、客観的な視点や意見をもらうことの重要性、そしてプロジェクトの強みを知ることができたと考える。

2つ目は、ゲーム基盤をもとにした機能拡張である。まずおこなったのは、休暇中にゲームプログラム班に作成してもらったゲーム基盤のコードを読むことである。機能拡張をおこなっていくため、各スクリプトがゲーム画面のどこに対応しているのか、何を動かしている部分なのかを理解する必要があった。その際、C #の知識が足りずに、メソッド等がわからないことが多々あり、適宜調べながらコードを読んでいったため、プログラミングスキルの上達の機会とすることができた。その後、詳細仕様に基づいて拡張機能の実装に着手した。実装については詳細仕様にある機能に応じてどんなメソッドが必要なのかを調べ、最適な方法を模索しながら実装を進めた。これについても、自身のプログラミングスキルの上達の機会となったと感じている。また、実装途中に何度もエラーが出るがあったが、自分で調べることやメンバに相談することなどを積極的におこない、対処法について身についたのではないかと考える。このように、後期の活動では、成果物以上に自身の成長ができる場にすることができた。

5.2.5.4 ブレインストーミング技術

前期の活動期間に担当教員である美馬のゆり教授から講義のような形でブレインストーミングの方法についてご教授いただいた。まずは発散的思考で、考えたいテーマについて思ったことをとにかく多く書き出してみることが必要であることを学んだ。この書き出す段階ではテーマに関係があることだけではなく、直接の関係が無くても、関係がありそうなことも書き出してみることで新しい発想への糸口になり得ることも教えていただいた。また、グループでおこなうブレインストーミングに関しては、誰かが出した意見を否定することなく、まずは受け入れることで、皆が意見を出しやすい環境にすることが大切であるということも教えていただいた。発散した後は、出た情報をカテゴライズして整理するという収束の段階に移っていく。整理した情報から再び発散的思考で新しく考えついたことを書き出してみる、といったようにこれらの手順を繰り返していくことで効果的なブレインストーミングができることを学ぶことができた。美馬のゆり教授の講義で学んだことをもとに、再び新聞データの活用法についてブレインストーミングをおこなった。講義後は明らかに質の高いブレインストーミングができたが、それでもやはり難しかった。ここで学んだブレインストーミングの考え方は、日常生活の様々な部分で通用する。この経験を無駄にせず、これからの人生に活かしていきたい。

5.2.5.5 最終成果発表会

最終発表では、中間発表での反省を活かし、スライド、ポスターとデモに加えて、デモ動画を用いてメンバとともに良い発表をおこなうことができたと感じている。成果物に関しては、「面白い」、「興味深い」など肯定的な意見とこうすればもっと良くなるのではないかとという客観的な意見もいただき、良いものができたのではないかと感じる。また、今回の最終発表では先に述べたようにタワーディフェンスゲームのデモを体験できる形で用意した。発表後は、主にデモ体験の対応をおこない、何人かの方に実際に体験していただくことできた。デモ体験についての良かった点としては、そこで、企業や大学外部の方に体験していただいた際に、面白いというような意見をいくつかいただくことができたことである。それに加え、プロジェクトのようなグループでおこなう活動で培った知識は、今後も大切になるなど助言もいただいた。このように充実したデモ体験の時間とすることができたが、反対に、悪かった点もある。それは単語の予測要素についてはあまり実感してもらえていなかったことである。一人あたりの時間が短いため、じっくり考える時間をとることができなかった。デモ体験で自分たちの伝えたいことを体験してもらうことの難しさを改めて実感したため、この失敗を今後の発表の機会に活かしたい。

(※文責: 嵯峨京介)

5.2.6 中村智椰の振り返り

5.2.6.1 前期の活動の振り返り

前期の活動では、はじめに成果物案決定のためのブレインストーミングをおこなった。この過程で美馬のゆり教授に教示いただいたブレインストーミングの効果的なおこない方を実際におこなった結果、今まで出たことのない意見がたくさん飛び交い、いいブレインストーミングができた。そのため、今後もこの方法を使っていきたい。

