IPA

2024 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

竹迫 良範
（株式会社リクルート データ推進室 アドバンスドテクノロジーラボ 所長）

2. クリエータ氏名

椎名 貫太（大阪電気通信大学 総合情報学部 デジタルゲーム学科 学部 4 年）

3. 委託金支払額

2,880,000 円

4. テーマ名

ジェネラティブ VJ ソフトウェアの開発を容易にするシステム

5. 関連 Web サイト

- 開発成果を公開している Web サイト：https://mitou.sainakey.com/
- ジェネラティブ VJ ソフトウェアの開発を容易にするシステム「Fluss」のリポジトリ：https://github.com/SainaKey/Fluss.BRP-public
- プリレンダ VJ ソフトウェア「SynapseRack」の公式 Web サイト：https://synapserack.com/
- 開発者個人 Web サイト：https://www.sainakey.com/

6. テーマ概要

本プロジェクトではジェネラティブ VJ と呼ばれるリアルタイム CG（コンピュータグラフィックス）技術を用いた映像パフォーマンスを行うためのソフトウェアの開発を容易にするシステム「Fluss」を開発した。ノードベースプログラミング環境によりノンプログラマやアーティストでもジェネラティブ VJ 表現のためのプログラムを実装することができるのが本プロジェクトの特徴である。

7. 採択理由

本提案は、WebGPU を用いてジェネラティブ VJ を実現する JavaScript プログラムを、探索しながら作成できるソフトウェアを開発することである。現状のジェネラティブ VJ にはノンプログラマとプログラマの間には大きな参入障

壁があり、本ソフトウェアがこれらの壁を乗り越えることによって、クリエイティブとエンジニアリングの双方を繋ぐ存在になると良い。

提案者は Unity での開発実績や BOOTH での自作 VJ ソフトの販売実績もあり、ノードベースとコードベースを行き来しながら新しい VJ 表現を探索できる価値をよく理解している。プロトタイプの開発では、いくつかのプリミティブな表現を地道に実装していくことになると思うが、このソフトウェアでインパクトのある表現を実現できないと最先端の VJ への導入は難しい。このプロジェクトにおいては、単なる機能の実装だけではなく、新しい表現の引き出しが増えることも必要である。

このソフトウェアで作られたジェネラティブ VJ 作品をきっかけに、ノンプログラマのクリエイターがプログラミングに興味を持ち、クリエイターの感性を持つ凄腕プログラマ人口が増えることを期待したい。

8. 開発目標

本プロジェクトの開発目標は以下の 3 つである。

(1) ジェネラティブ VJ ソフトウェアの開発を容易にすること

ジェネラティブ VJ ソフトウェアを開発するためにはプログラミングやソフトウェア開発、リアルタイム CG に関する知識が必要であるという部分を解決することである。本プロジェクトでは特にプログラミングを使った映像表現をノードベースというノーコードプログラミングにて実装できる環境「Fluss」を実装・提供することで、アーティストやノンプログラマがノーコードでジェネラティブ VJ ソフトウェアの開発を行えることを目指す。

(2) ジェネラティブ VJ の楽しさを多くの人が体験できること

多くの人が VJ パフォーマーとして、あるいは VJ パフォーマンスを見るお客さんとしてもジェネラティブ VJ の楽しさを体験できる。現状、ジェネラティブ VJ の認知度は低く、プログラマでかつ、アーティストでかつジェネラティブアートに興味がある人の間でしか話題になっていない。Fluss の開発を通して、こういった問題を多くの人を開発者かつ VJ パフォーマーとして促すこと、さらに VJ パフォーマーになるつもりが無い人にもジェネラティブ VJ のデモと発表内容を通してコンピュータ技術を用いた VJ の存在と楽しみ方を知ってもらうことを目指す。

