

Android 端末上で動作する状態遷移図に基づく対話型アニメーション作成環境

周 而晶¹ 岡本 秀輔² 新堀 道信¹ 鎌田 賢³ 米倉 達広³

概要: 対話型アニメーションは、ユーザからの入力に応じて内部状態を変化させ、表示内容を変えるアニメーションであり、コンピュータゲームもその一種である。状態遷移図を用いてアニメーションの動作を表現できることに着目して、Islay と呼ばれる対話型アニメーションの作成ツールが開発され、ユーザが GUI エディタで状態遷移図を描くことにより、対話型アニメーションを簡単に作成できるようになっている。本研究では、タブレット端末の特徴であるマルチタッチ操作の利点を生かすように、Android 端末上で動作する対話型アニメーション作成環境の開発を行う。

キーワード: Android, 対話型アニメーション, 状態遷移図, Islay

Interactive Animation Authoring Platform Based on State-transition Diagrams Runs on Android Devices

ERJING ZHOU¹ SHUSUKE OKAMOTO² MICHITOSHI NIIBORI¹ MASARU KAMADA³
TATSUHIRO YONEKURA³

Abstract: Interactive animations refer to animations that make internal state transitions and change their appearances according to the input from users, which may become a kind of definition of computer games. Based on the fact that the behavior of interactive animations can be defined by state-transition diagram, an interactive animation authoring tool called Islay has been developed. Islay allows users to create interactive animation simply by drawing state-transition diagrams on a GUI-based editor. Up to now Islay mainly runs on the PC, and we cannot use it on the portable devices that become popular recently. Therefore, in this research, a new version of Islay that runs on Android devices was developed, with the further purpose of making multi-touch event available as user input to the interactive animation created by Islay.

Keywords: Android, interactive animation, state-transition diagram, Islay

1. はじめに

アニメーションは、連続に変化する画像によって情報を

動的に表現する手段であり、マルチメディアコンテンツの重要な構成要素となっている。対話型アニメーションは、ユーザの入力に応じて内部状態を変化させ、表示内容を変えるアニメーションである。コンピュータゲームも対話型アニメーションの一種と言える。

現在、様々なツールで対話型アニメーションの作成が可能になっている。代表例は Adobe 社の Flash Professional[1] である。しかし、このようなツールを利用して対話型アニメーションを作成するためには、プログラミング言語を含め専門的な知識が必要になり、一般のユーザにとっては困

¹ 茨城大学大学院 理工学研究科情報工学専攻
Graduate School of Science and Engineering, Ibaraki University
Hitachi, Ibaraki 316-8511, Japan
² 成蹊大学 理工学部情報科学科
Faculty of Science and Technology, Seikei University
Musashino, Tokyo 180-8633, Japan
³ 茨城大学 工学部情報工学科
Department of Computer and Information Sciences, Ibaraki University
Hitachi, Ibaraki 316-8511, Japan

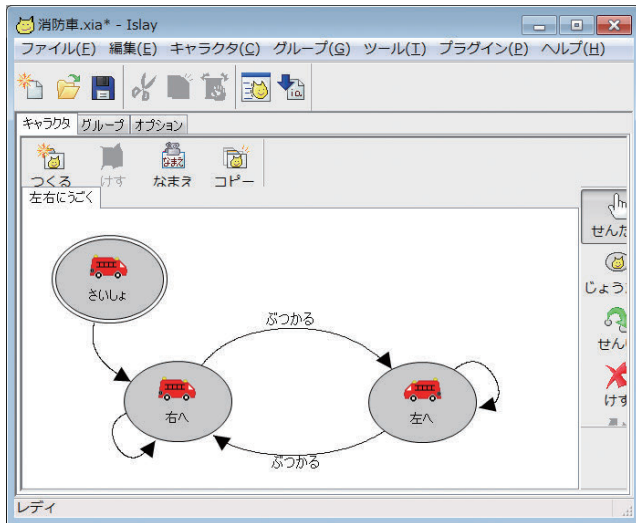


図 1 Islay エディタ

難である。

ゲームを含めた対話型アニメーションは有限状態機械として定義できる。状態遷移図を用いてアニメーションの動作を表現することに着目して、Islay（アイラ）と呼ばれる対話型アニメーションの作成ツールが開発された [2][3]。Islay では、GUI 環境で状態遷移図を作成することにより、アニメーションに登場するキャラクターの動作を記述する。状態遷移図は直観的に理解しやすいため、小学生でも Islay でゲームを作りながら、情報工学の基本である有限状態機械の概念を習得できる。

