

R Tコンポーネント操作マニュアル

チーム開発力の向上を目指した
RT ミドルウェアによる
ゲームコンポーネント

2013 年 12 月 4 日版

芝浦工業大学

デザイン工学部デザイン工学科

鈴木 弥絵・榎谷 奈々・近藤 貴大

佐々木 毅

更新履歷

2013 年 12 月 3 日 第 1 版

- 1 本コンポーネントの概要
 - 1.1 開発の背景
 - 1.2 今回テーマとしたゲームの説明と機能説明
 - 1.3 本稿で利用しているコンポーネント
 - 1.4 開発環境
- 2 本コンポーネントに必要な実機
- 3 コンポーネントの説明
 - 3.1 全体制御コンポーネント(Main_comp)
 - 3.2 アニメーションコンポーネント(Anime_comp)
 - 3.3 音声判断コンポーネント(C_Judgment_comp)
- 4 使用方法
 - 4.1 RTSystemEditor によるシステム構築の手順
 - 4.2 ゲームコンポーネントの使用手順
 - 4.2.1 部分エッジ画像コンポーネントへの画像の登録
 - 4.2.2 音声認識 RTC の使用方法
 - 4.3 接続方法
- 5 お問い合わせ

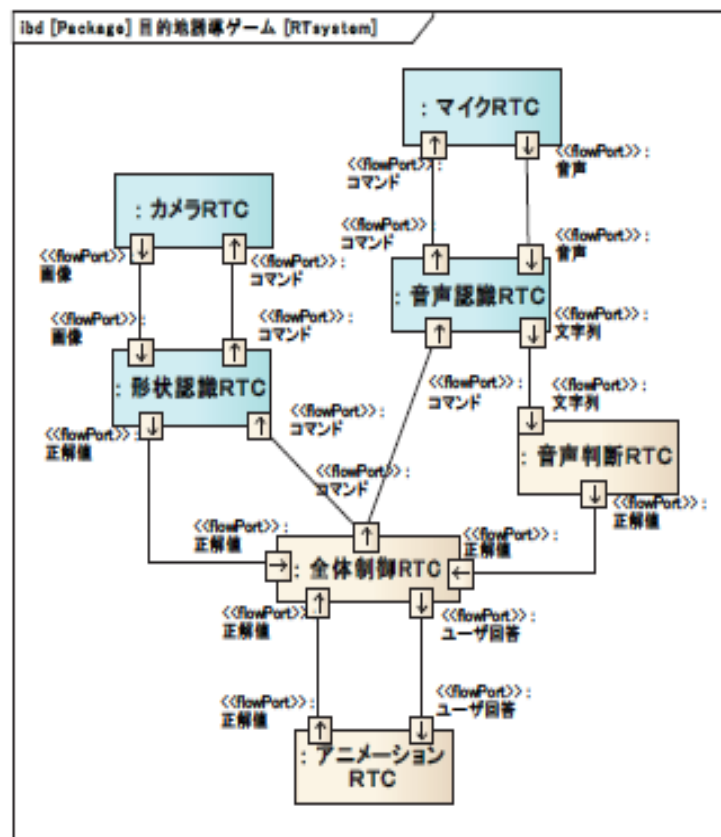
1. 本コンポーネントの概要

1.1 開発の背景

平成23年度の厚生労働省の雇用動向調査によると産業別入職・離職状況によると、情報通信業は生活関連サービス業・娯楽業に次いで入職超過率が低いのが現状です。25～29歳の離職の原因に着目すると、[収入の少なさ][会社の将来が不安]に続き3番目の理由として[仕事内容に興味が持てない]がありました。このようなミスマッチを防ぐためにも、在学中に実際の業務に近い形でシステムを作成する必要があると考えました。しかし、学生の知識ではモジュールの分割法やそれらを統合する知識が充分ではなく開発に時間がかかってしまうという問題点があげられます。また、技術力が足りないためアイデアを実現できないなどという問題点も挙げられます。そこで私はこれらの問題点に対してRT ミドルウェアの特徴が有効であると考えました。RTミドルウェアのコンポーネントという概念を用いれば、各機能を並行して開発ができるため前者の課題が改善できると考えます。またRTミドルウェアのコンポーネントには再利用性があるため、以前に作ったコンポーネントを利用することが可能です。これを用いれば、技術力が足りなくとも自分たちのアイデアを実現することができると考えました。今回はこの有用性を確かめるために実際にチームを組み、全員が一度は触れたことのあるゲームをテーマに開発を行いました。また、この有用性に関しては別途資料で報告いたします。