(3) ノンプログラマが本プロジェクトをきっかけにプログラマとして成長すること

提案者が VJ の感動をきっかけにプログラマとしてのキャリアを歩んだという経験から、本プロジェクトを通して VJ というエンターテイメントから感動を受

け、ジェネラティブ VJ として活躍し、プログラマとしてキャリアを歩むことを目指せるようになることを目標とした。

9. 進捗概要

(1) ノードベース UI の実装

Fluss は Unity から提供されている Graph View というノードベース UI を実装するための API を使用し、Unity エディタ拡張としてノードベースプログラミング環境を実現した。開発したプロトタイプのスクリーンショットを図 1 に示す。Graph View API について調査し、簡単なノードベースプログラミングを実装したが、より効率的にノードベースプログラミング環境を実装するため Graph View API をラップしたライブラリ NodeGraphProcessor を使用し開発を進めた。

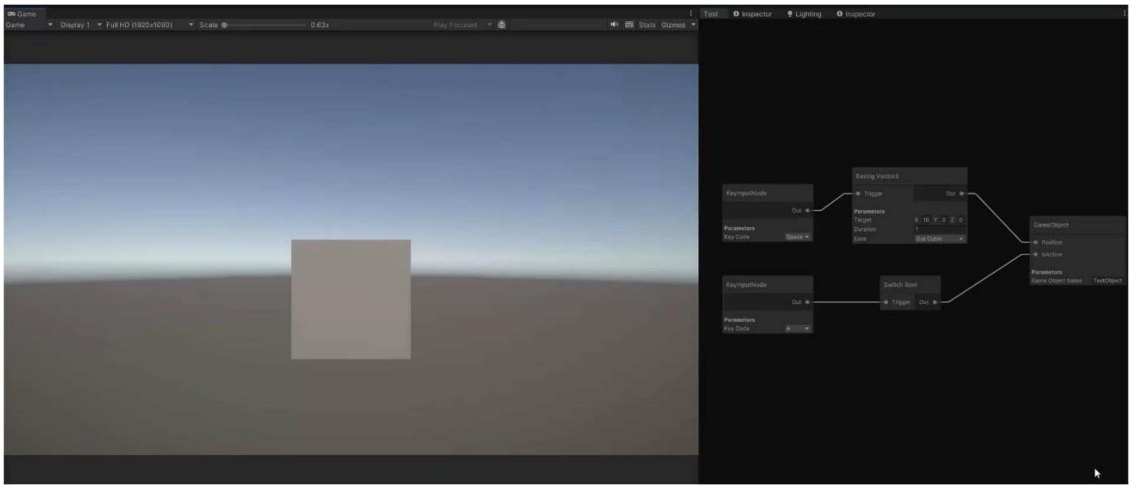

図 1：Unity エディタ拡張として実装したノードベースプログラミング環境

(2) 変数を参照する機能の実装

次に Unity でジェネラティブ VJ ソフトウェアのプログラミングを行うためには、変数と参照が必須である。Unity で扱われているデータはエディタ上で扱うのか、ランタイム（シーン）で扱うのかで大きく 2 つに分かれている。Unity の仕様として、これらはライフサイクルが違うため、エディタ上のものをランタイムで参照したり、逆にランタイムでエディタ上のものを参照したりすることは困難である。そのため、ランタイム上にある 3D オブジェクトをノードベースプログラミング環境のエディタ上で参照する機能を実装するためには工夫が必要だった。既に Unity Visual Scripting や、Unity Timeline といったエディタ上の機能がランタイム上のものを参照している機能があったため、それを調査し、変数を参照する機能の実装を行った。

(3) ホットリロードの実装

Unity C#にはコンパイル時間が長いという欠点がある。他のノードベースツ

ールではノードをつないだら即座に結果が反映され、プログラミングのイテレーションが早く回せるという利点がある。そこで、本プロジェクトも Unity 上でジェネラティブ VJ ソフトウェアの開発のイテレーションを素早く回せるようにし、よりよい開発体験を提供するためにホットリロード機能を実装した。

