

Summer Science Camp 2014

最終配布版

初めてのソフトウェア開発体験
～オブジェクト指向でゲーム開発に挑戦～

My First Experience of Software Development
～ Let's Challenge to Develop Games
by Object-Oriented Method ～

[第 3.1 版]

平成 22 年 4 月
平成 22 年 8 月改訂
平成 23 年 8 月改訂
平成 26 年 5・6・7 月改定中
東京工科大学
コンピュータサイエンス学部
担当教員：亀田弘之

本資料に対する質問等は、下記までご遠慮なく。

〒192-0982 東京都八王子市片倉町 1404-1

東京工科大学コンピュータサイエンス学部

亀田弘之（教授・学部長）

e-mail: kameda@stf.teu.ac.jp

Katakura 1404-1, Hachioji, Tokyo 192-0982 , Japan

School of Computer Science (SC)

Tokyo University of Technology (TUT)

Prof. & Dean of SC Hiroyuki KAMEDA

はじめに

我が国において情報システムは、金融、流通、製造あるいは電子政府などあらゆる分野に深く浸透し、社会基盤としての重要な役割を担っている

それ故に情報システムは安全・安心かつ持続可能なものであり、かつ、今日では地球環境に配慮してエコなものであることが強く求められている。一方周知のように、情報システム実現の主要な前提の1つである情報技術は、堰を切ったかのように急速に進展・高度化しておりその勢いは弱まる気配はなく、また、情報システムに対する顧客(ユーザ)の要求も絶えず目まぐるしく変わっている。例えば携帯電話は数カ月もすればモデルチェンジをしていることからそのことが伺い知れよう。その結果として、現在の情報システムは人間の手に負えないほど複雑化しかつ大規模化し、ソフトウェアの生産性や品質を向上させることは極めて難しくなっている。しかしながら、情報社会の国際化はもはや避けて通ることはできないため、我が国も国際競争に勝ち抜かなければならず、また、安心・安全な社会の実現のためにも常に高い製品品質を保証することが求められているため、この問題を解決することは社会的に極めて意義あることである。

このような状況に鑑み、これまで企業では長期間のOJT(On the Job Training)を通じてICT(Information Communication Technology)人材を社内教育により育成し、この困難な問題に対処してきた。しかしながら先にも述べたように、昨今の競争環境のグローバル化によって企業は投資の選択と集中を迫られ、高度ICT人材育成を大学に求めるようになってきた。このような社会情勢の変化により、「実践的かつ体系的なソフトウェア教育方法および教材の研究開発」と共に大学と企業のみならず高校をも含めた「ソフトウェア教育体制の整備」に可及的速やかに着手されなければならない国家的課題である。本実験はこのような問題意識のもと研究開発中の教育素材・教育方法の一例である。

なお本実験教材・教授方法等は、東京工科大学学内に文部科学省の援助により設立されているオープンリサーチセンターの1つ、タンジブル・ソフトウェア教育・研究センター(代表：CS学部中村太一教授)の研究の結果得られたものである。ここで「タンジブル・ソフトウェア教育」とは以下のものを2本柱とする先駆的なソフトウェア教育方法である。

(1) つくって学ぶものづくりソフトウェア教育

プログラムで欲しいものを自由につくり、動かし、夢を自ら実現し達成感と楽しさを体現することを通し高度な技術を学ぶ。

(2) 経験して学ぶ実践できるソフトウェア教育

要求獲得、要件定義、設計、製造、試験、保守・運用にわたる情報システム開発の全工程の方法論を実践的かつ体系的に学ぶ。

以上のことを踏まえて、本実験は準備されたものである。

なお、本実験の一部はタンジブル・ソフトウェア教育・研究プロジェクトの支援を得て準備されたものである。

実験の趣旨

本演習は、「オブジェクト指向プログラミングに基づく実践的ゲーム開発」を通じて、オブジェクト指向設計、オブジェクト指向プログラミング、ソフトウェア開発(ソフトウェア工学)に対する造詣ぞうけいを深めてもらうことを第一の趣旨として設計されている。またこれとともに、先にも述べた「つくって学ぶものづくりソフトウェア教育」、「経験して学ぶ実践できるソフトウェア教育」の枠組みのもと、PBL(Project Based Learning)という学びの方法をも諸君らに知ってもらうことも本演習の重要な趣旨きんめいの1つである。単にゲームを作って楽しむことだけが目的ではなく、また Java 言語のプログラミングをすることだけが目的ではないことを肝きんに銘めいじること。

