

# 2006 春学期 大岩研究プロジェクト 2 さうんど おんりい班 最終報告書

PM 三菱スペース・ソフトウェア 中川誠

環境情報学部 4 年 橋山牧人

環境情報学部 3 年 佐藤俊之

環境情報学部 3 年 堀正人

総合政策学部 2 年 渡辺士郎

## 1. プロジェクトの概要

さうんど おんりいプロジェクトは、(1) 学習プロジェクト、(2) ゲームプロジェクトの2つのサブプロジェクトから構成される。それぞれのプロジェクトごとに目標、成果物スコープが定義されている。

・参考資料 (添付資料 5.1: プロジェクト定義書)

### 1.1. 背景

株式会社ユードー社の南雲氏は、視覚にハンディキャップを有する人と健常者との両方が楽しむ時間を共有することができるゲームを開発したいと考えている。このプロジェクトでは、サラウンド音響を用いた映像の無いゲームの開発をして欲しいという南雲氏の依頼をもとに、「映像のないゲーム」を開発していく。

### 1.2. 目的

南雲氏から依頼された「映像のないゲーム」を開発する。

### 1.3. 目標

プロジェクトの目標について説明する。

参考資料 (添付資料 5.2: スコープ記述書)

#### 1.3.1. 学習プロジェクト目標

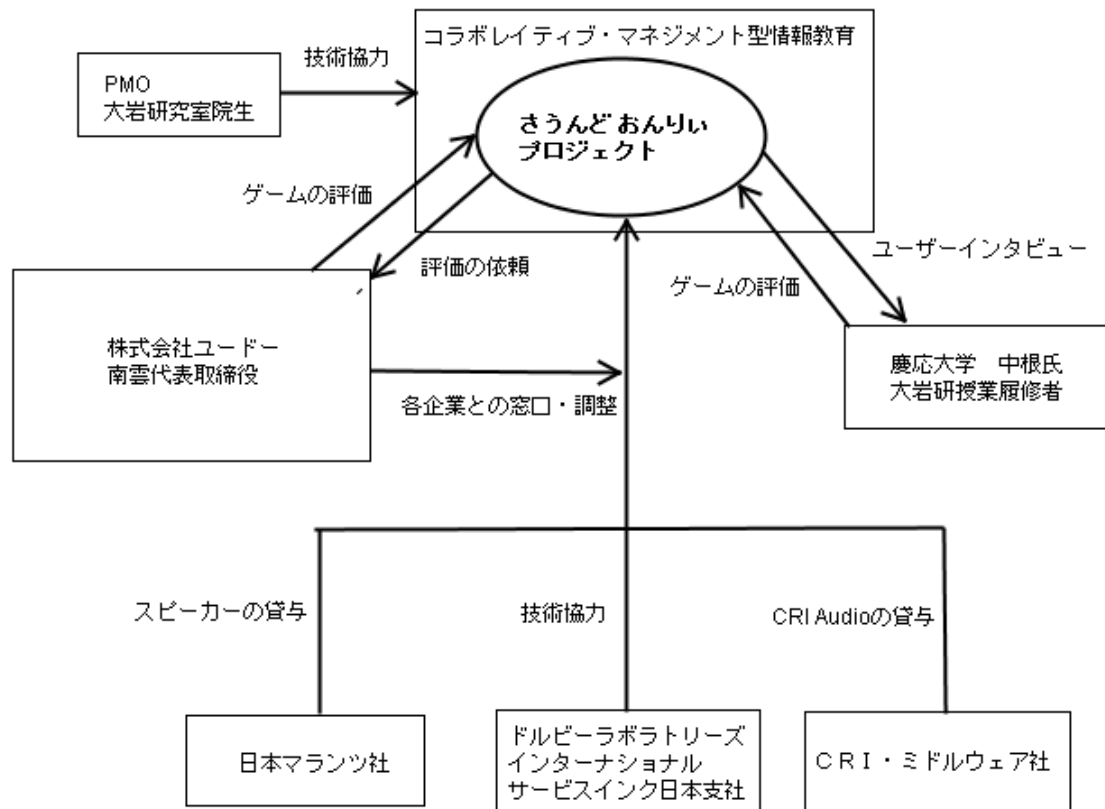
プロジェクト活動について学習するため、本プロジェクトでは、適切な計画を立案し、適切なプロセスでプロジェクト活動を実施することを目標とする。また、学生、PM がプロジェクト活動を通して、自己の目標を達成し、“成長”することを目標とする。

#### 1.3.2. 学習プロジェクト目標

視覚にハンディキャップを有する人、健常者がともに楽しい時間を共有することができる、「映像のないゲーム」を開発するための課題を解決すること目標とする。

#### 1.4. 体制

さうんど おんりいは大岩研究室の研究プロジェクトにおけるプロジェクトの 1 つである。プロジェクト・マネジメント・オフィス (PMO) や南雲氏を通じた各社から技術協力や機材の貸し出しをしてもらう。また、慶応大学内の視覚にハンディキャップを有する方へユーザーインタビューを行い、ゲームの評価をして頂く。



さうんど おんりいプロジェクトの体制

#### 1.5. 開発環境・設備

実装環境として、Visual Studio2003.NET を利用して、C++で行う。また、音をサラウンドで再生するときに、株式会社 CRI・ミドルウェア社の「CRI Audio」を利用する。さらに、Subversion によってプロジェクト内の UML やソースコード、ミーティングログを始めとするすべての資料をプロジェクトメンバー内で共有する。

音響設備として、開発用に 5.1ch 対応のヘッドホンを使用する。

## 2. 企画詳細

### 2.1. 企画決定までの変遷

#### 2.1.1. 企画構想

##### 2.1.1.1. 企画構想

今回、南雲代表取締役役に依頼されたのは「映像のないゲーム」の開発と、非常に漠然としたイメージのものであった。よって、企画の決定を行うのも今プロジェクトの一環であった。

企画の決定は以下の手順で行われた。

##### 2.1.1.2. 第1回企画考案会

まず、各プロジェクトメンバーが企画を持ち寄り、考案会という形で、プロジェクトミーティング内で各自5分程度の発表を行った。発表後、各案の良かった点を挙げていき、それぞれの良い点を抽出してそれを元に企画を決定させようとした。

この際、「何が出来そうか」という考えは排除して、単純にどういうものが作りたいか、どういうゲームなら遊びたいかという考えに基づいて案を考えることにした。初めからアイデアが制限されていては良い案を見落としてしまう可能性があるからだ。

このミーティングでは、宝探しゲームや、動き回る対象を追跡するゲーム、銃型コントローラーを用いたガンシューティングゲームと言った様々な案が出た。この内ガンシューティングゲームはハード面を実現するのが難しそうという理由で断念することになった。

##### 2.1.1.3. 第2回企画考案会

続いて、第1回企画考案会に出た「良かった点」を元に、再び各自で案を練り、第2回考案会にて発表を行うこととなった。考案会では、既存の宝探しゲームや追跡ゲーム、複数人での鬼ごっこゲーム等が立案された。

##### 2.1.1.4. 企画決定

第2回企画考案会で出た案を元に議論した結果、対戦型ゲームの方が健常者と視覚にハンディキャップを有する人が同時に楽しむことができるという考えもあり、鬼ごっこゲームに他の案の良いところを搭載させたものが企画として決定した。

また、対戦は不特定多数の人間が同時に遊べる環境ということでネット対戦で行おうということになった。

企画の詳細は **2.2 スコープ定義前の全体企画**にて記す。

#### 2.1.2. スコープ定義

##### 2.1.2.1. 目的

企画の構想と平行して、スコープの構想も行われた。今回の製作期間3ヶ月と短いため、実装する範囲を決定する必要があるからである。

#### 2.1.2.2. 経緯

第2回企画考案会において、各自、自分の案が今学期中にどれだけの範囲が実装できそうかということを発表した。先述の通り、話し合いの結果企画が決定したが、その後もスコープの定義についての議論を行った。

#### 2.1.2.3. スコープ決定

操作可能なのは鬼ごっこの「追う側」のみという点はすぐに決定したのだが、「逃げる側」を固定とするか、AIを搭載して逃げるようにするかという議論が行われ、最終的にとりあえずは根幹部分を作成しようということで逃げる側は固定することにした。また、ゲームを彩るトラップやボーダーラインといったオブジェクトは段階を踏んで実装しようということにし、スコープに優先順位を付け、作成の順番を付けた。

決定したスコープの詳細は **2.3 スコープ定義後の企画** にて記す。

#### 2.1.3. 学んだ点・苦労した点

##### 2.1.3.1. 企画の決定

今プロジェクトのモットーとして、全員が納得して作業に取り掛かりたいというものがある。自分の本意ではないものを作成しようとしてもモチベーションは上がらないし、ストレスも溜まってしまうからだ。なので、企画が決定するまで全員が納得するまで議論を行った。

当然、今プロジェクトに参加したメンバーは何らかの「やりたいこと」を持っていた訳なので、その「やりたいこと」ができなくなってしまうこともある。やはり、全員が納得する企画が決定するまでには時間がかかった。今回のことで、誰もが気持ちよく作業を行うことの重要性、及びその難しさを学んだ。

##### 2.1.3.2. スコープの決定

スコープを決定するのにあたり難しかったことは、殆どのメンバーが VisualC++ で作業をしたことが無く、またサラウンド音響に触れたことが無かった、ということが挙げられる。これらの原因で、自分たちがどこまで実装できるか分からないという不確定要素が発生した。スコープを決定するに当たって、まず何を達成すれば良いとするかということよりも、自分たちがどこまで達成することができるか、ということが焦点になって話し合いが行われた。

この決定したスコープ後の企画では、自分たちがまず達成すべきであろう範囲と、最終的に達成させたい範囲の二つに範囲を分けることにした。このようにスコープに段階をつけることにより、まず何からすればいいのかという明確なゴールラインを作成することができた。

### 2.2. スコープ定義前の全体企画

#### 2.2.1. ゲームの概要

複数対複数で対戦可能な「鬼ごっこゲーム」

#### 2.2.2. プレイヤーの役割

このゲームには「追う側」と「逃げる側」の2つの役割があり、プレイヤーはどちらかを選択してプレイすることになる。

追う側のプレイヤーの役割は時間内に逃げる側のプレイヤーを捕まえることで、逃げる側のプレイヤーの役割は時間制限まで追う側のプレイヤーから逃げ切ることや、追う側のプレイヤーをトラップにはめて撃退するということである。

複数対複数ということで、追う側のプレイヤー及び逃げる側のプレイヤーが複数存在することもある。

#### 2.2.3. ゲームの特徴

サラウンド音響を利用し、音による空間把握ができる。つまり、相手が発信する音で、相手の現在地がわかる。また、フィールドに点在するトラップも音を発し、周囲に何があるのかということを常に見張ることが可能である。

また、多数の人間が1つのフィールドで同時に遊ぶことになるので、ネット対戦という形で対戦を行う。

#### 2.2.4. ゲームを盛り上げる要素

一部に記述したが、トラップという要素がある。これは、逃げる側がトラップを置き、追う側がその設置したトラップに接触したときにある効果をもたらすというものである。この効果の案として、場所を移動させたり、動きを止めたり、即ゲームオーバーにするという案がある。

また、フィールドは平面であり、一定のマスに仕切られている。これは、音のみでは地面の高低や壁の表現をすることが難しいためである。また、果てしない平面では面白みが無いため、ボーダーラインという要素を追加する。一定の範囲外に出たプレイヤーはその場でゲームオーバーになると言うものである。また、この他に時間が経過すると進入禁止になるエリア（ボーダーラインを狭める）といった要素が考案された。

