2019 年度　卒　業　論　文

マルチエージェントシステムにおける タスク割り当て手法についての研究

指導教員：渡辺 大地 准教授

メディア学部 ゲームサイエンス プロジェクト
学籍番号　M0116332
渡部 大樹

2020 年 2 月

2019 年度　　卒　業　論　文　概　要

論文題目

マルチエージェントシステムにおける タスク割り当て手法についての研究

メディア学部 学籍番号：M0116332	氏名	渡部 大樹	指導教員	渡辺 大地 准教授

キーワード	マルチエージェント、タスク割り当て、メタ AI、 アクションゲーム、NPC

近年のアクションゲームのキャラクターは AI によって自律的に意思決定と行動選択を行っている。キャラクターが自律的に行動するための AI をキャラクター AI と呼ぶ。コンピュータの発展とともに大規模化するゲーム開発において、自動的に演出やシナリオ構築を行う AI は開発コストの削減のために非常に重要である。しかしイベントシーンなどの特定の行動を強制したい場面において、複数のキャラクターが AI によって行動すると重複や衝突といった競合が生じる可能性がある。

この競合を解消するために、キャラクターの役割分担などを行う手法としてメタ AI がある。メタ AI は一階層上からゲーム内の様々な AI をコントロールする AI である。ゲームからの命令と AI による行動の中間に位置し、命令を遂行するにあたって適切なキャラクターや適切なタイミングを指示する。

本研究ではマルチエージェントとタスクを割り当てによるメタ AI の役割分担を提案する。提案手法ではプレイヤーのタスクを奪わないキャラクターを目的としている。手法としては、タスクが発生するときに高いほど優先的に処理する優先度という値を設定する。その際にプレイヤーが実行中のタスクを認識し、そのタスクの優先度を下げることによってプレイヤーのタスクを奪わないようにする。NPC 全体でタスクを共有するタスクリストの実装によって、各 NPC への役割分担をすることができた。また優先度の設定によって、プレイヤーの目的に相当するタスクを NPC が奪わずに行動することができた。

目 次

図 目 次

表 目 次

第 1 章

はじめに

1.1　研究背景と目的

近年のアクションゲームのキャラクターは AI によって意思決定と行動選択を行っている。AI によってプレイヤーによる操作を必要とせずに自動的に動くキャラクターをノンプレイヤーキャラクター（以下 NPC と表記）と呼ぶ。NPC の AI におけるアルゴリズムについても多くの研究があり、目標の達成を最優先とするゴールベース AI や、キャラクターの身体的行動を木構造で表すビヘイビアベース AI などがある。これらのような NPC 自身が意思決定を行う AI をキャラクター AI と呼ぶ。

アクションゲームの NPC は敵の攻撃を一身に引き受けるタンク、敵への攻撃を担うアタッカー、味方の回復を担うヒーラーのように役割によって振る舞いが多様化している。ゲームデザイナーが振る舞いが異なる NPC の集団に役割分担をするとき、タンクは防御役、アタッカーは攻撃役、ヒーラーは回復役といったように各 NPC に役割を持たせる。そしてそれぞれのキャラクター AI は役割に応じた行動を優先的に選択する必要がある。しかしプレイヤーや敵との位置関係によっては NPC の移動が間に合わなかったり、無意味になったりと役割を全うできないことがある。例えばタンクは敵と味方の間に入り、敵の攻撃に対して防御をすることで味方をかばう

とする。そのためには敵の攻撃が味方に当たる前に敵と味方の間に移動する必要がある。敵の攻撃対象が頻繁に変化するような状況では、タンクは移動をしているだけで攻撃を防御する役割を果たすことができない。このようなゲームデザイナーによる役割分担では、状況によっては役割に応じた振る舞いができない場合がある。すべての NPC が役割をもって行動するためには、どの役割もこなせる NPC に対して、キャラクターの位置関係などのゲームの状況に応じて必要な役割を振り分ける必要がある。

NPC の役割分担をする手法として三宅ら [1][2][3][4] や上段ら [5] や里井 [6] のメタ AI がある。メタ AI は他の AI をコントロールすることを目的とした AI であり、適切なキャラクターに適切な役割を分担し、すべての NPC が役割をもって行動することができる。スクウェア・エニックスのファイナルファンタジー XV[7] では、イベントシーンでメタ AI によって NPC の動きを制御している。イベントシーンにて NPC を特定の場所に集めて待機させたい場合、キャラクター AI による NPC の行動を停止してゲームシステムが強制的に NPC の行動を命令する。そうするとイベントシーン直前の NPC の行動や位置関係によっては、NPC 同士が衝突したり、停止した NPC がイベントシーンに映り込んでしまったりする。ファイナルファンタジー XV のメタ AI はこのような行動の強制とキャラクター AI の間を調停する役割を持っている。例として味方 NPC の誰かにプレイヤーを先導してほしいとき、味方のそれぞれがキャラクター AI で判断すると全員が先導を始めてしまう。メタ AI は先導してほしいという指示をいったん引き受けて、手が空いているかどうかやプレイヤーとの距離などで適切な NPC を選択して指示をする。

また実際に役割分担によって複数の NPC を制御する例として Killzone3[8] のチーム AI がある。Killzone3 では司令官、分隊長、メンバの 3 層からなるチーム AI を実装しており、司令官が複数の分隊長へ命令し、さらにそれぞれの分隊長が複数のメンバに命令をする。メンバと分隊長は司令官に現状を報告し、司令官は報告された内容から命令を決定する。

またこれらの AI を構築するためのエンジンについての研究もある。三宅 [9] はデジタルゲーム

の大規模化にともなう AI の構造化のための設計図であるエージェントアーキテクチャについて述べた。全体のデータを管理するブラックボードとデータを操作するモジュールに分離していることで、モジュールの入れ替えによって様々な知能を構築することができる。長谷 [10] はエージェントアーキテクチャの他に、エージェント同士の協調行動のためのヒエラルキーについても述べている。ヒエラルキーはエージェントを制御する下層の Agent AI、AI の情報伝達や指示によってグループを制御する中層の Squad AI、他の AI を制御することでゲーム全体を制御する上層の AI Director の 3 層からなっている。

このようにゲームの NPC が自律的に行動するために、様々な AI に関する技術がある。しかしゲームには人が操作するプレイヤーが存在する。ゲームにおけるキャラクターの AI では、NPC 同士の協調だけではなくプレイヤーとの協調も考えなければならない。本研究では NPC とプレイヤーとの協調を「NPC がプレイヤーの役割を奪わないこと」と定義し、プレイヤーと協調する複数の NPC への役割分担をするメタ AI を目的とする。また複数の NPC への役割分担をするにあたって、複数の NPC をマルチエージェントとして扱い、役割分担をタスク割り当てとして扱ったマルチエージェントへのタスク割り当て技術を参考にする。

