2018 年度　卒　業　論　文

FPS ゲームの試合における観戦カメラ AI に関する研究

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
学籍番号　M0115293
廣里 直人

2019 年 2 月

2018 年度　　卒　業　論　文　概　要

論文題目

FPS ゲームの試合における
観戦カメラ AI に関する研究

メディア学部 学籍番号：M0115293	氏名	廣里 直人	指導教員	渡辺 大地 准教授

キーワード	FPS ゲーム、観戦カメラ、eSports、カメラワーク、注視点

昨今 eSports が流行っており、世界中でゲームの大会が開かれており、動画としていつでもどこでも見ることが可能である。大会の様子を動画にする上で、重要な要素としてプレイヤー達が操作しているキャラクターの視点とは違うものを映すことが出来る観戦カメラに着目した。大会には様々なジャンルのゲームが存在するが、FPS ゲームはキャラクターの数が多く、展開が早いため観戦カメラの操作は難しいという問題がある。その問題解決のために本研究では FPS ゲームの試合において 1 人のカメラマンとして観戦カメラが適切なポジションへの移動、適切なアングルに変更するという行動を自動制御する AI の作成を目的とした。手法としては自作した FPS ゲームにおけるキャラクター全ての座標と方向ベクトル、そして戦闘を行っているかを元に観戦カメラの注目地点の算出を行う。観戦カメラのカメラワークの決定にはキャラクター同士の距離と戦闘を行っているかでカメラワークを選択する。続けて同じカメラワークを選択していないかと撮影画面を壁で遮っていないかを判定し、カメラワークを決定する。算出した注目地点と決定したカメラワークを元に観戦カメラを動かす。評価実験として自作した FPS ゲームに対して AI 同士が戦い、その様子を提案手法のアルゴリズムを適用した観戦カメラで撮影し、動画とした。さらにもう 1 つ既存の観戦カメラのシステムに似たアルゴリズムを適用したカメラで撮影し、動画とした。この 2 つの動画と実際の大会の動画と比較することによって有用であるかどうかを検証した。検証結果として提案手法のアルゴリズムを適用して撮影した動画の方が有用であると言えたが、臨場感などが必要な場合は一人称視点のカメラワークも必要であると言えた。

目 次

図 目 次

第 1 章

はじめに

1.1　研究背景と目的

昨今 eSports と呼ばれる様々なジャンルのゲーム大会やプロリーグが世界中で行われており、その様子はインターネットを通じて動画としていつでもどこでも見ることが可能である。ゲームプレイヤーの様子や大会会場を映し出す実物のカメラ映像も重要な要素であるが、eSports において更に重要なる映像はゲーム画面である。各プレイヤーが操作しているゲーム画面や、操作画面とは別にゲーム内フィールドやキャラクター達の状況を映し出す専用のゲーム内のカメラが存在する。主に俯瞰視点や三人称視点での映像を映すこのカメラを観戦カメラと呼ぶ。ゲームの大会にも様々なジャンルが存在するが、ライブストリーミング配信サイトである Twitch[1] においては FPS ゲームが人気ジャンルの 1 つである。FPS ゲームの大会ではプレイヤーが操作するキャラクターの数が多く、展開が速い。手動で観戦カメラを操作しようとすると様々な状況が起きている中、どこの地点に注目するかと観戦カメラのカメラワークの決定を短い時間で考えて行動しなければいけない。加えて、ずっと同じ視点や動きでは単調になってしまうため、どのタイミングで観戦カメラを違う視点や動きに切り替えるかも考慮する必要がある。上記の理由より、手動による観戦カメラの操作難易度は高い。そのため FPS ゲームの世界大会では観戦カメラを操作す

るためのカメラマンが存在する [2]。しかし、大会の配信を行うために必要な人材や負担は少なくないため、専属のカメラマンを雇うことが出来ないことがある。よって、実況者や解説者が観戦カメラを操作する場合があり、操作がおろそかになってしまうことがある。その結果観戦カメラがうまく動かすことが出来ず、大会の運営や観戦者が満足のいく映像を作成することが出来ない場合がある。

映像を生成するための問題に対しては研究が盛んに行われている。窪田ら [3] は実際のサッカーの試合において配信の負担を軽くするために試合中のサッカーの映像の自動生成を行うということを研究した。実際のゲームに関する研究に関しても Burelli[4] はゲーム内のキャラクターの様子などを映画調な映像を作成する研究を行った。Yeung[5] は大会などでの実際の観戦カメラを操作するための UI に関する研究を行い、Gior[6] はゲーム内でのカメラを自動制御する研究を行った。Gleicher ら [7] は映像の自動生成に関しても映像を参考にして仮想空間における仮想カメラを動かす研究を行った。Galvane ら [8] は仮想カメラがどのような移動を行うかの経路を自動生成する研究を行い、Christie ら [9] は仮想カメラの自動制御に関する研究を行った。Unity[10] ではそれをシステム [11] として開発を行っている。仮想空間内での映像の演出に関する研究としては、Xu ら [12] は仮想空間内でのカメラワークや演出の方法を支援する研究を行った。他には、Li らは [13] は実際の映像制作チームのような役割分担を行いながら映像を創作する研究を行い、また Amerson ら [14] と Chen ら [15] は物語にそってカメラワークをリアルタイムで制御を行う研究を行った。

ゲーム内における観戦カメラを自動制御する既存の手法としては、Valve SoftWare[16] では観戦カメラをキャラクター視点に自動制御で切り替えるシステムを実際のゲームに実装した。しかし、視点が一人称視点限定であり、背後や左右で何が起きているか見ることが出来ない。なおかつ、視点が頻繁に入れ替わる場合などは何が起こっているかもわからないことがあるため、状況把握がし辛いという問題点がある。

本研究の目的は俯瞰視点における観戦カメラを AI で自動制御することである。研究の流れとしては FPS ゲームを自作し、全てのキャラクターを AI で制御して戦闘を行う。その戦闘の様子をアルゴリズムを適用した自動制御する観戦カメラによって撮影する。その動画と既存の観戦カメラシステムに似せたシステムを実装した動画の 2 つを作成した。その 2 つの動画と実際の大会の動画を比較することによって評価を行った。

結果としては本研究の手法も有用であると言えたが、被験者によっては本研究の手法よりも既存の観戦カメラシステムの方が有用と評価することもあった。

1.2　本論文の構成

本論文での構成を説明する。2 章では本研究の手法の説明をする。3 章では評価実験の結果を記載する。4 章ではまとめについて記載する。

第 2 章

提案手法

本研究の目的は俯瞰視点での観戦カメラの自動制御することである。方針としては FPS ゲームの大会の観戦において、観客が求める映像を戦闘を行っている状況を映しているものであると仮定し、その状況を求めるため、戦闘中のキャラクター間の距離とキャラクターが向いている方向に着目した。ゲーム内における観戦カメラに対し、観戦カメラの注目地点を求めるために各キャラクターの距離と方向ベクトルのなす角の余弦の値を用いて計算を行う。その後、求めた注目地点におけるカメラワークを決定することで観戦カメラの自動制御を実現する。