### 2.3. スコープ定義後の企画

上記の企画を元にスコープ定義を行った。今回、上記の企画の範囲内で実装する範囲は以下である。

#### 2.3.1. ゲームの概要

逃げる側の場所を固定し、プレイヤーは追う側のみとする。

プレイヤーは音だけを頼りに、フィールド上のトラップを避けつつ制限時間以内に目標物に到達するという一人用のゲームとする。（スイカ割りのようなもの）

#### 2.3.2. プレイヤーの役割

プレイヤーは目的地点へ辿り着くことを目標とする。

#### 2.3.3. ゲームを盛り上げる要素

トラップ及びボーダーラインは実装する。しかし、トラップは逃げる側が設置するのではなく、初めから置いてあるようにした。

#### 2.3.4. ストーリー

上記の企画を踏まえ、以下のようなストーリーも作成した。

「テロリストが地下駐車場に爆弾を設置したので、それを解除しに行かなきゃいけない。またテロリストがその電気を切ったので、暗闇を模索しながら行く。」

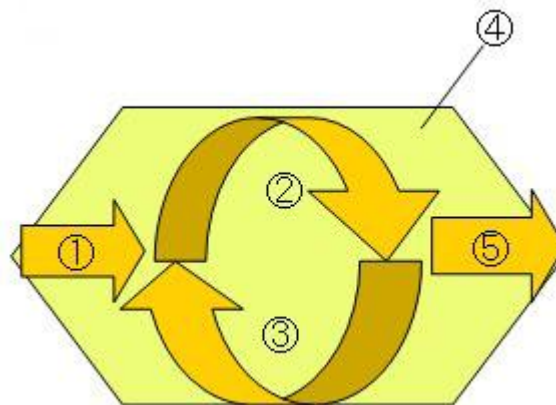
以上のプロセスを経て、企画・立案フェーズは終了した。

### 3. プロジェクトの進行内容

#### 3.1. プロジェクトの流れ

さうんど おんりいプロジェクトは、プロジェクトのマネジメント手法であり、世界中で広く認知されている知識体系である PMBOK に従って進めていく。メンバーの半分はプロジェクト未経験者で、残り半分は経験者である。PMBOK に従ってプロジェクトを進めることで、未経験者はプロジェクトがどのようなものであるのかということを体系的に学ぶことができる。また経験者は、過去に経験したプロジェクトとの違いを比較することで、プロジェクトの全体像が見え、プロジェクトの管理方法を学ぶことができる。以上が本プロジェクトを PMBOK に従って進めていく理由である。

PMBOK では以下の 5 つのフェーズがある。



図：3.1-1 PMBOK の 5 つのフェーズ

立ち上げフェーズ

計画フェーズ

実行フェーズ

監視・コントロールフェーズ

終結フェーズ

#### ・ 参考資料

- 添付資料 4.1 目標管理シート
- 添付資料 5.1：プロジェクト定義書
- 添付資料 5.7 WBS・スケジュール



## 3.2. 反復作業

### 3.2.1. 全体設計

#### 3.2.1.1. 方針

はじめに、最初の成果物となるスコープ第 1 段階の設計に取り掛かるのではなく、全体の設計を行うことにした。理由として、将来的な拡張性が挙げられる。一番小さいものをベースにすると、それを拡大する際に仕様変更等のリスクが生じることがある。それを防止するため、全体の設計から、それをベースに実際に実装する箇所だけを抜き出すという形式で実装を行うことにした。

#### 3.2.1.2. 作業概要

まず要求分析を行い、その結果を基に仕様を決定した。その仕様を基に更に設計を行い、反復第 1 回目の範囲の設計へと移行した。

#### 3.2.1.3. 作業詳細

##### 3.2.1.1.1. 要求分析

###### 3.2.1.1.1.1. 目的

どのような機能があればシステムを満足することができるのかということ进行分析した。

###### 3.2.1.1.1.2. 経緯

まずメンバー全員が各々考えたユースケース図を作成し、それを基に討論を行った。

この議論において、各人がユースケース図のみを作成し、それに対応するユースケース記述を作成していない、という点が明らかになった。確かに、ユースケース図だけで要求の理解に導くのは不可能であるという解釈を全員が持ち、ユースケース図を作成するときはユースケース記述も忘れないようにしよう、という意識が全体に浸透した。

またこの時、「ゲーム」というシステムの特異性(目的が曖昧)から、通常のユースケース図では表現が難しいという話になった。そこで、我々は 2 つの手段を用いて要求の分析を行うこととした。1 つは、ゲームに向いていると言われる組み込み UML について調査し、それにおけるユースケースの記述方法に基づいてユースケース図及びユースケース文書を記述する、という手段である。もう一方は、プレイヤーにゲームの説明や、ゲーム内で出来ることを著したマニュアルを作成する、という手段である。それぞれをメンバーが作成し、討論を基に双方の作成を行った。

##### 3.2.1.1.1.3. 成果物

###### 3.2.1.1.1.3.1. ユースケース

まず、作成したユースケース図について説明したい。組み込み UML においては、システム内に存在するオブジェクトもアクターの一部となり、それぞれのユースケースも記述する。通常と異なっている点として、ユーザが何を

したいかを分析するのではなく、どのような機能があればシステム自体が満足できるのかということ进行分析するという点が挙げられる。

なお今回作成したユースケース図、及びユースケース記述は 5.4.1.1 項の資料 3.2.1.3.1.3.1-1 と 3.2.1.3.1.3.1-2 に添付した。

#### 3.2.1.1.1.3.2. マニュアル

続いて、マニュアルの記述を行った。これもユースケース図と同様に複数のメンバーが作成し、検討を基に決定した。プレイヤーにとって理解がし易い、逆に言えばプレイヤーが理解するのに必要な範囲をマニュアルから分析するという狙いがある。なお、マニュアルは 5.5 項に添付した。

#### 3.2.1.1.1.3.3. 苦労点や発見点

ここで議論しているうちに、ユースケースとマニュアルを比較しての発見点が見つけられた。まず、ユースケースはどちらかというと製作者（システム寄り）の視点で書かれたものであるのに対し、マニュアルはプレイする側の立場に立って書かれたものである。プレイヤーが何かをするには、何をできるようにすべきか、といったゲームに必須となる動作の要求を分析するにはマニュアルは向いており、逆にシステム内でこういった作業が行われればプレイヤーがやって欲しいことを解決できるか、といった点进行分析するにはユースケースが向いている、という結論に達した。

この結果、実装にはユースケースが向いているかもしれないが、細かな仕様の確認をするためにマニュアルを用いるのも有効だと言う結論に達した。

### 3.2.1.1.2. 仕様

#### 3.2.1.1.2.1. 目的

どのような機能があればユースケースを要求を満足させることができるかをリストアップする。

#### 3.2.1.1.2.2. 経緯

先述のユースケース及びマニュアルに記した項目を全て満足させるために必要な機能を全て挙げ、1 つに纏めた「機能仕様一覧」を作成した。

#### 3.2.1.1.2.3. 成果物

作成した機能仕様一覧は 5.4.1.1 項にある表 3.2.2.3.4.3. -1 である。

#### 3.2.1.1.2.4. 苦労点や発見点

2 つの要求分析に用いたツールの整合性を取りつつ、必要だと思われる全ての機能及びその条件を記述するのに苦労し、当初は明らかに仕様の数が少なかった。最終的には、作成の結果、全員が何をつくるのかという点を共有することが出来た。

### 3.2.1.1.3. 設計

#### 3.2.1.1.3.1. 目的

決定した仕様を基に、全体の設計を行う。

#### 3.2.1.1.3.2. 経緯

設計も、これまでのプロセスと同様に全員の作成したモデルを基に討論した。その際、論点となったのは各種トラップを「トラップ」という抽象クラスのサブクラスとするかということ、及びプレイヤーと時限爆弾を「キャラクター」という抽象クラスのサブクラスにするか、と言った点である。これらのことは、キャラクタークラスを実装するメリットが現時点では見当たらないといった点から、とりあえずのクラス図を作成するという形になった。理由として、実際に実装をしてみないと分からないという点が考えられる。反復第2回の冒頭に再び全体設計を行うのだが、これらの点はその時に改めて議論され、最終的な結論へと達する。

成果物として、クラス図・オブジェクト図・シーケンス図を作成した。手順としてはまずオブジェクト図を作成し、ゲーム内にどのようなオブジェクトが登場するかを考えた。続いてクラス図を作成し、要求を実現するためにはどのようなクラスが必要となるかをまとめた。

最後にシーケンス図に全体的な処理の流れや各プロセスにおける処理の段階を記し、要求を実現するための手順を纏めた。当初はシーケンス図は1つだけであったが、そのままでは処理の流れが膨大であり、また条件分岐も含めて全て一枚にまとめてしまったので、非常に読みにくく、処理の流れもつかみづらいと言うレビューが入った。そこで、複数の図に分けて書いたという経緯がある。

#### 3.2.1.1.3.3. 成果物

作成したオブジェクト図、クラス図、シーケンス図は 5.4.1.1. 項の図 3.2.1.3.3.3-1 から 3.2.1.3.3.3-6 を参照のこと。

#### 3.2.1.1.3.4. 苦労点や発見点

上記のように、UML に詳しいメンバーとそうでないメンバーがあり、各々が図を作成し、それについて討論するだけでも、考え方などを参考にすることができ、勉強になった。

### 3.2.2. 反復 1 回目

#### 3.2.2.1. 方針

反復第一回目の方針は、まずゲームの根幹部分であるスコープ第 1 段階を作成しようというものである。特に、このゲームの特徴的な部分でもある、音響の出力によってキャラクターと対象物の相対的な距離と方向の把握を可能とする機能の実装に注力することとなった。

#### 3.2.2.2. 作業概要

まず要求分析を行い、その結果を基に仕様を決定した。その仕様を基に更に設計を行い、反復第 1 回目の範囲の設計へと移行した。設計が終了してから実装に取り掛かり、試験を行ってスコープ第 1 段階は終了した。

#### 3.2.2.3. 作業詳細

##### 3.2.2.3.1. 要件定義

###### 3.2.2.3.1.1. 目的

反復第 1 回目において、どこまでの機能を実装するかを決定する。

###### 3.2.2.3.1.2. 経緯

反復第 1 回目がどのような仕様を持つかは全員の話し合いを基に決定された。

###### 3.2.2.3.1.3. 成果物

仕様を絞りこんだ結果、全体仕様と以下の点が異なった。

- ・ゲーム内オブジェクト：プレイヤー、時限爆弾（目的地）
- ・機能は以下の点を達成することを目的とする

- (1) 音響出力：時限爆弾（目的地）が音を発し、位置がわかる
- (2) キャラクターの移動：プレイヤーの入力により、キャラクターが移動する
- (3) 描画：デモ用、デバッグ用に、キャラクターと目的地の存在地点が画される

要件を達成する目標として、ユーザーがプレイヤーを操り、音のみでプレイヤーを時限爆弾のある場所へと到達できることが挙げられる。具体的には、音の出力によって時限爆弾のある場所が把握できることである。

当初は、目標地点に到達したらゲームクリアにするといった案もあったが、音響出力の手段であるミドルウェアの使用法や使用具合が不透明であったため、まずは根幹部分を作成しよう、ということになった。

###### 3.2.2.3.1.4. 苦労点・発見点

どこまでを最初に作ると楽か、どこまでなら決められた期限までに実

装を完了できるか、といった点を基に話し合ったのだが、なかなか纏まらなかった。焦点となったのは当たり判定、及びそれに付随するクリア判定やゲームオーバーの判定であるが、実装しなければゲームとして成り立たないのではないかという意見も出た。しかし、とりあえずは動くものを作成することを目標にしよう、ということで上記のような仕様が出来上がった。

