Vol.2018-CG-172 No.8
Vol.2018-DCC-20 No.8
Vol.2018-CVIM-214 No.8
2018/11/7

高解像度ディスプレイにおける ドットグラフィックゲームの開発手法

牧野 聖久[†1]　松河 剛司[†1]

概要：Unity を用いて開発したドットグラフィックを用いたアクションゲームについて，高解像度ディスプレイにおいてドットグラフィックをピクセルパーフェクトで表示する手法や，追加カメラを用いて実装したドットグラフィックの表現を崩さない画面ズームの手法について述べる．

キーワード：ゲーム開発，ゲームエンジン, Unity，ドットグラフィック，ピクセルパーフェクト

Development Method of Dot Graphic Game Adapted to High Resolution Display

KIYOHISA MAKINO[†1]　TSUYOSHI MATSUKAWA[†1]

1. はじめに

近年, Unity や Unreal Engine といったゲームエンジンと呼ばれるゲームを開発する為のアプリケーションが無料で使用できるようになり，個人や学生でもゲームを気軽に開発できるようになった．前述のゲームエンジンは基本的に 3D 空間上で表現されるゲーム(以下，3D ゲーム)の開発を想定しているものだが, 3D グラフィックと比べて 2D グラフィックの方が軽量であったり，モデリングやレンダリングなどの 3D グラフィックを作成するために必要な技能と比べて，イラストやドットグラフィックなどの 2D グラフィックを作成する方が必要な技能が少なく容易であったりする為，個人や学生のゲーム開発では 2D 空間上で表現されるゲーム（以下, 2D ゲーム）の開発も広く行われている．

2D ゲームのグラフィックには四角形ピクセル(以下，ドット)の集合で描かれるドットグラフィックが用いられることが多い．ドットグラフィックとは，本来解像度の低いディスプレイでグラフィックを表現するために用いられた手法であるが，ディスプレイ解像度の基準が高くなった今，荒いドットでグラフィックを表現する必要はなくなってきている．しかし最近でも「UNDER TALE[1]」や「ショベルナイト[2]」といったドットグラフィックを用いたゲームがリリースされ，人気を博している．これは，ドットグラフィックが軽量なデータで低解像度でも表現できるという本来の目的とは別に，少ない情報量でプレイヤーに想像力を駆り立てるほか，古くからゲームに用いられていたことからゲームらしさを感じさせる効果のある表現技法として確立しているからであるといえる．このような理由から，ドットグラフィックのゲームは現在でも需要があり，高解像度ディスプレイでドットグラフィックを用いたゲームの開発が行われることは珍しくない．しかし，高解像度ディスプレイでドットグラフィックを表現しようとすると多くの問題が生じる可能性がある．本稿では，高解像度ディスプレイを用いてドットグラフィックを表現する際に起こる問題点と，その問題をゲームエンジンの Unity 上で解決する手法を述べる．

2. ピクセルパーフェクト

高解像度ディスプレイにおけるドットグラフィック表現のゲーム開発において重要となるのがピクセルパーフェクトである．ピクセルパーフェクトとはドットバイドットとも呼ばれ，ディスプレイを構成するピクセルの大きさとドットグラフィックを構成するドットの大きさが完全に一致している状態のことを指す．このピクセルパーフェクトという呼び方は, 2016 年に任天堂社から発売された「ニンテンドークラシックミニファミリーコンピュータ[3](以下，ミニファミコン)」という, 30 種類のファミリーコンピュータ向けソフトが内蔵されたハードでも，選択できる画面モードの一つとして用いられている．このモードを選択すると，くっきりとした正方形のドットでドットグラフィックが表示されるようになっている．しかしこの表示方法は擬似的にピクセルパーフェクトの状態を再現しているだけで，

†1 愛知工業大学大学院経営情報科学研究科
Aichi Institute of Technology Graduate School of Business Administration and Computer Science

Vol.2018-CG-172 No.8
Vol.2018-DCC-20 No.8
Vol.2018-CVIM-214 No.8
2018/11/7

実際にはドットサイズを拡大しており，ディスプレイのピクセルサイズとドットグラフィックのドットサイズが同じになるとは限らない為，本来のピクセルパーフェクトの意味とは違ってくる．