2.1 検証用ゲーム

本研究で用いるゲームのキャラクターは全員で 10 人であり、5 対 5 のチームに分ける。全キャラクターに 1 から 10 までの番号を振り分ける。ルールは敵チームのキャラクターを規定数まで倒しきることである。キャラクターは全て AI で自動制御する。キャラクターの AI は移動、攻撃の 2 パターンの行動がある。移動に関してはあらかじめ設定した地点を巡回する。その最中に敵チームのキャラクターを発見すると攻撃する。キャラクターは倒されると設定した地点に復活する。敵チームのキャラクターを規定数まで倒しきるとゲームは終了する。ステージは壁と床のみで構成し、床は観戦カメラの注目地点に影響し、壁は観戦カメラの注目地点とカメラワークの選

択の両方に影響する。

2.2　提案手法の処理の流れ

提案手法の流れとして、初めに各キャラクターの座標と方向ベクトルから観戦カメラの注目地点を算出する処理を行う。この処理を処理 1 とする。次にキャラクター間の位置関係や行動から観戦カメラのカメラワークを決定する処理を行う。この処理を処理 2 とする。その後、観戦カメラを切り替えるタイミングが来た場合はもう一度処理 1 と処理 2 を行う。図 2.1 はその流れをまとめた図である。

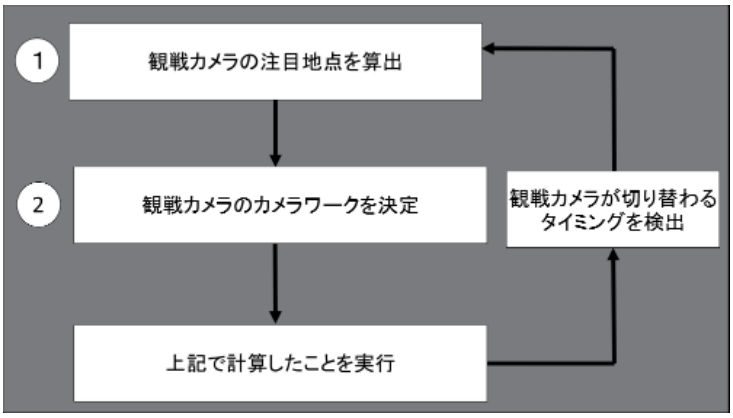

図 2.1　提案手法の処理の流れ

2.3　観戦カメラの注目地点

初めに各キャラクターの座標と方向ベクトルを取得する。次に各キャラクター間の距離を計算する。初めに任意のキャラクターを 1 人決める。任意のキャラクターから敵チームのキャラクターに向かって線を引き、その線が途中で壁か床と衝突していないか判定する。線が壁か床と衝突していない場合、任意のキャラクターと敵チームのキャラクターとの距離の計算を行う。線が壁か床と衝突している場合、任意のキャラクターと敵チームのキャラクターとの距離の計算を行わない。同様の処理を敵チームのキャラクター全員に対して行い、任意のキャラクターと敵チー

ムのキャラクター全員との衝突判定と距離を求める。一連の処理を全キャラクターで行うことによって、各キャラクターと敵チームのキャラクター全員との距離を算出する。

続けて、各キャラクターの方向ベクトルのなす角の余弦の値を計算する。2 つのベクトルからなす角の余弦の値を求める式として以下の式 (2.1) を用いる。式中の $\cos\theta$ がなす角の余弦の値であり、$\mathbf{A}$ と $\mathbf{B}$ はそれぞれベクトルである。

$$\cos\theta = \frac{\mathbf{A}\cdot\mathbf{B}}{|\mathbf{A}||\mathbf{B}|} \tag{2.1}$$

初めに任意のキャラクターを 1 人決める。次に任意のキャラクターと敵チームのキャラクターが向き合っているかの判定として、任意のキャラクターの方向ベクトルと任意のキャラクターから見た敵チームのキャラクターへの方向ベクトルを求め、なす角の余弦の値を算出する。任意のキャラクターの方向ベクトルを式 (2.1) 中の $\mathbf{A}$ に代入し、任意のキャラクターから敵チームのキャラクターへの方向ベクトルを式 (2.1) 中の $\mathbf{B}$ に代入する。

以下の図 2.2 は求めたなす角の余弦の値が 0 より大きい場合を示した図である。

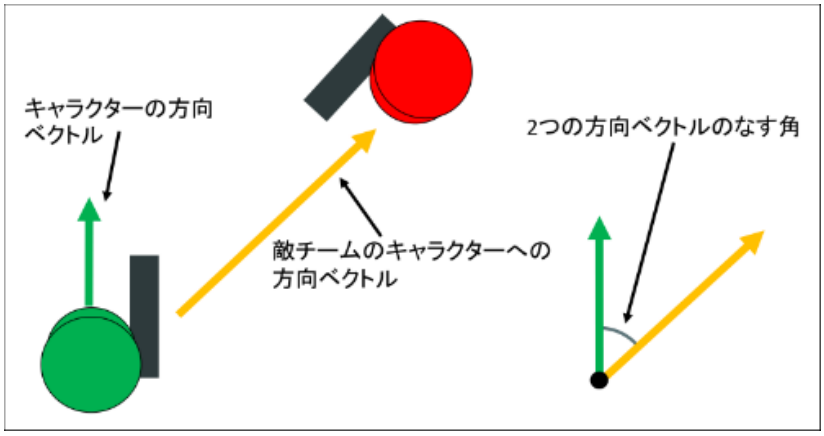

図 2.2　なす角の余弦の値が 0 より大きい時の状況

以下の図 2.3 は求めたなす角の余弦の値が 0 以下の場合を示した図である。

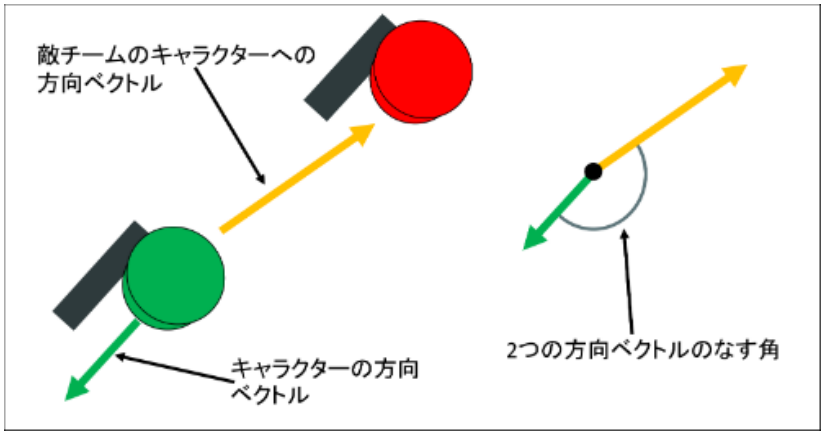

図 2.3　なす角の余弦の値が 0 以下の状況

同様に、敵チームのキャラクターの方向ベクトルと敵チームのキャラクターから見た任意のキャラクターへの方向ベクトルを求め、なす角の余弦の値を算出する。この場合、敵チームのキャラクターの方向ベクトルを式 (2.1) 中の $\mathbf{A}$ に代入し、敵チームのキャラクターから見た任意のキャラクターへの方向ベクトルを式 (2.1) 中の $\mathbf{B}$ に代入する。求めた 2 つのなす角の余弦の値がどちらも 0 より大きい場合、任意のキャラクターの方向ベクトルと敵チームのキャラクターの方向ベクトルのなす角の余弦の値の計算を行う。それぞれのキャラクターの方向ベクトルのなす角の余弦の値の計算にも式 (2.1) を使用する。その場合、それぞれのキャラクターの方向ベクトルを式 (2.1) 中の $\mathbf{A}$ と $\mathbf{B}$ に代入する。以下の図 2.4 は任意のキャラクターの方向ベクトルと敵チームのキャラクターの方向ベクトルが描くなす角の図である。