なおこの趣旨を意味あるものとするためには、学生諸君自らが本演習に積極的に興味を持ちつつ、自ら主体的にかかわり行動することが肝要かんようである。この点にも十分配慮してもらいたい。

学習目標

以下、本実験の学習目標を列挙する。学生の皆さんは下記の各項目内容を意識しつつ実験に取り組んでほしい。

1. 新しいソフトウェアをインストールする体験を積む。
2. Java 言語に親しむ。
3. オブジェクト指向設計についての知識と経験を増やす。
- ④ オブジェクト指向プログラミングのスキルを高める。
- ⑤ プログラミングを楽しむマインドを育てる。
- ⑥ ものづくりを通してエンジニアの素質を伸ばす。
- ⑦ コラボレーションに慣れるとともに、そのスキルレベルを上げる。
- ⑧ PBL(Project Based Learning)の体験をする。
- ⑨ 技術英語に親しむ。
10. UML(Unified Modeling Language)を使ってみよう。
11. デザインパターンを知ろう。
12. 主体的に行動するマインドを身につけよう。
- ⑬ 一般に、コンピュータを学ぶために必要なことは何かを見つけよう。
- ⑭ あなたにとって、コンピュータをさらに学ぶために何が必要か気が付こう。

実験の構成

本実験は大きく以下の3つの要素で構成されている。

第1部 Greenfoot の準備

第2部 Greenfoot による簡単なゲーム作成実験

第3部 ゲーム作成プロジェクト (PBL 演習)

まず、第1部では、“Class First” という着想に基づき開発されたプログラミング開発教育ツール BlueJ をさらに発展させた “Greenfoot” というオブジェクト指向設計・プログラミング学習環境の設定を行う。その際 Greenfoot のチュートリアルを参考にして、Greenfoot の概要を知る。第2部では、Greenfoot を利用して簡単なゲームを多数作成し、Greenfoot の利用の仕方を覚える。最後に第3部では、少人数のグループでゲーム開発会社を仮想的に立ち上げ、ゲームを実際に開発する。ゲームの開発はより実践的なものとなるように、実際の開発プロセス「要求定義、要件定義(企画書)、設計(外部設計・内部設計)、コーディング、試験、運用」をなぞる形式を採る。

なお以下の作業においては、多くのトラブルに遭遇することが予想されるが、それらはすべて「トラブルシート」に記録として残すこと。また、「気づきシート」には自分自身で気がついたことをその場ですぐか聞き込むこと。

第1部 Greenfoot の準備

1. 環境の構築

- Greenfoot の Web サイトに行く。
URL: <http://www.greenfoot.org/>
- Greenfoot をダウンロードする。
- Greenfoot を解凍する。
- Greenfoot が起動することを確認する。
- (Option)
 - Java の開発環境として Java の最新版とする。

(注1) 関連の URL や作業手順は当日説明があるのでそれを参考にすること。

(注2) USB メモリーから起動する方法もある。詳しくは Greenfoot のサイトを見ること。

2. Greenfoot の基本操作 (起動と終了)

課題 1 Greenfoot を起動してみよう。

手順 1-1 Greenfoot のアイコン  をダブルクリックすると、起動する。

手順 1-2 Greenfoot のウィンドウ右上で、[Scenario]→[Quit] を選ぶと、終了する。

[動画 2-1]も参照のこと。

3. Greenfoot の操作理解 (シナリオ Wambats による Greenfoot の概要理解)

課題 2 Greenfoot のチュートリアルを読んで、Greenfoot を用いた Java プログラミング開発方法の概要を学びなさい。

(ゲームのことは後で考える。まずは、開発環境に慣れることに専念する。)