1.2 今回テーマとしたゲームの説明と機能説明

キャラクターの進行方向の罫を回避し、ゴールを目指すゲームです。設定としてキャラクターは一方向に進んでいき、その道に障害が現れます。ユーザーはキャラクターをその障害を回避する選択肢を選びキャラクターをゴールまで導きます。もし、ユーザーが障害に合う答えを選択できなかったり、画面上に表示されている解答時間を超えてしまったりした場合はそこでゲームオーバーとなります。機能として1つめは通路の補助です。ユーザーは物体をカメラに認識させます。すると、その物体に応じて障害の部分に橋などが表示され、キャラクターをゴールへと誘導することができます。2つめはキャラクターの動きの補助です。マイクに向かって解答時間中に言葉で指示を与えると、キャラクターがはねたり、走ったりして障害を乗り越えます。この機能を実現するために、今回は大きく4つの機能に分けました。まずユーザーがカメラに映した物体を認識し、どれを選択したかを判断する『形状認識』です。次に、ユーザーがマイクに向かって発した指示が何かをマイクから受け取り、なにを命令したのかを判断する『音声認識』です。それらの情報の回収の始まりや終わりを支持し、またアニメーションに指示をだす『全体制御』、その指示に従いアニメーションを変化させる『アニメーション』の4つです。



1.3 本稿で利用しているコンポーネント

前述した機能を実現するために以下の 2 つのコンポーネントを再利用しました。

- 部分エッジ画像認識モジュール(**partialedge**)
カメラに何を選択したのかを判断するコンポーネント。
http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID235/
- OpenHRI (Julius)
マイクに向けて発した命令がなにかを判断し、文字列にするコンポーネント。
<http://openhri.net/download/>
- OpenHRI (PortAudioInput)
マイクから音声を取得するコンポーネント。
また、機能をそろえるために以下のコンポーネントを作成しました。
- 全体制御コンポーネント(**Main_comp**)
システム全体の制御のコンポーネント。おもに解答の正解・不正解を判断する。
- アニメーションコンポーネント(**Anime_comp**)
解答により、画面上のアニメーションを変化させるコンポーネント。
- 音声判断コンポーネント(**moji_comp**)
音声認識のコンポーネントの出力を画像認識と同じ形に認識するコンポーネント。

1.4 開発環境

本コンポーネント群は Windows にて動作確認を行っています。

- Windows7 (32bit 版)
- RTミドルウェア:OpenRTM-aisti1.1.0-RELEASE
/

- コンパイラ:Microsoft Visual C++ 2010 Express
- Eclipse: Eclipse 3.4.2 + OpenRTM Eclipse tools 1.1.0-RC3
- CMake: CMake 2.8.8

また、今回開発したコンポーネントのうち、アニメーションコンポーネント (**Anime_comp**)については依存ライブラリとして OpenCV を利用しています。

- OpenCV 2.4.6

また、今回再利用させていただいた部分エッジ画像認識モジュール(**partialedge**)については依存ライブラリとして OpenCV と Python 版の RT ミドルウェアを利用しています。

- OpenCV 2.2
- RT ミドルウェア:OpenRTM-aist-Python-1.0.0-RELEASE

2 本コンポーネントで必要な実機

本コンポーネントではマイクと USB カメラの 2 つを用います。このコンポーネントを使用する前にこの 2 つが PC に接続していることを確認してください。

マイクは特に種類に指定はありませんが、できるだけ口元に持ってくる事のできるものにしてください。また USB カメラは OpenCV のキャプチャ関数と接続可能なものならどのカメラでもかまいません。また、物体を映し出す背景として黒の背景が好ましいため、黒い紙や、黒い布を用意しておいてください。