これまで複数のプラットフォーム上で動作する Islay アニメーションの作成及び実行環境が開発されていたが、スマートフォンやタブレット端末にはまだ対応していない。一方、タブレット端末は、電子教科書やアクティブラーニングのために、教育現場に導入されつつある。そこで、本研究では、従来の Islay に基づいて、タブレット端末の特徴であるマルチタッチ操作の利点を生かしながら、Android[4] 端末上で動作する対話型アニメーションの作成環境の開発を行う。

2. PC 向け Islay の概要

2.1 アニメーションの作成

アニメーションの作成に使われるのは図 1 のようなグラフィカルエディタであり、マウス操作だけで対話型アニメーションを作成できる。状態を表す丸 (○) と遷移を表す矢印 (→) を図面上に追加し、状態遷移図を描くことによって、アニメーションに登場するキャラクターの振る舞いを定義する。各キャラクターは 1 つの状態遷移図に対応し、グループにより管理されている。複数のキャラクターを同じグループに入れることで、同時に登場させることができる。状態遷移図の各状態には、キャラクターの表示画像及びアクションを指定し、状態間の遷移には、アニメーションの実

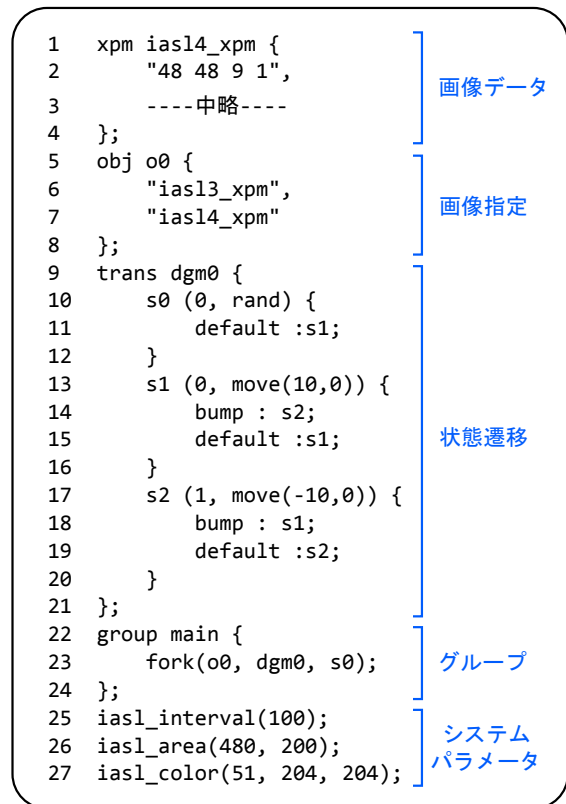


図 2 アニメーション定義ファイルの例

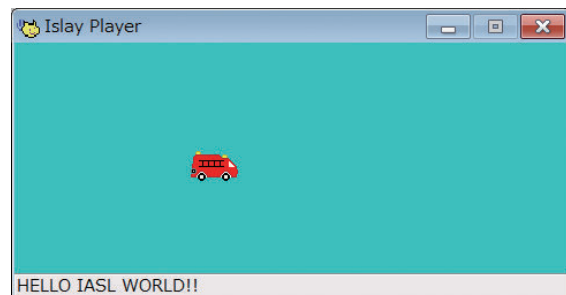


図 3 Islay プレーヤ

行中に発生するイベントを条件として指定する。キャラクターのアクションとして、キャラクター自体の移動や他のキャラクターの生成などの動作を指定できる。また、遷移条件となるイベントには、ユーザからの外部入力やキャラクター同士の衝突などの内部イベントを指定できる。それによって、ユーザとのインタラクションが取れるような対話型アニメーションを実現できる。Islay エディタで作成したアニメーションはアニメーション定義ファイルとして保存される。ファイルの中身は、図 2 に例示されるような独自形式のスクリプト言語で記述される。