【学習目標】

- 自分で Greenfoot を起動できる。
- 自分で Greenfoot を終了できる。
- クラス World の下位クラス (ここでは仮に、myWorld を呼ぶ) を作成できる。
- World の下位クラス myWorld の描画サイズを調整できる。
- クラス Actor の下位クラス (ここでは仮に、myActor を呼ぶ) を作成できる。
- Actor の下位クラス myActor のインスタンスを生成できる。
- シナリオ Wombat の仕組みを理解する。

(注) シナリオ Wombat の詳細は、この時点では理解する必要はない。

課題 3 上記課題 2 で学んだことを、グループで教え合い、自分の理解が正しいことを確認するとともに、友人の理解が間違っているときにはそれを指摘し、教え合いなさい。

第2部 Greenfoot による簡単なゲーム作成実験

以下では、Greenfoot によるオブジェクト指向プログラミング手法を学ぶ。第2部の学習を通じて Java プログラミングのウォーミングアップをしよう。

それでは、以下の各作品を順に作成していき、ゲーム作成のための基礎的知識・スキルを身につけていきます。各作品を作りながら、もっといい作品、もっと面白いゲームに発展させることができないか絶えずアイデアを練ること。

課題0 次の作品 No. 1 を制作しよう。

[作品 No. 1 テントウムシがいっぱい]

課題1 次の作品 No. 1 を作成せよ。

[作品 No. 1 テントウムシワールド]

- ・作成プログラムの例示：(ビデオ教材の視聴)
- ・基本シナリオ： とある土の庭にテントウムシ(ladybug)のテンテンが一匹おり、地面の上を這っている。土の庭には何もなく、テンテンは画面(庭)の端にぶつかってもそのまま強引に進もうとする。(従って、その場で動きが見かけ上停止することもある。)
- ・学習内容：
 - ・ Greenfoot における基本用語(World クラス、Actor クラス)を理解する。
 - ・ サブクラス(subclass)の生成方法を知る。
 - ・ 背景画像の設定方法(変更方法)を知る。
 - ・ getX, getY, setLocation などのメソッド(method)の使い方を知る。
 - ・ Greenfoot におけるプログラミング手順を習得する。

手順 1-1 土の庭(テントウムシのテンテンの存在場所)を作成する。

手順 1-2 テントウムシクラスを定義する。

手順 1-3 テントウムシを土の庭に登場させる。

手順 1-4 テントウムシが動くようにプログラミングする。

手順 1-5 実際に実行してみる。

手順 1-6 総合テストを実施する。

- (ア) 満たすべき要件を列挙する。
- (イ) 各要件を満たしていることを確認するためのテスト方法を考える。
- (ウ) 実際にテストをし、要件を満足していることを確認する。

(注1) 以上の課題は、教員のプロジェクターによる説明を見ながら一步一步進める。
説明事項は必ずメモしておくこと。

(注2) ここでいう“総合テスト”とは、最終的に作成されたソフトウェアが当初の“要件定義(仕様)”に合致したものを確認する開発工程のことを意味している。

課題2 次の作品 No.2 を作成せよ。

[作品 No.2 大海原に漂うヨット]

- ・作成プログラムの例示：(ビデオ教材の視聴)
- ・基本シナリオ： 青々とした大海原にヨットが浮かび波に揺られて動いている。なおヨットの動きは酔歩(ランダムウォーク, random walk)とする。
- ・学習内容：
 - ・ getRandomNumber メソッドの使い方を知り、乱数を利用する。
 - ・ Greenfoot におけるプログラミング手順になれる。
 - ・ actor メソッドにロジックを記述する練習を積む。

手順 2-1 ヨットが漂うための海(背景世界)を作成する。

手順 2-2 ヨットクラスを定義する。

手順 2-3 ヨットを海に登場させる。

手順 2-4 ヨットが動く(酔歩する)ようにプログラミングする。

手順 2-5 動作を確認する。

手順 2-6 総合テストを実施する。

(ア) 満たすべき要件を列挙する。

(イ) 各要件を満たしていることを確認するためのテスト方法を考える。

(ウ) 実際にテストをし、要件を満足していることを確認する。

発展課題 2-6 海に潮の流れを設定する。潮は左から右へゆっくり流れるものとする。

課題3 次の作品 No.3 を作成せよ。

[作品 No.3 自動車を上手に運転しよう]