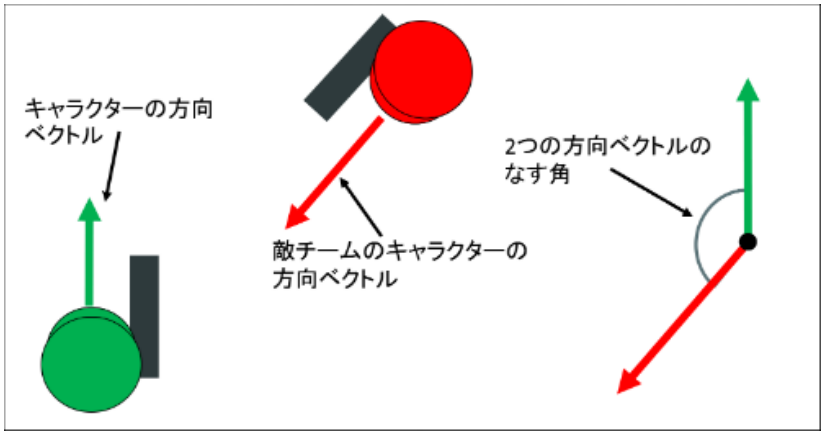

図 2.4　各キャラクターの方向ベクトルが描くなす角の図

求めた 2 つのなす角の余弦の値が少なくともどちらかが 0 以下の場合、任意のキャラクターの方向ベクトルと敵チームのキャラクターの方向ベクトルのなす角の余弦の値の計算を行わない。同様の処理を敵チームのキャラクターの全員に対して行い、任意のキャラクターと敵チームのキャラクター全員との向き合っているかの判定となす角の余弦の値の値を求める。一連の処理を全キャラクターで行うことによって、各キャラクターと敵チームのキャラクター全員とのなす角の余弦の値を算出する。

次に 1 人のキャラクターと敵チームのキャラクター全員との距離と方向ベクトルのなす角の余弦の値を比較する。距離と方向ベクトルのなす角の余弦の値は数値が小さい順で順位付けを行う。そして 1 人のキャラクターにおける、距離と方向ベクトルのなす角の余弦の値のそれぞれの順位から合計順位を求める。合計順位は以下の式 (2.2) の r の値が小さい順に並べる。以下の式中 a は敵キャラクターとの距離の順位であり、b は敵キャラクターの方向ベクトルとのなす角の余弦の値の順位である。

$$r = \frac{a+b}{2} \tag{2.2}$$

合計順位が 1 位になった敵キャラクターをインタレストキャラクターとする。同様の処理を全キャラクターに対して行う。

そして、全キャラクターとそのキャラクターのインタレストキャラクターの距離と、方向ベクトルのなす角の余弦の値、戦闘を行っているかの判断を行い、それぞれ 3 つの順位付けを行う。その後、3 つの順位の平均から総合順位 1 位のキャラクターとそのキャラクターのインタレストキャラクターを求める。全キャラクターとそのキャラクターのインタレストキャラクターで計算した距離と方向ベクトルのなす角の余弦の値を数値が小さい順で順位付けを行う。そして、戦闘を行っているどうかは判別として射撃を行っているかどうかで判断する。射撃を行っている場合は戦闘を行っているものとして順位を 1 位に設定し、戦闘を行っていないものの順位を 10 位に設定する。この 3 つの順位の平均から総合順位を求める。総合順位が 1 位のキャラクターとインタ

レストキャラクターを求める。

最後に総合順位 1 位のキャラクターとインタレストキャラクターの交戦予定位置を取得する。求めた総合順位 1 位のキャラクターとインタレストキャラクターの中間地点を交戦予定位置として取得する。交戦予定位置は観戦カメラの注目地点とする。以下の図 2.5 は本節で述べた処理の流れを示した図である。

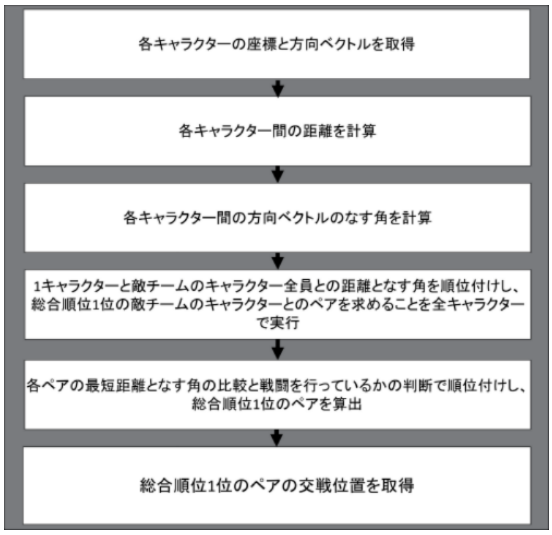

図 2.5　観戦カメラの注目地点の算出する処理の流れの図

2.4　観戦カメラのカメラワーク

注目地点におけるカメラワークに関しては Web サイト [17] の情報と実際の FPS ゲームの大会においてよく使われているカメラワークを元に基本の 5 パターン用意した。カメラワークの選択の際、観戦カメラと注目地点との間を壁が遮っていないかの判別のため、注目地点から観戦カメラに向かって線を引き、途中で壁と衝突していないかで判断する。衝突していない場合は総合順位 1 位のキャラクターとインタレストキャラクターの距離が近いか、戦闘が起きているか、同じカメラワークが連続していないかからカメラワークを選択し、決定する。衝突している場合は衝突していない場合と同じ処理か衝突地点へと観戦カメラを移動する、もしくは選択肢から除外し

てカメラワークの再選択ということを行う。

2.4.1　観戦カメラのカメラワークの種類

各パターンのカメラワークに関しては Web サイトの情報を元に実際の FPS ゲームの大会におけるカメラワークを見て、ピックアップしたものを模倣するような動きを 5 パターン抽出した。この 5 パターンの動きを三人称カメラワーク、全体カメラワーク、パンカメラワーク、ドリーカメラワーク、ドリーインアウトカメラワークと定義する。次からは各パターンのカメラワークに関しての詳細を記述していく。

初めに三人称カメラワークについて述べる。総合順位 1 位のキャラクターの背後に対し少し上に補正した座標に観戦カメラを配置し、少し見下ろす形のアングルでキャラクターの背後を常に撮る。このカメラワークで観戦カメラが映す図とキャラクターと観戦カメラを横から見た図のイメージ図が図 2.6 である。

