



2009 年度上期未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当PM

安村 通晃 PM(慶應義塾大学 環境情報学部 教授)

2. 採択者氏名

チーフクリエイター: 矢田 裕基(明治大学 理工学部情報科学科)
コクリエイター : なし

3. プロジェクト管理組織

リトルスタジオインク株式会社

4. 委託金支払額

2,971,076 円

5. テーマ名

物理演算を用いた作曲インタフェース

6. 関連Webサイト

なし

7. テーマ概要

近年、独特なインタフェースと楽しくインタラクションしていくうちに音楽ができあがっていく作曲ツールが登場してきた。
それらは、演奏中に LED の発光によるパフォーマンスが行えたり、タッチパネルを積極的に採用されたりと、そのインタフェースデザインにおいても使いやすさよりは直感

性や楽しさが重視されているように見える。

これらのツールによる作曲・演奏プロセスは出力される音と操作の関係が視覚的に見えるため鑑賞者にとっても楽しい。

一方、計算機の性能向上にともない、最近では物理演算を使ったインタラクティブなコンテンツが増えてきた。

これらのコンテンツは、直感的に物体を自由に配置したり動かしたりすることができ、慣れてくると複雑な動きをする物体も作成することができる。

それは、見ているだけで楽しい仕掛けを構築するにとどまらず、独自のゲームや 8bit の計算機まで作ってしまうことすら可能である。

意図通りの動作を行う系を構築するのは困難だが、その困難さゆえにうまく動作させられたものは喝采を得るし、それは視覚的に関係が見えやすいからこそであると考えられる。

そこで、私は、この二つを組み合わせ、ユーザが物理演算による作品を作りつつ、音楽を作ったり奏でることのできるツールを作ることができるシステムを提案する。

8. 採択理由

物理演算に従ったオブジェクトの動きとその衝突によって音を出そうという非常に興味深い提案である。

繰り返し、早送り、逆戻しなども可能となっている。どちらかという、偶然性を利用するのに近い方式なので「作曲」とは呼ばない方が良いのではないかと。もし「作曲」ということに拘るのなら、望みの音がきちんと出せるかとか、リズムがちゃんととれるのかなど、多くの課題が残り、それを未踏期間中にクリアできるかどうか厳しいように思う。

つまり、「作曲」を全面に出すのではなく、物理法則に伴ったオブジェクトの動きや衝突から、心地の良い楽しい音楽がどれだけ出せるか、ものの動きとそれに伴う音という視点で、開発を行なうのが良いと思う。ただ、意外性だけでは不十分であり、音楽として、あるいは、音として充分楽しめるレベルにまでもって行って欲しい。

システムをある程度動く段階に早めにもっていき、自分以外の人にも「作品」を作ってもらい、成果報告会には、その中の優れた作品をいくつかお披露目してもらいたい。結果の作品を Web 上に残す仕組みも考えておくと良いであろう。

9. 開発目標

近年、音楽と視覚的要素が密接に関係している楽器やコンテンツが人気を博している。例えば、TENORI-ON による演奏は、聴こえる音以外にも、譜面や音に対する効

果を LED の発光で見せることで、演奏を耳と目で同時に楽しむことができる。

他にも、ピアノロールに絵を描く MIDI アニメや、操作しなくてもクリアできるように改造しかつ効果音で演奏を行なう全自動マリオなどの動画がある。これらの動画は、ニコニコ動画や YouTube などの動画共有サイトで人気のジャンルとなっている。これらのコンテンツが人気となる要因として、一つは作曲・演奏の過程と出力される音と操作の関係が視覚的に見えるため、鑑賞者にとっても楽しいコンテンツになっていることが挙げられる。また、動画を見れば、その作品の技術力の高さが一目でわかるということも要因の一つである。音楽に同期した作品を構成するのは労力を要するが、それゆえに意図通りに演奏されたときには人を感動させる。

一方、計算機の性能向上にともない、最近では物理演算を使ったインタラクティブなコンテンツが増えてきた。例として、ソニー・コンピュータエンタテインメントが発売した物理演算を利用したゲームであるリトルビッグプラネット、学習のための物理演算ソフトである Phun や Box2DFlashAS3 を利用した Flash コンテンツなどがある。

これらのコンテンツは、直感的に物体を自由に配置したり動かしたりすることができ、慣れてくると複雑な動きをする物体も作成することができる。これらは NHK のテレビ番組「ピタゴラスイッチ」のような、見ているだけで楽しい仕掛けを構築するにとどまらず、独自のゲームや 8bit の計算機まで作ってしまうことすら可能である。もちろん前述の作曲インタフェースの話と同様、意図通りの動作を行なう系を構築するのは困難だが、その困難さゆえにうまく動作させられたものは喝采を得るし、それは視覚的に関係が見えやすいからこそであると考えられる。