2.2 アニメーションの実行

アニメーション定義ファイルをアニメーションとして実行するには、Islay プレーヤが必要になる。図 1 に示す状態遷移図で表現されたアニメーションを実行する Islay プレーヤの画面は図 3 のようになる。図 4 は、その処理の流

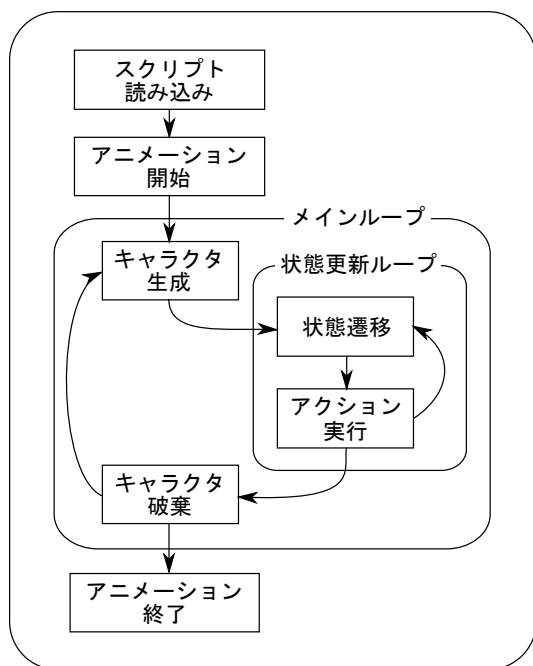


図 4 Islay プレーヤ内部処理の流れ

れを示す。Islay プレーヤは、まずスクリプトファイルを読み込み、アニメーションを開始する。次にキャラクタを生成し、状態更新ループに入る。状態更新ループでは、一定の時間間隔毎に各キャラクタの遷移条件を調べ、その条件が当てはまるなら状態を更新する。最後にすべてのキャラクタを破棄し、アニメーションを終了する。

3. Android 向け Islay

本研究で開発するシステムの全体は、Android アプリケーションとして実装され、アニメーションの作成及び実行に使われるエディタとプレーヤから構成されている。従来の PC 上で動作する Islay の基本仕様を継承し、マルチタッチを含めたタッチジェスチャに対応するように設計されている。これにより、Android 端末のタッチスクリーンの特性を生かしたマルチタッチ操作による対話型アニメーションの作成と実行が可能になる。

4. 実装

現段階では、プレーヤ機能の実装が完成されており、既存のアニメーション定義ファイルの実行が可能になっている。また、エディタの開発は詳細設計まで完成し、実装に着手している。

4.1 エディタ

エディタは Android 端末上で動作し、ハードウェア環境は PC と異なり、マウスによる操作は実現できない。そのため、マウスの代わりに、タッチジェスチャを使ってもエディタの操作ができるように、エディタの操作方法やユーザーインターフェースの構成に対し再設計を行った。図 5 に