- ・作成プログラムの例示：(ビデオ教材の視聴)
- ・シナリオ： 3軒の家しかない田舎町がある。その町中を1台の自動車がのんびりと走り回る。自動車はキーボードでコントロールする。
↑キー(up-key)、↓キー(down-key)、→キー(right-key)、←キー(left-key)のどれかが押され続けると、その間それぞれ上(up)、下(down)、右(right)、左(left)に自動車が移動する。
- ・学習内容：
 - ・ キーボードによるインスタンス(オブジェクト)の制御法を知る。

- isKeyDown メソッドの使い方を知る。
- 打鍵されたキーを検知する方法を知る。

• 参考コード :

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
public class aaa extends Actor
{
    public void act()
    {
        if(Greenfoot.isKeyDown("up")){
            setLocation(getX(),getY()-1);
        }else if(Greenfoot.isKeyDown("down")){
            setLocation(getX(),getY()+1);
        }else if(Greenfoot.isKeyDown("left")){
            setLocation(getX()-1,getY());
        }else if(Greenfoot.isKeyDown("right")){
            setLocation(getX()+1,getY());
        }
    }
}
```

【アドバイス】 矢印キー以外のキーを用いるように変更してみよ。

詳しくは Greenfoot の Web ページ (<http://www.greenfoot.org/>) を参照のこと。また、上記のコードの意味(特にロジック)は完全に理解すること。

課題 4 次の作品 No. 4 を作成せよ。

[作品 No. 4 毒蛇に気をつけろ!]

- 作成プログラムの例示 : (ビデオ教材の視聴)
- 基本シナリオ : とある未来少年トミーが謎の庭に入り込んでしまった。
庭には毒蛇が 5 匹おり、トミーは毒蛇に捕まらないように逃げ回る。
- 要求仕様 : 毒蛇は酔歩するものとし、トミーの位置はキーボードで制御する。
- 学習内容 : ここまでに習った知識を応用する練習をする。
 - キーボード制御と乱数利用を組み合わせる。

課題 5 次の作品 No. 3 を作成せよ。

[作品 No. 5 カエルワールド]

- 作成プログラムの例示：(ビデオ教材の視聴)
- 基本シナリオ： 青々と生えた芝生の上にカエルが5匹いる。カエルはじっとしているが、マウスカーソルをカエルに近づけてクリックするとカエルは跳んで逃げる。
- 要求仕様：カエルの逃げ方は乱数を利用するように設計する。
- 学習内容：
 - getX, getY, setLocation メソッドの応用
 - MouseClicked メソッドの使い方を知る。

- 参考コード：

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
public class Snipper extends Actor
{
    public void act() {
        // Add your action code here
        if(Greenfoot.mouseClicked(this)){
            frogMove(5);
        }
    }
    public void frogMove(int step){
        int a = Greenfoot.getRandomNumber(10);
        int b = Greenfoot.getRandomNumber(10);
        a = a%step - 2;
        b = b%step - 2;
        setLocation(getX() + a*20, getY() + b*20);
    }
}
```

【アドバイス】 このコードを解析し、動作を日本語で記述して見なさい。

このコードはあくまでも参考であり、必要に応じて各自が書き直してもよいものとする。

課題6 すでに作った作品 No. 5 に以下の変更を加えよ。

[作品 No. 6 鳴き声を上げるカエルワールド]

- ・作成プログラムの例示：(ビデオ教材の視聴)
- ・変更要求： カエルはクリックされると逃げるだけでなく、“ケロケロ”と鳴き声を上げる。なお、効果音ファイルは指定された Web サイトからダウンロードできるが、自分で他を探しても、あるいは、自作してもよい。
- ・学習内容：
 - ・ playSound メソッドの使い方を知る。