(4) 様々なノードの実装

ジェネラティブ VJ ソフトウェアの開発に必要な機能を持ったノードを開発していった。実装したノードの一部を挙げると「ランダムな数字を出力するノード」「キーボードの入力を受け付けるノード」「GUI の入力を受け付けるノード」「数値をアニメーションさせるノード」「複数のカメラを切り替えるノード」「3D オブジェクトの座標を書き換えるノード」などである。

(5) ノードベースプログラミング環境「Fluss」の完成

Unity に標準で搭載されているノードベースプログラミング環境 Unity Visual Scripting で「オブジェクトを回転させる」という実装をした時のスクリーンショットを図 2 に示す。Unity Visual Scripting でもジェネラティブ VJ ソフトウェアの開発は行えるが、ノードの数が多く、複雑になっている。

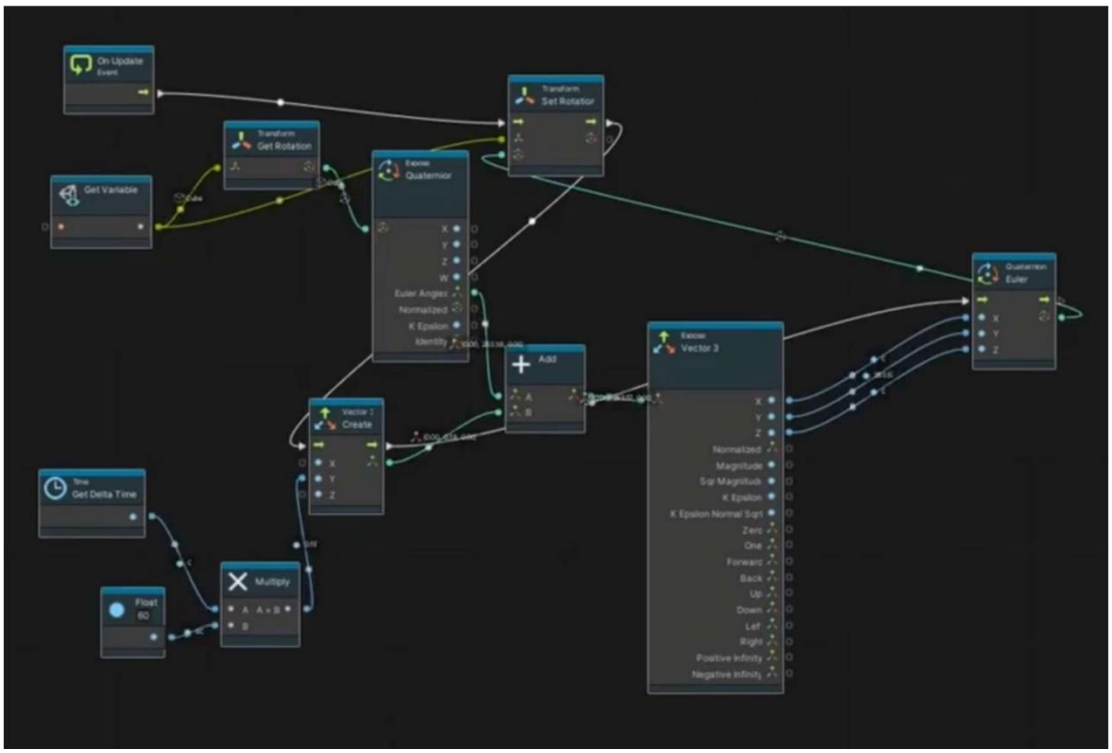

図 2：Unity Visual Scripting で実装した様子

Fluss で同じ動作を実装した時のスクリーンショットを図 3 に示す。図 2 に対して本プロジェクト Fluss は、たった 3 つのノードで「オブジェクトを回転させる」という実装を行えている。