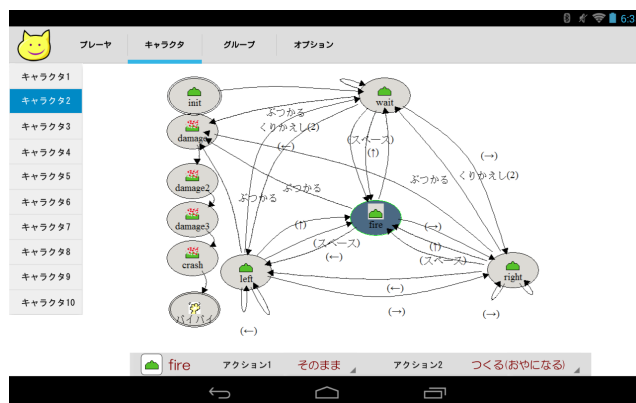


図 5 Android 向け Islay エディタ UI のモックアップ

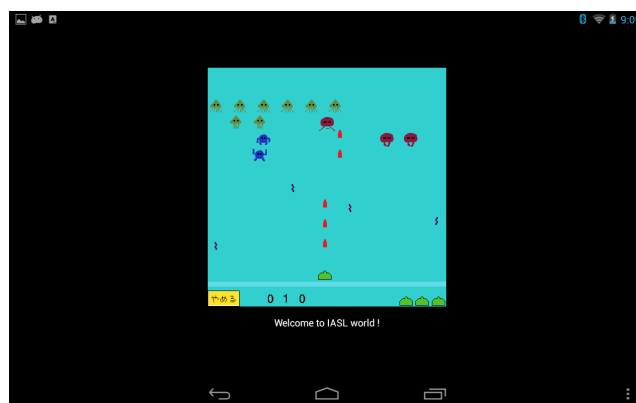


図 6 Android 端末上でのアニメーション実行画面

ユーザーインターフェースのモックアップを示す。

状態を表す丸 (○) に対するマウスのシングルクリックは、そのままタッチで置き換え、右クリックは長押しで置き換えた。遷移を表す矢印 (→) をマウスカーソルで選択することに比べて、タッチで選択することは難しい。そのため、1 本目の指で遷移元の状態をタッチしながら、2 本目の指で遷移先の状態をタッチして遷移を選択するように改良した。該当する遷移が複数あるときには、2 本目の指を動かすことによって、その方向の遷移を選べるようにした。

4.2 プレーヤ

エディタで生成したアニメーション定義ファイルをプレーヤが読み込んで、スクリプトの内容に対して構文解析を行う。その後、アニメーションの構成要素を抽出し、アニメーションを実行する。また、アニメーションの実行中でユーザからのタッチなどのイベントを取得し、アニメーションの遷移条件に反映させる。アニメーション実行中の画面を図 6 に示す。

4.2.1 構文解析の実現方法

本研究では、Java の正規表現パッケージ (java.util.regex[5]) のクラスを利用して、Islay エディタで作成したアニメーション定義ファイルの構文解析を実現している。図 2 に示されている 5 種類のセクション