自分たちがどこまでやれるのかわからない場合、要件1つを決めるのにも苦戦するのだという勉強になった。

#### 3.2.2.3.2. 要求分析

##### 3.2.2.3.2.1. 目的

要件定義の結果を受け、要件を満足する機能の洗い出しを実施した。

##### 3.2.2.3.2.2. 経緯

このような要件を絞ったバージョンでは、システムが行える事は当然のことながら全体設計で分析した物とは異なる。その変更点をユースケース図に反映させた。

##### 3.2.2.3.2.3. 成果物

ユースケース図を作成した(5.4.1.2項の図3.2.2.3.2.3-1)。なお、ユースケース記述は全体設計のものと同一であるので、そちらを参照していただきたい。

##### 3.2.2.3.2.4. 苦労点・発見点

特になかった。

#### 3.2.2.3.3. 設計

##### 3.2.2.3.3.1. 目的

スコープ第1段階の実装のため、詳細設計を行った。

##### 3.2.2.3.3.2. 経緯

上記の要求分析の決定を受け、設計を行った。設計の成果物として、全体分析時に作成したクラス図・オブジェクト図・シーケンス図の仕様を絞ったバージョンを作成した。

これらも、全体設計におけるモデルに疑問点が生じる度に変更を加えられた。

##### 3.2.2.3.3.3. 成果物

オブジェクト図、クラス図、シーケンス図を作成した。5.4.1.2項の図3.2.2.3.3.3-1から3.2.2.3.3.3-5を参照のこと。

##### 3.2.2.3.3.4. 苦労点・発見点

実装の途中で疑問点が加わるたびに修正を行った。その度に全ての図

を変更する必要があったということ、またちゃんと図の間の整合性が取れてなければならないといった点で苦労をした。

#### 3.2.2.3.4. システム仕様決定

##### 3.2.2.3.4.1. 目的

スコープ第 1 段階に存在する機能の仕様を挙げ、満足させるべき項目のリストアップを行う。

##### 3.2.2.3.4.2. 経緯

上記の設計の結果、明確化されたシステム仕様の一覧を作成した。全体の要求仕様と比較すると、今期は実装しない部分や、仕様を変更して実装する部分が出てくる。これらの差分を色分けすることにより、今回の開発範囲を把握できるようにした。

##### 3.2.2.3.4.3. 成果物

仕様一覧は 5.4.1.2 項の表 3.2.2.3.4.3 -1 として添付した。

##### 3.2.2.3.4.4. 苦労点・発見点

特に無かった。

#### 3.2.2.3.5. 実装

##### 3.2.2.3.5.1. 目的

実装を行い、スコープ第 1 段階の成果物を作成する。

##### 3.2.2.3.5.2. 経緯

上記の成果物を以て、反復第 1 回目の設計を終了とした。ただ、実装の途中にも細かな仕様の変更や設計の変更は行われた。数多くのレビューを経た、最終的なモデルが上記のモデルであることを明記しておく。

設計の変更が行われた理由として、実際に実装して見なければ分からなかった点、特にミドルウェア付近の扱いがある。実装するにあたって疑問点が出る毎に設計のレビューを行い、その後に実装を再開するという形式によって実装は行われた。

##### 3.2.2.3.5.3. 成果物

映像ないゲーム（第 1 版）

##### 3.2.2.3.5.4. 苦労点・発見点

実装は、1 人のエキスパートを中心に進められることになった。しかし、全てを 1 人に任せると何の学習にもならないので、全員に作業を分担させる必要があったのだが、どこまでが各自で作業することが可能なレベルであるかが難しく、あまり作業の分担ができなかった。

#### 3.2.2.3.6. 試験

##### 3.2.2.3.6.1. 目的

実装による成果物が、要件を満たすものであるか確認する。

##### 3.2.2.3.6.2. 経緯

実装の終了後、試験を行った。試験は、まず試験手順書を作成し、各項目に対応する箇所を埋めていく、という形式で行われた。

##### 3.2.2.3.6.3. 成果物

試験手順書・成績書試験手順書は 5.4.1.2 項に表 3.2.2.3.6.3-1 に添付した。

##### 3.2.2.3.6.4. 苦労点・発見点

今回はシンプルな機能しか持ち合わせていなかったため大きな問題にはならなかったが、試験項目があまり詳細まで行き渡ってなかったため、変更を加えた後もまだ細分化して評価できる項目があった。

#### 3.2.3. 視覚にハンディキャップを有する人に対するインタビュー

##### 3.2.3.1. 目的

視覚にハンディキャップを有する方に現状の成果物をプレイしてもらってその内容を評価してもらう。

##### 3.2.3.2. 経緯

今回のインタビューは、以前より協力をお願いしていた慶應義塾大学大学院、政策・メディア研究科の特別研究助手である中根雅文氏にお願いした。実際のインタビューは 2006 年 6 月 28 日の 13 時 30 分頃から慶應大学 SFC のイオタの 308 で行われた。

インタビューの内容は大きくわけて以下の 3 つだった。

- (1) 現状のゲームをプレイしてもらい、評価してもらう
- (2) ゲームをより良いものにするにはどうすれば良いかを乞う
- (3) 視覚にハンディキャップを持つ人とコンピューターゲームの関係が一般的にどういうモノかを聞く

##### 3.2.3.3. 成果物

以上のようなインタビューを経て、我々が反復第 2 回目に反映することになった仕様は以下の 2 点である。

- (1) ステージ単位でのハイスコアを記録・提示する
- (2) ステージ開始前にチュートリアルを設置する

#### 3.2.3.4. 苦労点や発見点

結果として、あまり新しい情報が得られることは無かった。主に操作方法やゲーム性に関する質問を投げたものの、ゲームそのものがまだ単純だったからか、ゲーム一般にも言えるような回答しか得ることができなかった。

例えば「ゲームをより面白くするにはどうすれば良いか」という質問に対し、中根氏は「スコアなど、人と競えるものがあれば良い」と答えたが、それは別に今回のゲームに限った話ではなく、ゲームでは一般的に使われている手法である。また、音の種類や定位を認識するためにチュートリアルがあった方が良い、という意見も出たが、それもやはり事前にチーム内で考慮されていたことで、特別真新しい話ではなかった。

ただ、1つだけ面白い話を聞くことができた。その内容は、視覚にハンディキャップを有する人でも一般的なコンピューターゲーム（の一部）はプレイすることがある、ということだ。詳しく話を聞くと、やはり視覚情報が得られないため、（敵の動きなどで）ランダムな要素が絡んでくるゲームは難しいが、パターン化できるゲームだったら充分プレイ可能である、とのことだった。例としては横スクロール型のアクションゲームなど、敵の出現位置や動きがパターン化されているゲームなどが挙げられた。念頭に置いておくこととしては貴重な意見だった。

全体としては、「これだけはこの人に問いたい」というような事項がなかったため、効果的とは言えないインタビューとなってしまった。実際に追加された仕様が一般的なゲームで使われている手法であることを考えると、やはりインタビューを行うには早すぎたのではないか、あるいはそもそもインタビューは必要でなかったのではないか、と感じた。



### 3.2.4. 反復2回目

#### 3.2.4.1. 方針

第2段階の反復作業は、第1段階の成果物があるので、それを基に拡張していく事にした。そして第2段階の反復作業の設計・実装は第1段階の成果物を作っていく際に気付いたことや第1段階の成果物を視覚にハンディキャップを有する人に評価をしていただいた結果を盛り込んだ内容とした。これは当たり前のことだが、第1段階と第2段階とはっきり作業をわけているのは実装以前にわからなかった問題や第1段階の実装をしてみて生じた問題を第2段階で解決するためである。これは、今回の映像のないゲームという前例のないものを作るにあたり、PMが最初から問題が起こるだろうということを先のことを想定した方法であり、ベストではないかもしれないがベターな方法であったといえる。

#### 3.2.4.2. 作業概要

まず第1段階の成果物が出来たことによる追加された要求分析を行い、その結果を基に仕様を決定する。その仕様を基にスコープ第2段階の範囲の設計へと移行する。設計が全部終了すると実装に取り掛かり、試験を行ってそれを満足させればスコープ第2段階の実装は終了となる。としたいところであったが、実際は日程の関係や「面白い」ゲームを作ることを優先するために設計が終わった部分から実装に移ることとなった。

#### 3.2.4.3. 作業詳細

##### 3.2.4.3.1. 要件定義

###### 3.2.4.3.1.1. 目的

スコープの第2段階において、どこまでの機能を実装するかを決めた。

###### 3.2.4.3.1.2. 経緯

スコープ第2段階がどのような仕様を持つかは、ユーザーインタビューの結果や第1段階の実装の結果判明した問題点について話し合いをし、その結果決定された。具体的には目的地から遠くにいと音の聞こえ方の変化が小さいためわかりにくいことや、キー入力の改善、トラップ、ボーダーラインの追加などである。

###### 3.2.4.3.1.3. 成果物

- ・ゲーム内オブジェクトとはキャラクター、時限爆弾(目的地)、ボーダーライン、トラップをさす。
- ・機能は以下の点を達成することを目的とする。

- (1) 音響出力：時限爆弾(目的地)、ボーダーライン、トラップがそれぞれ音を発し、位置がわかる。
- (2) キャラクターの移動：プレイヤーの入力により、キャラクターが移動する(第1段階と同じであるが、第1段階で発覚したキーの入力の

改善やキャラクターの移動量の調整もここに含まれると言える)。

- (3) 描画：デモ用、デバッグ用に、すべてのゲーム内オブジェクトの存在地点がフィールドに描画される。
- (4) 状態：第 1 段階では起動した時点でゲーム状態であったが、タイトル状態、ステージセレクト状態、チュートリアル状態、ゲーム状態、ヘルプ状態、ゲームクリア状態、ゲームオーバー状態が描画(デモ・デバッグ用に)かつ音声で知らされる。ここでは状態と書いたが、これは他のシステムでいう画面という言葉で説明されるものである。以後もわかりやすさを優先して「～画面」と記述する。
- (5) ステージ読み込み：外部でつくった txt 形式ファイルを読み込み、様々なゲーム内ステージを作成できるようにする。
- (6) スコア、ハイスコア：ステージ読み込み同様、ゲームを面白くするためにクリア時間を利用したスコアを作る。

要件を達成する目標として、プレイヤーがキャラクターを操り、タイトル画面からゲームクリア or ゲームオーバー画面まで音のみで遷移できること。これが最低条件で、加えて具体的ではないが「面白い」ゲームに仕上げる。この二点を目標とした。

#### 3.2.4.3.1.4. 苦労点・発見点

最終的にどこまで出来るか、といった点を基に話し合ったのだが、第 1 段階同様なかなか纏まらなかった。第 1 段階の成果物を受けて色々修正すべき点などはあっさり決まったが、今回の焦点となったのは「面白い」ゲームとはそもそも何か、という人によって感覚が違うためそれを文章や数値で表すのが難しい問題であった。

もし「面白い」ということを要件に盛り込むことが出来るのなら、世の中の存在するゲームは殆ど面白くなるはずであるが、現実にはそんな事はない。このゲームを私たちは「音だけ」という点を「斬新」で面白いと評価したが、それはあくまで私達の主観であるので、それ以外にも何かゲームらしい要素が必要だろうということになった。

ある班員は基本的にどんなゲームでもリスクとリターンがあり、例えば競馬でも勝ちそうな馬は儲けが低いが勝てそうにない馬は儲けが高いという、低リスクには低リターンしかかえってこないし高リスクには高リターンがかえってくるものであると主張した。虎穴にいらずんば虎児を得ずである。

よって、このゲームでそのリスクとリターンを天秤にかけるようなことをどのように仕様に盛り込むかという議論になった。

Q. このゲームにおけるリターン(目標)とは何か

A. ゲームクリアもしくはゲームクリア時のハイスコア

Q. このゲームにおけるリスク(目標が達成できないこと)とは何か

A. ボーダーラインやトラップにひっかかったり時間切れで死ぬこと

これらの考えから接触すれば制限時間が増加するアイテムの追加を決めた。これは、残りの制限時間がそのままスコアとなる仕様としたので、そのままクリアするかそれとも危険を冒してアイテムをとりに行ってスコアの上昇を狙うか、というプレイヤーにリスクとリターンを迫ることが出来る方法であった。この仕様にたどりつくまでにかなりの議論を要した。

