

Gen

戀津魁 安藤健翔 松本竹生 日置優介 細川慎一 神山大輝 竹内亮太

『Gen』は、飛び回る大量のパーティクルを操り、敵を倒していくゲームである。「大群を操ることの楽しさ」に着目し、指先で直に触れるような感覚でパーティクルを自在に操作する。また、プレイの結果を利用しジェネラティブ・アートを生成する事により「絵を創造する楽しさ」を織り込んだ。大量のパーティクルを操る気持ちよさと、ジェネラティブ・アートの生成を組み合わせ、今までにないゲームプレイを生み出した。

Gen

KAI LENZ KENSHO ANDO TAKEO MATSUMOTO YUSUKE HIOKI
SHINICHI HOSOKAWA DAIKI KAMIYAMA RYOTA TAKEUCHI

"Gen" is games that will defeat the enemy by manipulate a large number of particles that fly around. Focusing on the "joy of manipulating the horde", we created to be able to manipulate the particles as in direct touch with own fingertips. In addition, generate a Generative art by utilizing the results of the play in order to add "joy of create a picture". By combining Generative art and the fun of manipulate particles, we created a new user experience.

1. はじめに

1.1 概要

インディーゲーム開発集団 Yack Lab.のゲームアプリ作品第2弾として、『Gen』を開発した。企画初期段階のブレーストーミングにより、コンセプトとして「大群を操ることの楽しさ」を設定し、更に、ゲームの目的設定を行う際にジェネラティブ・アートの要素を追加し、「絵を創造する楽しさ」をコンセプトに織り込んだ。

コンセプトに設定した2つの楽しさの提供を実現するために、「大量のパーティクルを操作し敵を倒していく」というゲームを開発した。ユーザー体験としてコンセプトを検討した結果、指先で直接触れて操るような形が望ましいという結論を得た。また、ユーザーの触れられる面積がより広い方が好ましいため、Apple社のiPadをプレイ環境として設定し、Objective-Cによる開発を行った。

1.2 ジェネラティブ・アート

ジェネラティブ・アートとは、マット・ピアソンら[1]によると「ジェネラティブ・アートは、プランや材料や道具を使って組み上げるものではありません。あたかも花や樹木のように、育てられるものです。しかし、その種子は土壌や水分ではなく、ロジックとエレクトロニクスです。」と説明されている。プログラムによる生成でありながら、ランダム性のある要素を利用した予測不可能な面を持つアートである。

『Gen』においては、ユーザーのプレイ結果というランダム性を利用してジェネラティブ・アートを創出することで、コンセプトである「絵を創造する楽しさ」を実現した。

1.3 類似作品と差分

同様のゲームとして、QApps LLCによる『Gravitarium Plus』[2]が存在する。Gravitarium PlusはiPhone及びiPad向けのアプリで、タッチパネルを介してパーティクルを操作し、またDrawモードではパーティクルの軌跡を利用して描画を行う事ができる。しかし、ゲームとしての目的がなく、描画についてもユーザーの介入する余地が少ない。

『Gen』においては、ゲーム中に敵を配置しそれをパーティクルによって倒すという目的を持たせ、また敵の倒し方に応じてジェネラティブ・アートを生成するという結果を設定した。これによって、単純に「大きな画面で直接触れるように大群を操る」という楽しさ、「次々と出現する敵を倒す」というアクション性、「プレイ結果に応じたジェネラティブ・アートが創出される」という創造性といった、多彩な楽しさを提供している。

2. ゲームシステム

2.1 パーティクル操作

ゲーム画面内では大量のパーティクルが縦横無尽に飛び回り、画面をタッチすることでタッチ位置に集まってくるような動作をする。図1に操作例を示す。

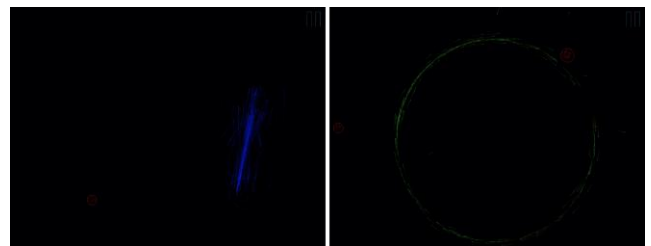


図1 タッチによるパーティクルの操作

†1 Yack Lab.

マルチタッチを有効化し、触れている指の本数に応じた動きをするよう実装した。例として1本指で触れた場合にはその点を中心に集まり、2本指では2点間を往復する直線の動きになる(図1左)。3本の場合は3点を結んだ三角形の外接円の形となるように動く(図1右)。また、本数に応じてパーティクルの色も変更している。

各パーティクルは現在座標と目的地座標、これまでの座標の履歴を保持しており、それらを使い移動及び描画を行う。描画はOpenGLを利用し行っているため高速であり、数百個のパーティクルを60FPSで描画可能である。

この高速な描画と指先に追従するような動きにより、「大群を操ることの楽しさ」を実現している。

2.2 敵とのインタラクション

1章で述べた通り、『Gen』では敵が出現しそれを倒すことを目的のひとつとしている。敵はパーティクルを数個内包した状態で画面内に出現し、一定時間をかけ移動し画面外に消えていく。その間に指先でパーティクルを操り敵に当てることで倒していく。