このように，ゲーム開発においてのピクセルパーフェクトという言葉は「ディスプレイの大きさに関わらず，ドットグラフィックを構成する全てのドットサイズが同じかつ正方形であり，すべてのドットが縦と横に均等に並べられた線(以下,グリッド)に沿って並べられているかのように，ズレが生じないよう整列している状態」を指すものとして扱われるように変化したといえる．本稿でも，ピクセルパーフェクトは上記の意味として扱う．

3. 高解像度ディスプレイにおける ドットグラフィック表現の問題点

低解像度ディスプレイなら本来の意味でのピクセルパーフェクトを行えば，ドットグラフィックが崩れることはなく問題ないが，高解像度ディスプレイでは擬似的にピクセルパーフェクトを行わないと多くの問題点が出てくる．

高解像度ディスプレイでドットサイズとピクセルサイズを同じ大きさにしようとすると，画面全体に対してドットグラフィックを使ったオブジェクトがとても小さくなってしまうという問題が起きてしまう．かといって，ドットグラフィックのオブジェクトを拡大して小ささの問題を解決しようとすると，中途半端な拡大をすることで 1 ドットが正方形にならないといった問題や，ドットグラフィックの拡大縮小により異なるドットサイズが一画面上に存在するようになるといった問題，ドット単位ではなく，ピクセル単位でオブジェクトが移動するため，グリッドに沿ったドットの並びを崩すといった問題のように多くの問題が出てきてしまい，グラフィックの統一感を崩す原因となってしまう．スマートフォン向けのゲームでは，ドットグラフィックを使ったオブジェクトに拡大縮小を行なっているため，オブジェクトごとに拡大率が違うため画面内のドットサイズが統一されていないゲームや，ドットグラフィックの移動が整列を意識していない，ドットグラフィックに回転をかけて使用しているためドットの四角形が斜めになっているなど,ドットグラフィックの配置がドットを並べる為のグリッドを意識していないゲームもリリースされている．このように，ピクセルパーフェクトの手法を用いずにドットグラフィックを表現してしまうと，画面全体のドットグラフィックの統一感が乱れているように感じてしまう．高解像度ディスプレイ上でドットグラフィックのオブジェクトに拡大縮小を行なった例を図 1 に，位置の移動を行なった例を図 2 に示す．また，比較としてピクセルパーフェクトが正しく行われている状態の例を図 3 に示す．

図 1　ピクセルパーフェクトをせず
拡大縮小を行なったドットグラフィック

図 2　ピクセルパーフェクトをせず
位置の移動を行なったドットグラフィック

図 3　ピクセルパーフェクトを行なった
ドットグラフィック

Vol.2018-CG-172 No.8
Vol.2018-DCC-20 No.8
Vol.2018-CVIM-214 No.8
2018/11/7

SQUARE 社(現 SQUARE ENIX 社)からスーパーファミコン向けに発売されたゲームソフト「クロノ・トリガー(1995 年)」の, PC 向けにリリースされた移植版[4](2018 年)では, 移植にあたって高解像度ディスプレイに対応させる HD 化が行われた. しかし, リリース初期のこの移植版ではドットグラフィックを平滑化したオブジェクトと平滑化をかけずそのまま拡大されたドットグラフィックのオブジェクトが混在しているためグラフィックの統一感を損ねていた. また, ユーザーインターフェース用のスプライトデータやフォントには新規に高解像度のグラフィックが使用されており, これもまた, グラフィックの統一感を損ねる要因となっている. このようなことが原因となり,リリース初期は質の悪いグラフィックの移植であると低い評価を受けていたが, 後のアップデートで「オリジナルグラフィック」という原作に近いドットグラフィックに変更できるオプションが追加された. このような前例から, 高解像度の 2D グラフィックを用いるなら高解像度のグラフィックで統一するべきであり, ドットグラフィックを用いるならくっきりとしたドットグラフィックで統一することで, 良質なグラフィックであると評価されることにつながると言える.

4. ゲーム開発においてドットをピクセルパーフェクトで表示する手法

高解像度ディスプレイにおいてドットグラフィックを表現するためにピクセルパーフェクトで表示することは重要である. そのため, ゲーム開発ではドットグラフィックをピクセルパーフェクトで表示する手段が必要であり. ここでは Unity 上でピクセルパーフェクトを行う二つの手法について以下に述べる.