#### 3.2.4.3.2. 要求分析

##### 3.2.4.3.2.1. 目的

第1段階の成果物の完成をうけて、実装に必要な機能を洗い出し直した。

##### 3.2.4.3.2.2. 経緯

第2段階の反復作業における成果物は、第1段階の結果などをうけて当然のことながら全体設計で分析した物とは異なってきた。その変更点をユースケース図に反映させるということであった。

##### 3.2.4.3.2.3. 成果物

第2段階目のユースケース図は5.4.2.1.項の資料3.2.4.3.2.3.-1に添付した。

##### 3.2.4.3.2.4. 苦労点・発見点

ユースケース図にどこまで記述するかという点についてプロジェクト員だけでは判断できなかったので大岩研究室の松澤氏に意見を求めるなど、班員だけでは出来ないことがあったが、全員が同時に集まるのが難しいなど日程に調整などに苦労した。

#### 3.2.4.3.3. 設計

##### 3.2.4.3.3.1. 目的

スコープ第2段階の実装のため、詳細設計を行った。

##### 3.2.4.3.3.2. 経緯

上記の要求分析の決定を受け、設計を行った。設計の成果物として、第1段階の反復作業時に作成したクラス図の修正を行った。また、ゲームを実装するのにわかりやすい状態遷移図の作成も行った。これらは、できあがるたびに実装にすぐ移るという形であった。よってじっくり修正をかける時間がなくモデルに疑問点が生じる度に適時変更が加えられた。

##### 3.2.4.3.3.3. 成果物

第2段階のクラス図は5.4.2.2.項の3.2.4.3.3.2.3.-1に、状態遷移図は3.2.4.3.3.2.3.-2に添付した。

#### 3.2.4.3.3.4. 苦労点・発見点

実装の途中に疑問点が加わるたびに修正を行った。特にクラス図でのトラップオブジェクトの上に抽象クラスを作るかどうかという点で議論がわかれた。現在の仕様ではトラップが一つ(KillTrap)しかないため、わざわざ抽象化する必要性を感じられないというメイン実装者の意見があったが、最終的には今後の拡張性を考えてトラップを抽象化し、第2段階の範囲ではボーダーラインとトラップとアイテムはキャラクターと接触して何らかの反応を起こすという同様の機能であるので、抽象化されたトラップクラスに属する形となった。

#### 3.2.4.3.4. システム仕様決定

##### 3.2.4.3.4.1. 目的

スコープ第2段階に存在するシステム仕様を挙げ、満足させるべき項目のリストアップを行う。

##### 3.2.4.3.4.2. 経緯

上記の設計の結果、明確化されたシステム仕様を決定した。

##### 3.2.4.3.4.3. 成果物

仕様一覧は5.4.2.3.項の3.2.4.3.4.3.-1に添付した。

##### 3.2.4.3.4.4. 苦労点・発見点

特になし。

#### 3.2.4.3.5. 実装

##### 3.2.4.3.5.1. 目的

実装を行い、スコープ第2段階の成果物を作成する。

##### 3.2.4.3.5.2. 経緯

上記の成果物を以て、反復第2回目の設計を終了とした。ただ、第2段階は設計ができた部分から実装に移ると決まっていたので、実装の途中にも細かな仕様の変更や設計の変更は行われた。数多くの議論を経た、最終的なモデルが上記のモデルであることを明記しておく。

設計の変更が行われた理由として、「人に使ってもらうソフトウェア」であるので、実際に実装してみてユーザーインターフェース的によろしくない点があれば、実装主導で設計の変更が行われたからである。

##### 3.2.4.3.5.3. 成果物

映像ないゲーム「さうんどおんりい」(第2版)

##### 3.2.4.3.5.4. 苦労点・発見点

実装は、第1段階同様基本的には一人のエキスパートを中心に進められ

た。

なお、今回の「映像のないゲーム」プロジェクトにおいて仕様がわかりやすいゲームにはむいていない、そもそも完全に UML を理解していないなどの理由から、実装に対して UML はあまり必要でなかったのでは、という考えがメンバーたちにあった。

しかし、UML はプログラム設計のために使用されるが、全員がそれを通じてシステムの内容が理解できるのであれば、例えば実装終了後に行われる UML でも意味があると言えるのではないだろうか、と感じた。

#### 3.2.4.3.6. 試験

##### 3.2.4.3.6.1. 目的

実装による成果物が、仕様を満たすものであるか確認する。

##### 3.2.4.3.6.2. 経緯

実装の終了後、試験を行った。試験は、まずシステム仕様一覧を参考とした試験手順書を作成し、各項目に対応する箇所を埋めていく、という形式で行われた。項目として特に重要なのは第 1 段階の反復作業以降に行われた部分である。

##### 3.2.4.3.6.3. 成果物

試験手順書に対応する試験成績書は 5.4.2.3. 項の 3.2.4.3.6.3.-1 に添付した。

##### 3.2.4.3.6.4. 苦労点・発見点

第 2 段階の反復作業の試験手順書には当初 53 個の機能がリストアップされていたが、必要ないため削った機能、後から追加された機能、仕様の変更など、終盤になってかなりの作業が立て込み、(現在のものは修正されているが)試験手順書の試験項目に若干のズレがみられた。

### 3.3. 成果物の評価

#### 3.3.1. 発注者評価

##### 3.3.1.1. 概要

7月13日、発注者である株式会社ユードー（以下、ユードー社）の南雲代表取締役（以下、南雲氏）に、成果物が発注の基準を満たしている物かどうか評価をして頂きたく、ユードー社を訪問して評価会を行った。

この時点ではまだゲーム性を発展させている途中であり、チュートリアルやガイド用の音声は無かったが、ゲームの本質的な部分は出来上がっていた。また、どのようにしたらゲームを面白くすることができるだろうかということを、ゲーム製作のプロフェッショナルである南雲氏から意見を頂きたいという目的もあった。

##### 3.3.1.2. 評価の様子

評価には南雲氏のほかに、ドルビーラボラトリーズインターナショナルサービスインク日本支社から2名、CRI・ミドルウェア社から3名、ユードー社から1名が参加した。

評価会ではまず参加していただいた皆様にゲームをプレイしてもらった。そのテストプレイでは、大半の方が前後の音が把握できないという意見を言っていたが、南雲氏は簡単にクリアできた。スピーカーの性能が悪くても、近づけば音が大きくなり、遠ざかれば音が離れるということに気付けばプレイできるということが分かった。

その後、ゲーム性を発展させる意見を数多く頂いた。チュートリアルの内容や難易度の設定は、この時に頂いた意見を反映した部分もある。他に、今期の開発範囲外の意見（対戦の形式や、3Dのゲームを作成する場合の計算方法）も頂き、今後の発展の参考にすることができた。

##### 3.3.1.3. 評価

南雲氏の評価として、「技術的な問題もあるし、音だけのゲームが面白いのか不安だったが、この日出来たものを見て発展できそうだと感じた」という意見を頂くことができた。

今後の発展材料として「映像の表現によるプロモーションのおかげで売れたゲームもあったが、映像のないゲームとして情緒を感じさせるようなゲームを作成することができれば良い」という意見を頂くことができた。

##### 3.3.1.4. 結論

発注者である南雲氏に満足していただけたということ、数多くの発展材料があり我々の成果物が今後進化をすることができるということがあり、我々のゲームプロジェクト目的であった「視覚にハンディキャップを有する人、健常者がともに楽しい時間を共有することができる、「映像のないゲーム」を開発するための課題を解決すること目標とする」（1.3.2 参照）を達成することができたと考えている。

### 3.3.2. 健常者の評価

#### 3.3.2.1. 概要

成果物の評価をしてもらうため、大岩研究室が開催している授業の受講者を対象に7月19日～26日の間、ユーザインタビューを行った。ゲームを5分程度プレイし、用意したアンケート用紙に回答する、という形式で行われた。

#### 3.3.2.2. 手順

ユーザインタビューを行うため、まずユーザインタビューを行うための場所を確保するために研究室のスペースを借用し、また機材としてデスクトップPCを1台借用した。PCにはヘッドホン及び我々の成果物を搭載し、実際にプレイできる環境を用意した。続いて、先述の授業の受講者を対象にアンケートに協力を求めるメールを送信した。

#### 3.3.2.3. アンケート内容

アンケートには以下の項目が含まれている。1～4用意した回答から選択する形式になっている。以下が、項目と回答の対応である。

##### 問1. 映像を用いないゲームをプレイした感想

- ・ 非常に面白い
- ・ 面白い
- ・ 普通
- ・ つまらない
- ・ 非常につまらない

##### 問2. 映像を用いないゲームをまたプレイしたいと思うか

- ・ 是非プレイしたい
- ・ 機会があればプレイしたい
- ・ どちらでも良い
- ・ あまりプレイしたくない
- ・ 二度とプレイしたくない

##### 問3. 面白かったと感じた部分（複数回答可）

- ・ 映像を用いないという新しいゲームを体験できる
- ・ 音がサラウンドで聞こえる
- ・ 音声による説明が充実している
- ・ ハイスコアを競うことができる
- ・ ゲームの難易度・バランスがちょうど良い
- ・ その他（記入可）

##### 問4. 改善・追加するべきだと感じた部分

- ・ マップ上のトラップや時限爆弾などの聞こえ方
- ・ 音声によるガイド
- ・ ゲームの操作性
- ・ ゲームの難易度・バランス

- ・ 他人と競う部分（スコア以外）
- ・ その他

問 5. その他、自由意見

#### 3.3.2.4. 結果

合計 9 名の人にアンケートを回答してもらうことができた。

問 1 の回答結果は、「非常に面白い」が 1 名、「面白い」が 3 名、「普通」が 3 名、「つまらない」が 2 名であった。普通から高めの評価を受けることができていたことがわかった。

問 2 の回答結果は「是非プレイしたい」が 4 名、「機会があればプレイしたい」が 3 名、「あまりプレイしたくない」が 2 名であり、映像の無いゲームに面白いと感じる要素があったと感じてもらえることができたことがわかった。

問 3 の回答結果は「映像を用いないという新しいゲームを体験できる」が 6 名、「音がサラウンドで聞こえる」が 5 名であり、既存のゲームとは異なる点を高評価してもらうことができた。

問 4 の回答結果は「マップ上のトラップや時限爆弾などの音の聞こえ方」が 7 名、「音声によるガイド」が 6 名、「ゲームの操作性」が 5 名、「ゲームの難易度・バランス」が 3 名であり、まだまだ発展の余地が残っていることが分かった。特に音の聞こえ方に問題があるという意見が多くあったが、5 分程度のプレイでは慣れることが難しく、またヘッドホンの影響もあるため、発展に反映するのが難しい部分であると感じた。また、ガイド部分や操作性については、今後複雑なゲームになるにつれて重要となる箇所なので、更に考慮する必要があると感じた。

問 5 の回答としては、チュートリアルについてや難易度、環境についてといった、上記の回答をフォローする物が多かった。

#### 3.3.2.5. 結論

「面白い」という意見や「またプレイしたい」という意見が多く、我々の成果物は「面白そうなものである」と結論付けることができた。

また、初めてプレイした人がどのように感じているかという貴重な意見を頂くことができた。優秀なゲームに必要な要素として「何度もやってもらえる」ものでなければならない、というものがある。ゲームとしての質を向上するため、今後の発展材料となる意見を数多く得ることができた。



## 4. 個人レポート

### 4.1. 個人レポート（橋山）

#### 4.1.1. プロジェクトを通しての活動内容

私は、株式会社ユードーの南雲代表取締役とともに「映像のない」ゲームを開発していくための問題を解決するためのプロジェクトである「さうんど おんりい」プロジェクトに所属して、活動を行った。ここでは、その活動の詳細を述べていく。