敵にパーティクルが当たっている間、1フレームにつき1つつつパーティクルが敵内部に取り込まれる。敵が内包できる数には上限が設定してあり、パーティクルを取り込んだ敵は少しずつ膨らみ、上限に達した瞬間破裂するように消え内包しているパーティクルを放出する。敵を倒すことによってあらかじめ内包していた数だけパーティクルが増え、逆にパーティクルを取り込ませておきながら倒しきれず画面外に出られた場合にその分失う事となる。

この方式により、敵を倒すごとにパーティクルが増加する。これによってゲームの進行に従い「大群を操ることの楽しさ」がより強化される。

2.3 結果画像の生成

これまで述べてきた通り、結果画像の生成にはユーザーのプレイ結果を利用する。具体的には、敵を倒した際に得られる次の各種情報などを利用する。

- 敵を倒した座標
- 出現から倒すまでにかかった時間
- 何本指でパーティクルを操作し倒したか

これらの情報を利用し描画を行うため、ユーザーによる結果への干渉を行う事ができる。例として倒すまでにかかった時間によって結果が変わる場合、敵を即座に倒す、あるいは逆に画面外へ出ない範囲で可能な限り遅くに倒すという工夫が可能になる。あるいはそもそも全ての敵を倒さず、敢えて見逃すことによる干渉も考えられる。

このように、ジェネラティブ・アートのランダム性にユーザーの行動を利用することで弱いランダム性として活用でき、ユーザーの意図を反映させた結果画像の生成が可能となる。アルゴリズムを変更しプレイ結果の利用方法を変えることで、更に多彩なジェネラティブ・アートの生成が可能となる。また、ユーザーが能動的に結果に干渉する意

思を持たずとも、プレイ毎に異なる結果が得られる。この方式によって、「絵を創造する楽しさ」を実現した。

図2及び図3に、同じアルゴリズムで生成した異なるプレイ結果のジェネラティブ・アートを示す。

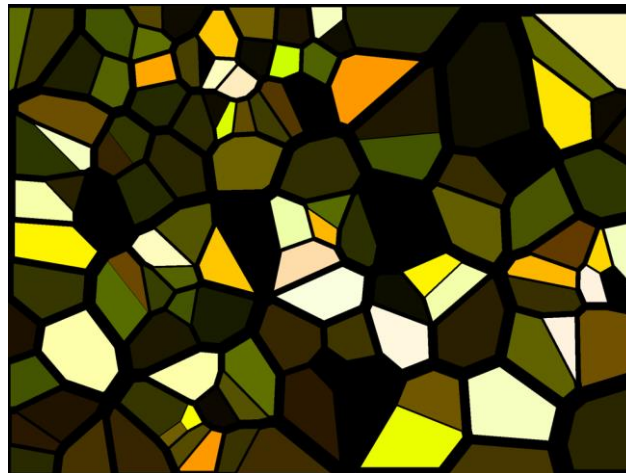


図2 プレイ結果によるジェネラティブ・アート(A)

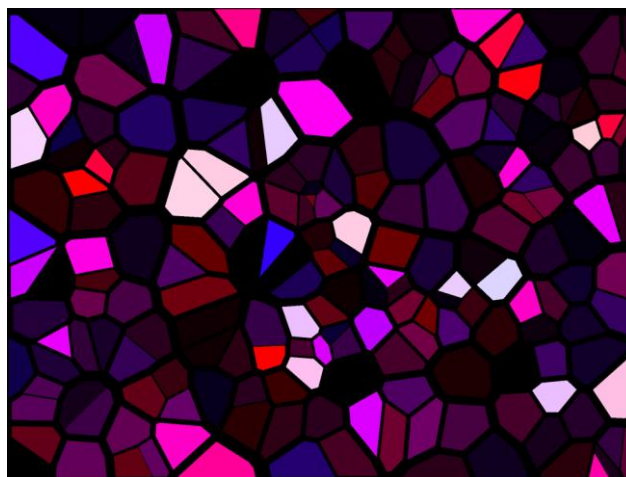


図3 プレイ結果によるジェネラティブ・アート(B)

3. おわりに

『Gen』は2016年8月現在まだ開発中ながら、東京ゲームショー2015のセンス・オブ・ワンダー ナイト、WIRED CREATIVE HACK AWARD 2015、Gotland Game Conference、芸術科学会東北支部によるアート&テクノロジー東北2016と多くの場で評価されており、新しいユーザー体験の創出に寄与できたと考えられる。

今後は結果画像の共有によるCGMやSNSとしての楽しみ方についても検討し、アプリのリリースに向け実装を行っていく。

参考文献

- [1] マット・ピアソン, 久保田晃弘, 沖啓介: ジェネラティブ・アート—Processingによる実践ガイド, ビー・エヌ・エヌ新社 (2012).
- [2] QApps LLC, Gravitarium Plus - The Best Dancing Particles Show & More, <https://itunes.apple.com/us/app/id438667778>