3 コンポーネントの説明

3.1 全体制御コンポーネント(Main_comp)

全てのコンポーネント間へ動作の指示をだし、ゲーム内のアニメーションを判断するコンポーネントです。

・ InPort

名称	型	説明
main_ain	TimedLong	Anime_comp からゲーム内の罠の情報を受け取る
main_sin	TimedLong	Moji_comp からユーザーの発した言葉が正解か不正解かの値を受け取る。

・ OutPort

名称	型	説明
main_ain	TimedLong	アニメーションコンポーネントへユーザーの解答により、画像の差し替えの命令を送る。正解していた時は 1、不正解のときは 0。
main_ein	TimedDoubleSeq	ユーザーの解答開始時に情報取得開始のために partialedge へコマンド(=1)をおくる。
main_sin	TimedLong	ユーザーの解答開始時に情報取得開始のためにマイクコンポーネントへコマンド(=1)をおくる。

・サービスポート

名称	型	説明
main_ein	TimedLong	アニメーションコンポーネントへユーザーの解答により、画像の差し替えの命令を送る。正解していた時は1、不正解のときは0。

3.2 アニメーションコンポーネント(Anime_comp)

ゲームの解答により、キャラクターの動きに変化をつけ、背景を動かすコンポーネントです。一定の距離進むと罠の情報を Main_comp に渡し、Main_comp から受け取ったコマンドによりゲームを終了したり、アニメーションに変化をつけキャラクターを動かしたりします。

・ InPort

名称	型	説明
anime_in	TimedLong	Main_comp からユーザーの解答情報を受け取る

・ OutPort

名称	型	説明
anime_out	TimedLong	Main_compへ罠の情報を渡す。(落とし穴のときは5を、木のときは7を渡す)

・サービスポート

なし

3.3 音声判断コンポーネント(C_Judgment_comp)

Julius から出てきた文字列が選択肢の中のどれかを判断するコンポーネントです。
ユーザーが選択肢の中でどの言葉を発したのかをフラグに置き換え出力します。

・ InPort

名称	型	説明
C_Judgment_in	TimedLong	julius_comp から受けとったどれが一番確立高いのかが書かれた配列を受け取る

・ OutPort

名称	型	説明
C_Judgment_out	TimedLong	Main_comp へ一番確立高かったものがジャンプかそれ以外かを送る。(ジャンプのときは 1、それ以外は 0 を渡す)

・ サービスポート

なし

4 使用方法

4.1 RTSystemEditor によるシステム構築の手順

RTSystemEditor を用いたシステム構築は通常以下の手順で行われます。

1. ネームサーバの起動
2. RTSystemEditor の起動
3. ネームサーバへの接続
4. コンポーネントの起動
5. システム構築と実行

これらの手順はどのようなシステムでも共通ですので、操作方法はOpenRTM-aist のホームページ (<http://www.openrtm.org/>) を参照してください。以下の節では手順5 に関して、ゲームコンポーネントの使用手順を説明します。

4.2 ゲームコンポーネントの使用手順

ゲームコンポーネントの使用手順をまとめると以下ようになります。

0. 各ファイルをインストールし、前準備を行う。
 - (a)部分エッジ画像認識モジュールのマニュアルに沿ってゲーム内で使用する画像の登録を行う。
 - (b)sample.xml と sample2.xml の二つを HRIaudio のファイルの中に入れる。
 - (c)HRIaudio から portaudioinput.exe を実行する
 - (d)ダウンロードした各コンポーネントを立ち上げる
1. 他のコンポーネントに接続する
2. コンポーネントをアクティブ化する

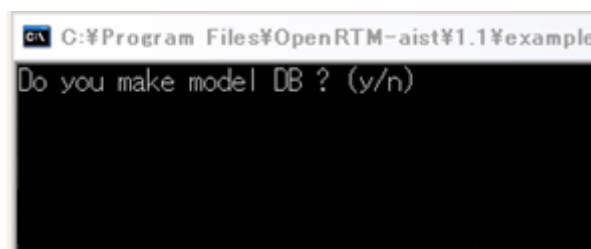
以下の節からは接続する前段階として行わなければならない 2 つの使用手順を説明します。