##### 4.1.1.1. 企画決定まで

まず、「映像のないゲーム」の企画をメンバーで持ち寄って、それを発注者である南雲代表取締役に確認してもらうことになった。私は、映像を用いずにゲームを行うということを考えたとき、真っ先に探検をするといったゲームが思いついた。これは、私が自然の中など未知の領域を探検することが好きだからである。例えば、川のせせらぎや虫の鳴き声、葉っぱを踏みしめる足音などから、自分が森にいて近くに川が流れているところが想像できる。そして、音を頼りに森の中を探検して、色々なものを発見するというゲームである。しかし、森の中ではゲームの目標が設定しにくいので、宝探しという分かりやすい目標に変更した。それに伴い、舞台も森の中から古代遺跡に変更した。

プレイヤーは宝を探しに遺跡に入ったが、途中で落とし穴に落ちてしまい、持っていた照明器具を壊してしまう。真っ暗闇の中、プレイヤーは金属探知機の音だけを頼りに宝を目指す、といったストーリーである。遺跡の中では水がしたたる音やコウモリの羽ばたく音などが聞こえている。また、金属探知機はトラップや宝に反応して、色々な音を鳴らす。音の種類と大きさから、自分の近くに何があるのかを判断して、トラップだったら避けて、宝だったらその方向に向かう。

メンバー全員の企画自体は南雲代表取締役に認められたので、他のメンバーの企画と比べてどの企画を採用するかみた。私の企画では、壁があることをプレイヤーにどのように知らせるか、という問題があった。壁が音を反射することで存在を知らせるという意見もあったが、その場合は音の種類が多くなりすぎて煩雑になりすぎる。検討の結果、堀氏の提案した対戦型ゲームに他の3人のゲームの要素を少しずつ取り入れたような企画にすることが決定した。企画詳細については、本書の「2. 企画詳細」を参照していただきたい。

##### 4.1.1.2. ユースケース vs. マニュアル

企画が決定したことで、実際に要求分析に入った。メンバー全員でユースケースを書いたが、いまいちどんなものかゲームの全体像が見えてこない。そこで、UMLは人に伝わる手段であるという認識をメンバーで共有して、ユースケースの代わりにマニュアルを書いてみるという作戦になった。メンバーを2グループに分けて、1グループはマニュアルを書き、もう1グループは組み込みUMLを使ってユースケースを書くことにした。これは、普通のUMLよりも組み込みUMLのほうがゲームの開発に近いのではないか、という疑問を検証するためである。

私は、組み込みUMLを用いたユースケースを書くグループとなった。研究室にあ

った組み込み UML の本を借りて、そこにあったキャンディソーターのユースケースを参考に「映像のないゲーム」のユースケースを書いた。キャンディソーターでは、キャンディソーターを動かす人間をアクター、キャンディソーター内のモーターや IC チップが副アクターとなっていた。ここから、「映像のないゲーム」をプレイする人間をアクター、「映像のないゲーム」の中で動作するキャラクターやトラップなどを副アクターと捉えることで、前回書いた UML より分かりやすいものが書けたのである。

各グループが持ち寄ったユースケースとマニュアルを元に、どちらがゲームの開発に適しているか、という議論が行われた。組み込み UML で書いたユースケースは前回よりも分かりやすくなったが、それでも、まだ伝わりにくい。また、マニュアルは伝わりやすいのだが、詳細な仕様を書かないためユースケースが必要となる。結論として、マニュアルとユースケースの両方を用いて要求分析を行うということになった。ゲーム開発が今までのソフトウェア開発のプロセスとは異なると感じた最初のポイントであり、組み込み UML のことも知る良いきっかけとなった。

#### 4.1.1.3. スコープ記述書の作成

全体分析と同時に、このプロジェクトでどこまで作るか、という範囲を決めるためスコープ定義書を作成した。この仕事は私が担当となったが、スコープ定義書を書くのは初めてだったので PMBOK を参考にしてどのようなことを書けばいいのかを調べた。その結果、プロジェクトの目標や成果物スコープ、成果物の受け入れ基準などを書いたスコープ定義書の第 1 稿が出来上がった。今後、プロジェクトで「映像のないゲーム」を作っていくときに、このスコープ定義書がすべての基準になるとのことであったので、そのような重要な文書を作ったことでプロジェクトの管理に貢献できたのではないかと考えている。

#### 4.1.1.4. 反復作業 1 回目

全体としての分析が終わったところで、いよいよスコープを絞った反復作業 1 回目に入った。ここではゲームの根幹部分を作って、とりあえず動くものを完成させようという方針であった。そのため、プレーヤーと目標（時限爆弾）が存在して目標から音が聞こえてくる、という部分のみを作ることにした。

ここで実装を行うにあたって、Visual Studio .NET 2003 を使うこととなり、メンバー全員が研究室でこれをインストールすることにした。しかし、なぜか私のパソコンにだけインストールすることができず、ウェブ上で色々原因を探っても一向に分からなかった。VS は Windows のコアな部分と絡んでいることが多いため、インストールやアンインストールに失敗したら、OS を再インストールするのが一番早いという意見が多かった。結局 1 週間ほど試行錯誤したが解決しなかったため、2 日ばかりで OS を再インストールして環境を構築した。プロジェクトの進行に影響が出る前に、OS を再インストールする決断が出来てよかったと思う。

また、実装時のソースコードやその他文書をメンバーで共有するために SVN を利用してファイルのバージョン管理をすることとなった。当初、VS 上から Subversion

を利用する予定であったが、VS のプラグインでは日本語ファイルが文字化けしてしまうため、TortoiseSVN というクライアントを利用してバージョン管理をすることにした。私はSubversionの設定やTortoiseSVNのインストール方法などをまとめた。

実装のための準備を終えて実装に入った。私は、デバッグ用の画面を作成して、キー操作によってキャラクターが動くという部分を実装した。C++に不慣れだったため、メンバーの堀氏や大岩研究室の篠崎氏に協力して頂き、なんとか実装することができた。また、音を鳴らすときに使っていた CRI Audio の仕様で不明点があったため、メンバーの堀氏とともに CRI・ミドルウェアに訪問した。今後不明点や質問があった場合は、調整役であった南雲代表取締役を通さずに、メンバーが直接 CRI・ミドルウェアの担当者に問い合わせても良いという許可を頂いた。気軽に質問ができるという状況を作ることが出来たことは、今後の開発において非常に心強いことであった。

#### 4.1.1.5. ユーザーインタビュー

反復作業 1 回目が終わったのを受けて、慶応大学の中根氏にその時点で出来上がっていたゲームを遊んでいただき、意見や感想をもらった。私は中根氏とのスケジュールの調整を行ったが、メンバー全員の都合がなかなか合わずに、インタビューの実施時期が遅くなってしまったことが反省点である。

実際のインタビューでは、「映像のないゲーム」のこと以外にも、最近のゲームではどのようなものをプレイするのかということを聞いた。すると、「電車で GO」という答えが返ってきた。以前、横浜市立盲学校の生徒達も同じように「電車で GO」がプレイしたいと言っていたことを思い出した。「電車で GO」は電車が止まる時の音などがパターン化されている。そのようなゲームであれば、映像がなくても十分にプレイできるということを改めて知ることが出来た。

#### 4.1.1.6. 反復作業 2 回目

中根氏のインタビューを元に、ゲームを面白くするスコアやゲームを分かりやすくするチュートリアルの実装を中心に反復作業 2 回目が始まった。さらにメンバー内でプレイした結果、いくつかの改善点や追加仕様も決まった。

反復作業 2 回目では、要求分析を担当した。それまでの設計を大岩研究室の松澤氏にレビューしてもらった。詳細クラス図を評価するためには、その機能を網羅した詳細なユースケースが必要であるというレビューを踏まえて、ユースケース図と文書を書き直した。

#### 4.1.1.7. 成果物評価会

反復作業 2 回目がある程度終盤に近づいた頃、発注者である南雲代表取締役をはじめとして、ドルビー社や CRI・ミドルウェア社の方々と交えた成果物評価会を行った。ここでは、音の聞こえ方に関する問題と、ゲーム性を向上させるための工夫について色々な意見を伺うことができた。中でも、印象に残っているのは、5.1ch サラウンドのスピーカーを使って対戦を行うときの方法である。私は、プレイヤーが全員 5.1ch サラウンド対応のヘッドフォンをかけてゲームを行うか、通信対戦に

するくらいしか解決方法が思い浮かばなかった。しかし、人間のいる位置に応じてスピーカーから聞こえてくる音の聞こえ方を変えることができるといった技術があることを教えて頂いた。例えば、ゲーム内で音の発するものを投げたとする。投げた側の人は音が遠くなるように聞こえて、投げられた側の人は音が近づいているように聞こえるということが、同じ空間にいても可能なのである。他にもゲーム性の向上において役に立つ情報をたくさん頂き、とても有意義な評価会となった。

#### 4.1.1.8.最終発表会に向けて

最終発表会のプレゼン資料や最終報告書の目次案は6月半ばから作り始めていたため、多少余裕があった。しかし、最終発表会のプレゼンを直前になって変更することにした。今までは、最終報告書に書くことをまとめた活動報告であったが、新しいプレゼンでは今後同じようなゲームを開発する人たちに向けての提言が中心となっている。これは、単なる活動報告では聞いている方が退屈してしまうのではないかと考え、他のプロジェクトとは異なるプレゼンをしようというメンバーの意向である。

この決定を行うために、5 時間に及ぶミーティングを行った。誰に対してどのようなプレゼンを行うのか、ということ念頭に置き、10 分という限られた時間で自分たちが伝えたいことを最大限にアピールするためのプレゼンを考えた。

また、最終報告書ではプロジェクトの概要と流れを記述した。これは目標の設定とも関連しているため、次項で詳しく述べる。

#### 4.1.2.目標の設定と達成について

##### 4.1.2.1.設定した目標（目標管理シートより抜粋）

（１）今までゼロから物を作った経験があまりないので、チーム開発を通してソフトウェア（今回はゲーム）開発プロジェクト全体の流れを把握する。本プロジェクトを知らない第三者に概要が説明できることが、プロジェクトの流れを把握しているという目標の評価基準となる。また、アウトプットとして、最終報告書にプロジェクトの流れを記述する。

（２）PM の補佐をしながら、プロジェクトの管理方法を学習する。本プロジェクトでは、PMBOK の定義の監視・コントロールフェイズについて学習することを目標とする。これを達成するために、スケジュール管理やコミュニケーション管理について、PM に具体案を提示する。プロジェクト終結時に、私の提案がプロジェクトにどれだけ貢献したかということの評価基準とする。アウトプットとしては、最終報告書の個人レポートとして、プロジェクト内でどれだけプロジェクトの管理が補佐できたかを記述する。

##### 4.1.2.2.目標の達成について

（１）について、最終報告書のプロジェクトの概要という項目にプロジェクトの全体像をまとめた。

また、プロジェクトの流れについては PMBOK におけるプロジェクトの流れを整理し

た。最終プレゼンテーションでプロジェクトの概要を説明することで、プロジェクトの流れを把握しているかどうか評価できると思われる。

(2) について、ユーザーインタビューの調整、終盤スケジュールの作成などスケジュールに関しては大きく貢献した。また、メールや実際に会うことで頻繁にコミュニケーションを取り、情報の誤解や行き違いが起こらないように努力した。また、報告作業を積極的に行うことで新規履修者の参考になるようにした。

#### 4.1.3. 感想

##### 4.1.3.1. プロジェクト全体を通して

今回、このプロジェクトが始まる前に、株式会社ユードーの南雲代表取締役と下準備を進めてきた。ドルビー社や CRI・ミドルウェア社など各企業にも訪問して、春休みを費やして企画してきた。実際に4月になって学生が集まるかどうかという心配があったが、その心配は杞憂に終わった。