成果物が決定した後は、自然言語処理・自動生成班として、イベントを自動生成するための準備に取り組んだ。そこでは、自動生成にはどのような技術が必要なのか、また、機械学習の素材として新聞のデータを使うので、どのように新聞のデータの特徴を活かすのか考えた。そこで、イベン

トを自動生成する方法として Python の MeCab というライブラリを利用することに決まったので、Python の学習をした。後期では実際にイベントの生成に取り組もうと考えた。

中間発表会の前には、ポスターを作ったが、ほとんど一人で作るようになったので、もっと協力を要請する必要があると感じた。後期ではこの点に特に気をつけようと思った。ポスターはポスターセッション用のポスターと、前期におこなった内容をまとめた読む用のサブポスターを作った。ポスターセッション用のポスターはバイリンガルで書く必要があったため、文字数が多くなるのが想定された。そのため、前期のうちにそれぞれがやった内容をしっかり書きながらも、図示などを取り入れることで認知負荷を軽減させようと思力した。前期におこなった内容をまとめた読む用のサブポスターでは、「未来へつなぐ新聞ビッグデータ」という新聞に重きを置いたプロジェクトにちなんで、新聞調のサブポスターにした。新聞調のサブポスターを作るにあたり、たくさんの新聞を読み、新聞の作り、工夫点を学び取り入れることで、より新聞身の強いサブポスターに仕上げた。また、新聞のような文字がたくさん入っているものでは、少しのずれで認知負荷が大きくなる可能性があったため、上下の位置を揃えたり、特に目立たしたいところは認知負荷が大きくなるような色づかいを心がけた。

中間発表会では、ポスターセッションをおこなうのにオーディエンスがしっかり近くで見れるように発表前に前に来るように呼び掛けたり、しっかり聞き取りやすいように、また理解しやすいように大きな声で発表する、ポスターをしっかりと指し示す、発表内容は暗記するのではなく、自分の言葉で伝えるなどの工夫をした。その結果、「大事なところを強調して話せていたのでわかりやすかった」や「話す構成が工夫されていた」などと非常に好評であった。しかし、「スライドのほうが見やすい」、「モニターなどに移した方がいい」などの意見もあったので、成果発表会では、その意見をぜひ参考にしたいと思った。また、新聞調のサブポスターに関しては「新聞をモチーフにしてるのがユニークだった」や「サブポスターの内容、意見が一致している」、「サブポスターが目をはきかつ、見やすい」と非常に好評であった。

5.2.6.2 夏期休暇中の活動の振り返り

夏季休暇中の活動では、自然言語処理・自動生成班の取り組みとして過去のプロジェクト学習にて作成されたソースコードを実行、その内容、結果の把握をおこない、プロジェクトメンバに共有した。過去のプロジェクト学習にて作成されたソースコードから、新聞ビッグデータをどの様に整形しているのか、どのように扱っているのかを学んだ。私が担当したソースコードは「頻出単語」、「DataAnalysis1」、「地域性」、「DataAnalysisMonthDict」、「DataAnalysisMonthsNEologd」、「hoge」と名前が付けられた6つのファイルだ。これらのソースコードを読み、実行することで GoogleColaboratory の使い方や GoogleDrive との連携方法、Python での文字コードの指定方法、ファイルの開き方、書き込み方、ディレクトリの移動の仕方、ダウンロードしたものや作成したファイルの保存場所などを学ぶことができた。また、ライブラリ内関数のアップグレードや開発環境の違いなどにより、多くのエラーが出たが、そのすべての原因を調べ、解決策を提案した。私が担当した6つのファイルのうち「頻出単語」、「地域性」、「DataAnalysisMonthDict」、「DataAnalysisMonthsNEologd」の4つのファイルでは、主に新聞のデータを読み取ったのち、そのデータから名詞の頻出語を取り出し頻出回数が多い順にソートするという処理をおこなっていた。ソートした後は、txt ファイルや json ファイルといった形式で出力していた。「DataAnalysis1」というファイルでは、新聞のデータを読み取ったのち、そのデータを1年ごとに分けて csv ファイルとして出力していた。「hoge」というファイルは、指定した txt ファイルに書いてある文章を要約するものだった。