【アドバイス】

- ・ファイル形式は wav や mp3 が使える。
- ・書き方 playSound(ファイル名);
ファイル名はダブル引用符で囲むことを忘れないこと。

課題7 次の作品 No. 7 を作成せよ。

[作品 No. 7 マウスカーソルが大好きなアリ]

- ・作成プログラムの例示：(ビデオ教材の視聴)
- ・基本シナリオ： 画面上にアリが 10 匹いる。アリたちはマウスカーソルに向かって動き、カーソルの近くにきたらそこに留まる。アリたちは常にマウスカーソルのそばにいるようにカーソルが動くときそれに伴って動きまわる。
- ・学習内容：
 - ・ getMouseInfo メソッドの使い方を知る。
 - ・マウスの位置を知る方法について知る。
- ・参考コード： サンプルプログラム balloons 内の dart.java のコードを見よ。

課題8 次の作品 No. 8 を作成せよ。

[作品 No. 8 子猫のマーやはネズミのチュー太郎が苦手？]

- ・作成プログラムの例示：(ビデオ教材の視聴)
- ・基本シナリオ： 温かい春のある日、子猫のマーやはうとうとと眠りかけていた。今日はゆっくりと寝ていたい心境にある。ところが、ネズミのチュー太郎が子猫のマーやのそばにすぐ寄ってくる。マーやはゆっくり寝ていたいので、チュー太郎がすぐそばに近づいてきたら、少し離れた所に逃げ出す。
- ・学習内容： いままでの知識を応用する。

課題9 すでにつくった作品 No. 2 に以下の変更を加えよ。

[作品 No. 9 大空に漂う UFO]

- ・作成プログラムの例示：(ビデオ教材の視聴)
 - ・変更要求：
 - ・ヨットを UFO に変更する。UFO は酔歩のままよい。
 - ・背景の海を、大空に変更する。
 - ・空には雲が2つあり、右から左に流れている。
- 【アドバイス】 UFO や背景の大空は GIMP などの描画ツールを用いて作成せよ。
2つの雲の流れ方は、同じでも異なってもよい。

課題10 自分で好きなものをできる限りたくさん作ってみよう。

[作品 No. 10～]

- ・作りたいものたくさん作ること。ただし、作ったものの中には1つ以上オリジナリティのあるものが含まれていること。大がかりなものはここでは考えないこと。
- ・作品ができるごとに TA にデモをして見せて感想と OK をもらうこと。
- ・Greenfoot の Web サイトも参考にすること。
 - (ア) <http://www.greenfoot.org/scenarios/index.html>
 - (イ) Greenfoot Gallery (<http://greenfootgallery.org/>)
- ・いい作品ができたなら、どんどん upload しよう。

第3部 ゲーム作成プロジェクト(PBL 演習)

ここまでは、Greenfoot の使い方について学んだ。以下ではその知識と経験およびそこで得たスキル(技能)をもとに、自分たちだけの、東京工科大学オリジナルのゲームを作成してみよう。なお以下の課題は、実験の第2日目～第3日目にかけて行い、最終日(第3日目)はプレゼンテーションの日とする。

課題10 ゲーム Greeps を動かしてみよう！

- (1) ゲーム Greeps を自力で動かしてみよう。
- (2) ゲーム Greeps の動作を理解しよう。

—これからが本番だ！ 力を合わせてゲームプログラムを作ろう！—

・**状況設定**: 君たちは ICT のプロの卵として選ばれし者たちである。以下の指令よく読み、確実にこなしてほしい。

- ・指令1 チームをつくろう。
- ・指令2 役割分担を決めよう。
- ・指令3 自分たちだけのゲームをつくろう。
(ゲーム作成のためには、そのための作業工程がある。その説明は教員からある。)

課題1 チームを作る。(教員の指示に従うこと)

課題2 チーム内の役割分担を決める。

役割および決め方はチーム内で決める。例えば、チームリーダー、記録係、タイムキーパー(時間管理担当)、プログラミング作成担当、ドキュメント作成・管理担当、シナリオ(レベルデザイン)担当、芸術担当(音楽担当、画像担当)、ソフトウェア試験担当 など。