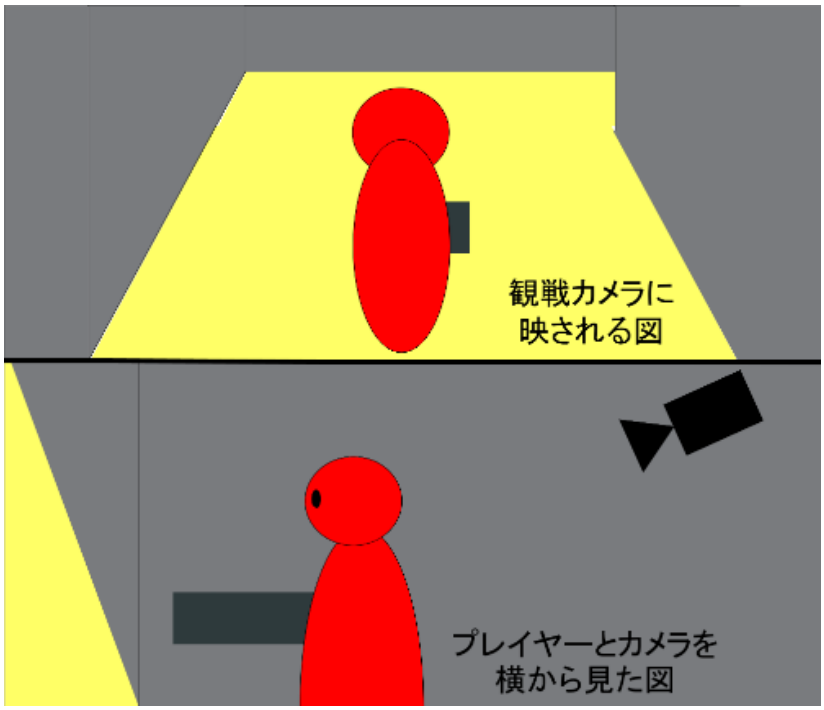

図 2.6　三人称カメラワークのイメージ図

次に全体カメラワークについて述べる。算出した注目地点を真上から見下ろすような座標に観戦カメラを配置する。観戦カメラのアングルに関しては真下に向くように変更する。このカメラワークで観戦カメラが映す図と観戦カメラを横から見た図のイメージが図 2.7 である。

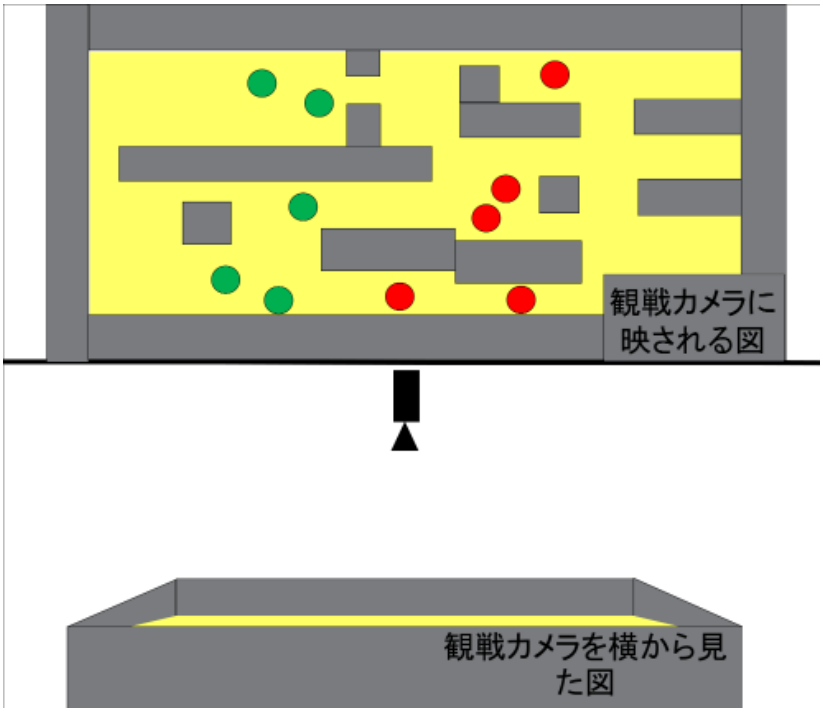

図 2.7　全体カメラワークのイメージ図

続けて、パンカメラワークについて述べる。座標の決め方に関して、まずはキャラクター達を横から撮るような地点に配置する。アングルに関して、現在地点から総合順位 1 位のキャラクターへの方向ベクトルを $\mathbf{S}$ とし、現在地点から敵チームのキャラクターへの方向ベクトルを $\mathbf{E}$ とする。総合順位 1 位のキャラクターへの方向ベクトルから敵チームのキャラクターへの方向ベクトルまで変化している方向ベクトルを $\mathbf{C}$ として、以下の式 (2.3) を用いて求める。式中 t は媒介変数であり、0 から 1 の間で変わっていく。

$$\mathbf{C} = \frac{\mathbf{S} + (\mathbf{E} - \mathbf{S})t}{|\mathbf{S} + (\mathbf{E} - \mathbf{S})t|} \tag{2.3}$$

総合順位 1 位のキャラクターの地点へのアングルを t=0 の時の観戦カメラのアングルとし、敵チームのキャラクターの地点へのアングルを t=1 として計算を行う。後は t の値を 0 から始め、

フレーム毎に t を一定値で上昇させていくことでカメラワークとしている。このカメラワークでの観戦カメラが映す図とキャラクターと観戦カメラを横から見た図のイメージが図 2.8 である。

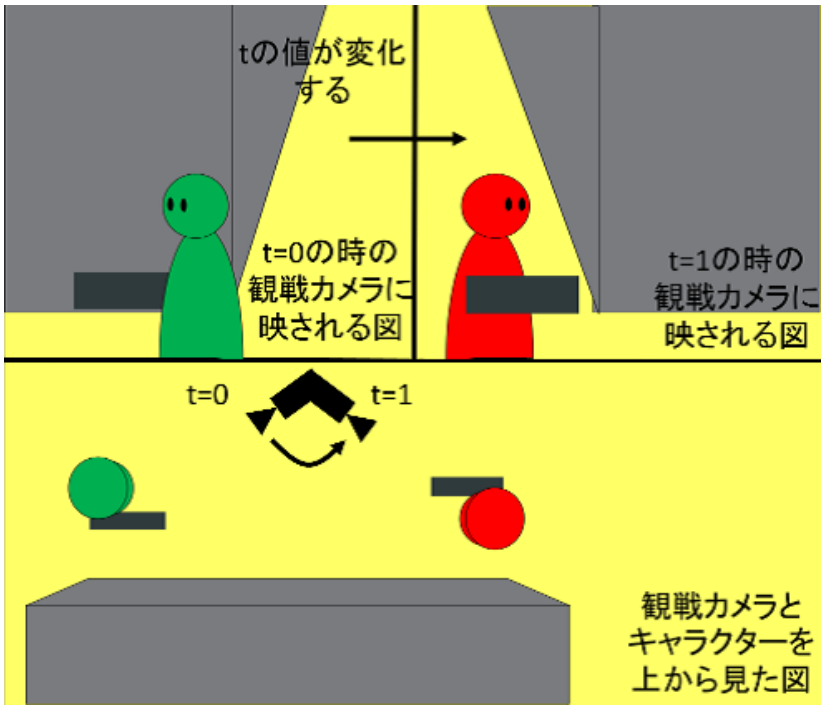

図 2.8　パンカメラワークのイメージ図

次に、ドリーカメラワークについて述べる。この処理ではベジエ曲線を作成し、それに則って観戦カメラが移動を行う。まずはベジエ曲線の作成するためのアンカーポイントを 4 個設定する。初めに総合順位 1 位のキャラクターとその敵チームのキャラクター間を 5 等分する座標を求める。そこで求めた 4 個の座標に対して、上方向への補正を行う。補正として 4 個の座標のうち、各キャラクターに近い 2 つの座標の y 成分を $y + H$ とし、残りの 2 つの座標の y 成分を $y + I$ とする。H と I は補正を行うための数値であり、$H < I$ は常に成立するものとする。補正を行った 4 つの座標をアンカーポイントの座標とする。そのアンカーポイントの座標を元にしてベジエ曲線 [18] を描く。以下の式 (2.4) 中の $\mathbf{P}$ は計算した観戦カメラの座標であり、$\mathbf{A}$ はアンカーポイントの座標であり、t は 0 から 1 の間で変化する変数である。

$$\mathbf{P} = (1-t)^3\mathbf{A}_0 + 3(1-t)^2(t)\mathbf{A}_1 + 3(1-t)(t)^2\mathbf{A}_2 + (t)^3\mathbf{A}_3 \tag{2.4}$$

t の値を変化することによって曲線上の座標を求め、観戦カメラを求めた座標に移動する。観戦カメラの視線は座標が変わっても注目視点に注目する。このカメラワークで観戦カメラが映す図とキャラクターと観戦カメラを上から見た図のイメージが図 2.9 である