そこで、音楽映像表現と物理演算システムの共通点をヒントに、物理演算を用いた作曲インタフェースである「Gravity Piano」とこのシステムをより発展させた物理演算の要素を取り入れた作曲・演奏環境システム「Gravity Pianist」を開発する。Gravity Pianist は、Gravity Piano を基に多くの機能を追加した作曲・演奏環境である。

Gravity Pianist では、ピアノという鍵盤楽器を分解・変形させ、物理法則の支配する空間に再配置する。配置した鍵盤は物理法則によって動き出し、鍵盤同士もしくは鍵盤と空間上の他の物体に衝突すると割り当てられた音が鳴る。これを利用して、音楽を演奏するというシステムとした。

10. 進捗概要

物理エンジンと音の組み合わせで、ある種の作曲を可能にするシステムの提案であり、非常に魅力を感じた。採択時には、後藤 PM からは、きちんとした曲が作れるシステムが良いとの意見もあったが、私としては、ランダム性の方が面白そうと感じていた。ブースト会議までも開発は順調に進み、かなり手作りではあるが、スーパーマリオブラザーズの主題歌を演奏がすでに、できていた。1 回目のプロジェクトレビュー(10 月

3日)でも、開発は順調に進んでいる、との印象を受けた。ただ、その直後(10月8日)、プロジェクト管理組織の方から、今回開発のシステムに良く似たシステムが Web 上にあるとの連絡を受けた。これは、同日(10月8日)にチームラボがリリースしたスケッチピストン 4(<http://tablet.wacom.co.jp/bambooplayroom/>)である。これは、広い意味で物理演算と音の生成を結びつけているという点で共通性があるだけで、実は、今回開発のものとはかなり違ってはいるものの、2回目のプロジェクトレビューにおいては、この類似品との区別をどうするかを話し合った。その結果、矢田君の方のシステムは、ランダム性ではなく、明確な音作り(=作曲)を目指すことになった。そのため、作曲支援のための機能を重点化していった。そこで、最初に考えていた、単純に物理演算と音の生成の方を、Gravity Piano と呼び、後半の作曲支援に重点を置いた方のシステムを Gravity Pianist と呼ぶことになった。

11. 成果

本プロジェクトでは、(1) 構築: 鍵盤楽器を分解し再構築する、(2) 演奏: 物理演算によってオブジェクトを動かす、音楽を奏でる、(3) 配信: 演奏の映像を動画に変換する、(4) 楽器: 持ち運びのできる楽器、という4つの要素からなっている。主要部分は、最初の2つである。以下、この4つの要素を説明する。

(1) 構築

構築は Windows アプリケーション(Gravity Pianist と呼ぶ)として実装している。このアプリケーションでユーザは四角形や円などのオブジェクトを画面上に自由に配置できる。これらのうち、特定の音が割り当てられているオブジェクトを「鍵盤オブジェクト」と呼ぶ。鍵盤オブジェクトは、後述の「演奏」で別のオブジェクトと衝突すると、割り当てられた音が鳴る。各オブジェクトについては、オブジェクトの質量・角度・反発係数やオブジェクトを空間に固定するかどうかなどの設定が行える。これらの動きに関する値を調整したり後述の支援機能を利用したりすることで、発音のタイミングを調整する。

GravityPianist は、レイヤーリスト、音色情報、BPM 情報の3つの情報をもつオブジェクトからなる。さらに、レイヤーは、オブジェクトリスト、背景色、視点、重力加速度、カメラオブジェクト、開始フレーム、終了フレームといったデータを保持している。

オブジェクトは、GUIにより、空間的に配置したり、値を変更したり、削除したりできる。配置できるオブジェクトとは現状では、長方形オブジェクト、円形オブジェクト、回転オブジェクトの3種類である。編集可能なオブジェクトの値としては、反発係数、摩擦係数、重さ、固定するかどうか、色、音の高さ、鳴らし方(順次か同時かランダ

ムか)、音の長さ、初速度、回転速度などである。これらは、メニューから編集ができる。

作曲支援機能として、先読み機能がある。これにより、オブジェクトの軌道と衝突時のオブジェクトの輪郭が表示される。また、何フレーム目かも分かる。また、BPM(Beats Per Minute)機能により、スライダーを動かすと、BMP 値が変化し、またそれに合うフレーム数が表示されるので、演奏の長さの制御ができる。さらに、レイヤー機能により、複数の空間に分けて、それぞれの背景色とかオブジェクトの配置などを、別個に設計することができる。

(2) 演奏

演奏は、Gravity Pianist のデータ形式を、Box2DX という物理エンジンで利用できる形に変換して、物理演算を行ない、演奏を行なう。図 2-5-1 に、ドミノピアノの演奏中の画面例を示す。