4.2.1 部分エッジ画像コンポーネントへの画像の登録

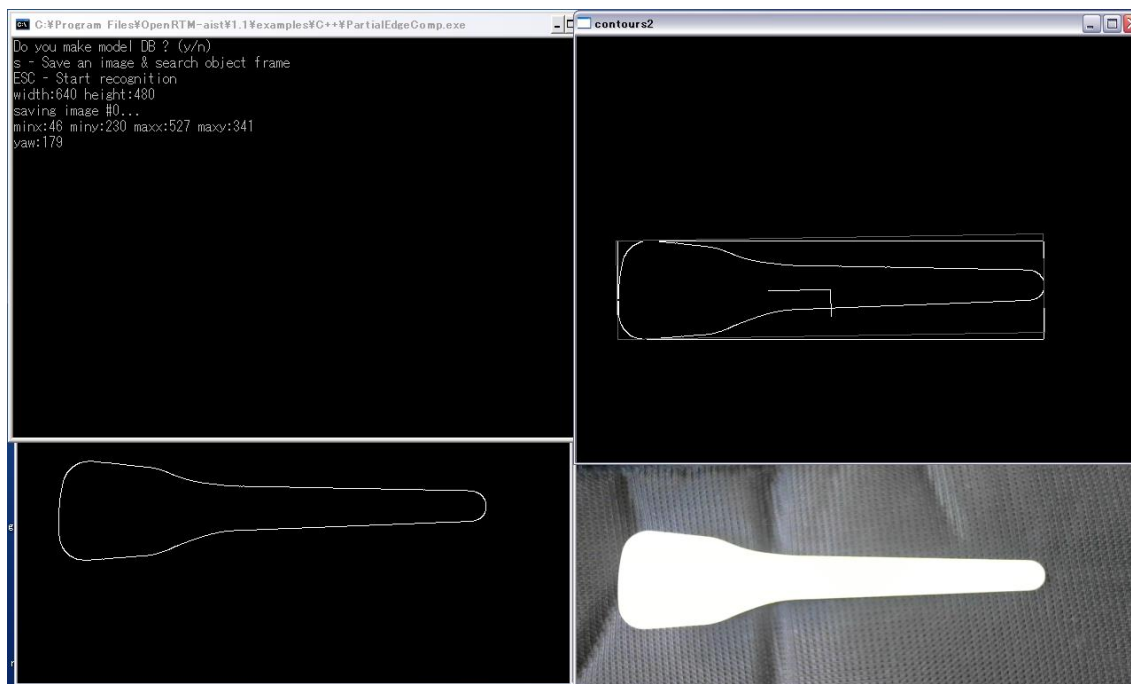
詳しくは部分エッジ画像コンポーネントの操作マニュアルを読んでください。ここでは流れを説明します。

1. PartialEdgeComp.exe を立ち上げる。
2. コンソールパネルに『Do you make model DB?(y/n)』と表示されるので y を入力する。
3. 対象物を黒い背景の上に置きカメラ画像の中心に来るように置く
4. 『S』キーを押すと登録される。
5. 画面登録が終わったらエスケープキーを押す
6. 一度コンソールを閉じてください

PartialEdgeComp.exe を立ち上げると下図のようにコンソールパネルに表示されます。



y を入力すると下図のようになります。



これらの画像が **Train** と書かれたディレクトリに保存されるため、その中に保存されていたら成功です。保存されていたらコンソールを一度閉じてください。

4.2.2 音声認識 RTC の使用方法

Julius を起動するときには **Windows** の場合普通にファイルを開くのではなくコマンドプロンプトから実行する必要があります。通常は **OpenHRI** のホームページのマニュアル通りなのですが、今回特別なライブラリを使用しているのでこちらで紹介します。