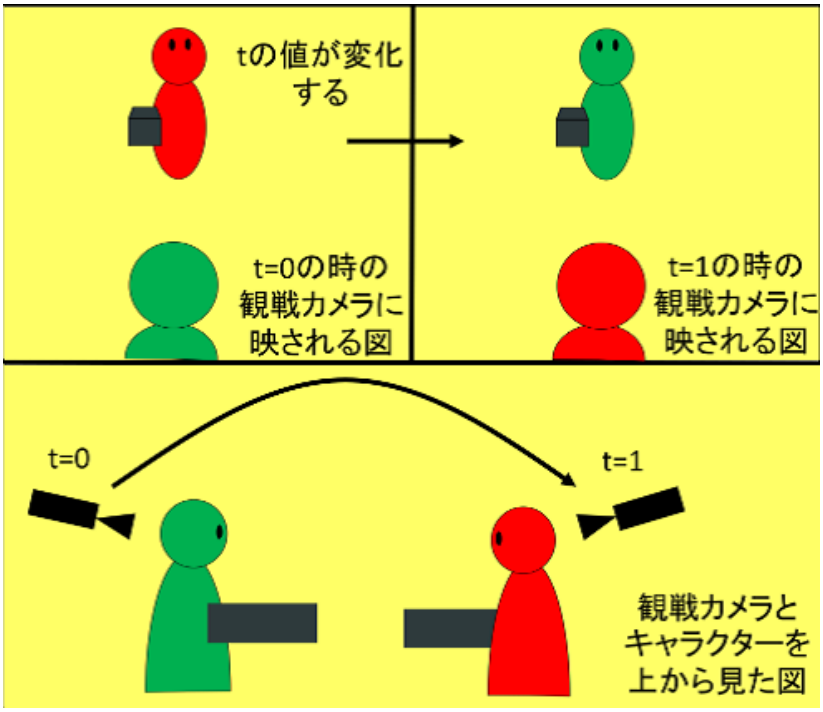

図 2.9　ドリーカメラワークのイメージ図

最後にドリーインアウトカメラワークについて述べる。常に総合順位 1 位のキャラクターと相手キャラクターを真上から観戦カメラで映す。式 (2.5) は 2 人のキャラクターが画面内に収まるように、観戦カメラと注目地点の距離を変える式である。以下の式 (2.5) の d は注目地点からどれだけ距離を取る必要があるかという値であり、r は総合順位 1 位のペアの距離の半分の値である。m は観戦カメラの上下左右おける画面端の中で一番近い画面端から両キャラクターまでどれだけ

距離を空けるかに関する変数である。F は観戦カメラの視野角の度数法の値を代入している。

$$d = \frac{2(r+m)}{F} \tag{2.5}$$

このカメラワークで観戦カメラが映す図とキャラクターと観戦カメラを上から見た図のイメージが図 2.10 である

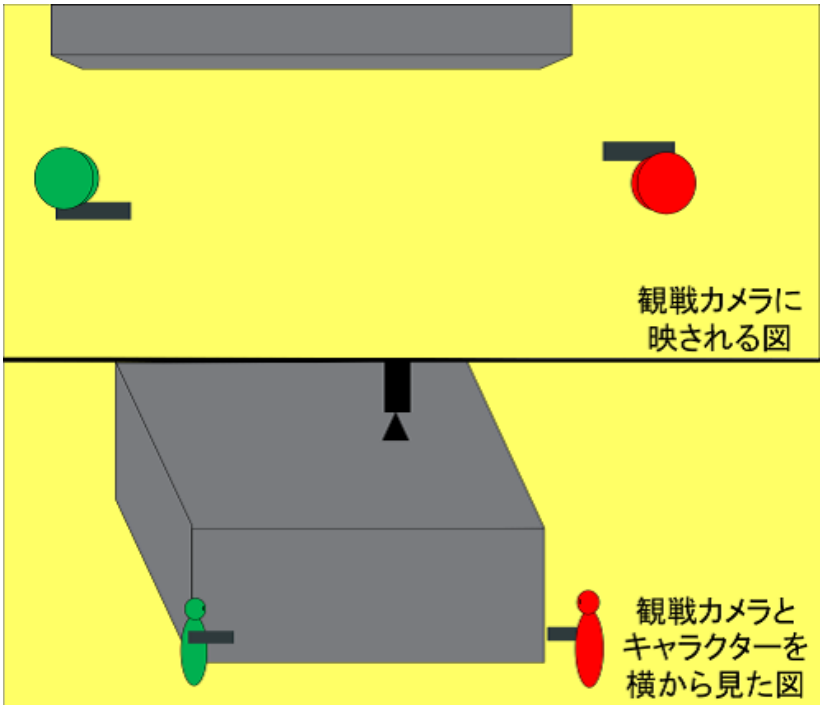

図 2.10　ドリーインアウトカメラワークのイメージ図

2.4.2　観戦カメラのカメラワークの選択

アルゴリズムの詳細な説明としては前節で求めた総合順位 1 位のキャラクターとインタレストキャラクターの距離が近いかどうかによって判別し、その後戦闘が起きているかどうかを判別する。距離が近いかの判別方法はあらかじめ設定した値より低いかどうかで判断する。距離が近い場合はドリーカメラワークか、パンカメラワークの内、前回のカメラワークとは違うものを選択する。距離が近くない場合は戦闘が起きているかどうかの判別を行う。戦闘が起きていない場合

は三人称カメラワーク、全体カメラワーク、ドリーインアウトカメラワークのいずれかの内、前回と前々回のカメラワークとは違うものを選択する。戦闘が起きている場合は三人称カメラワークかドリーインアウトカメラワークのどちらかの内、前回のカメラワークとは違うものを選択する。決めた時間まで経過した場合、もしくは観戦カメラと注目地点の間に壁がある場合は観戦カメラのカメラワークを再選択する。以下の図 2.11 はアルゴリズムの流れをフローチャートにしたものである