5.2.6.3 後期の活動の振り返り

後期の活動では、夏期休暇に分析した過去のプロジェクト学習にて作成されたソースコードで学んだものを用いて、シミュレーションゲーム班やタワーディフェンス班などから要望されたデータを形態素解析を用いて作成した。このときにはエレクトロニクス工房に置かせていただいている、おとしプロジェクト学習にて購入した PC を用いてソースコードの作成、実行、データの作成などをおこなった。私がおこなった技術的挑戦は、「頻出語の抽出」、「Word2Vec」、「TF-IDF」、「読み込み・書き出し・整形」だ。私たちのプロジェクトには新聞データが 30 年以上もあるため、データの読み込み・書き出し、リストの作成などを非常に多く利用する必要があり、これらを 1 つのメソッドとしてまとめるために、特に整形の技術を多く利用した。「頻出語の抽出」の技術は、すべての年についておこなうのに 20 分ほどかかり、さらに「TF-IDF」の技術は半年分の頻出語と半年分の記事すべての値を計算するのに 2 時間半ほどかかるなど、非常に大規模な操作をおこなうことになった。また、これにより講義内でのデータの作成数が非常に少なく、課題として非常に多くの時間を費やす必要が生まれてしまった。今後機会があれば、ソースコードの高速化や、コンパイル言語であるため Python よりも実行速度の速い C++ などを用いて簡単に素早く形態素解析をおこなえるようにしていきたい。

成果発表会の前には、発表用のスライド作成と、ポスターの作り方および Adobe Illustrator の使い方をポスター作製をおこなうメンバに教えた。発表用スライド作成では、メンバの変化や提言を述べるスライドの作成とスライドのデザインを担当した。その過程で総務省統計局が出している「人口推計 - 2023 年 (令和 5 年) 10 月報 -」を使い、有権者の年齢ごとの人口グラフの作成、それを踏まえた提言などを述べた。また、発表用スライドのデザインでは担当教員のアドバイスを従い、デザイン 4 原則を特に気をつけ、また、目立たせたいところは周りとは違う色を使いつつ認知負荷が高まりすぎないようにするなど、細かなところにも気を配った。ポスターの作り方および Adobe Illustrator の使い方をポスター作製をおこなうメンバに教えるときには、プロジェクトワーキンググループから配布されている A1 のポスターのテンプレートを用いて、トンボやぬりたしなどの説明のほか、ガイド線の引き方、アウトライン化の仕方などを教えた。また、ポスターで書かれている文章の添削や配置確認などもおこなった。また、最後には文末の文字位置のずれなどの修正もおこない、ポスターの認知負荷を軽減できるように努めた。前期の中間発表の準備のときは協力をもっとしなければならぬと反省したが、成果発表会の準備ではその反省を活かして、みんなで協力して準備をおこなうことができたと思う。

成果発表会では割り振られた場所がミュージアムの入り口で、非常に人も目につきやすい場所だったので、より興味を持ってもらいやすくするためにポスターを外から見える場所に置く、多くの人に来ても見やすいように弧を描くように配置するなど、配置の工夫について考えた。また、予想より多くの人会場におり、非常に声が通りにくい環境だったので、できるだけ伝わりやすいように、大きな声でスライドを指し示しながら発表した。また、1 回目の発表で、「シミュレーションゲームのパラメーターの変化でどのように新聞データを活かしているのか」という質問が多くあったため、2 回目以降の発表では、「頻出語の抽出」と「Word2Vec」を用いているということ、情報系の方でない人にもわかりやすいように説明を加えた。その結果、「言葉の関連性をベクトル距離で求めていると驚いた」という意見があり、しっかり伝えられた。また、スライドに関して、「目的が一貫していてわかりやすかった」、「スライドは要点をまとめられていた」などと非常に好評だった。