課題3 チーム名(Virtual Company name; VC name)を決める。

課題4 チームシートを作成する。

課題5 プロジェクトを立ち上げる。

課題6 ブレインストーミングをし、ゲームのアイデアを練る。

課題7 企画書を作成する。

課題8 作成するソフトウェアを分析し、要求定書と要件定義を作成する。

課題9 ソフトウェアの設計をする。

課題10 ソフトウェアをコーディングする。

課題11 ソフトウェアの動作検証をする。

課題 12 みんなで遊んでみる。

課題 13 プレゼン資料にまとめる。(ゲーム会社の社長に売り込みをするためのプレゼン資料を作成すること。)

なお、本課題遂行にあたっては教員から追加説明および追加資料が配布されるので、それらも十分活用すること。

Good

Luck!

<<参考文献>>

I. Greenfoot 関連のサイト

[1-1] Greenfoot の公式サイト(<http://www.greenfoot.org/>) (2009 年 4 月 2 日時点).

[1-2] Greenfoot ギャラリー (<http://greenfootgallery.org/>)

II. UML を活用するシステム開発関連

[2-1] “コンピュータシステム開発入門,” 松永・中村・亀田, オーム社(2008).

III. UML 自体に関する文献

[3-1] “UML モデリング技能認定試験 レベル(L1)問題集,” 技術評論社(2004).

[3-2] “UML モデリング L1 [第 2 版],” Shoeisya(2004).

[3-3] “UML モデリング L2 [第 2 版],” Shoeisya(2004).

[3-4] “Mode’lisation object avec UML,” Pierre-Alain Muller et Nathalie Gaertner, Eyrolles(2000).

IV. ソフトウェア開発関連の文献

[4-1] Head First ソフトウェア開発 頭とからだで覚えるソフトウェア開発の基本, Dan Pilone & Russ Miles, オライリー・ジャパン(2009).

[4-2] ずっと受けたかったソフトウェアエンジニアリングの新人研修, 大森久美子・岡崎義勝・西原琢夫, SHOEISHA(2009).

[4-3] 日本のソフトウェア産業がいつまでもダメな理由, 久手堅憲之, 技術評論社(2008).
(特に、第 3 章の 3 は一読の価値あり。“英語オンチの日本人が作るソフトウェアは、市場も国内に限定される。” 本書 P.114 より引用)

V. ソフトウェアテスト関連の文献

[5-1] ソフトウェアテストの常識, 秋本芳伸・岡田泰子, ディー・アート(2006).

(学術的ではないが、役に立つことがたくさん書かれています。)

[5-2] 演習で学ぶソフトウェアテスト 特訓 150 問, JSTQB テスト技術者認定 Foundation 対応, 正木威寛, 技術評論社(2006).

VI. ゲームプログラミング&ツール関連

[6-1] “Game Programming Gems (日本語版),” ボーンデジタル.

このシリーズは現在 Gem1 ~ Gem7 まで出ており、一冊約 1 万円ほどしますが、ゲームプログラマーになりたいければ必携の文献です。亀田も必死で読んで勉強中です!

[6-2] ゲーム者のための AI 入門, David M. Bourg & Glenn Seemann, O’Reilly(2005).

VII. 画像描画関係

- [7-1] “はじめての GIMP,” はせべれいこ, 工学社(2006).
- [7-2] <http://www.gimp.org/> (GIMP の公式サイト)(2009 年 4 月 2 日時点).
- [7-3] <http://www.libpng.org/pub/png/> (png の公式サイト) (2009 年 4 月 2 日時点).

VIII. 音楽関係

- [8-1] “ミュージックメディア入門 MIDI 検定 4 級対応,” 日本シンセサイザー・プログラマー協会(編集・著作), 音楽電子事業協会(AMEI)(2005).
- [8-2] “MIDI 検定 3 級公式ガイドブック,” 音楽電子事業協会(1998).
- [8-3] Windows サウンドプログラミング, 田辺義和, SHOEISHA(2001).