1.2 論文構成

本論文は全 5 章で構成する。本章では背景と目的について述べ、第 2 章ではマルチエージェントとタスク割り当てについて述べる。第 3 章では提案手法について述べ、第 4 章では検証と考察について述べる。そして最後の第 5 章でまとめについて述べる。

第 2 章

マルチエージェントとタスク割り当てについて

本章ではマルチエージェントとタスク割り当てについて説明する。2.1 節ではマルチエージェントについて説明し、2.2 節ではタスク割り当てについて説明する。2.3 ではマルチエージェントとタスク割り当てを、メタ AI に適用する際の問題点について説明する。

2.1　マルチエージェント

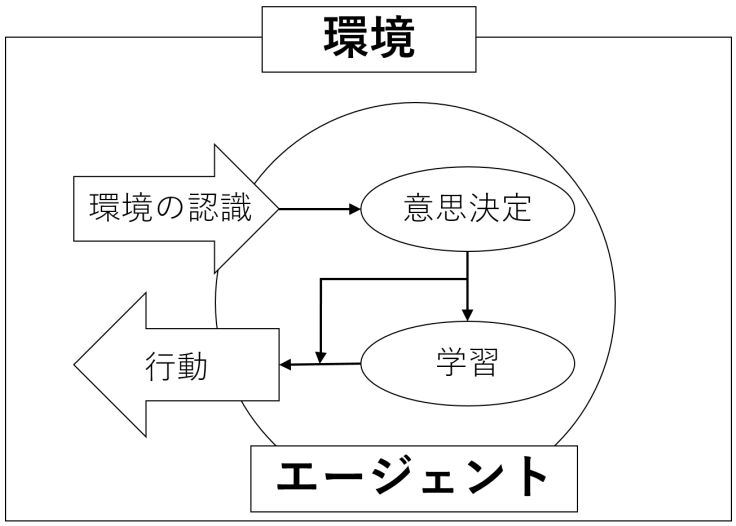

図 2.1　エージェントモデル

マルチエージェントとは、複数のエージェントによる集団のことである。エージェントとは周囲の環境の認識と、自律的な意思決定によって行動する機能を備えたオブジェクトを指す。それらに加えて学習機能を備えるものもあり、エージェント自身が意思決定と行動を環境に適したものに近づけていく。図 2.1 はエージェントの機能のモデルを表している。

マルチエージェントはエージェント同士の相互作用によって、様々な要素が絡み合った複雑な状況を解析するために用いる。それぞれのエージェントが環境や他のエージェントと情報共有をしながら、エージェント同士で協調する。これにより単一のエージェントでは達成できない複雑な事象を、マルチエージェント全体で達成する。例として横断歩道での往来における衝突の回避や、生物の群れの動きなどのシミュレートに用いる。

またマルチエージェントを用いた研究として、徳田ら [11] は駐車場の混雑状況を駐車場の空き状況やドライバーの目的地などの情報をもとに予測し、ドライバーに他の駐車場を利用するよう交渉する手法を提案した。提案手法の有無それぞれに 50 回シミュレーションを行い、手法の有効性を示した。布施ら [12] はインタラクティブデバイスによるマルチエージェントシステムにおけるエージェント間の協調形態を「中央制御型」、「直接協調型」、「間接協調型」の 3 種類に大別し、それぞれの利点と欠点について調査した。これらの協調形態を切り替えて利用する従来の手法に対して、複数の強調形態を利用してエージェントが目的達成のためのプランニングをするシミュレーションについて検討した。岩田ら [13] は不均一な移動速度の複数のエージェントが巡回清掃する環境において、エージェント間での直接の通信を行わず移動速度の違いによって分業する手法を提案した。移動速度と距離によって報酬が変化し学習するシミュレーションを行い、一部の環境において分業を促していることを示した。

また三宅 [14][15] はクロムハウンズを例に、コンピューターゲームにおけるマルチエージェントについて述べている。クロムハウンズはフロムソフトウェアが発売した最大 6 人でチームを組んで通信しつつ、メカを操作して相手チームと戦うオンラインアクションゲームである。このゲー

ムはプレイヤーと協調する NPC の AI を目指しており、目標を小目標に細分化して達成までの行動を決定するゴール指向型プランニングによって動いている。エージェントの目標とチームの目標から行動を決定し、集団としての知能を実現している。

本研究では NPC をエージェントとして扱い、複数の NPC の集合をマルチエージェントとして扱う。

2.2 タスク割り当て

タスクとは全体の目的を達成するための作業を細分化したものである。仕事や課題といった意味の他に、コンピュータが処理する作業の最小単位という意味を持つ。タスク割り当てとはタスク処理の効率化を図るために、適切なエージェントを選択するアルゴリズムを指す。

タスク割り当ての手法に関連する研究は様々なものがある。飯嶋ら [16] はタスクを一定数保持し、エージェントにタスクそれぞれに対する不得意度と優先度を宣言させた。それらの情報からタスクを処理するにあたって、エージェントのリソースを無駄なく利用する割り当ての効率化を提案した。江川ら [17] はリソースを多く要するタスクを、処理時間が最も短くなるエージェントに割り当てる残りタスク数優先という手法を提案した。これを実際の作業環境に導入し、実証実験を行うことで提案手法の有効性を示した。上原ら [18] は RoboCup Rescue において消防司令所エージェントが消防エージェントの最短経路における道路閉塞による実行コストを、道路啓開エージェントの実行コストによって修正しタスクを割り当てる手法を提案した。これにより 2 種類のエージェントの割り当ての組み合わせを最適化した。

2.3 ゲームに適用する際の問題点

従来のタスク割り当て手法の多くはエージェントとタスクのマッチングに焦点を置いている。タスクを一時的に保持し、エージェントとのマッチングの選択肢を広げることによって、より適

切な割り当ての実現を目的としている。

しかしリアルタイムに状況が変化するアクションゲームにおいて、一時的なタスクの保持を行うとキャラクターの行動決定に遅れが生じる可能性がある。例としてモンスターハンタークロス[19] では、アイルーという味方 NPC が 2 体までクエストに同行することができる。アイルーは戦闘中にプレイヤーを回復したり、乱入してきたボス級モンスターを追い払ったりする。しかしプレイヤーの体力や敵モンスターの数などの状況変化は突発的であり、状況変化したときにアイルーが別の行動をしていると状況変化に応じた行動ができない。これが原因でクエストに失敗することもあるため、状況変化に即時対応できる必要がある。