5.2.6.4 まとめ

今回のプロジェクト学習で学んだことのまとめとして、グループワークの難しさを感じた。日ごろのコースの授業のグループワークでは、全員がグループワークに慣れており、また同じコースにいる人たちのため、自分たちのあたりまえが簡単に通じてしまう。しかし、プロジェクト学習では全く違うコースの人が集まっており、日ごろの講義でグループワークをたくさんおこなっているわけでもなかった。そのため、「あたりまえ」として連絡を怠ると周りが何をしているのかわからなかったり、経験者だけが物事に取り組んでしまうなどの問題が発生し、円滑にグループワークを進めることが難しくなってしまうと痛感した。今後のグループワークではぜひ積極的にコミュニケーションをとっていききたい。

(※文責: 中村智椰)

5.2.7 杉山孝太の振り返り

前期の活動では、まず新聞で何を作るのかを決めるため、新聞に関する事柄のブレインストーミングをおこなった。ブレインストーミングと成果物案のアイデア出しに時間をかなり使ったが、それにより独特な着眼点からプロジェクトの目標を決めることができた。ブレインストーミング自体に慣れていなかったため最初は上手くできていなかったが、教員の指導のもとおこなっていく過程で、ブレインストーミングの工程に慣れ、幅広く考えを巡らせることができたと考える。ブレインストーミングによって得た経験や技術はアイデアを考える上で有用なので、今後も大切にしていきたい。成果物が決定してからはゲームプログラミング班として活動をした。Unity を触った経験はあったが、ほとんど忘れていたため使い方を一から勉強した。また、開発効率のため複数人での同時開発を目指し、GitHub を利用しての開発方法を学んだ。Unity との相性や、そもそも Git や GitHub を上手く扱う技術の習得に時間がかかるなどの問題があり、前期の最終段階ではツールを用いずに手動で開発データを共有することになった。また、ゲームのイメージが固まっていない段階から開発を始めてしまい、メンバ間のイメージ共有や、ゲームへの理解が不十分だったため開発が滞ってしまうことがあった。また、ゲーム開発の途中経過をメンバに共有することが必要だったが、前期ではこまめな共有ができなかった。ゲームの問題点や改善点などをフィードバックしてもらうために、こまめに途中経過を共有してメンバとコミュニケーションを取る必要があったと思う。結果的に前期ではゲーム開発自体はあまり進まなかった。中間発表のポスター制作では、一部のメンバに負担が集中してしまい、私もポスター制作に積極的に参加していなかったことが一番の反省点だったと感じる。

後期では班分けを変更し、政治シミュレーションゲームの制作をする班で活動した。前期で問題となっていた開発体制は、Google Drive で制作途中のプロジェクトファイルを共有するという方法に決定した。これにより共同の開発者とのファイル共有が問題なくできるようになり、開発体制が安定した。前期では共同開発のためのツールとして真っ先に GitHub を利用しようとしたが、2人での小規模な共同開発だったので結局は GitHub を用いる必要性が大きくなかったと感じた。GitHub を使おうとした経験は無駄ではないが、前期はそのための勉強にかなり時間を使ってしまった。開発に際してツールや手法が適切であるかどうかをまず考えることも大切だと改めて実感した。班内では共有が安定し開発と修正のサイクルを上手く回すことができたと思う。しかし、プロジェクトでの共有は共有方法が明確に決まっておらず、共有形式や頻度もバラバラだったので、他班のメンバにとってはゲーム開発の進捗が分かりづらくなってしまった。また、前半では途中経過の共有の連絡をプロジェクトの時間以外ではしていなかったため、それも進捗を分かりづら