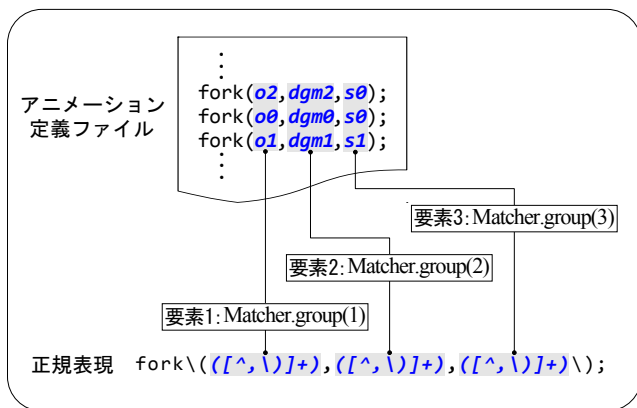


図 7 正規表現によるアニメーション要素抽出の例

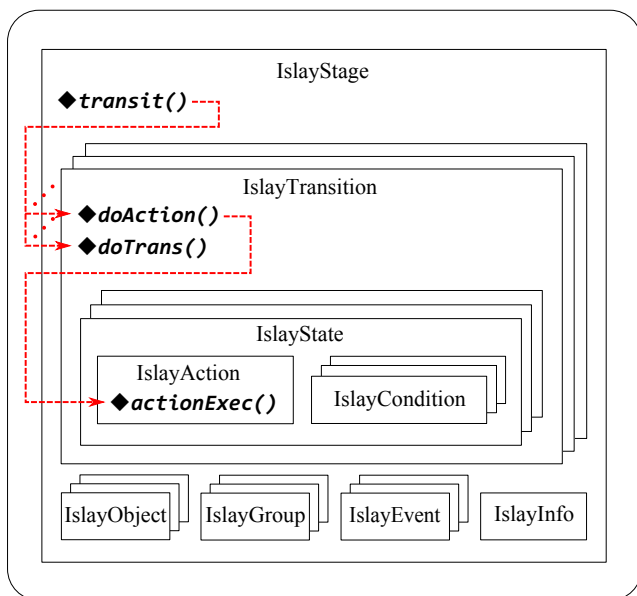


図 8 プレーヤクラスの構成

それぞれに対して、予め用意した正規表現のパターンを用いて、ファイル全体レベルとセクションレベルに分けて、アニメーションの構成要素になるテキストとマッチングを行う。

マッチされた各要素は正規表現のグループ分けによって区切られ、Matcher インスタンスの group() メソッドを呼び出すことで、グループの中身を取得する [6]。セクションレベルでのマッチングの一例を図 7 に示す。パターンマッチングにより抽出したアニメーションの構成要素は、後述のプレーヤクラスのインスタンスとして生成される。

4.2.2 プレーヤの内部仕様

プレーヤ全体はオブジェクト指向モデルに従って設計されており、クラスの構成は図 8 のようになっている。図に示されたクラスの包含関係は内部にあるクラスがその外部のクラスのフィールドとしてインスタンス化されていることを表し、各クラスはそれぞれ独立している。

プレーヤの基盤となるのは IslayStage クラスであり、他のプレーヤクラスのインスタンスがすべてこのクラスの

フィールドから直接または間接に参照できる。IslayStage 以外のクラスはそれぞれ図 2 に示すアニメーション定義ファイルの各セクション及びセクションの中身に対応し、構文解析による取得したアニメーションの構成要素をパラメータとしてコンストラクタに渡して、インスタンス化されている。クラスのフィールドには Java の HashMap を利用し、同じクラスのインスタンスを複数個格納している。

アニメーションに登場するキャラクタはそれぞれ IslayTransition クラスのインスタンスであり、メインループでそこにある transit メソッドが周期的に呼び出される。transit メソッドでは、すべてのキャラクタインスタンスに対して doAction メソッドと doTrans メソッドの呼び出しを行い、さらに、doAction メソッドで現在状態のインスタンスの actionExec メソッドを呼び出している。これにより、各状態で指定されたアクションの実行及び条件判定を元に他の状態への遷移が実現されている。

5. おわりに

本研究では、Android 端末上で動作する状態遷移図に基づく対話型アニメーション作成環境の設計を行い、アニメーションを実行する部分の実装まで完成した。これにより、Islay エディタで作成したアニメーションを Android 端末上で実行できるようになった。今後の課題としては、設計仕様通りにエディタ部分の実装を完成し、システム全体の有効性の評価を行う予定である。

参考文献

- [1] Adobe Flash Professional CC, <http://www.adobe.com/jp/products/flash.html> (最終アクセス: 2013/10/24).
- [2] 岡本秀輔, 鎌田賢, 中尾隆司: 状態遷移図にもとづく対話型アニメーション作成ツールの提案, 情報処理学会論文誌: プログラミング, Vol.46, No.SIG 1(PRO 24), pp.19-27, (2005/01).
- [3] 中川昌幸, 岡本秀輔, 鎌田賢, 米倉達広: 状態遷移図に基づいた Flash Animation 記述環境, インタラクシオン 2006, インタラクティブ発表, (2006/03).
- [4] Android, <http://www.android.com/> (最終アクセス: 2013/10/24).
- [5] java.util.regex, <http://developer.android.com/reference/java/util/regex/package-summary.html> (最終アクセス: 2013/10/24).
- [6] Java 正規表現の使い方, <http://www.javadrive.jp/regex/> (最終アクセス: 2013/10/24).