また従来のタスク割り当てに関する研究の手法は、発生したタスクを全て完遂することを前提としている。それらの手法をそのままゲームに適用すると NPC が全てのタスクを行ってしまい、プレイヤーの役割を奪ってしまう可能性がある。プレイヤーが主体となってゲームを進行したい場合に、NPC が全てのタスクをこなしてしまうとプレイヤーのゲーム内での役割が減少してしまう。例としてゴッドイーター 3[20] では、プレイヤーの味方である NPC だけでミッションをクリアしてしまうことがある。ゴッドイーター 3 では敵の一部位を集中攻撃して弱体化する結合崩壊や、ダメージの蓄積によって一定時間行動不能になるダウン状態を狙って戦闘を有利に進めることができる。ゴッドイーター 3 以前のシリーズ作品では、プレイヤーがこれらのギミックを意図的に狙う必要があった。しかしゴッドイーター 3 では NPC がギミックを積極的に狙い、NPC だけで戦闘に勝利できるようになりプレイヤーの役割が減ってしまった。

第 3 章

提案手法

本章では提案するマルチエージェントへのタスク割り当てによるメタ AI について説明する。3.1 節ではメタ AI について説明し、3.2 節では提案手法を適用するゲームについて説明する。3.3 節では本研究で提案するタスク割り当てのアルゴリズムについて説明する。

3.1 メタ AI について

メタ AI とはゲーム内で使用している様々な AI をコントロールするための AI である。キャラクターへの指示の他にも敵の配置や、マップの自動生成といったレベルデザインにも大きく関わる。

ゲームのキャラクターはイベントシーンなどにおいて、特定の行動をとらなければならない場面が多数存在する。イベントを円滑に進めるためにはキャラクター AI によるキャラクターの行動を停止させて、ゲームシステムからイベントに従った行動をキャラクターに強制する必要がある。しかし複数のキャラクターがそれぞれ所定の位置に移動したいといった場合に、キャラクター AI による衝突回避などの行動が停止しているとキャラクター同士が衝突するといったイベントにそぐわない振る舞いをする恐れがある。また 1.1 節で述べたような、アタッカーやタンクなどの役割をキャラクター AI によって行うと、そのときのゲームの状況に不要な役割のキャラクターは停

止したりランダムに行動したりするため、必要な役割を持ったキャラクターの行動の邪魔になることがある。

メタ AI はこのような状況で、ゲームシステムからの命令とキャラクター AI を両立する役割をもつ。メタ AI はゲームシステムからの命令を一度預かり、状況に応じたキャラクターとタイミングを選択して命令する。イベントシーンの場合は、キャラクター同士の経路が重ならないように、キャラクターそれぞれへの命令のタイミングを調整する。プレイヤーの回復の場合はプレイヤーとの距離や、それぞれのキャラクターの行動から、適切なキャラクターを一人選択し命令する。

3.2 環境とするゲームについて

このゲームではキャラクターの位置関係に応じて必要な役割がタスクとして発生し、それらを別々の NPC に割り振って役割分担をすることを目標としている。そのためゲームの最終目標であるボスを倒すためには敵の攻撃を防いだり、装置を起動したりするといった複数種類のタスク達成が必要なルール環境とした。タスク割り当てによって NPC が行動する様子を可視化するために、C#と Fine Kernel ToolKit[21] を用いてアクションゲームを作成した。図 3.1 は作成したゲームの実行画面である。

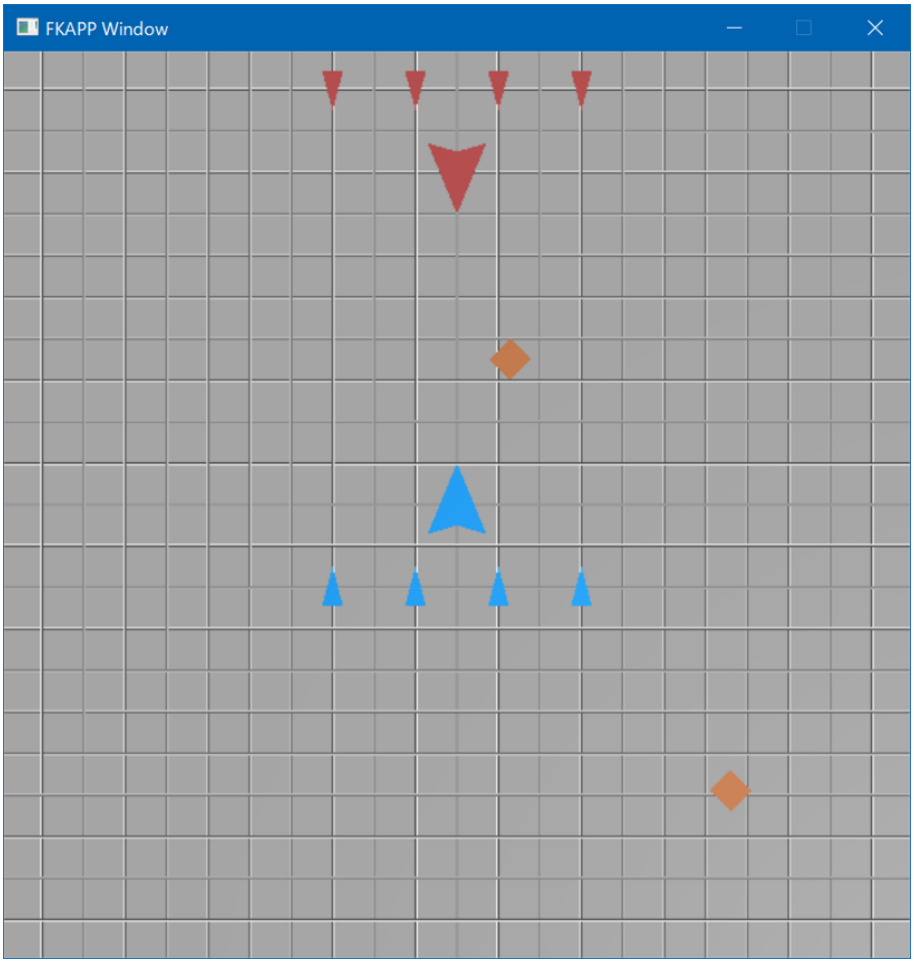

図 3.1　ゲームの実行画面

3D 空間上に配置したプレイヤーキャラクターを、リアルタイムにマウスで移動と攻撃の操作を行うアクションゲームである。プレイヤーを含む 10 体のキャラクターがフィールドという空間内を前後左右に移動しながら、体当たりによって攻撃し合う。フィールドは灰色の半径が 100 の円形の領域になっており、キャラクターがフィールドの外に出ようとしたとき、そのキャラクターをフィールドの内側に弾く。プレイヤーの視点となるカメラはプレイヤーキャラクターを上空から見下ろしており、プレイヤーを中心とした 1 辺が 100 の正方形の範囲を見ることができる。キャラクターは 1 体のプレイヤーと 4 体のサポーターで構成された自陣営と、1 体のボスと 4 体のエネミーで構成された敵陣営に分かれている。自陣営のキャラクターモデルは青色で、敵陣営のキャラクターは赤色となっている。図 3.2 は各キャラクターのモデルである。左からプレイ