くする要因となった。ゲームの共有方法や頻度を明確に決めていれば、もっと他班のメンバからのフィードバックを受け取ることができ、プロジェクト全体でのゲーム開発が上手くできるのではないかと感じた。

期末発表の準備では主にポスター制作に取り組んだ。ポスターを Illustrator で作成するのは初めての経験だったが、プロジェクト内の経験者に教えてもらいつつ、制作途中にフィードバックを受け取りながらポスターを完成させることができた。前期では負担の集中が反省点だったが、今回は誰かに負担が集中することもなくポスター制作に携わったメンバ全員が積極的に取り組めていて、自分も積極的に参加することができた。反省点としてはデザインへの理解が不十分だったため、制作途中で修正点が多く、進行ペースが遅かったことにより完成が期末発表前日になってしまったことが挙げられる。期末発表では発表箇所を分担して発表をおこなった。前期は参加していないため初めての発表となったが、つつがなく発表できたと思う。しかし、姿勢や発声などは反省点が多く発表後に練習不足を感じた。発表準備も含めて、プロジェクトの成果を不特定多数の人に向けて発表するという貴重な経験ができたので、この経験を卒業研究などに活かしていきたい。

プロジェクトで新聞に触れることによって、新聞の特徴も多く知ることができた。新聞は網羅性が高く、社会で起こった大きな出来事の大抵は新聞を見れば把握できる。また、レイアウトも洗練されており、右上から左下に記事を押さえて流し、直感的に記事の区切りと順番が分かるようになっている。見出しも出来事を簡潔に分かりやすく表現しており、一目見れば社会の全体像を何となく把握できるような一覧性も、新聞というメディアの大きな特徴である。私は、こういった網羅性や一覧性が他のメディア媒体より優れた点だと感じた。新聞に触れたことで、新聞のメディアとしての価値を改めて認識できたと思う。

プロジェクト学習を通して自分自身の変化も大きく感じた。プロジェクトの序盤ではアイデア出しのために積極的に新聞を読むようにしていたが、その後も大学では空いた時間に新聞を読むことが多くなった。プロジェクト学習前は新聞にあまり関心がなく、普段目にするメディアはインターネット記事や SNS が殆どであり、新聞を読む機会は帰省時に時々地方紙を読む程度だった。新聞を読むようになってからは、普段自分の関心がない情報にも触れることができ、日本や世界の情勢を以前よりある程度感じることができるようになった。また、驚くべきことは、以前まで私のメインの情報源であったインターネット記事では全く触れることがなかった情報を知ることができたことである。もちろんインターネット記事でも新聞が取り上げるニュースを扱ってはいるが、一定以上の大きな出来事を除いて、私がインターネット記事では見えていない情報を新聞では得ることができた。その原因は、情報収集のほとんどをインターネットで行うことによる情報の偏りだと感じる。そもそもユーザーが見る情報を取捨選択するインターネットでは、自分の関心のある領域のニュースは能動的に見ることが多いが、逆に関心のない領域ではその記事を見ようとはあまり思わない。また、ニュースアプリなどでは、ユーザーの属性で表示するニュースを変えるターゲティングが有効になっていると、ユーザーの関心がないと判断されたニュースはそもそも表示されなくなってしまう。これによりインターネットでは、自分の興味のある情報だけを見る状態であるフィルターバブルや、自分の意見と同じもしくは似ている意見ばかり触れてその意見を信じ込んでしまうエコーチェンバーといった現象が問題視されている。私もプロジェクト学習前はこういった状態に陥っていたと感じる。特に新聞はその日のトピックについて網羅的に掲載しているなので、ある意味無差別に情報が入ってくる。私は、幅広く情報が集められる新聞の有用性を知ることができ、特に関心のない話題について知らないことが多い自分は読むべきものだと思った。そのため、新聞を読む習慣はプロジェクトが終わっても続けることにした。