実際にプロジェクトが動き出してからは、とにかく動くものを作ることが最重要であった。いくら面白い企画であっても実際に音だけでゲームを行うことが確認できなければ、それは画餅になってしまう。幸い、プロジェクトメンバー全員が週に15時間以上このプロジェクトに貢献できると回答していたので、時間的な余裕はあったと思う。初めのうちは新規履修者2名にプロジェクトがどのように進んでいくかということを知ってもらうために、率先して進捗報告をしたり、メールを送ったりした。プロジェクトメンバー全員が個性的で自己主張をするため、良い意味で常に議論が絶えないプロジェクトであった。

PMの中川氏も積極的にメンバーとコミュニケーションを取ったり、資料をレビューしてくれたりした。管理は今までのどのプロジェクトよりも厳しかったが、毎週出されるアクションアイテムやリビジョンされるスケジュールは、自分たちの立ち位置を明確にしてくれた。今、何をやらなければいけないのかということを各メンバーが認識しながら作業を行うことができたため、スケジュール的にも余裕ができたのである。

また、新規履修者であった堀氏がC++に精通していたため、一番困難だと思われた実装の面もスムーズに行うことができた。6月の半ばでとりあえず最低限遊べそうなものが出来たので、一安心した。

今回のプロジェクトで特に気を使ったのは時間の配分である。4年になって授業は減ったものの、大学院の入試準備や2輪の免許取得、研究室の仕事など、本プロジェクト以外にも時間を割かなければいけないことが多かった。昔から時間に追われたときに自分が使う方法は、やることをリストアップして優先順位をつけることである。優先順位をつけた結果やらなければいけないことは、いつまでにやる必要があるかを確認して、いつやるかを決めるのである。やると決めたら徹夜してでもその日中に終わらせることで、過密スケジュールでもなんとか作業をこなしていった。

7月18日に大学院の合格発表が行われ無事合格することができたので、「映像のないゲーム」の開発は大学院の私の研究として引き継がれる。そのことを考えても、今回のプロジェクトで発注者に一定の評価をもらえたものを開発できたことは非常

に大きな成果である。

また3ヶ月という限られた時間ではあったが、自分たちが作ったゲームに対して視覚にハンディキャップを有する人やプロのゲーム開発者の人たちから意見をもらうという貴重な経験ができた。映像を用いずにゲームを開発して面白いゲームができるのか、という問いを持ち続けながらこのプロジェクトを続けてきたが、発注者やユーザーの評価を受けて「映像のないゲーム」は面白いゲームになるだろうと確信することができた。今後も、このプロジェクトのメンバーが望めば、一緒に映像のないゲームについての開発・研究を進めていきたいと思う。

#### 4.1.3.2. PMBOK について

今回、世界中で広く認知されているプロジェクトマネジメントの知識体系であるPMBOKに従ってプロジェクトを進めてきた。PMからこのことを聞いたとき、最初は何故PMBOKを使うのか、PMBOKを使ったらプロジェクトはどのように変化するのかという疑問が頭の中を渦巻いていた。

しかし、プロジェクトが進むにつれてPMBOKに従っているということが頭の中から消えていった。前回までのプロジェクトでやってきたことと、今回のプロジェクトでやっていることに大きな差を感じないのである。そのことを改めて考えた時に、PMBOKは何も特別なことをしていないということに気づいた。

PMBOKは世界中で日常的に使われているプロジェクトマネジメントにおける知識体系をまとめただけである。私たちは日常生活の中でプロジェクトと呼ばれるものを行うときには無意識のうちにPMBOKに書かれているような内容を実行しているのである。つまり、逆に言うと私たちがプロジェクト内で意識せずに行っていることに名前をつけて体系づけてまとめたものがPMBOKなのである。

これは私個人の意見だが、PMBOKとはプロジェクトを行っていて何か問題が発生したときにその解決策として最も一般的な解を示しているだけであり、それに従うかどうかは本人の自由である。プロジェクトの初心者は分からないことが多いためPMBOKを参考にすると効率が良い。PMBOKに従ってプロジェクトを行うことで、一般的なプロジェクトというものを知ることが出来る。その上で、自分に合った別の方法があればそちらを採用して、自分なりのプロジェクトマネジメントを作り上げる。そのときに、参考としてPMBOKを利用するのが良いのではないかという結論に至った。

#### 4.1.3.3. 最後に

本プロジェクトの発注者であり、協力企業との調整を行ってくださった株式会社ユードーの南雲代表取締役、プロジェクトを統括してくれたPMの中川氏、一緒にプロジェクトを行ったメンバーの佐藤氏、堀氏、渡辺氏、その他レビューや実装のときに協力して頂いた大岩研究室とPMOの皆様、他本プロジェクトに協力して頂いたすべての人々に感謝します。ありがとうございました。

## 4.2. 個人レポート（佐藤）

### 4.2.1. 目標の確認と達成の如何

#### 4.2.1.1. 目標の確認

私がこのプロジェクトに望むにあたり、目標としていた点が2つある。それを今もう一度確認してみる。

- 今までに体験したプロジェクトは全て「元あるモノの改良」だったということもあり、何も無い段階からモノを作る場合の手順・手法を学びたい。最終的にはプロジェクトで取った手法やマネジメント方法について纏めたレポートを記述することを目標とする。
- 3期目ということ、また就職活動を目前に控えているということもあり、制作活動を通じて自分の適性を把握したい。プロジェクトを通じて、自分のプロジェクト内での役割と仕事内容、またその進行度合を記録し、自らの仕事を振り返っての評価を記したレポートを作成し、自分の適性を確かめることを目標とする。

では、続いて自分がどれだけこの目標を達成できたかを確認したい。

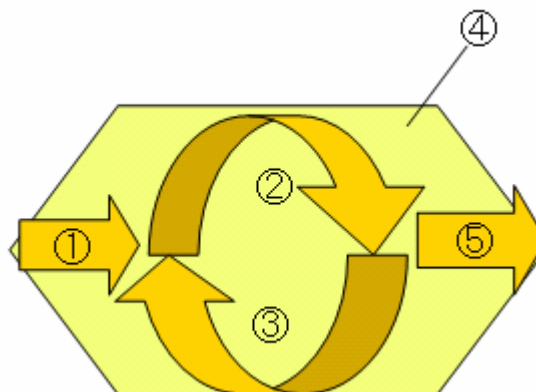
#### 4.2.1.2. プロジェクトで取った手法について

##### 4.2.1.2.1. 用いた手法についての簡略なまとめ

ここでは、今回さうんど おんりいプロジェクトで用いられた手法について、私自身の体験も交えて纏める。

## 1. PMBOK

今プロジェクトは PMBOK に従って活動が行われた。PMBOK とは、プロジェクトのマネジメント手法であり、世界中で広く認知されている知識体系である。



PMBOK には、5つのフェーズがある。立ち上げ 計画 実行 監視・コントロール 終結である。まず立ち上げフェーズから入り、計画フェーズと実行フェーズをループして実施し、目標を達成した時点で終結フェーズへと移行する。これらの全てのフェーズを監視するのが監視・コントロールフェーズとなっている。

今回の我々のプロジェクトでは、企画・立案が立ち上げフェーズ、学習・準備及び設計が計画フェーズ、反復実装作業が実行フェーズ、そして最終的なドキュメントの作成といった作業が終結フェーズに属すると考えている。また、計画とのずれが生じていないか、設計通り行っていないのではないかとといった点を管理するのが

監視・コントロールフェーズであり、意識はしなかったものの常に行われていたということになる。

各フェーズが終了する毎に、成果物を製作する。例えば、我々のケースで言うと、企画・立案フェーズの成果物としてスコープ記述書があり、計画フェーズの成果物として設計書、実行フェーズの成果物として製作したさうんど おんりいのゲームが成果物といったような成果物がある。これらは、フェーズを終了させたという証拠となり、曖昧さ回避のためにも必要な物であると考えている。

## 2. WBS

また、今回のプロジェクトでは WBS によるスケジュール管理が行われた。WBS とは Work Breakdown Structure の略で、プロジェクト全体の大きな仕事を細かい作業に分割したものである。各 WBS の達成時期や担当者を決定することにより、進捗管理や作業漏れが無くて済み、リスクが軽減される。また、一目で現在の進捗が把握できるため、何をすべきかが分かりやすく、作業が円滑になる。

## 3. EVM

EVM の活用によるスケジュール管理も行われた。EVM とは Earned Value Management の略であり、プロジェクトのパフォーマンスを費やしたコスト（時間）で定量的に測定し、作業の進捗や達成度の管理を行うプロジェクト管理法である。これを活用することにより、進捗の状況をグラフという形で目に見えて確認することができる。

### 4.2.1.2.2. 用いた手法について学んだこと

#### 1. PMBOK

PMBOK で用いた手法についてであるが、はっきり言って全てを理解できていないと思っている。しかし、このようなプロセスでプロジェクトが進められるのであるということが分かっただけでも良かったと思うし、はっきりとはではないが「何も無い段階からモノを作る場合の手順・手法」がどのようなものであるかと言うことは把握できたと思っている。

特に、プロジェクトが一連の流れで回っているのだと言うことが理解できたことは大きいと思っている。異なる作業の連なりなのではなく、内容は違っているが、外観的には全て同じプロセスの繰り返しであり、大差が生じることはないということを理解できたということ。この経験は、他のプロジェクトに取り組む際に、前と全然違う性質のプロジェクトであるとか、細かな作業の内容にとらわれず、何をすればプロジェクトの目標を達成できるかということが理解できているため、作業への取り組み方が以前とは異なるのではないかと思う。

また、自分が現在どのフェーズにいるかということを意識し、それぞれに達成条件を設けるということにより、プロジェクト内にはっきりとした役割意識があったと思う。やはり、何事もゴールがはっきりしていると遂行がしやすいものであるということを再確認した。



## 2. WBS

今回のこのWBSによるスケジュール管理であるが、かなり便利なものであると感じた。進捗具合による記入をするのが面倒ではあるという欠点はあるが、何を誰がどこまで行うかということが簡単に見ることができるし、どの項目の事項を達成することが目標の達成に必要なのかということも理解することができる。自分たちのスケジュール管理のためだけではなく、ドキュメントを記述する際にも助けとなることであるし、急な変更に対する対応性もある。今後のプロジェクトでも是非活用して欲しいと思った。

## 3. EVM

EVMに関しては、たまに中川さんが見せてくれるという程度であり、そこまでじっくりと効果を意識して閲覧したことは無かったが、どれだけの効率で作業を行い、また進捗はどれくらいなのかということが理解することができたのはよいことだと思う。しかし、反復第二回に入った時点付近の費用対効果がそれまでに比べれば優れている、ということは分かったのだが、その理由が知りたいと感じた場合、どのようにすればその理由を抽出できるのかが知りたいと思った。しかし、現段階で私がコストについて理解できているとはいいいがたく、まだ実用するには勉強が必要だと感じた。

### 4.2.1.2.3. 用いた手法について学んだこと

例えばの話であるが、私が社会人になり、プロジェクトマネージャーをすることになったとすれば、現段階では私は今回中川さんが取った手法と全く同じ手法を取ろうと思うだろう。それは、私が把握しているマネージングの方法が他にないということ以外に、今回この手法を用いることによって生じたマイナス点が、私の立場からは感じられなかったということが挙げられる。どのようにすればプロジェクトを遂行することができるのか、という漠然なイメージではあるが理解できたということは今後の役に立つと考えている。