ヤー、サポーター、ボス、エネミーの順となっている。

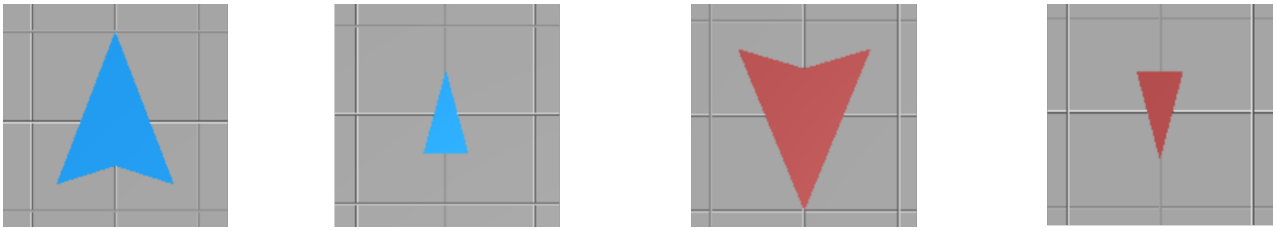

図 3.2　キャラクターモデル

各キャラクターには体力という数値を設定しており、プレイヤーの体力が 0 になると自陣営の敗北で、ボスの体力を 0 にすると自陣営の勝利となる。攻撃方法は体当たりのみで、相手陣営のキャラクターの体力を減らすためには移動して接近し、体当たりを当てる必要がある。自陣営のプレイヤーと敵陣営のボスは体力を 20 に設定しており、自陣営のサポーターと敵陣営のエネミーは体力を 15 に設定している。体当たりを 1 回当てられるごとに体力が 1 減少し、0 になったキャラクターはフィールドから削除する。

キャラクターは体当たりをしたときと、体当たりを受けたときに硬直状態となる。硬直状態は移動や体当たりといった行動の一切が行えない状態であり、硬直時間という一定の時間が経過することで解除する。硬直状態の最中に新たに硬直状態が付与された場合は、硬直時間は加算ではなく上書きする。体当たりをしたときは硬直時間が 135 になり、体当たりを受けたときは硬直時間が 45 になる。連続で体当たりをすることを防ぐために、体当たりをされたときよりも体当たりをした時のほうが硬直時間を長く設定している。

またフィールドには敵陣営に属する装置というオブジェクトが 4 つ設置してあり、ゲーム開始時にフィールド上のランダムな場所に配置する。自陣営のキャラクターがこれに体当たりを 5 回当てると起動状態となる。敵陣営のボスの体力が 10 以下になると、フィールドに設置した装置を全て起動状態にしなければ、ボスの体力をそれ以上減らすことができない。敵陣営のボスの体力が 11 以上の場合でも装置を全て起動することは可能だが、自陣営のサポーターは装置の起動を行

わない。図 3.3 は装置のモデルである。左が非起動状態で、右が起動状態である。

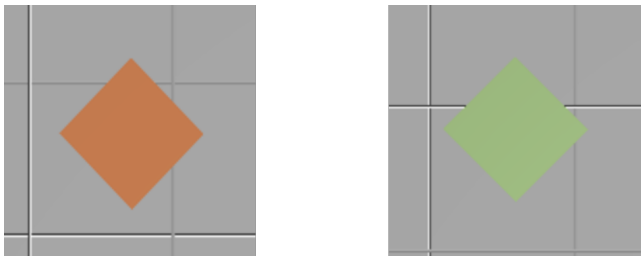

図 3.3　装置モデル

自陣営のサポーターは罠の設置という行動が可能であり、フィールドに罠というオブジェクトを設置できる。この罠に接触した敵陣営のキャラクターは硬直状態となり、自陣営のキャラクターが体当たりを当てる機会を増やすために硬直時間は 450 と最も長く設定している。図 3.4 は罠のモデルである。

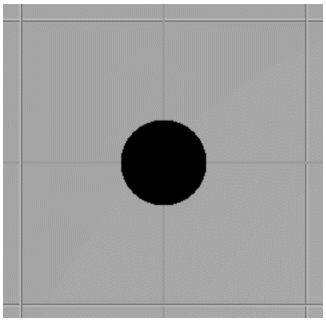

図 3.4　罠モデル

表 3.1 はこのゲームの詳細をまとめたものである。

表 3.1　ゲームの詳細

ゲーム内での名称	詳細
キャラクター	プレイヤー : 1 体, 自陣営, 自身で操作するキャラクター, 自陣営の要 サポーター : 4 体, 自陣営, AI で行動するキャラクター, 罠を設置できる ボス : 1 体, 敵陣営, AI で行動するキャラクター, 敵陣営の要 エネミー : 4 体, 敵陣営, AI で行動するキャラクター
オブジェクト	装置 : 4 機, 敵陣営, 全て起動しないとボスの体力を 10 未満に減らせない 罠 : 自陣営, 敵陣営のキャラクターが接触すると 450 の硬直時間を与える
フィールド	ゲーム内で使用する範囲 半径 100 の円形
カメラ	プレイヤーを中心として上空から見下ろす視点 1 辺が 100 の正方形の範囲を描画

3.2.1　キャラクターについて

プレイヤーは自身が操作するキャラクターである。自陣営の要であり、プレイヤーの体力が 0 になるとゲームに敗北する。左クリックでマウスカーソルの位置に向かって移動する。右クリックでマウスカーソルがある方向に向いて体当たりをする。

サポーターはプレイヤーの味方となるキャラクターであり、基本的には敵陣営への攻撃とプレイヤーを敵陣営の攻撃から守る役割を担う。特殊な行動である罠の設置が可能であり、敵陣営のキャラクターの進路上にいるときは、罠を設置して足止めをする。

ボスは敵陣営の要であり、ボスの体力を 0 にするとゲームに勝利する。ボスは自身を攻撃してきた自陣営のキャラクターを攻撃する。ボスの体力が 10 以下になると、フィールドに設置した装置を全て起動状態にしなければボスの体力をそれ以上減らすことができない。