加えてプロジェクト学習の過程では、社会問題などについても考えることが多く、自分の社会へ

の関心が高まる活動となった。また、チームで長期間プロジェクトに取り組むのは自分にとって初めての経験だった。メンバとのコミュニケーションや計画の立て方など、学ぶことが多かった。また複数人での活動は、コミュニケーションの面でのオーバーヘッドや、そもそも共同で動くことの難しさなど困難なことも多かったが、幅広い思索や分担による負担軽減などチームで動いてこそその良さも実感できた。特に私は前期の終盤は体調を崩して休んでしまったため、迷惑をかけたか助けられたりしたことがとても多かったが、それも含めてプロジェクト学習は貴重な良い経験だったと思う。

(※文責: 杉山孝太)

5.2.8 村澤雄斗の振り返り

5.2.8.1 前期の活動

このプロジェクトでは、まず初めに新聞に関する問題点や、関連する事項などをブレインストーミングを行って書き出した。このブレインストーミングの結果から、プロジェクトでどんな課題をどのように解決するか決めた。

その後、プロジェクト内で3つの班に分かれて作業をおこなった。私はゲームデザイン班に所属した。ここでは、制作するゲームの仕様を検討した。すでにリリースされているゲームの公式ホームページや攻略情報サイト、YouTubeに投稿されている動画などを参考にしながら、どのような仕様にするか検討した。新聞データの使い所や、面白いゲームにするための方法、政治に関心を持ってもらう方法を考えた。特に、政治に関心を持ってもらうためにはどのような仕様にすれば良いか深く検討した。ここで、ゲームが他のメディアとことなる点として、プレイヤーが操作を行うことで物語や世界に介入し、それらを体験した気分になる点に着目した。この特性を活かしたゲームになるように、仕様を検討した。検討した仕様をゲームプログラミング班や自然言語班と共有しながら、より良いゲームを作成するために検討を進めた。

中間発表では準備を初めるのが遅く、急いで準備を進めることになってしまった。この反省から、後期では無理のない計画を立てて実行できるようにしたい。

5.2.8.2 夏季休暇中の活動

夏季休暇中は、ゲームデザイン班として、シミュレーションゲームのUIや画面遷移の基盤を考えた。

決定した画面はメイン画面、イベント画面、コマンド画面、パラメータ画面の4つである。これら4つの画面は、プレイヤーがゲームをプレイしやすいように設計した。これらの画面はUIデザインツールのFigmaを利用して作成した。次のターンへ進むボタンや資金と人口の表示などのUIについては、既存のシミュレーションゲームを参考にして作成した。また、ゲーム内にプレイヤーが被選挙人となって選挙を行うシステムを実装する予定であったが、ゲームの目的や開発時間などを鑑みて、選挙システムは実装しないことに決定した。

5.2.8.3 後期の活動

後期は、シミュレーションゲーム班として活動した。私は、ゲームデザインやUIの設計等を担当した。まず初めに、ゲームのクリア条件と目的を、班のメンバと検討した。このとき、担当教員からのアドバイスや意見を参考にしながら検討した。既存のゲームとの違いを出すためにどうするかや、最初に決めたプロジェクトの目的を達成するためにどうするかを検討した。いくつかの案の

中から、最終的に「函館市の人口減少を食い止めるゲーム」を作ることになった。

ゲームの目的が決定した後、ゲームに実装するコマンドを考えた。コマンドを作成するときに気をつけたところは、コマンドの名称をなるべく現実に存在しそうな政策にした点である。このために、新聞やニュースサイトから情報を入手し、参考にした。それぞれのコマンドの名前に合わせて変化するパラメータを追加した。

さらに、ゲームバランスの調整を随時おこなった。ゲームをより面白いものに、プロジェクトの目的を達成できるように工夫をおこなった。具体的には、パラメータの値を調整したり、コマンド実行時のパラメータの変化量の調整などをおこなった。