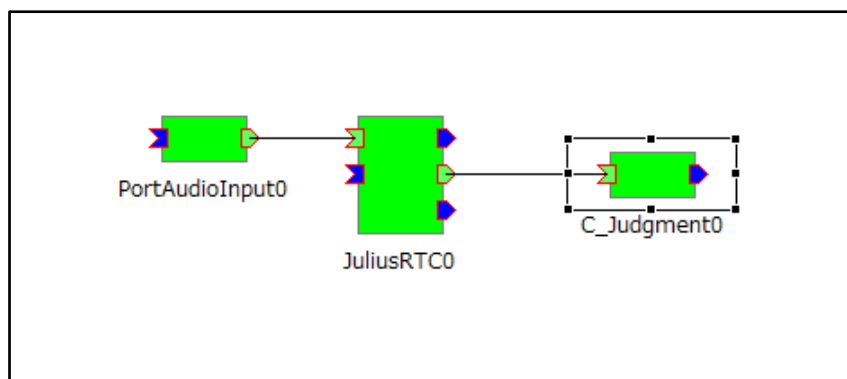
1. **sample.xml** と **sample2.xml** の二つを **OpenVoice** のファイルの中に入れる。
2. コマンドプロンプトで **julius.exe** と **sample.xml** を立ち上げる
3. **C_Judgement.exe** と **OpenHRIAUDIO** 内にある **portaudioinput.exe** を立ち上げる
4. コンポーネントをアクティブ化する
5. 他のコンポーネントと接続する

まず始めに **julius** の実行ファイルが入っている場所に、今回ダウンロードした **sample.xml** と **sample2.xml** を移動してください。その後、コマンドプロンプトから実行するのですが、その際に **juliuscomp.exe** とともに **sample.xml** を実行してください。

```
コマンド プロンプト
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\y10140>cd C:\Program Files\OpenHRVoice
C:\Program Files\OpenHRVoice>juliusrtc.exe sample.xml
```

上図のように2つ同時に実行します。
実行したら下図を参考にコンポーネントを接続してください。



接続しマイクに言葉を話すと以下のように表示されます。

```
C:\Users\y10140\workspace\C_Judgment\build\src\Debug\c_judgmentcomp.exe
場所text:105
場所":115
結果: jump
正解!
場所rank:82
場所text:105
場所":114
結果: bye
不正解
場所rank:82
場所text:103
場所":114
結果: hello
不正解
場所rank:82
場所text:104
場所":114
結果: jump
正解!
場所rank:82
場所text:105
場所":115
結果: jump
正解!
```

現在判断される言葉は『こんにちは』『さようなら』『ジャンプ』の3つです。「こんにちは」といった場合下図のように表示されます。結果のところに『hello』その下に不正解と表示されれば成功です。

```
場所text:103  
場所":114  
結果: hello  
不正解  
場所rank:82
```

また「さようなら」といった場合は下図のようになります。結果のところに『bye』その下に不正解と表示されれば成功です。

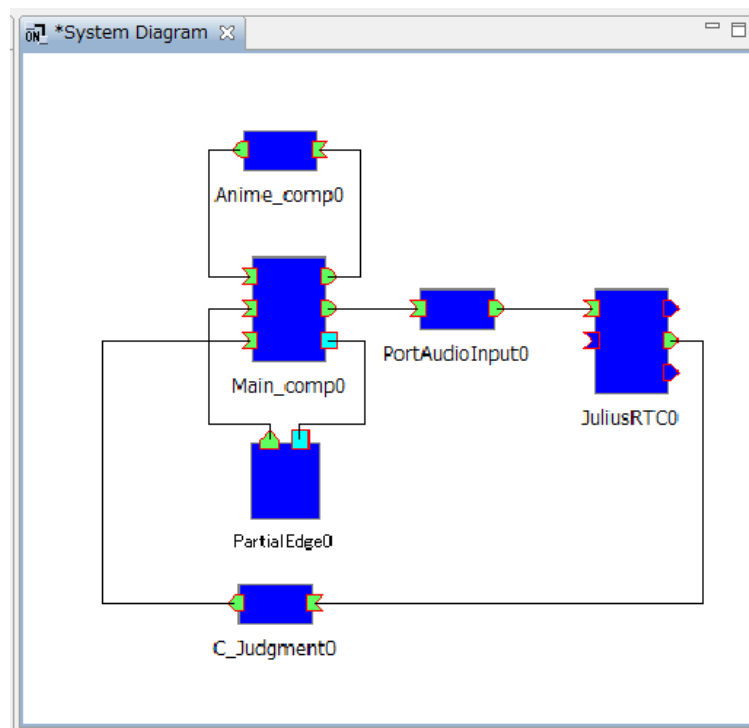
```
場所text:105  
場所":114  
結果: bye  
不正解  
場所rank:82
```