エネミーはボスの味方となるキャラクターであり、ボスと同様に自身を攻撃してきた相手陣営のキャラクターを攻撃する。

3.3　タスク割り当てアルゴリズムについて

本研究で提案するマルチエージェントへのタスク割り当て手法について述べる。図 3.5 はタスク割り当てのおおまかな流れを示したものである。最初にキャラクターや装置の位置関係を取得し、それに応じた優先度を設定したタスクが発生する。発生したタスクを優先度が高いものから順に、全て割り当て終わるまで適切なエージェントに対して割り当て処理を行う。割り当て処理が終わったら、エージェントがタスクに従って行動する。この一連の流れを本研究ではターンと呼ぶ。

本研究では提案するタスク割り当て手法を自陣営のサポーターの AI に適用する。

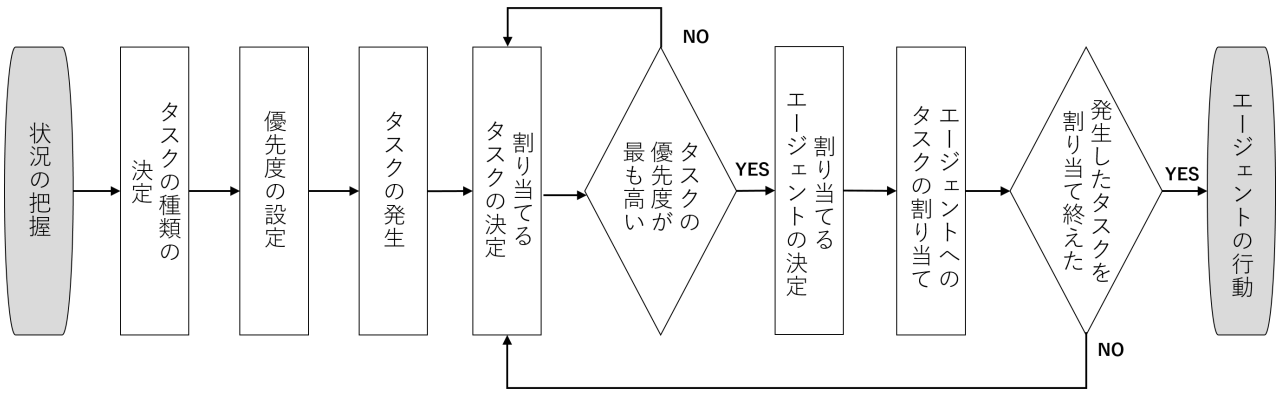

図 3.5　タスク割り当ての概要

3.3.1　タスクの種類

ゲーム中に発生するタスクは以下の 3 種類存在する。

1. 敵陣営のキャラクターへの攻撃
2. 罠の設置
3. 装置の起動

敵陣営のキャラクターへの攻撃は指定された敵陣営のキャラクター 1 体に体当たりを行う。罠

の設置は、罠の対象としたキャラクターの進路上に罠を設置する。装置の起動は敵陣営のボスの体力が 10 以下になると発生するようになり、フィールドに設置された装置に体当たりを行う。

3.3.2　タスクの発生

このゲームの実行中は常にタスクの発生判定を行う。発生するタスクの種類は条件によって分岐する。

敵陣営のキャラクターへの攻撃は、フィールドにいる敵陣営のキャラクターそれぞれに、そのキャラクターを攻撃対象とするタスクが 1 つずつ発生する。所定の位置への罠の設置は敵陣営のキャラクターの進路上に、そのキャラクターの攻撃対象ではない自陣営のサポーターがいる場合に発生する。装置の起動は敵陣営のボスの体力が 10 以下のとき、起動状態ではない装置それぞれに、その装置を起動状態にするタスクが 1 つずつ発生する。

タスクの発生判定の際に、発生条件を満たしたタスクが複数ある場合はそれらすべてが同時に発生する。ここで発生したタスクをタスクリストという全体で共有するタスク一覧に格納する。図 3.6 はタスク発生のモデル図である。

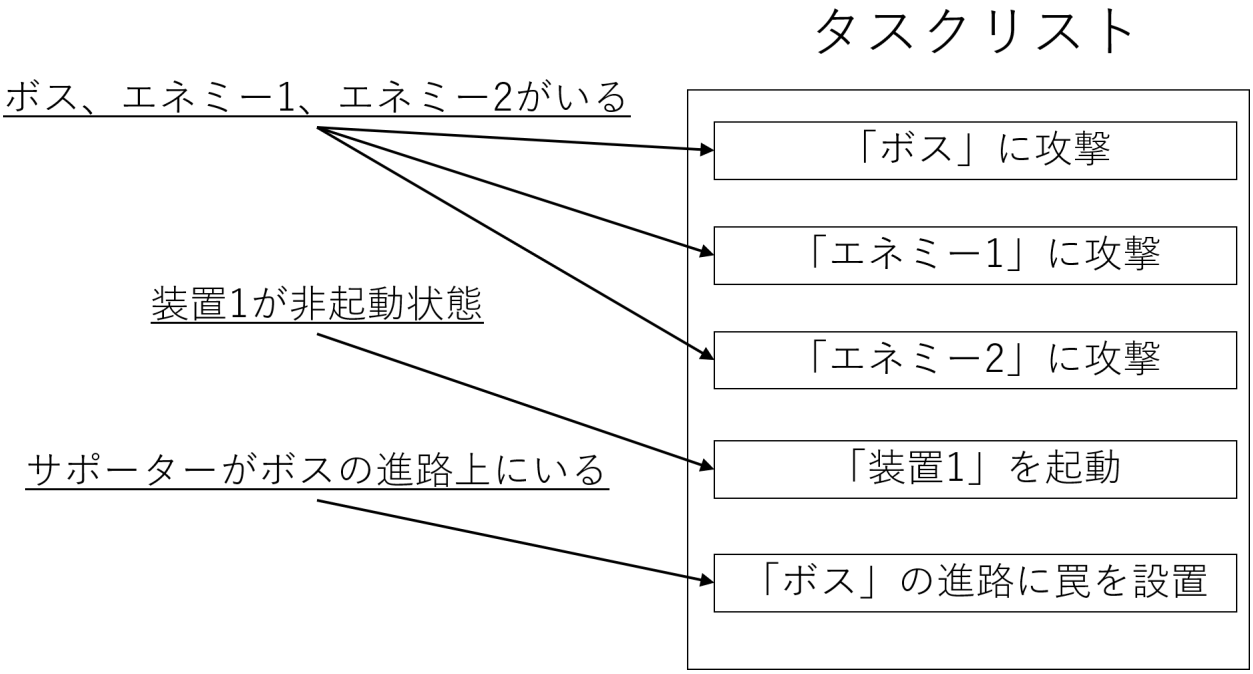

図 3.6　タスク発生のモデル図

3.3.3　タスクの優先度

タスクを発生させるときに、優先度という値を各タスクに設定する。優先度はタスクの種類ごとに初期値を設定しており、プレイヤーの行動によって変動する。優先度が高いほど優先的に割り当てる。