4.1 位置を変更する手法

ドットグラフィックを使ったゲームオブジェクトの位置をドットが整列するような位置に移動させることで擬似的にピクセルパーフェクトな表現をすることが可能である. まず下準備として, ゲームオブジェクトに用いるスプライトデータの設定を行う必要がある. まずスプライトデータのサイズ設定をドットの大きさと一致させる.1 ドットを 1×1Unit(Unit は Unity で用いられる距離や大きさの単位)の大きさとする事で, 例えば 32×32 で作られたドットグラフィックなら32×32Unitの大きさとなる. これで, ゲームオブジェクトに使用するスプライトデータ側の設定は完了となる.

これにより, 1 ドットの大きさが 1Unit となったため, ゲームオブジェクトの位置を整数のみになるよう制限することで, ドットの位置を整列させ擬似的にピクセルパーフェクトを行うことができる. しかし本来, Unity におけるゲームオブジェクトの移動位置は整数だけではなく小数点以下の値も使用するため, 位置の値を整数に変換するスクリプトを作成し, ゲームオブジェクトに追加する必要がある. スクリプトの内容は,フレームが呼び出される際に処理が行われる関数(LateUpdate)内で現在のオブジェクト位置座標を取得し, 取得した値を整数に変換する. 整数に変換した値をオブジェクトの位置座標に再度代入することで, 位置の整数化を行うことができる. しかし, これだけだと物理演算などを用いて位置を動かす場合に, 少しの移動だけだと同じ位置に戻され続け, 移動することができなくなってしまう. そのため, 整数に変換する前の座標を一旦格納し, フレームが表示し終わった際に処理される関数(OnRenderObject)内で元の座標をオブジェクトの位置座標に再度入れ直すことで, 本来の座標を維持しながら画像を表示する瞬間だけ整数に変換することができる. これにより位置座標を変換することで, 擬似的にピクセルパーフェクトを行うことが可能となった. しかし, 一瞬だけでもオブジェクトの座標を動かしてしまうため, オブジェクト同士の接触判定が一瞬離れ, うまくいかなかったり, 全てのオブジェクトに位置を整数に変換するオブジェクトを当てないといけないため, 手間が発生してしまったりするという問題点を抱えていた.

また, この手法では位置を変更しているだけなので, ゲームオブジェクトに半端な拡大縮小や回転を行うとピクセルパーフェクトを崩してしまうという問題もある.

4.2 Pixel Perfect Camera を使う手法

Unity のバージョン 2018.2.3 アップデートから, Pixel Perfect Camera という機能が公式に追加された. Pixel Perfect Camera は文字通りグラフィックをピクセルパーフェクトに変換して表示するカメラフィルタ機能であり, カメラオブジェクトにコンポーネントとして追加することで使用することができる. Pixel Perfect Camera を追加したカメラの Asset Pixels Per Unit の項目に, スプライトデータの設定でつかった Pixel Per Unit の値を入力し, Reference Resolution の項目に解像度の値を入力することで, 指定した解像度に合わせカメラで写したものが自動的にピクセルパーフェクトに変換される. カメラで写したものしかピクセルパーフェクトに変換されないため, スコアやステータスといった UI などが変換されないので注意が必要だが, UI のレンダーモードをワールド空間(World Space)にして, ゲームオブジェクトと同じ空間で扱うことで, 同じようにピクセルパーフェクトに変換することができる.

Pixel Perfect Camera はカメラフィルタ機能なので, 前述した位置を変更して擬似的にピクセルパーフェクトを行う手法とは違い, オブジェクトの位置を変更する必要がないので接触判定に問題が起こることがない, またゲームオブジェクトの拡大縮小や回転を行なっても, 自動的にグリッドに沿ったドットグラフィックに変換してくれるので, オブジェクトの拡大縮小や回転も表現方法の一つとして用

Vol.2018-CG-172 No.8
Vol.2018-DCC-20 No.8
Vol.2018-CVIM-214 No.8
2018/11/7

いることができるようになっている．例として，オブジェクトに回転を行いピクセルパーフェクトになっていないドットグラフィックを図 4，同じグラフィックに Pixel Perfect Camera を用いる事で，ピクセルパーフェクトに変換したドットグラフィックを図 5 に示す．このように，Pixel Perfect Camera の登場よって Unity におけるドットグラフィック表現を用いたゲーム開発の問題のほとんどが解消された．