「ジャンプ」といった場合は下図のようになります。結果のところに『jump』その下に正解と表示されれば成功です。

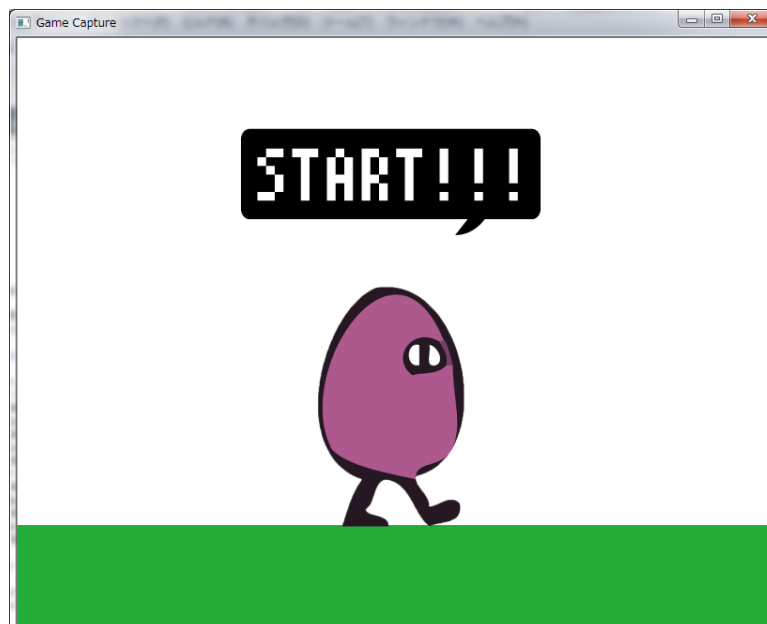
```
場所text:105  
場所":115  
結果: jump  
正解!  
場所rank:82
```

4.3 接続方法

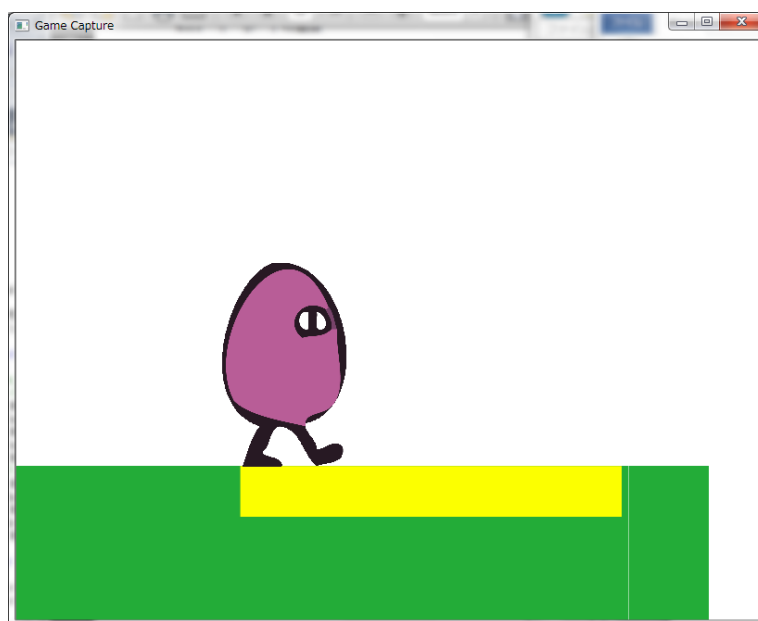
そして残りのインストールしたファイルを実行してください。そして、システムエディタを用いて下図のようにコンポーネントを接続し、システムを構築してください。