初期状態では優先度の高さを高い順に罠の設置、装置の起動、敵陣営のキャラクターへの攻撃と設定している。罠の設置には相手キャラクターが設置地点に到達する前に完遂させる必要があるため、優先度を最も高くして確実に割り当て処理をする。装置の起動は自陣営の勝利条件である敵陣営のボスの体力を 0 にするために必要なため、敵陣営のキャラクターへの攻撃よりも優先する。

装置の起動と相手キャラクターへの攻撃がプレイヤーの目標とするタスクとなった場合、それらのタスクの優先度を下げて、プレイヤーのタスクを奪わないようにする。プレイヤーの目標と

は、プレイヤーが一定時間内に体当たりした相手キャラクターまたは装置を対象としたタスクを指す。図 3.7 はプレイヤーの目標によってタスクの優先度が変化する様子を示したものである。

タスクの種類	優先度
「ボス」に攻撃	1
「エネミー1」に攻撃	1
「エネミー2」に攻撃	1
「装置1」を起動	2
「ボス」の進路に罠を設置	3

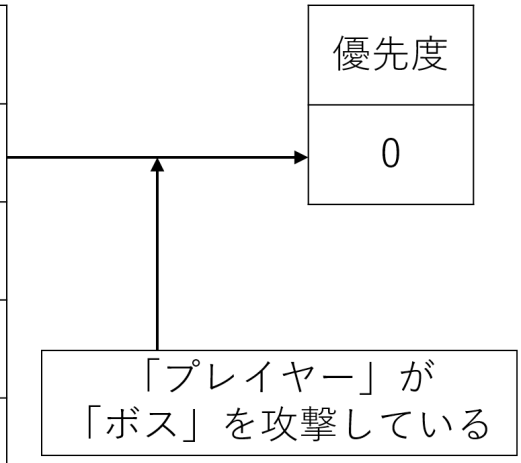

図 3.7　優先度の変化のモデル図

3.3.4　タスクの割り当て

本研究で使用するゲームでは自陣営のサポーターに対してタスク割り当てを行う。

最初にそのターンに発生したタスクをタスクリストに加える。タスクリストの中から最も優先度が高いものを選択する。選択したタスクを割り当てるサポーターを選択する。タスクの種類が敵陣営のキャラクターへの攻撃の場合は攻撃対象となるキャラクターに最も近いサポーターを選択する。罠の設置の場合は、罠の対象となるキャラクターの進路上にいるサポーターを選択する。装置の起動の場合は、対象となる装置に最も近いサポーターを選択する。優先度が最も高いタスクを選択したサポーターに割り当て、割り当てたタスクをタスクリストから削除する。

この割り当てを繰り返し、最終的に全てのサポーターに優先度が高いタスクを割り当て終わったら、タスクリスト内に残ったタスクを破棄する。最後にサポーターがタスクに従った行動をして、サポーターが持つタスクを破棄してターンの最初にもどる。図 3.8 は割り当ての手順を示し

た図である。

サポーターが持つタスクを破棄する際に、サポーターがそのタスクを達成したかどうかは判定しない。例としてキャラクターへの攻撃の場合は、実際に攻撃する前の移動をしただけであってもタスクを破棄する。そのため複数の敵陣営のキャラクターがサポーターから見てほぼ同じ距離にいる場合は、サポーターの攻撃対象が頻繁に切り替わることがある。

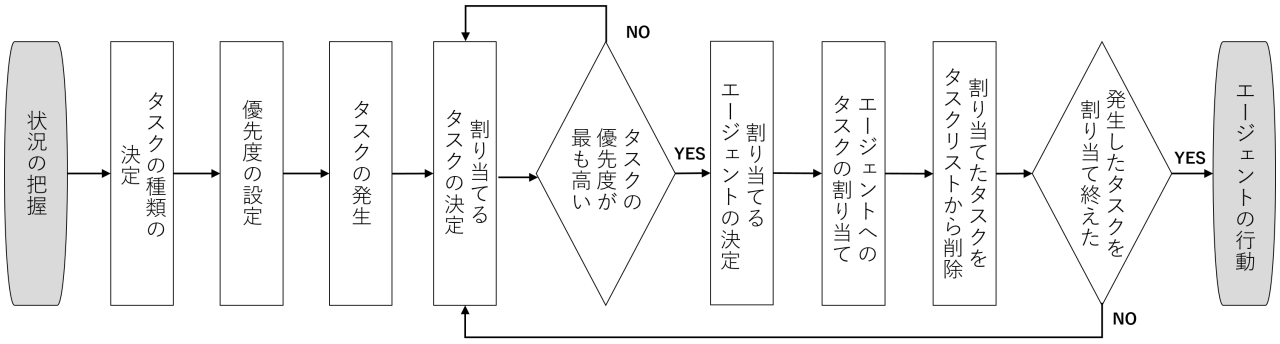

図 3.8　タスクの割り当ての手順

第 4 章

検証と考察

本章ではタスクの割り当てとサポーターの行動について検証し、それについて考察を述べる。検証では複数のサポーターに別々のタスクを割り当てているか、サポーターがプレイヤーの目的に相当するタスクを奪っていないかの 2 点に注目する。

4.1　検証

敵陣営のキャラクターへの攻撃については正常にタスクが発生し、それぞれのサポーターが別々の相手キャラクターを攻撃対象とする様子を確認できた。図 4.1 はその様子を示したものである。このタスクではサポーターと攻撃対象の敵陣営のキャラクターとの距離によって、サポーターの行動が異なる。サポーターは距離が一定以上離れている場合は攻撃対象に向かって移動し、一定未満の距離まで近づいたら体当たりを行う。図中の赤い線はゲーム中に表示されており、サポーターとそのサポーターが攻撃対象としている敵陣営のキャラクターを繋いでいる。