5.2.8.4 最終発表について

最終発表会前には、発表に使うポスターの作成を担当した。ポスターは Adobe Illustrator を用いて作成した。ポスターの作成にあたって、Adobe Illustrator の使い方をデザインコースに在籍するプロジェクトメンバに教わった。まず、ポスターに書く項目を決め、それをどのように配置するか検討した。配置が決定した後、それぞれの項目に何を書くかを検討した。文章で書くのが最適か、図を用いるのが最適かなどなるべく見やすいポスターにするために検討を重ねた。ポスターの項目をすべて埋めることができたなら、更に良いポスターにするために推敲をおこなった。メンバとポスターの改善を行い、最終発表会までにポスターを作成することができた。しかし、ポスターを作り始めるのが遅かったため、ポスターの完成が最終発表会の直前になってしまった。この反省から、計画を建てて臨機応変に作業を進めることが重要だと実感した。今後の生活や活動に活かしたい。

最終発表会は、スライドとポスター、タワーディフェンスゲームのデモ、ゲームのプレイ動画を使って発表をおこなった。主にスライドを使って発表をおこなった。想定以上に会場に人が多く、声が通りにくかった。そのため、なるべく大きな声で発表をするように心がけた。中間発表の時よりも効果的な発表が行えたと感じている。しかし、最終発表直前までポスターの作成を行っていたことなどにより、発表練習をあまりすることができなかった。その結果、言葉に詰まってしまう場面があった。スライドがあるとはいえ、練習はするべきだったと感じた。

5.2.8.5 全体の活動を通して

全体を通した振り返りとして、いくつか感じたことがある。1つ目に、グループで活動することの難しさである。1人や、2、3人程度ではなく、今回のプロジェクト学習は8人での活動だったため、難しかった。認識の違いが起きたり、仕事量の偏りが起きたりしてしまった。これらの問題を起こさないために、定期的な連絡を行い、コミュニケーションを取ることが非常に重要だと感じた。また、グループ活動におけるスケジュール管理の重要性も感じた。一人で活動していればスケジュール管理に失敗したとしても自分が苦勞するだけである。しかし、グループ活動では自分だけでなく、他のメンバにも影響が出てしまう。適切なスケジュール管理ができるように、工夫をしなければならぬと感じた。

2つ目に、新聞に触れる機会が増えたと感じる。これまでは、SNS で時たまニュースを見るぐらいだった。しかし、プロジェクトで新聞に触れ、新聞を読んでいくうちに新聞やネットでニュースを読むことが習慣になった。また、ゲームで取り扱った人口減少問題など、政治や社会の問題についての関心が高まった。実際にニュースを見て、自分で少し考えてみるようになった。

3つ目に、情報を伝える工夫が重要であることを感じた。特に、最終発表会のポスターを作っているときに感じた。今まで何気なく見ていたポスターだったが、読みやすいように、情報が伝わり

やすいように様々な工夫が凝らされていることを知った。情報を伝えるためにポスターやスライドなどを作るときは、この気づきを活かして効果的なものを作れるようにしたい。

(※文責: 村澤雄斗)

5.3 全体の振り返り

私たちは、このプロジェクト活動を通していくつかの大きな変化があった。

第一の変化は、発言数の増加である。プロジェクト活動が始まったばかりの頃は、ブレインストーミングやミーティングの際にも、あまり発言が多く見られなかった。しかし、プロジェクト活動を重ねるにつれて個人々の発言数は明らかに多くなっていった。これは、活動を重ねていくにつれて、各々の活動に対する主体的な意識が向上した事や、自分の仕事に対する責任感を持ったからだと思われる。活動が進み、プロジェクトメンバをいくつかの班にわけたことにより、その班における自分の役割というものを強く認識させることができた。このような変化により、成果発表会などの準備の段階においても、デモ動画の作成やオーディエンスがゲームのデモプレイをできるようにするといった取り組みをおこなうことができた。結果として、成果発表会の発表評価も平均 8.1 と高い評価を得ることが出来た。