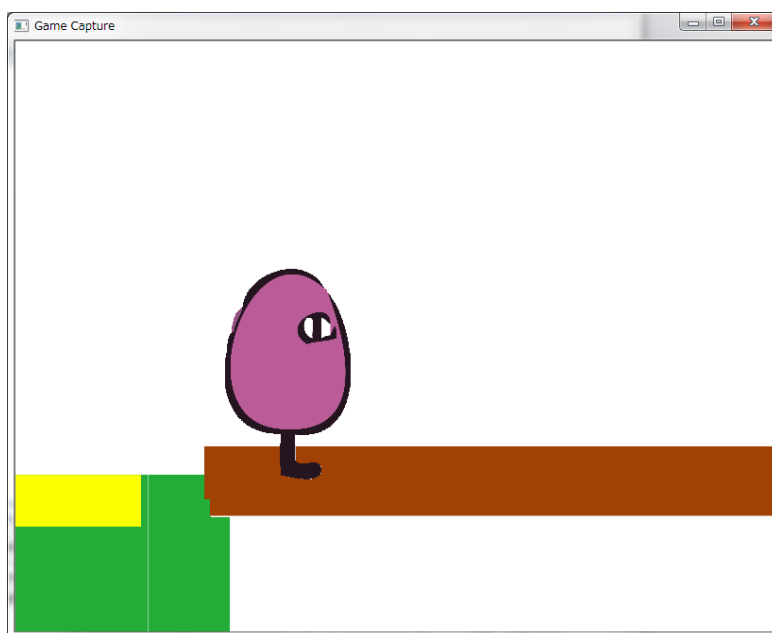
ALLActivateを行うと以下のような画面が現れます。



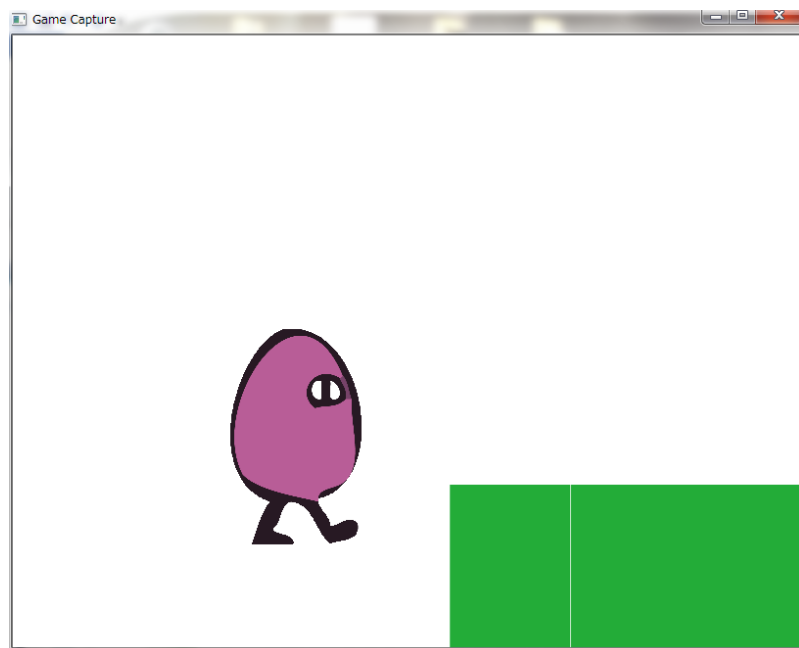
このような画面が表示されるので **Esc** キーをクリックすると以下のような画面になり、ゲームがスタートします。