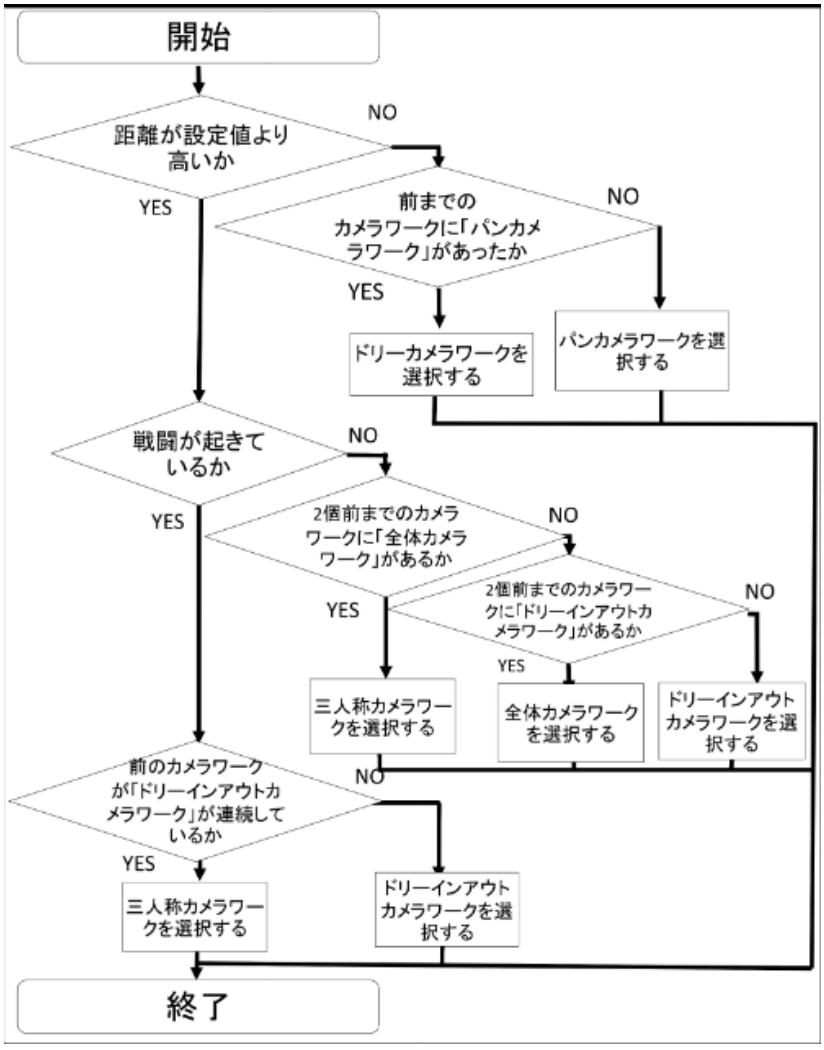

図 2.11　どのカメラワークを選択するかのフローチャート図

選択したカメラワークを実際のカメラワークとして決定する前に、観戦カメラと注目地点の間

を壁が遮っていないかの判定を、選択したカメラワーク毎の始点から終点まで線を引き、途中で壁とぶつかっていないかで判断する。選択したカメラワークが三人称カメラワークかパンカメラワークである場合、始点はキャラクター、終点は観戦カメラである。選択したカメラワークが全体カメラワークかドリーインアウトカメラワークである場合、始点は注目地点、終点は観戦カメラである。選択したカメラワークがドリーカメラワークである場合、始点は注目地点、終点はベジエ曲線の全アンカーポイントである。

衝突していない場合は選択したカメラワークを実際のカメラワークとして決定する。衝突している場合は選択したカメラワークの種類によって処理を変える。選択したカメラワークが三人称カメラワークである場合、三人称カメラワークを選択肢から除いてカメラワークの再選択を行う。選択したカメラワークが全体カメラワークとドリーインアウトカメラワークである場合、そのカメラワークを実際のカメラワークとして決定する。選択したカメラワークがパンカメラワークである場合、観戦カメラが線と壁の衝突地点に移動する。選択したカメラワークがドリーカメラワークである場合、ベジエ曲線のアンカーポイントを線と壁の衝突地点に移動する。

2.5　観戦カメラが切り替わるタイミング

観戦カメラが壁によって遮られ、注目地点を映すことが出来ない場合、もしくはあらかじめ設定した数値の時間まで経過した場合に観戦カメラの種類が切り替わる。しかし、観戦カメラが戦闘を映している場合、観戦カメラは切り替わらない。

第 3 章

結果と評価

本章では Unity で作成した FPS ゲームにおいて AI 同士で戦闘を行わせ、その様子を提案手法で述べたアルゴリズムを実装した結果と、それによって人が手動で操作する観戦カメラに近づいたかどうかを検証した。

3.1　実行結果

実行結果として観戦カメラの注目地点と観戦カメラのカメラワークについてそれぞれ記載していく。

3.1.1　観戦カメラの注目地点

観戦カメラの注目地点の算出結果としては戦闘が起きるであろう地点に観戦カメラの注目地点には向けることに関しては想定通りであった。以下の図 3.1 は戦闘予測が出来ている状況を示す図である。

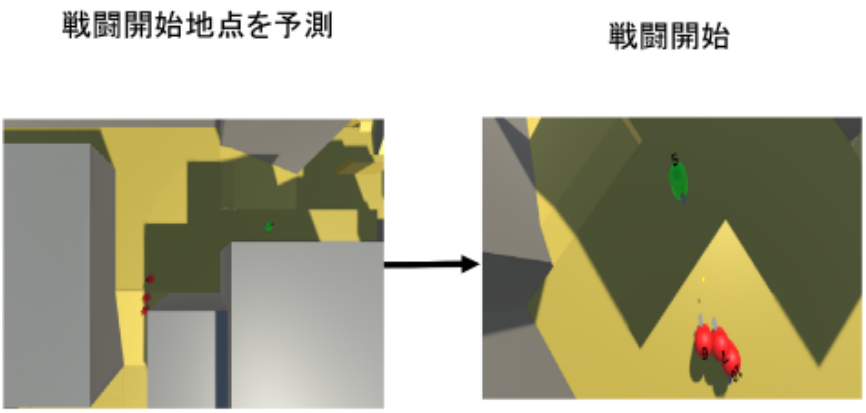

図 3.1　戦闘が起きる前に予測出来ている状況

しかし、観戦カメラの注目地点の算出とカメラワークの決定を行った直後に戦闘が終わってしまい、他の地点での戦闘を映すことが出来ない状況も少なくなかった。以下の図 3.2 は戦闘が終わってしまった地点を映している状況を示す図である。

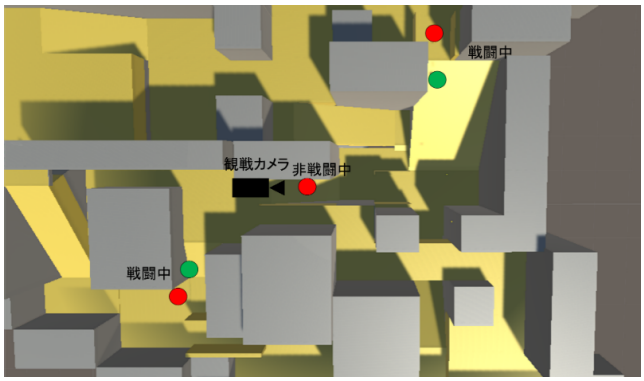

図 3.2　戦闘が終わってしまった地点を映している状況

3.1.2　観戦カメラのカメラワーク

観戦カメラのカメラワークに関しては概ね予定していたものが出来た。詳細は個別に説明していく。観戦カメラのカメラワークの選び方や切り替わり方もキャラクターを適切に映すことが出来る観戦カメラのカメラワークを選ぶことが出来た。