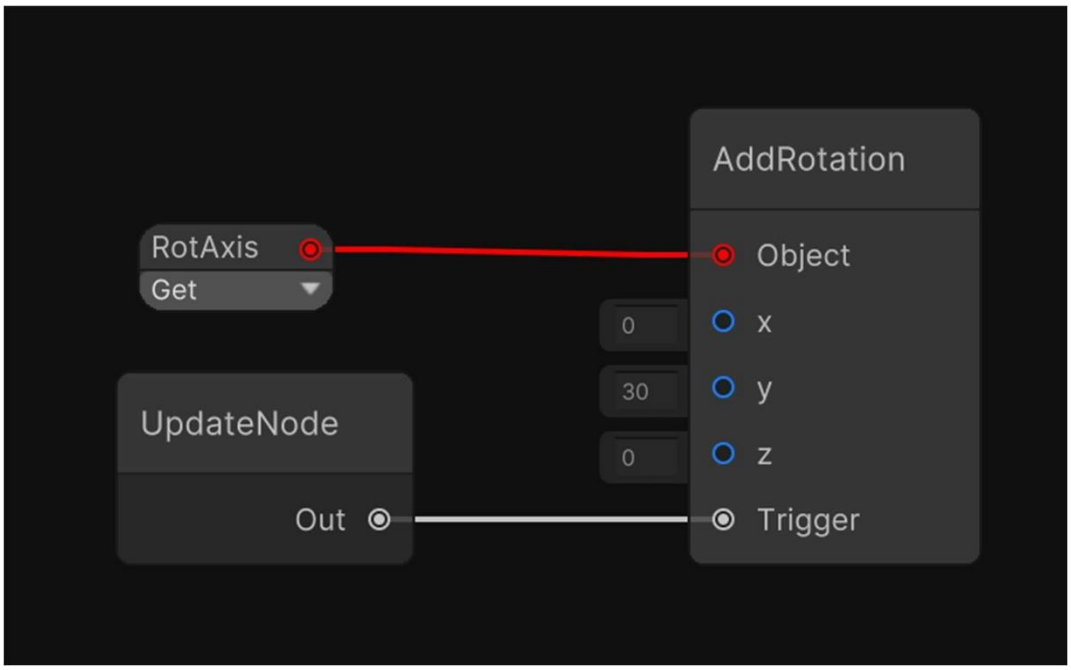

図 3：Fluss で実装した様子

このように Fluss はジェネラティブ VJ ソフトウェアの開発に特化したノンプログラマやアーティストが扱える容易なノードベースプログラミング環境となった。ジェネラティブ VJ ソフトウェア開発においてはソフトウェア開発やゲームプログラミングのように「なんでもできる」必要はないが、容易さを優先した結果「誰が作っても同じような表現になる」ことは問題である。Fluss はそのようなバランスを調整し、容易でかつ、プログラマブルであることを実現した。

10. プロジェクト評価

成果報告会でデモの実演を行うために、このプロジェクトの成果物である Fluss を使用してジェネラティブ VJ ソフトウェア開発した。コンセプトとして、ジェネラティブな表現があること、クラブのような激しい映像の変化があること、また本プロジェクトと直接は関係しないがリアルタイム CG のリッチさを表現することを目標として映像制作のプログラミングを実施した。ジェネラティブな表現としては、図 4 のように白い光る柱がランダムに生成される演出を実装した。リアルタイム CG のリッチさは、LTC（Linearly transformed cosines）という面光源のグローバルイルミネーションを表現する技術を使用することで実現した。