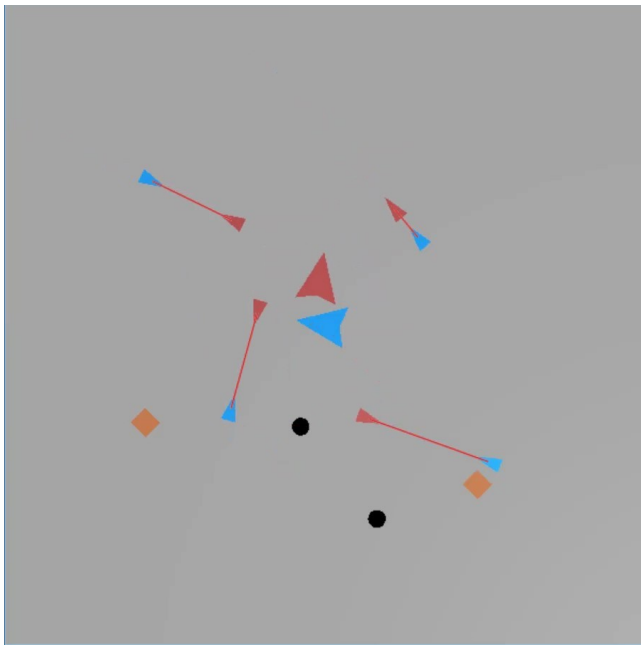

図 4.1　サポーターが別々の相手キャラクターを攻撃する様子

罠の設置については正常にタスクが発生し、サポーターが罠を設置する様子を確認できた。図 4.2 はその様子を示したものである。図中の赤い丸の中にいるサポーターが罠を設置している。

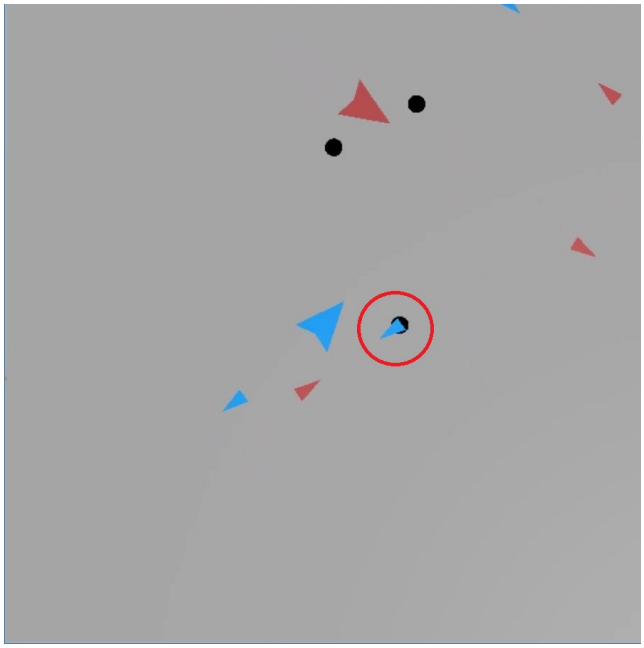

図 4.2　サポーターが罠を設置する様子

装置の起動については正常にタスクが発生し、それぞれのサポーターが別々の装置に体当たり

する様子を確認できた。図 4.3 はその様子を示したものである。このタスクでも敵陣営のキャラクターへの攻撃と同様にサポーターは、サポーターと装置の距離が一定以上なら近づき、一定未満なら体当たりをする。図中の赤い線はゲーム中に表示されており、サポーターとそのサポーターが体当たりしている装置を繋いでいる。

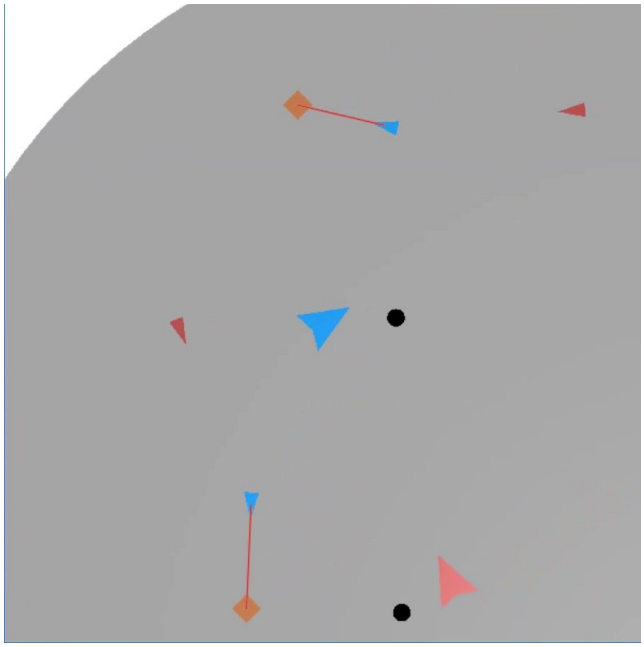

図 4.3　サポーターが別々の装置に体当たりする様子

サポーターがプレイヤーの目的に相当するタスクを奪わなかったかについては、プレイヤーが目的としている相手キャラクターを、サポーターは攻撃しない様子を確認できた。図 4.4 はその様子を示したものである。図中のピンク色の相手キャラクターはプレイヤーの目的となっている。相手キャラクターへの攻撃のタスクは通常の場合、サポーターに最も近い距離にいる相手キャラクターを攻撃の対象とする。図中の赤い丸はピンク色の相手キャラクターがサポーターに最も近い距離にいても攻撃せず、次に近い距離にいるボスを攻撃している様子を示している。

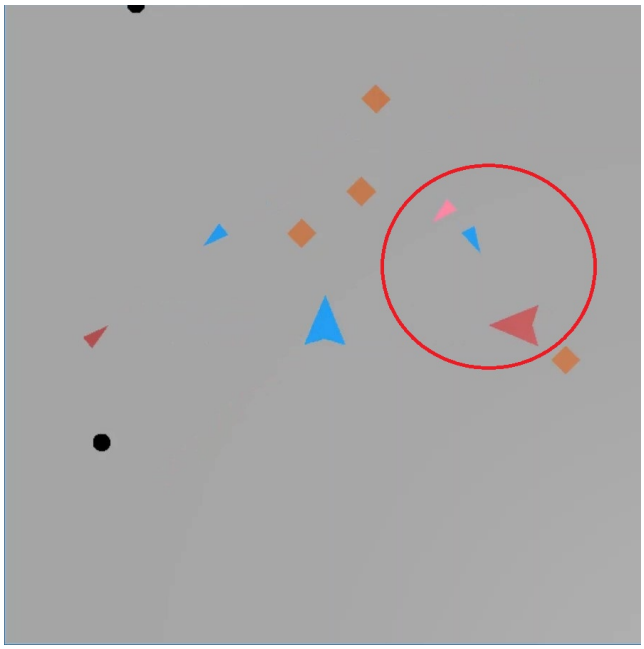

図 4.4　プレイヤーの目的に攻撃しないサポーターの様子

4.2　考察

タスクの割り当てによってサポーターへの役割分担を行った。またサポーターはプレイヤーの目的に相当するタスクを奪わずに行動した。

しかし罠を設置については、正常に設置することはできたが相手キャラクターが罠にあまり当たらなかった。このことからタスクが発生する条件や、敵陣営のキャラクターを罠に誘導するタスクを発生させるといった、罠にかかりやすくする改善策についても着目する必要がある。また今回の手法では、1 つのタスクを 1 体のサポーターのみで処理するため、タスクの処理に必要なサポーターの処理能力が足りない場合も想定する必要があると考えた。