三人称カメラワークに関しては現在使われている大会や既存のシステムとは何も変わらない

ものであると言えた。以下の図 3.3 は実際に FPS ゲームで実行した観戦カメラが映した画面である。

図 3.3　三人称カメラワークが映している画面

全体カメラワークに関しても既存のシステムとは違いがないと言えた。以下の図 3.4 は実際に FPS ゲームで実行した観戦カメラが映した画面である。

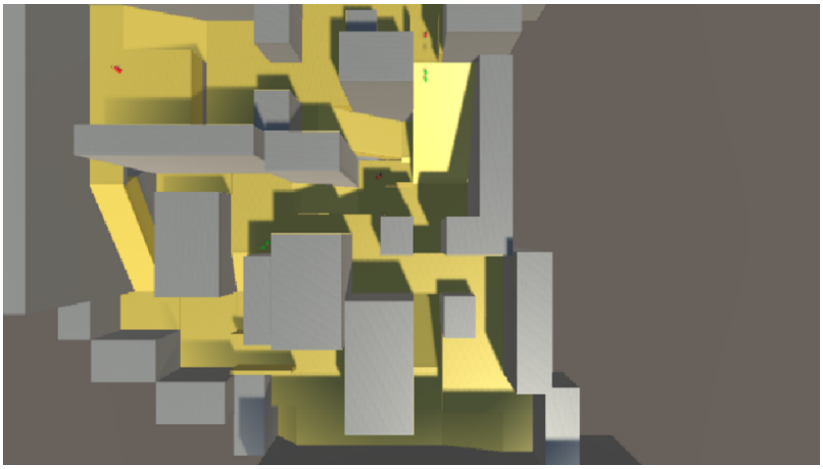

図 3.4　全体カメラワークが映している画面

パンカメラワークに関しては想定したカメラワークと同じものが出来たと言えた。更なる発展としては俯瞰視点以外のカメラワークを実装し、もう少し映画などで使われるようなカメラワークも実装出来るとさらに良くなると考えた。以下の図 3.5 は実際に FPS ゲームで実行した観戦カメラが映した画面である。

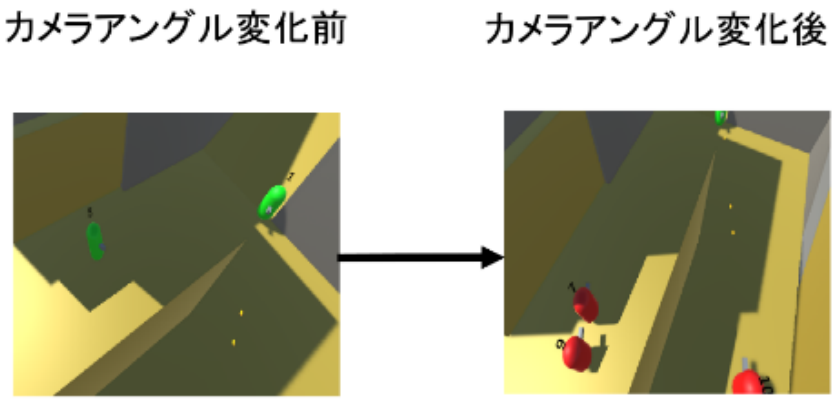

図 3.5　パンカメラワークが映している画面

ドリーカメラワークに関しては想定していたカメラワークと同じ観戦カメラの移動を作成することが出来たが、ステージの形状によって観戦カメラがどのような円を描くかを変えることが出来たならば、さらによくなると考えた。以下の図 3.6 は実際に FPS ゲームで実行した観戦カメラが映した画面である。

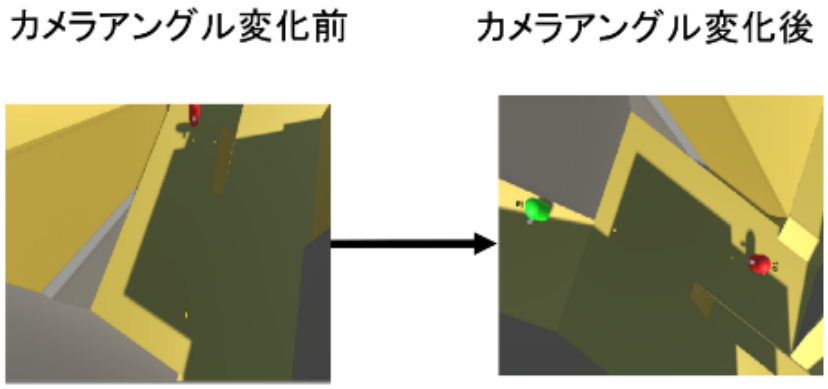

図 3.6　ドリーカメラワークが映している画面

ドリーインアウトカメラワークに関しては真上から見下ろす視点とはまた違うカメラワークであり、適切なカメラワークであると言えた。以下の図 3.7 は実際に FPS ゲームで実行した観戦カメラが映した画面である。

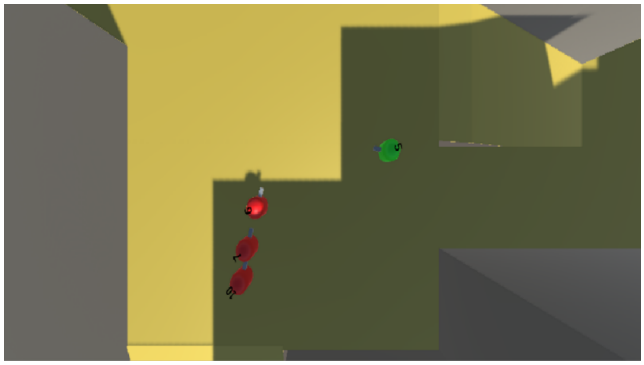

図 3.7　ドリーインアウトカメラワークが映している画面

3.2　評価実験

3.2.1　実験方法

本研究の評価実験として、提案手法を適用して撮影した動画が既存の手法を真似た手法を適用した動画と比べて、実際の大会の動画に似ているか、面白いか、状況把握しやすいかということを検証した。実験の方法として、はじめに様々なカメラアングルを含んだ実際の大会の動画 [19] から抜粋した 2 分間の動画を見てもらった。次に自作した FPS ゲームの試合に対して、提案手法を適用して撮影した 2 分間の動画と、既存の手法を真似た、戦闘が発生した地点に切り替わり、一人称視点しか映さない手法を適用して撮影した 2 分間の動画をそれぞれ見てもらった。その後、実際の大会の動画と自作した FPS ゲームの試合を撮影した 2 つの動画についてのアンケートを取った。アンケートの内容は大会の動画とどちらの動画の方が似ているか、なぜその動画が似ているかと思ったか、どちらの動画の方が面白いか、どちらの動画の方が状況把握できるかと自由記入欄という項目に答えてもらった。

3.2.2　実験結果

FPS ゲームの経験や大会の動画をあまり見ない人や日頃から FPS ゲームで遊んでいる人など様々な被験者 12 人にアンケートを答えてもらった。評価実験の結果として、大会の動画と似て