図 4　ピクセルパーフェクトをせず
回転を行なったドットグラフィック

図 5　Pixel Perfect Camera を使うことで
ピクセルパーフェクトに変換されたドットグラフィック

5.　ピクセルパーフェクトを維持したズーム

5.1　Pixel Perfect Camera が抱える問題点とその解決手法

Pixel Perfect Camera は便利であるが，使うにあたっていくつか問題点がある．その問題点の一つとして，Pixel Perfect Camera をコンポーネントに追加したカメラの Size の値を変えられない事が挙げられる．Unity においてカメラの Size の値とはカメラで写す範囲の値であり，この値を変更することでズームインやズームアウトを表現できる．3D ゲームの場合であればほとんどの場合用いられる投影方法が近いものが大きく，遠いものが小さく見える透視投影であるため，Size の値が変えられなくとも対象物にカメラが近づくことでズームイン，離れることでズームアウトを行うことができる．しかし，2D ゲームは基本的に距離が変わっても物の大きさが変わらない平行投影が用いられるため，その手法は使うことができない．そのため，ズーム機能と Pixel Perfect Camera を同時に使うための工夫をする必要があった．そこで，シーン内に平面のオブジェクトを生成し，そのオブジェクトに Pixel Perfect Camera で写している画面を投影する，そしてそのオブジェクトをもう一つ用意した別のカメラで写すことにより，平面オブジェクトの大きさを変更すればピクセルパーフェクトで表現されたドットグラフィックを崩さずに擬似的なズームを行うことができると考えた．

5.2　ズーム機能の実装手法

(1) 平面オブジェクトを生成する．
(2) Pixel Perfect Camera を使用したカメラ映像をムービーテクスチャとして平面オブジェクトへ割り当てる．
(3) 平面オブジェクトを撮影するカメラを生成する．
(4) 平面オブジェクトの大きさを，生成したカメラが撮影する範囲と同じ大きさにする．

これにより，平面オブジェクトを拡大すればズームイン，縮小すればズームアウトを擬似的に行う事ができる．この手法を図にしたものを図 6，図 7 に示す．

しかし，生成したカメラが写す範囲と平面オブジェクトのサイズが完全に同じだと，ズームアウトした時に Pixel Perfect Camera を追加したカメラに写っていない範囲が新しく生成したカメラの方に写ってしまう．Pixel Perfect Camera を追加したカメラで写す範囲を広くし，その映像を投影する平面オブジェクトの大きさもそれに応じて大きくすれば，ズームアウトした際に写すことのできる範囲に余裕ができるため問題を解決することができる．

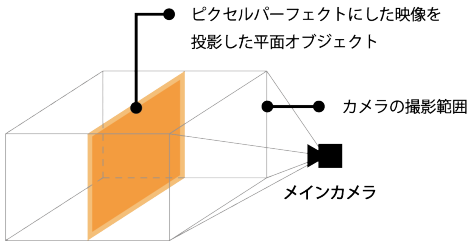

図 6　ズーム前の状態

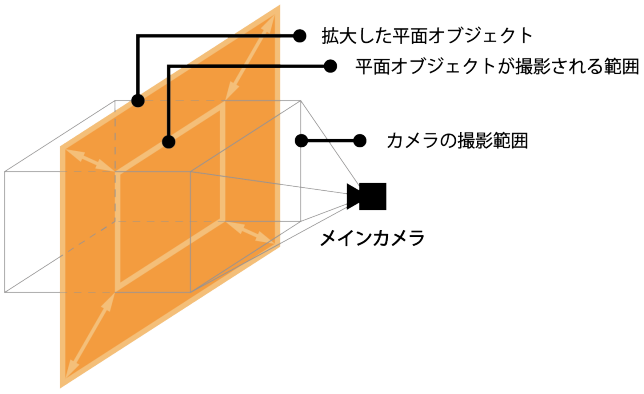

図 7　ズームイン時の状態

Vol.2018-CG-172 No.8
Vol.2018-DCC-20 No.8
Vol.2018-CVIM-214 No.8
2018/11/7

6. ピクセルパーフェクトを用いた際のフォントの表現手法

6.1 文字を扱う際の問題点と解決手法

画面全体をピクセルパーフェクトで表現するために，フォントのドットサイズを解像度が低めのドットグラフィックにあわせると，文字が荒くなりすぎて読みづらくなってしまうといった問題が起こる．また文字表示にドットグラフィックとは関係のない高解像度のフォントを使う手法もあるが，前述したクロノ・トリガーの移植版でも取り上げたように，この手法ではドットグラフィックと高解像度グラフィックが混在するため，グラフィックの統一感が失われてしまうという問題がある．ドットサイズをドットグラフィックに合わせ，ピクセルパーフェクトを行なう手法の例を図 8，高解像度フォントを用いる手法の例を図 9 に示す．