第 5 章

まとめ

本研究ではプレイヤーと協調する複数の NPC への役割分担をするメタ AI を提案した。C#と Fine Kernel ToolKit を用いて、提案したタスク割り当て手法をメタ AI として実装したアクションゲームを作成した。タスクリストによって NPC 全体でタスクを共有することで、複数の NPC が同じ役割を持つことなく役割分担をすることができた。またタスクに優先度を設定し、プレイヤーの目的に相当するタスクの優先度を下げることによって、NPC がプレイヤーの目的を奪わずに行動することができた。

しかし罠の設置については、設置した罠が相手キャラクターに当たらない状況があった。そのため、タスクが発生する条件やタスクの種類などについても着目し、より必要性の高いタスクが発生するようにすべきと考えた。

今回の手法では 1 つのタスクを 1 体の NPC に割り当てて役割分担を行うため、1 つのタスクに複数の NPC が協力して実行する状況を想定していない。タスクの数が NPC の数を下回ったときに、手の空いた NPC が他の NPC を手伝うといった状況についても検討していきたい。

謝辞

本研究を進めるにあたってご指導いただいた先生方、先輩方や相談にのってくれた友人たちに感謝いたします。

参考文献

[1] 三宅陽一郎. ディジタルゲームにおける人工知能技術の応用 (<特集>ゲーム AI). 人工知能学会誌, Vol. 23, No. 1, pp. 44–51, jan 2008.

[2] 三宅陽一郎. ディジタルゲームにおける人工知能技術の応用の現在 (<特集>エンターテイメントにおける AI). 人工知能, Vol. 30, No. 1, pp. 45–64, jan 2015.

[3] 三宅陽一郎, 今村紀之, Ingimar Gudmundsson, 小松智希, 下川和也, 上段達弘, 白神陽嗣, 高橋光佑, 並木幸介, Fabien Gravot, Prasertvithyakarn Prasert, Hendrik Skubch, Matthew Johnson, 南野真太郎, 横山貴規. 大規模ゲームにおける人工知能　―ファイナルファンタジー XV の実例をもとに―. 人工知能, Vol. 32, No. 2, mar 2017.

[4] 三宅陽一郎, 水野勇太. 人工知能（メタ AI）を用いたゲームデザインの変革. https://cedil.cesa.or.jp/cedil_sessions/view/1757. 参照:2019.01.19.

[5] 上段達弘, 下川和也, 高橋光佑, 並木幸介. FINAL FANTASY XV におけるレベルメタ AI 制御システム. https://cedil.cesa.or.jp/cedil_sessions/view/1544. 参照:2019.01.19.

[6] 里井大輝. 感情を揺さぶるメタ AI ～ゲームへの実装方法とバランス調整への応用事例～. https://cedil.cesa.or.jp/cedil_sessions/view/2013. 参照:2019.01.19.

[7] FINAL FANTASY XV（ファイナルファンタジー 15） — SQUARE ENIX. http://www.

jp.square-enix.com/ff15/. 参照:2020.01.19.

[8] Killzone 3. https://www.killzone.com/nl_NL/killzone3.html. 参照:2020.02.28.

[9] 三宅陽一郎. 3-1 ディジタルゲームにおける人工知能エンジン. 映像情報メディア学会誌, Vol. 68, No. 2, pp. 125–130, 2014.

[10] 長谷洋平. 汎用ゲーム AI エンジン構築の試みとゲームタイトルでの事例. 人工知能, Vol. 32, No. 2, mar 2017.

[11] 徳田渉, Miguel A. Lopez-Carmona, 金森亮, 伊藤孝行. 車両に対する自動交渉による駐車場割当手法の提案. Technical Report 13, 名古屋工業大学, University of Alcala, 名古屋大学, 名古屋工業大学, mar 2015.

[12] 布施太章, 諏訪博彦, 栗原聡. 環境の動的変化に応じて協調形態を柔軟に変化させるマルチエージェントプランニングの検討. Technical Report 3, 電気通信大学, 電気通信大学, 電気通信大学, jan 2014.

[13] 岩田裕登, 杉山歩未, 菅原俊治. 巡回問題における能力の異なる複数エージェントの自律的な行動決定手法. Technical Report 2, mar 2019.

[14] 三宅陽一郎. ゲーム AI とマルチエージェント（上）. https://www.slideshare.net/youichiromiyake/ai-79541209. 参照:2019.01.19.

[15] 三宅陽一郎. ゲーム AI とマルチエージェント（下）. https://www.slideshare.net/youichiromiyake/ai-79541241. 参照:2019.01.19.

[16] 飯嶋直輝, 齋藤健吾, 早野真史, 菅原俊治. 継続的に発生する優先度つきタスクの効率的割り当て手法の一解法について. Technical Report 9, 早稲田大学基幹理工学部情報理工学科, 早稲田大学基幹理工学研究科情報理工学専攻, 早稲田大学基幹理工学研究科情報理工学専攻, 早稲田大学基幹理工学研究科情報理工学専攻, feb 2016.

[17] 江川知樹, 小野良太, 山下倫央, 川村秀憲. タスク優先度に基づくマイクロタスクの逐次割り

当て手法の提案と評価. Technical Report 6, 北海道大学大学院情報科学研究科, 北海道大学大学院情報科学研究科, 産業技術総合研究所人工知能研究センター, 北海道大学大学院情報科学研究科, feb 2016.

[18] 上原淳, 片上大輔, 新田克己. RoboCup Rescue における異種エージェントを考慮したタスク割り当て. Technical Report 2(2006-ICS-142), 東京工業大学総合理工学研究科　知能システム科学専攻, 東京工業大学総合理工学研究科　知能システム科学専攻, 東京工業大学総合理工学研究科　知能システム科学専攻, jan 2006.

[19] CAPCOM：モンスターハンタークロス 公式サイト. http://www.capcom.co.jp/monsterhunter/X/. 参照:2020.01.19.

[20] ゴッドイーター 3 — バンダイナムコエンターテインメント公式サイト. https://ge3.godeater.jp/. 参照:2020.01.19.

[21] Fine Kernel ToolKit System Top Page. https://gamescience.jp/FK/. 参照:2020.01.19.