### 4.2.1.3. 自分の適性について

#### 4.2.1.3.1. 今期行った主な作業

まず、今期のさうんど おんりいプロジェクトにおいて自分がどれだけの作業を行うことができ貢献することが出来たかを考えるため、私が携わった主な作業内容を挙げる。

- ・ 企画・スコープ範囲決定

1. 企画案を持ち寄り、発表を行った
2. 全員の案を下に議論を行い、企画とスコープの決定に参加した

- ・ 全体設計
  1. インスタンス図・ユースケース図・クラス図・シーケンス図を作成した
  2. 各自の作成した UML 図を下に、最終的な UML 図の作成を行った
  3. 作業中に変更となった点を設計図に反映させた
- ・ 反復第一回目
  1. 全体設計における設計図を元にした、反復第一回目の設計図の作成
  2. スペック決定の議論への参加
  3. マップ内オブジェクトクラスと言った、小範囲の箇所の実装
  4. システム仕様一覧の製作
  5. 試験項目の製作
  6. 反復第一回目の成果物ドキュメントを作成した
- ・ 反復第二回目
  1. 音の連続再生やタイトル画面といった小範囲の箇所の実装
  2. システム仕様一覧の製作
  3. 試験項目の製作
  4. ユーザーレビューの準備
- ・ その他
  1. 進捗報告資料制作・発表
  2. 中間発表資料製作・発表
  3. 視覚にハンディキャップを有する人への評価時、ビデオで手元やプレイ中画面の撮影をした
  4. 発注者評価のための発表資料の作成
  5. 進捗報告の資料作成・発表担当（2 度）

#### 4.2.1.3.2. 作業のまとめ

上記の通り、今プロジェクトでは満遍なく異なる作業を行った。各分野の作業に置いて、私がどれだけの仕事のできたのかを自己分析してみる。

##### 4.2.1.3.2.1. 企画

企画においては、私は第 1 回企画考案会では「鬼ごっこゲーム」、第 2 回企画考案会では「捕獲ゲーム」を考案した。

「鬼ごっこゲーム」は、1~2 人のプレイヤーで遊ぶもので、一方が逃げ、一方が追いかける、というゲームである。全体企画に採用された部分も多かった。

「捕獲ゲーム」は 1 人用ゲームで、逃げる対象を追いかけるというゲームであった。対象は AI を搭載し、また障害物も自分を追いかけてくるというゲームであった。

統括として、どちらもオリジナリティという点では低かった。私は発想力が豊かなほうではないことは自分でも分かっていた。授業の最終課題でも、ありきたりな物や元ある物のリメイク・再現が多く、オリジナルの何かを作るのは苦手分野であった。

キックオフミーティング時、チーム名を決める話し合いが合ったのだが、他のメンバーが面白いアイデアをポンポン出している中、自分も必死に考えたが何も

思い浮かばず、歯がゆい思いをしたこともあった。

まとめとして、「もともとあるそれっぽいものを考案するのはできるが、斬新なアイデアを考案する発想力は無い」ということが言える。

#### 4.2.1.3.2.2. 設計

今回、設計は全体設計と反復作業第1回目において手がけた。自分がどれだけUMLや設計について理解できているか不安なところはあったが、割と問題なく設計をすることができた。完璧ではなかったが、自分の理解が以前に比べて格段に深まっているということが理解できたのは収穫であった。

ただ、そこまで複雑なプログラムでなかったということは作用していると思う。例えば、対戦に対応したモノを作らなければならない場合、設計を上手く行うことができたかは不安なところがある。

まとめとして、「今回は規模の問題で正確な技量はわからないが、全く話にならないというレベルではない」ということがわかった。

#### 4.2.1.3.2.3. 実装

今プロジェクトにおける実装は殆ど堀君に担当させてしまい、あまり手がける部分は無かった。数少ない手がけた部分を思い起こしてみると、C++に触れるのが初めてということもあり、予想よりも時間がかかったというケースが多くあった。例えばステージ選択を作成する、というタスクでは、他の授業で学んだCの技法を使用したためシステム終了時にエラーが出てしまう、というケースがあった。C++に対応した記述法を探すのに結構な時間がかかり、慣れの問題もあるだろうが、効率のよう実装ができたとは言えなかった。

ただ、振られた実装タスクは全てこなしているもので、これも全く話にはならないというレベルではなかったと思う。しかし、最終的な成果物のソースコードの全てを理解しているわけではない。

まとめとして、「今回のプロジェクトで実装のスキルアップは殆ど無かった。レベルとしても、全く分からないというレベルでも良く理解できているというレベルでもない」という結論に至った。

#### 4.2.1.3.2.4. 発表

今プロジェクトでは、中間発表の担当をしたことが一番の思い出である。資料を何度も作成しなおし、練習もみっちりやったおかげで、時間通ピッタリに発表を終わらせることができた。

しかし、発表内容は「つまらない」ものであり、もっと考える余地があったと思う。相手は何を見たいのか、自分たちは何を見せたいのか、ということ考えた資料作りをしなければならないということ学んだ。

最終プレゼンテーションの内容決めのミーティングでも、「君は何を見せたいの」という問いに対し、全く何も思い浮かばなかった。成果物を見せ、その経緯を見せるというものが最終プレゼンテーションであるという既成概念が出来上がっていたからだと思う。もうちょっと柔らかい頭というか、せめて自分がどう

したいのかということは理解しておきたいと思った。

まとめとして、「お堅いプレゼンテーションならできるが、型を破ったプレゼンテーションはできない」ということがわかった。

#### 4.2.1.3.2.5. ドキュメンテーション

ドキュメンテーションであるが、反復第1回目の成果物ドキュメントや、ユーザーレビューのための資料の作成などを行った。

成果物ドキュメントの方は、メンバーに言われるまで自分たちの意見を入れようというアイデアは思い浮かばなかった。読む立場のことを考えたドキュメント作成が行えていないということである。

ドキュメント作成は苦ではなく、それまでの作業を思い起こしてそれを記述するという作業は楽しく感じた。

まとめとして、「つまらないドキュメントを作成するのはうまい」という結論に達した。

#### 4.2.1.3.2.6. その他

最終ドキュメントのマージング作業や、試験項目の書き出しといった、割と機械的な作業は、確かにつらいとは感じたが、スムーズに行うことができたと思う。

#### 4.2.1.3.3. 自分の適正について

上記のことから、自分は「斬新な考えを考案したり、既存と異なる手段を取ることが苦手である。しかし、機械的な作業や形式の決まっている仕事をするとは得意」な人間であることがわかった。

自分で書いておいてなんだが、悲しいとは思う。どうせなら逆のタイプの人間になりたいと思うが、努力次第でなんとかできる問題なのかと言えば私はそうではないと思う。発想力が豊かになる本などがあれば是非読んでみたいが、小学生のナゾナゾにも引っかかる自分が発想力の豊かな人間になれるとは思えない。

なので、私は既存の手段を用いてタスクをクリアするという、地道な方向を歩むべきなのではないかと思っている。手段を問わず何かをしる、という指令があるとしたら、既存の手段を用いてどうにかする。一応、最低限の「自分なりの考え」は反映させつつ、周囲が思っている通りの成果が挙げられる、そのような人間になることができればと思う。

このタイプは決してネガティブな考えに基づいて生まれたものではなく、努力次第で幾らでも貢献することができる、周りにいたらありがたいタイプだと思う。無難かも知れないが、仕事をこなすことができる人間、それが私が将来的に辿り着くべき終着点なのではないかと考える。もちろん、発想力や独創性はもっと欲しいと思っていることには変わりはないが。

#### 4.2.2. プロジェクトを通しての感想

一言で言えば、とても楽しいプロジェクトであったと思う。プロジェクトメンバーは全員頼りになる存在であったし、PMの中川さんも凄く真剣に取り組んでくれ、また

非常に気を使ってくれたので、とても良い環境で仕事できたと思う。

この「音のないゲーム」は今後橋山君が研究していくということになっているそうだが、将来自分が携わっていたということが自慢できるような物に仕上げて欲しいと思う。決してプレッシャーをかけるわけではないが。

何も無い段階からモノを作成するということ、企業の方を PM にしての作業、そして 3 期目としての自覚を持つての取り組みと、あらゆる要素が新しいプロジェクトで、自分の目標を全て達成できたわけではないが、目標にしていた事柄以外にも非常に多くのことを学べたと満足している。

#### 4.2.3. 最後に

共にプロジェクトに取り組んだ橋山君、堀君、渡辺君、そして中川さんに素晴らしいプロジェクト活動が行えたことを感謝したい。

「1 学期間、ありがとうございました。」

## 個人レポート（堀）

### 4.2.4. プロジェクト総括

今回、私はプロジェクトを通して2つのことを見極めたいと思っていた。1つはゲームの設計におけるUMLの有効性、もう1つは集団でコーディングを行う上で重要なことである。また、プロジェクト管理手法がどういったものかも興味があった。それぞれ考えたこと、得たことを見ていきたい。

#### 4.2.4.1. ゲームの設計におけるUMLの有効性

結論から言うと、プロジェクトの設計段階で利用したUMLの大部分で、その有効性を実感することはできなかった。我々は結果的にユースケース図、オブジェクト図、クラス図、シーケンス図、状態遷移図の5種類を作成したのだが、ユースケース図と状態遷移図以外はほとんど有効利用されことなく終わってしまった。そのため、小規模なゲームのプロジェクトであれば要件定義でユースケース図・ユースケース記述を、設計段階で状態遷移図を記述し、それ以外は（設計段階では）むしろ使わない方が効率的に進む、という結論に達した。

その理由を、各図の作成状況・利用状況と併せて以下にまとめた。

##### 4.2.4.1.1. ユースケース図

今回の制作にあたって、我々はまずユースケース図を作成した。ゲームは通常の情報システムとは方向性が異なるためユースケース図が有効かは怪しかったが、今になって考えるとユースケース図・ユースケース記述は要件定義に役立っていたと考えられる。というのも、ユースケース図やユースケース記述を作成する過程でゲームの要件が少しずつ固まっていったからだ。

目標としていたゲームがかなり単純だったため、あえてユースケース図を利用する必要は無かったのかもしれないが、その理念である「ユーザーの立場からの分析」という手法はゲームの要件定義に充分有効だと感じた。

つまり、あえてユースケース図という形態を取る必要は無いが、ユーザーの立場から要件を定義していく、という手法は使える、という結論に達した。

##### 4.2.4.1.2. オブジェクト図

オブジェクト図は反復第1回目の設計段階で制作したが、これは作りっぱなしで活用されることはほとんど無かった。正確に言えばクラス図の作成において使われたが、そのクラス図が有効利用されなかったため実質活用されていない。

##### 4.2.4.1.3. クラス図

クラス図は反復第1回目と2回目の両方で制作され、何度か作り直された。しかしこの作り直された、というのが問題で、結局クラス図は実装に合わせる形で改変されていたのだ。だとすれば、設計段階からクラス図を利用する必要はなかったのではないだろうか、というのが私の見解だ。設計ではクラス図は制作せず、実装が固まった時点で実装状況のリファレンス用としてクラス図を制作すれば充分だった。

その理由は、クラス図を先に書いたからといって実装の細部に渡って見通すことはできないからだ。今回のプロジェクトでも、いくら時間をかけてクラス図を書いたところでいざ実装してみると具合の悪い部分はいくつか発生し、そのたびにクラス図を修正する手間が発生していた。初期段階に記述したクラス図に限っては概念的な整理しか行っていなかったため、そのクラス図を実装に反映したがために無駄な抽象クラスをいくつか作成してしまった。

つまり、クラス図は実装ではほとんど役に立たず、むしろ手間を増やすだけの存在になっていたのである。

そのため、設計段階ではクラス図は作成せず、実装が固まった時点でリファレンス用として作成するやり方が適切だ、という結論に達した。

#### 4.2.4.1.4. シーケンス図

今回使用した図の中で最も影が薄かったのはこのシーケンス図で、反復第1回目で作ったものの、本当に作っただけで以後は完璧に放置されていた。理由は、シーケンス図はゲームという形態と合わなかったからだと考えられる。