(3) 動画

作った作品は、動画(AVI ファイル)に出力できる。

(4) 楽器

以上述べたものは、Windows PC 上で動作するが、この「楽器」部は、iPhone を用いた作曲・演奏システムで、iPhone のタッチパネルでオブジェクトを操作したり、加速度センサーにより、仮想重力を変化させたりできる。図 2-5-2 に、「楽器」として、iPhone を用いた演奏の例を示す。

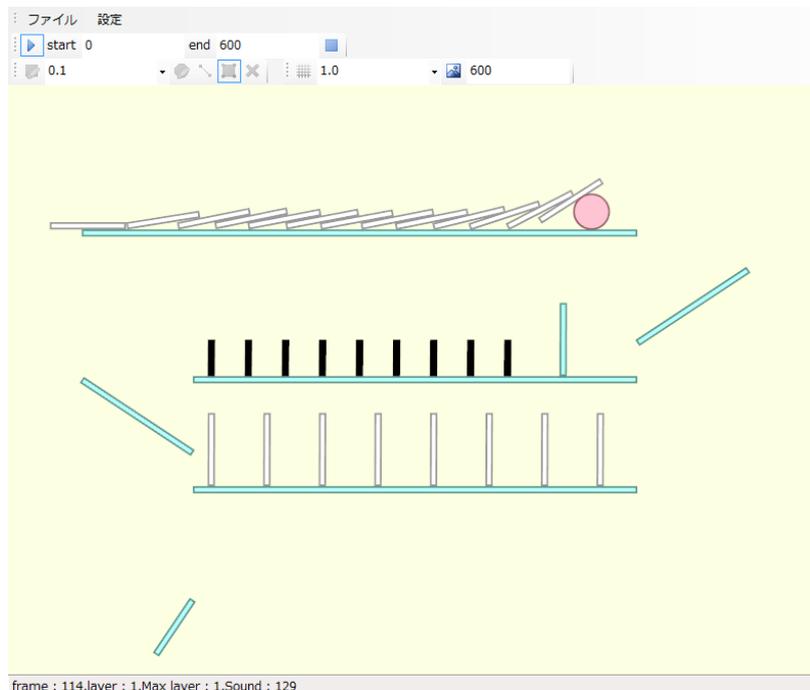


図 2-5-1 ドミノピアノの演奏の様子

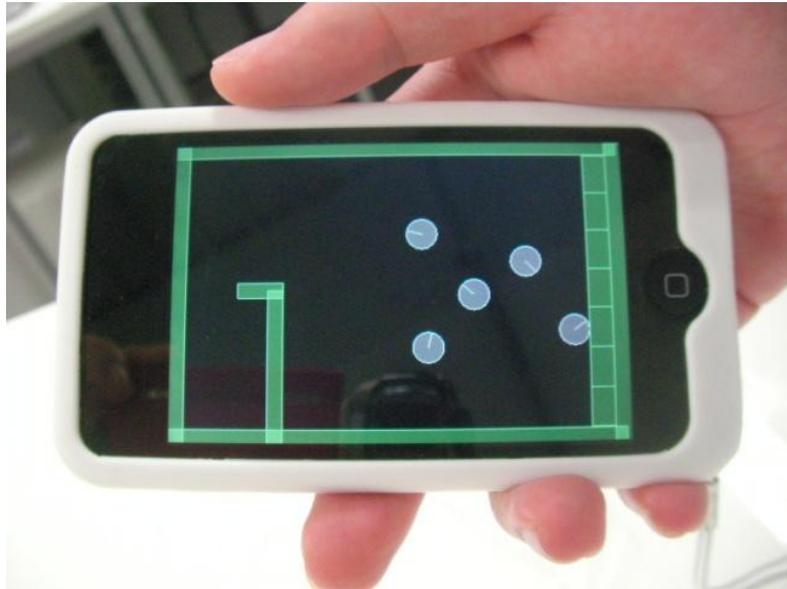


図 2-5-2 iPhone を用いた「楽器」演奏

12. プロジェクト評価

物理エンジンを用いたアニメーションに音の生成の組み合わせると、意外な音、予期せぬ音が出ることもあるし、逆に、音楽は聞き慣れたものであったとしても、その音を出すためのオブジェクトが視覚的に楽しいものもできる可能性がある。このようなことを実現するシステムとして、矢田君は、Gravity Piano と Gravity Pianist の2つのシステムを開発した。前者は、単純に物理エンジンと音の生成を組み合わせただけのものである。これを開発している最中に、ちょっと良く似たシステムが現れたこともあり、Gravity Pianist を新たに作成した。こちらは、作曲支援の要素が含まれたもので、期待する音を出すための支援機能が含まれている。このシステムを今後、Web 等で公開することにより、物理的なオブジェクトを用いた楽しい楽曲が数多く出て欲しいと思う。今後のさらなる発展を期待する。

13. 今後の課題

作曲支援機能のうち、テンポをコントロールする部分は今後の課題である。また、本システムをベータ公開して、作品を集めてくる仕組みを作ると面白いだろう。