グラフィックの統一感を維持しつつ読みやすい文字を表示するために，文字表示に半分サイズのドットのフォント(以下，半ドットフォント)を使う手法で解決できると考えた．この手法は「ケロブラスター[5]」などの統一感のあるドットグラフィック表現が行われているゲームで見られる手法である．この手法を用いた場合でも，高解像度のフォントを使う場合と同じくピクセルパーフェクトではなくなってしまうが，ドットサイズが半ドットなため，グリッドに沿った並びを大きく乱すことはない．半ドットフォントの手法を用いた例を図 10 に示す．

6.2 半ドットフォントの実装手法

(1) フォントオブジェクトのみを写すカメラを生成する．
(2) 生成したカメラのコンポーネントに Pixel Perfect Camera を追加し，ゲーム全体を写すカメラの二倍の解像度を設定する．
(3) 上記カメラ映像をムービーテクスチャとして平面オブジェクトへ割り当てる．

これにより作成したフォントのみを映している平面オブジェクトと，ドットグラフィック全体を映している平面オブジェクトを重ね，平面オブジェクトを撮影するためのカメラで撮りなおす事で，ピクセルパーフェクト表現が行われているグラフィックと半分のサイズでピクセルパーフェクト表現が行われたフォントを合成する事ができる．この手法を図にしたものを図 11 に示す．この手法により，半分のドットサイズでピクセルパーフェクト表現を行なったフォントを実装できるようになった．

図 8　ドットサイズをグラフィックに合わせたフォント

図 9　ドットを意識しない高解像度フォント

図 10　半ドットで表現されたフォント

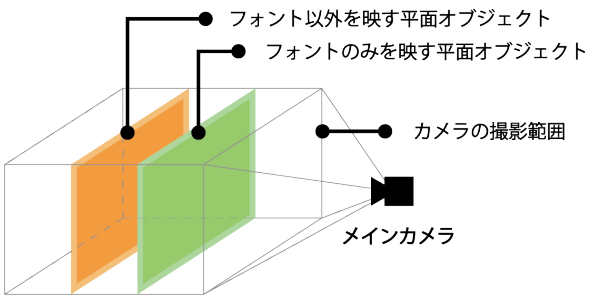

図 11　ピクセルパーフェクトなグラフィックと半ドットフォントの合成手法

Vol.2018-CG-172 No.8
Vol.2018-DCC-20 No.8
Vol.2018-CVIM-214 No.8
2018/11/7

7. まとめ

本研究では，高解像度ディスプレイにおけるドットグラフィックを用いたゲーム開発を行う際に起こる問題点と，それを解決する重要性，並びにゲームエンジンの Unity 上で問題点を解決し，ドットグラフィックを綺麗に見せる手法について述べた. Pixel Perfect Camera というグラフィックをピクセルパーフェクトで表現する事ができるフィルタ機能の実装により，Unity でドットグラフィックのゲームを開発する環境が一変し ,グラフィックに乱れのない綺麗なドットグラフィックのゲームが開発しやすい環境になった. また, Pixel Perfect Camera の問題点を独自の手法で解決することにより，ズームインズームアウト機能や読みやすいフォント表現を実現することを可能にした.

参考文献

[1] "UNDERTALE ". Toby Fox. https://undertale.jp/, (参照 2018-10-15).
[2] "SHOVEL KNIGHT ショベルナイト "Yacht Club Games. https://www.nintendo.co.jp/3ds/aksj/, (参照 2018-10-15).
[3] "ニンテンドークラシックミニ ファミリーコンピュータ". 任天堂. https://www.nintendo.co.jp/clv/index.html, (参照 2018-10-15).
[4] "CHRONO TRIGGER クロノ・トリガー". SQUARE ENIX. https://www.jp.square-enix.com/chronotrigger/, (参照 2018-10-15).
[5] "ケロブラスター". 開発室 Pixel. http://publishing.playism.jp/keroblaster, (参照 2018-10-15).