表 3.1　評価実験の結果

アンケート項目 / 動画の種類	似ているか	面白いか	状況把握出来たか
提案手法を適用した動画	10	10	12
既存の手法を適用した動画	2	2	0

いる動画としては本研究の手法を適用した動画の方が割合が多かった。選ばれた理由としては観戦カメラのカメラワークの動き方が似ているや、映し方が似ている、提案手法を適用した動画の方が印象に残ったというものであった。逆に一人称視点限定の動画を選んだ理由としては一人称視点の方が印象に残ったというものであった。どちらの動画の方が面白いかと、どちらの動画の方が状況把握出来たかという質問に対しても本研究の手法を適用した動画の方が割合が多かった。表 3.1 は評価実験の結果を表にしたものである。

3.3　考察

本研究の手法では座標や方向ベクトルといったどのようなゲームにも存在する数値を参照して、注目地点の算出とカメラワークの決定を行っているため、多くの FPS ゲームにも実装できると考えられる。問題点としては武器の種類や特殊能力といった特殊な数値を考慮することが出来ないことである。

評価実験のどちらの動画の方が実際の大会の動画の方が似ているかという項目では、提案手法を適用した動画の方が似ているという割合が多かった。そして、選ばれた時の理由として、俯瞰視点でのアングルが実際の観戦カメラのカメラワークに似ていることや、試合の流れの見せ方が似ているというものがあった。以上より、本研究の手法を適用した動画のカメラワークの方が、既存の手法を真似た手法を適用した動画のカメラワークより、大会の動画のカメラワークに似ていると言える。加えて、どちらの動画の方が面白いかという項目と、どちらの動画の方が状況把握しやすいかという項目のどちらも提案手法を適用した動画の方が割合が多かった。特に、どちら

の動画の方が状況把握しやすいかという項目では被験者全員が提案手法を適用した動画の方を選んだため、状況を把握するのは三人称視点での動画の方が良いとわかった。しかし、どちらの動画の方が面白いかという項目で一人称視点の動画の方を選んだ被験者の意見として、臨場感があるということもあがった。よって多くの時間は三人称視点でのカメラワークを行い、臨場感や没入感が必要な場合は一人称視点でのカメラワークの方がよいのではないかと考察した。

第 4 章

まとめ

本研究では 1 人のカメラマンとして現在の大会などで手動で操作されている観戦カメラを自動で再現することが目的である。そのため、観戦カメラがどの地点に注目すべきかのアルゴリズムとしてはキャラクター間の距離や方向ベクトルを元に算出し、観戦カメラがどう動くかはその時点でのキャラクターの状態やキャラクター間の距離によって決定づけるという手法を提案した。想定していた観戦カメラのカメラワークとしては戦闘が起きるであろう地点やキャラクターに対して観戦カメラの視点を向け、カメラワークとして飽きがこないように同じ観戦カメラのカメラワークが続かないようにし、戦闘シーンのような見所を撮り損ねることのないものである。結果としては大会と見比べても遜色ないものであると自己評価した。評価実験の結果として、提案手法を適用した動画の方が実際の大会の状況には似ているが、状況によっては一人称視点に切り替えた方がよいものになると考察出来た。今後の目的としては、全体の観戦カメラのディレクションも自動で制御を行うシステムも実装することが出来たならばさらに大会での需要も上がると考えた。三人称視点での観戦カメラのカメラワークも今以上にバリエーションを増やすことや戦闘以外での注目地点の検出も行うことによって、本研究の手法では検出の難しい戦略的行動に対しても注目地点を算出することが出来るのではないかと考えた。アルゴリズムを変えていけば様々な映像を生み出すことが可能であるため、現在以上の映像を作成できる手法を今後は提案して

いく。

なお本研究は芸術科学会 NICOGRAPH2018 における”FPS ゲームの試合における観戦カメラ AI に関する研究”[20] として発表した内容を含む。

謝辞

本研究を行うにあたり、ご指導くださった渡辺先生をはじめとする指導教員の方々、ご支援ご協力をいただきました. 心より感謝いたします。また評価実験のアンケートにお答えいただいた方々にも深く感謝します。

参考文献

[1] In Twitch.tv. すべてカテゴリー twitch.tv. `https://www.twitch.tv/directory`. 参照:2018/12/21.

[2] クドータクヤ. Rizest の縁の下の力持ち、現場スタッフから見た esports シーンとは 後編-esports を支える人々-. `https://alienwarezone.jp/post/751`, 2018. 参照:2018/12/21.

[3] 窪田進太郎, 青木康雄, 熊野雅仁. ディジタルカメラワークを用いたボールと選手の状況認識に基づくサッカー映像の自動生成. *MIRU2005*, Vol. 2005, 画像の認識・理解シンポジウム, pp. IS3–117, 2005.

[4] Paolo Burelli. *Interactive Virsual Cinematography*. PhD thesis, IT University of Copenhagen, 2012.

[5] Eva M. Yeung. Viewer interface first person shooter streaming. Master's thesis, MIT, 2016.

[6] John GIors. The full spectrum warrior camera system. In *Game Developers Conference 2004*, 2004.

[7] Micheal Gleicher and Andrew Witkin. Through-the-lens camera control. *Proceeding SIGGRAPH '92*, Vol. 26, No. 2, pp. 331–340, 1992.

[8] Qunetin Galvane, Marc Christie, Christophe Line, and Remi Ronfard. Camera-on-rails:

Automated computation of constrained camera paths. In *the 8th ACM SIGGRAPH Conference on Motion in Games*, pp. 151–157. ACM, 2015.

[9] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. *Computer Graphics*, Vol. 27, No. 8, pp. 2197–2218, 2008.

[10] Unity Technologies. Unity. `https://unity3d.com/jp`, 2018. 参照:2018/12/21.

[11] Unity. Cinecast. `https://create.unity3d.com/cinecast`. 参照:2019/02/11.

[12] Chanchan xu, Guanzheng Fei, and Honglei Han. Cambridge: A bridge of camera aesthetic between virtual environment designers and players. In *the 16th ACM SIGGRAPH International Conference*, p. Short Paper 54. VRCAI, 2018.

[13] Tsai-Yen Li and Xiang-Yan Xiao. An interactive camera planning system for automatic cinematographer. In *the 11th International Multimedia Modelling Conferenc*, pp. 310–315, 2005.

[14] Daniel Amerson and Shaun Kime. Real-time cinematic camera control for interactive narratives. In *the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pp. 369–369. ACM, 2005.

[15] Chia-Hou Chen and Tsai-Yen Li. Context-aware camera planning for interactive storytelling. In *Computer Graphics, Imaging and Visualization (CGIV), 2012 Ninth International Conference on*, pp. 43–48, 2012.

[16] ESEA. Inc esea. gotv for couter-strike: Global offensive released all new spectator tools revealed. `https://www.youtube.com/watch?v=xIH0DB0HQjA.`, 2012. 参照:2018/12/21.

[17] patrimonio. 【動画ビギナー必見!】知っておきたい基本のカメラワーク. `https://pixta.jp/channel/?p=15795`. 参照:2018/12/21.

[18] Pomax. A primer on bézier curves. `https://pomax.github.io/bezierinfo/ja-JP/`.

参照:2019/02/04.

[19] overwatch League. Full match — dallas fuel vs. seoul dynasty — stage 1 week 1 day 1. https://www.youtube.com/watch?v=FB5NWzsA7AQ&t=1515s. 参照:2018/12/21.

[20] 廣里直人, 渡辺大地, 阿部雅樹. FPS ゲームの試合における観戦カメラ AI に関する研究. *NICOGRAPH2018*, 2018.