まず道が上図のように黄色くなっている部分に差し掛かったら、カメラに 4.2.1 で登録した物体をかざしてみてください。すると下図のように橋がかかります。

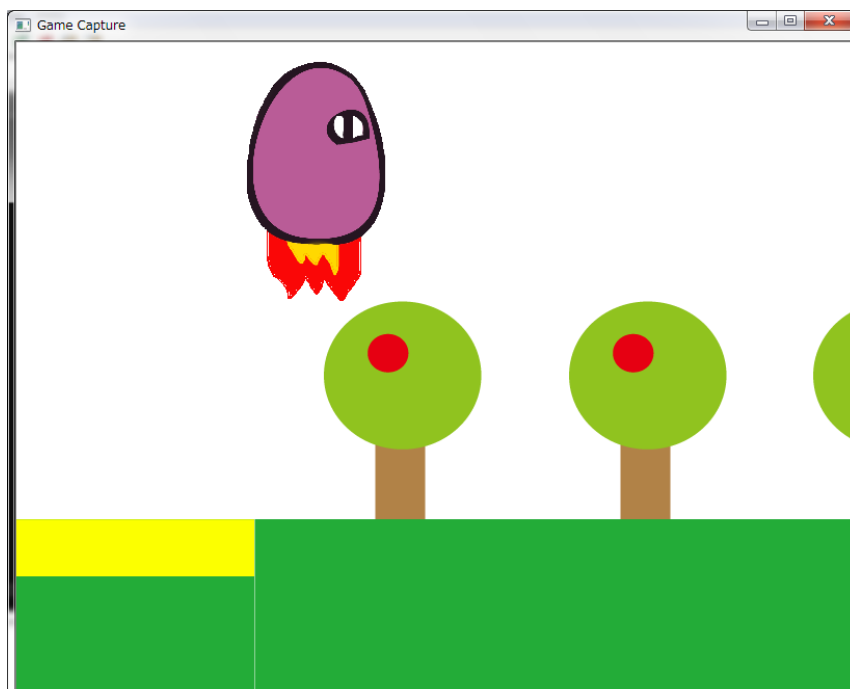


もしこの部分で何もかざさなかったり、間違えた選択肢を選択すると下図のようにキャラクターが落ちていきます。

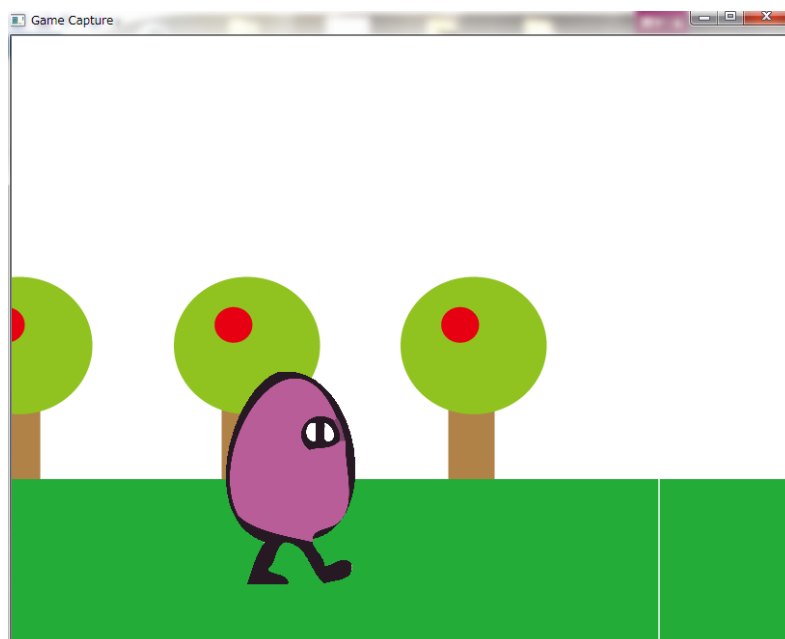


成功した場合はゲームが続きます。歩いている先に再び道が黄色くなっている部分があるのにその部分にキャラクターが乗ったら今度はマイクに向かって命令してください。今回の正解は「ジャンプ」になります。

ジャンプと唱えるとキャラクターは障害物を飛び越えます。



もしそれ以外の言葉を命令するとキャラクターは落ちて行ってしまいます。



このようにキャラクターが動けば、成功です。

5 お問い合わせ

本コンポーネントにつきましては、まだ改善の余地があるものと考えております。ご要望、バグ報告、マニュアルの記述の不備等に関しましては、芝浦工業大学デザイン工学部デザイン工学科の鈴木までご連絡ください。

【問合せ先】

〒108-8548

東京都港区芝浦3-9-14

芝浦工業大学 鈴木弥絵

y10140@shibaura-it.ac.jp