第二の変化は、私たち自身の政治に対する興味の増加である。私たちはこのゲームをプレイすることによって、プレイヤーの政治への興味を高める事を目的として活動を進めていた。しかし活動を振り返ってみると、そんな活動を通して私たち自身の政治への興味もまた、高まっているということに気がついた。プロジェクト活動が始まったばかりの頃は、投票に行っていないという人も多く、政治に関する話にもあまり興味を示していなかった。しかし、最近では各活動のミーティング中に政治の話をする機会が非常に多くなっていった。それだけでなく、活動日の新聞の内容を多くのメンバが把握し、その内容についてのディスカッションをすることもあった。こういった活動を通して、改めて現在の政治における投票制度についてメンバでディスカッションをしていたところ、「そもそも若者の総人口が少ないので意見が通りにくいという社会構造的な問題があるのではないか」という考えに至った。単に、若者の投票率を上げるだけではなく、若者の意見をより強いものにする工夫が必要なのではないかという提言を成果発表会でおこなった。こうした考えは、特に政治・統計に詳しい人にとっては、考えるまでもなく当たり前のことであるかもしれない。しかし、政治に興味を持っていなかった私たち「若者」が自分たちの考えを巡らせ、こうした提言に至ったということは、それ自体に大きな価値があるのではないだろうか、私たちは考える。

(※文責: 永井清盛)

付録 A 謝辞

本プロジェクトでは北海道新聞から提供を受けた新聞記事データを活用した。プロジェクトテーマの構想やシステムの試作において、同社の多大なるご協力を頂戴したので感謝申し上げます。

参考文献

- [1] 総務省, “総務省 | 国政選挙における投票率の推移”. 参照: 2024 年 1 月 15 日. [Online]. Available at: https://www.soumu.go.jp/senkyo/senkyo_s/news/sonota/ritu/index.html
- [2] 内閣府, “特集 1 日本の若者意識の現状～国際比較からみえてくるもの～”. 参照: 2023 年 11 月 22 日. [Online]. Available at: https://www8.cao.go.jp/youth/whitepaper/r01honpen/s0_1.html
- [3] 総務省, “総務省 | 国政選挙の年代別投票率の推移について”. 参照: 2024 年 1 月 15 日. [Online]. Available at: https://www.soumu.go.jp/senkyo/senkyo_s/news/sonota/nendaibetu/index.html
- [4] Serious Game Project, “九州大学シリアスゲームプロジェクト”. 参照: 2024 年 1 月 15 日. [Online]. Available at: <https://macma-lab.heteml.net/html/game/game.html>
- [5] 財津康輔, 松隈浩之, 前田真, “環境問題をテーマとしたシリアスゲームの教育的効果とユーザー特性の関連”, “デジタルゲーム学研究”, 5 巻, 2 号, pp.31-42. Sep. 30, 2011.
- [6] 函館市, “「第 2 期函館市活性化総合戦略（令和 3 年度改訂版）」および「函館市人口ビジョン（令和元年度改訂版）」《概要版》, p.1, Dec. 14, 2022.
- [7] IBM, “自然言語処理とは”. 参照: 2024 年 1 月 15 日. [Online]. Available at: <https://www.ibm.com/jp-ja/topics/natural-language-processing>
- [8] 国立国語研究所, “『UniDic』国語研短単位自動解析用辞書”. 参照: 2024 年 1 月 15 日. [Online]. Available at: https://clrd.ninjal.ac.jp/unidic/about_unidic.html
- [9] T. Sato, 「mecab-unidic-neologd/README.ja.md at master · neologd/mecab-unidic-neologd」, GitHub. 参照: 2024 年 1 月 15 日. [Online]. Available at: <https://github.com/neologd/mecab-unidic-neologd/blob/master/README.ja.md>
- [10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space.” arXiv, Sep. 06, 2013. doi: 10.48550/arXiv.1301.3781.
- [11] Radim Rehurek, “What is Gensim?”. 参照: 2023 年 12 月 21 日. [Online]. Available at: <https://radimrehurek.com/gensim/intro.html#what-is-gensim>