IX. より進んだ情報

- [9-1] <http://www.bluej.org/> (BlueJ の公式サイト) (2009 年 4 月 2 日時点).
- [9-2] “Objects First with Java -A Practical Introduction using BlueJ(4th Ed.),”
David J. Barnes & Michael Kölling, Prentice Hall/Pearson Education(2008).
(BlueJ の本)
- [9-3] “Game Programming in Introductory Courses With Direct State Manipulation,”
Michael Kölling and Poul Henriksen, Proceedings of ITiCSE'05, Lisbon, Portugal,
June 2005. (Greenfoot の論文)
- [9-4] “greenfoot: Combining object visualisation with interaction,” Poul Henriksen and
Michael Kölling, in Companion to the 19th annual ACM SIGPLAN conference on
Object-oriented programming systems, languages, and applications (OOPSLA),
pages 73-82, Vancouver, BC, CANADA, November 2004. (Greenfoot の論文)
- [9-5] “A Direct Interaction Tool for Object-Oriented Programming Education,” Poul
Henriksen, Master Thesis, The Maersk Mc-Kinney Moller Institute for Production
Technology, University of Southern Denmark, 2004. (Greenfoot の論文)

謝辞 本実験の着想および設計に関して熱心な討論と適切なお指導・ご助言をいただきました中村太一教授(東京工科大学コンピュータサイエンス学部)、高嶋章雄氏(現、湘北短期大学講師)およびタンジブル・ソフトウェア教育研究プロジェクトのメンバ(岩下講師・菊池講師・木下教授ら)に謝意します。また Greenfoot を開発しオープンソースとして公開した University of Kent (英国)と Deakin University (オーストラリア)に謝意を表す。また、実験教材開発に協力をしてくれた宮田孝二君(東京工科大学コンピュータサイエンス学部 OB)と山崎正貴君(東京工科大学コンピュータサイエンス学部 OB)にも謝意を表す。

付 録

サイエンスキャンプでは使いませんでしたが、本格的に行う場合にはこのようなドキュメント作成も行います。参考までに添えておきます。

ゲーム作品製造メモ(第 版)

提出日：平成 26 年 8 月 日

項 目	記 述
作品 ID 番号 作品名	作品番号 No. _____
ゲームコンセプト	(ゲームのジャンル、基本シナリオ、ユーザイメージなど)
外部設計	
内部設計	
コメント (もしあれば)	

トラブルシューティングシート

症状・トラブル内容	実施した対処内容	今後の対応
(例)プログラムをコンパイルしたら XXX というエラーメッセージが出た。	プログラファイル p.java の 25 行目に変数宣言を追加した。	使われる変数が分かりやすいように、整理して宣言する。

チームシート(第 版)

提出日：平成 26 年 8 月 日

チーム名
チーム構成(氏名、役割) ・ ・ ・ ・ ・
チームの行動指針
チームの達成目標
活動に関する注意点(授業時間外の打合せ場所・時間など)
その他

ソフトウェアテスト用紙

提出日：平成 年 月 日

No.	試験対象要件項目	試験方法	試験結果	事後対応	備考(実施日等)
	(例)アクタはクリックされると爆発する。	実際にクリックする。	2回目以降反応せず。	ソースコードを修正。	修正後要再度確認すること。

チームシート

提出日平成 年 月 日

気づきシート

平成 年 月 日

気付いたこと	備考

プロジェクト憲章作成シート 提出日 平成 年 月 日

- ・目的
- ・方針
- ・要求仕様決定

ゲームコンセプトの決定

いつどこで誰がどんな風にこのゲームを楽しむ。

ゲームの種類は？

ゲームの仕様の決定

仕様書の作成

ユースケース分析

ユースケース図の作成

外部設計

外部仕様の決定

外部仕様書の作成

内部設計

内部仕様の決定

内部仕様書の作成

システム設計

外部設計書

作成： 年 月 日（ ）

提出： 年 月 日（ ）第 版

(ここに画面設計情報(画面遷移設計図)などを書いてください。
また、必要ならば紙を追加してください。)

友情シート

氏名： _____平成 年 月 日

(プロジェクト貢献に対する感謝を具体的に記述する。)