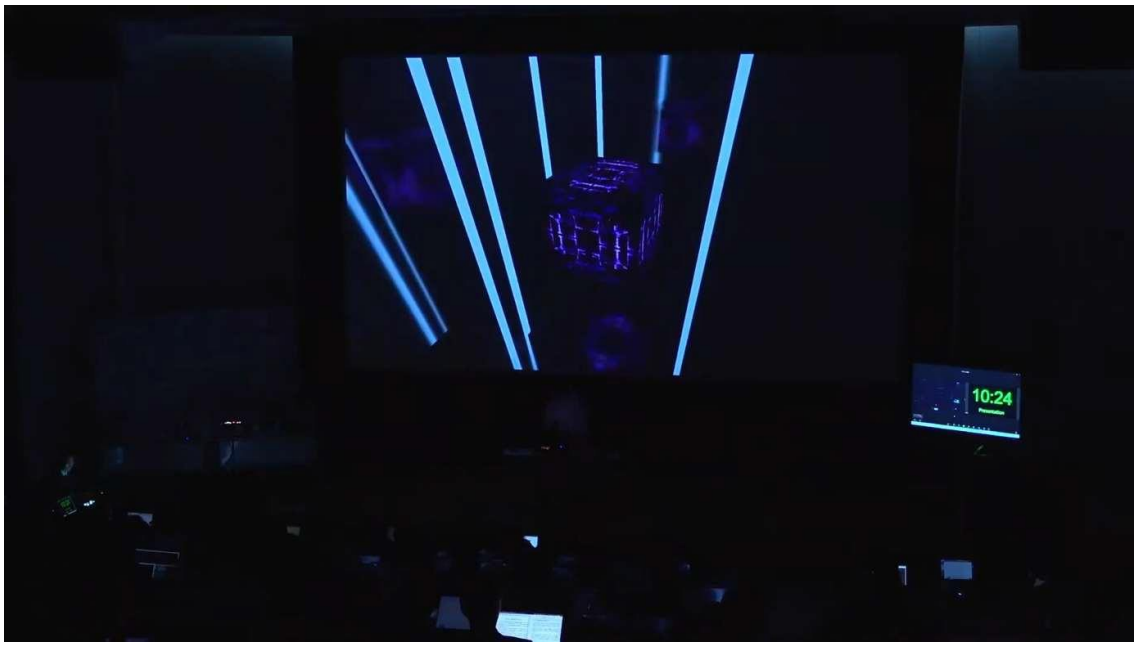

図 4：成果報告会でのジェネラティブ VJ デモの様子

最終的にプロジェクト期間中に Unity 上で動作するノードベースのジェネラティブ VJ ソフトウェア開発環境 Fluss を実装することができ、デモも実施することができたので、開発に成功したと言える。Fluss はオープンソースソフトウェアとして MIT License のもとで GitHub 上に公開し（https://github.com/SainaKey/Fluss.BRP-public）、誰でも使える状態になっている。

提案当初の「WebGPU ベースの Web3D 表現ライブラリ開発」や「Web アプリケーション開発」は、技術的な限界や目標達成の難しさから断念し、代わりに、より目的に適した Unity エディタ拡張としての実装に方向転換した。技術選定や機能仕様の検討において、複数のプロトタイプを開発し、進捗報告会議でのフィードバックを通じて検討を重ねた。最終的に、初期計画からプロジェクトをピボットする形となったが、目的達成のために最適と判断したアプローチにたどり着いた。プロジェクト進行する上で、しっかりと技術検証を重ね、目的達成のために当初考えていた手段を変えるといった柔軟な対応ができたことは今後の人生においても重要な経験になったと思う。

11. 今後の課題

Unity エディタ拡張として実装したため、Fluss を使うためには Unity の習得が必要であり、依然として参入障壁は残っている。より広い層への普及には、この障壁を下げるための工夫がさらに必要である。Fluss を OSS として公開しているが、より多くのユーザに使ってもらうためにはさらなるドキュメント整備が必要である。今回の未踏の成果物を紹介する Web サイト

（https://mitou.sainakey.com/）を公開しているが、使い方ガイドやチュートリアルを充実していくと普及につながっていくであろう。

　ノードの追加や新機能の開発など、Fluss の継続的なアップデートも必要である。特に、ジェネラティブ VJ ソフトウェアとして表現の幅を広げられる機能強化があるとアーティストとしても活用しやすくなる。こういった新機能も含めて椎名氏自身が Fluss を使って様々なジェネラティブ VJ を実施することで、ツールの有用性を実証し、魅力を伝えていくことが重要である。