ゲームは基本的に自律的に活動するオブジェクトが大半である。ユーザーの操作に呼応して反応を示す部分もあるが、大半はユーザーの意思や操作とは無関係に行動する。同様に、1つのオブジェクトは独立して活動する場合が多いため、他のオブジェクトと呼応して何かをする場合も少ない。そのため、時系列でオブジェクト同士のメッセージ活動をまとめるシーケンス図は作ったところで役に立たなかったのだと考えられる。

また、そもそもシーケンス図に記述された内容があまりに当たり前だったので、あえて参照する必要が無かった、というのも大きな理由だったと考えられる。

#### 4.2.4.1.5. 状態遷移図

状態遷移図は反復2回目になって初めて作成された。反復1回目では状態が1つしかなかったため、作る必要性を感じなかったからだ。

結果として、この状態遷移図は後の実装にかなり役立つことになった。理由は、状態を整理したことで独立したオブジェクトが明らかになり、実装作業を分担することができたからだ。

それまでは明確に状態を整理していなかったため、他の箇所の変更により実装作業がムダになる可能性が高かった。そのため、私が1人で大半の実装作業を行っていたのだが、状態遷移図のおかげで他と干渉しない・されない独立したオブジェクトが明らかになり、その部分の実装作業を他のメンバーにふることができた。

#### 4.2.4.2. ゲームの設計における UML の有効性まとめ

まとめると、ゲームの設計においてはユースケース図、状態遷移図が有効で、オブジェクト図、クラス図、シーケンス図があまり有効でない、という結論になった。

#### 4.2.4.3. 集団でのコーディング

前項では UML による事前設計の必要性に疑問を投じたが、やはり設計そのものは必要だ、ということも今回実感することになった。その理由が、集団でのコーディングである。設計書が無ければ集団で同時並行的にコーディングをすることができないのだ。

「とりあえず作って見ないとわからない・使ってみないとわからない」という面が強かった今回のプロジェクトも、複数人でコーディングを行った部分はごくわずかで、大半のコードは私 1 人が書くことになった。磐石な設計ができてなかったため、他のメンバーと同時に進行しても結局ムダなコーディングになる可能性が高かったからだ。そして実際、反復第 2 回目では設計の根幹であるライブラリが変わったため、それまでのコードで不要になる部分が 200 行ほど出てきた。設計が脆かった結果である。

結局のところ、今回のようなノウハウの少ないソフトを作る場合はどうしても大きな変更点多発するため、集団でのコーディングには不向きである。1 人、できれば 2 人のプログラマーで色々と検討をしながらコーディングを行い、使用するライブラリ等と大枠の状態遷移とを確立させてから個別の独立した部分を集団でコーディングする、というのが最も現実的なやり方だろう。

今回に限ってはそもそもゲームとして成り立つかわからなかったためゲームプレイの部分から作りはじめたが、理想を言えば先にタイトル、ステージセレクト、チュートリアル、ゲームプレイ、ヘルプなどの状態遷移間の関係を設計した方が良かった。そうすれば各状態は独立したモジュールになるので、分割して集団でコーディングすることもできたはずだ。

#### 4.2.4.4. プロジェクト管理手法

私は今回初めてプロジェクト管理手法 (PMBOK) なるものに触れたのだが、その雑感と、今回学んだ手法そのものの概要をまとめて行きたい。

##### 4.2.4.4.1. 雑感

まずなにより、その特殊な用語の多さに辟易した。アクションアイテム、WBS (Work Breakdown Structure)、PV (Planned Value)、フィージビリティスタディ、など、知らなければ全く意味のわからない略語や用語が頻発していた。進捗報告会などでも「アクションアイテムとは何か」といった質問も出てきて、そのわかりづらさが目立っていたと考えられる。起源が海外なのではない面もあるが、もう少しわかりやすい用語に直せないものかと思った。

とはいえ、色々と小難しい用語で説明されてはいるものの手法の内容そのものはわかりやすいと感じた。(1)プロジェクトの目標と期限を決め、(2)期限までに扱う範囲を決め、(3)その上で人や資源を配分し(つまりスケジュールを練り)、(4)適宜監視・調整を行いつつプロジェクトを終結させる。感覚的に理解していることを厳密に定義し、体系だててまとめたただけだ。工数やコストの計算など、ビジネス的な要素に馴染めなかったくらいだろうか。



#### 4.2.4.4.2. 手法の概要

前項でも軽く説明したが、PMBOK とは我々が感覚的に理解しているプロジェクトの管理方法をまとめたものだ。大体のプロジェクト それこそダムの製作から飲み会の開催まで に共通している要素を抽出し、各要素・各ステップを厳密に定義し、その上でコストの見積りやスケジューリングで使える手法をくっつけた、というものだと考えていいだろう。

例として、学校のレポートを執筆する、というプロジェクトを考えてみる。「現代日本のソフトウェア業界が抱える問題点を 1 つ挙げ、その考えうる解決方法を 3000 字以内でまとめよ。提出は 2 週間後」、というレポートが出たとする。

##### 4.2.4.4.2.1. 立ち上げ

これからそのレポートを執筆するわけだが、実はこのレポートが出題された時点ですでにプロジェクトで言う「立ち上げ」は完了している。

立ち上げでは本来プロジェクトの認可を得るため、プロジェクト憲章や成果物記述書などを出力しなければならないのだが、今回は執筆者がレポート執筆の認可を得なければならない存在は居ないため(あえて言えば、その存在は執筆者本人である) それらの作成は全て省略できる。また、プロジェクトの目標や制約も(1)ソフトウェア業界が抱える問題点と解決方法が挙げられている、(2)2 週間後に提出、(3)3000 字以内、と必要なものは与えられているので、新たに制約を考える必要も無い。

##### 4.2.4.4.2.2. スコープの定義

次に考えなければならないのは、今回のプロジェクトで扱う範囲(スコープ)、言い換えればレポートで扱う具体的な題材を決定することだ。例で言えば、ソフトウェア業界が抱えるどの問題を扱うか、という点だ。これはもちろん、与えられた制約条件の中でできる範囲でなくてはならない。レポート側で提示されたテーマと文字数に沿っていることは当然のことながら、自分が執筆に取れる時間などの資源や、最終的に目標とする品質も考慮しなくてはならない。可能なら過去のレポートなども調査し、どういったレポートの評価が高く、どういったレポートの評価が低いか、なども調べた方が良い。一度決めた題材を教授にレビューしてもらえればなお良い。

ちなみに、品質の悪い成果物をスコープとして定義したとしても、自分がそれを認可し、設定した目標を達成できてればそれはプロジェクト的には成功である。この場合で言えば、単位さえくれば良いというスタンスならスコープの品質を低く設定し、自分の時間と労力というコストを低く抑えて執筆する、というのも手ではある。

また本来スコープ定義が終わった段階でも認可が必要なのだが、立ち上げと同じく認可するのは自分なので、自分がそれで良ければ問題は無い。

##### 4.2.4.4.2.3. 計画

次に行うのは計画だ。計画、つまりスケジューリングのことだが、具体的に

やるのは作業全体の設計と資源の配分である。例で言えば、いつ目次を作成するのか、いつ資料を調査するのか、いつ執筆を行うのか、といったことの決定である。大抵我々はこのあたりをあまり厳密に決めず、「何曜日の何時ごろに作業する」くらいのことしか決めないが、ビジネスでのプロジェクトではWBSという詳細な作業項目と、その実施スケジュールを定義する。そのことでスケジュールが明確になるとともに、各作業担当者の責任追求もしやすくなるからだ。しかし、今回の例ではスケジューリングも作業も全て自分 1 人で行い、その責任も自分で負うため、あえてWBSの形で文書化する必要は無い。

なお、PMBOK ではこの計画を非常に重視しており、プロジェクトを左右する活動だとしている。そのため、いくら期限が迫っていたとしても計画は入念に行う方が結局は良い、ということになっている。

#### 4.2.4.4.2.4. 実行・監視

計画まで完了すれば、あとはその計画に基づいて実行するのみである。目次を考え、図書館へ行って資料を漁り、執筆を行い、推敲を重ねる。作業内容としてはこのようなものになるだろう。その際、常に監視をすることを忘れてはならない。つまり、自分の現状とスケジュールとを比べ、どの程度の遅れがあるかを自覚し、その調整を行うことである。例えば、必要な資料が手に入らなくて、調査が長びきそうだからその分先に書けるところから執筆を始める、といったようなことだ。

また、推敲作業を行うことも非常に重要だ。予定されている内容を網羅しているかを確かめ、もし足りないようなら追加する。あるいは不要な部分を削除する。これらの作業はプロジェクトで言うところの試験にあたり、全体の品質を保証・向上する重要な手順となる。本来は試験の手順書なども作成し、その結果も文書として残す必要があるのだが、今回の例ではやはり不要だろう。

#### 4.2.4.4.2.5. 終結

以上のような手順を踏み、実行と監視を繰り返すとやがてプロジェクトは終結する。正確に言えば、設定された期限に合わせてプロジェクトは組まれているので、期限が近づくとプロジェクトは終了する。最後に残っているのは成果物を提出することと、その評価を行うことである。今回の例でも文字通り、レポートの提出とその評価が行われる。ビジネス上のプロジェクトでは評価も含め、未来のために様々なことを分析し、文書として残しておくが、今回の例では特にそういったことを行う必要もないだろう。もちろん、そういったことも行うのがベストである。

#### 4.2.4.4.3. プロジェクト管理手法まとめ

以上が、PMBOK から学んだプロジェクト管理手法の概要である。多くの点を省略したが、大筋としてこのようなものだとは認識した。

#### 4.2.4.4.4. 参考文献

- ・『PMBOK ガイド 2000 年版』 2002 Project Management Institute Inc.

### 4.3. 個人レポート(渡辺)

#### 4.3.1. 学習プロジェクト

##### 4.3.1.1. 学習プロジェクトにおける個人目標の確認とその目標に至った背景

私がこのプロジェクトに望むにあたり、学習プロジェクトとして目標としていたのは「システム開発に利用する UML(Unified Modeling Language)の基礎的な理解を目標とし、インターネットで公開されている無料 UML 技術認定試験 (<http://naha.cool.ne.jp/overdrive/UmlExam/index.htm>)を利用して理解度を示し、それについてレポートをまとめる」ということであった。

この目標に至った背景として、大岩研究会 1 期目である私は、システム(今回はゲームだったが)開発において右も左もわからない状態であり、今回のゲーム開発プロジェクト「さうんど おんりい」の開発環境である VC++や設計に利用する UML の勉強会に参加した。そこではじめて UML を知り、UML は作る予定のシステムについて共通の認識を得るものだとなった。結果的には「さうんど おんりい」は今までにないものかつゲームであるため、UML 全てが役立ったとは言い難かったが、設計かつコミュニケーションの重要性を私に教えた重要なツールであった。

##### 4.3.1.2. 学習プロジェクトにおける個人目標の達成

テストを受けた結果、48 問中 40 問正解で、合格という一定の基準を得られた。



##### 4.3.1.3. 学習プロジェクトの感想

このテストの中にあった、

**あなたは、ソフトウェア開発プロジェクトのプロジェクトマネジャーです。先日、品質監査のために、設計に関する成果物ドキュメントである約 30 枚の UML の図を提出しました。ところが監査の結果は「成果物ドキュメントに問題あり」という判定でした。最も考えられる問題点は？**

- 1)UML の図の枚数が少なすぎる
- 2)UML の図の枚数が多すぎる
- 3)UML の図で書けないことが成果物ドキュメントとして提出されていない
- 4)内部設計が行われていない

という問題をみて、直接は関係ないが、ユースケース図を初めて書いたときにユースケース記述なしで図だけで表わそうとし、「情報量が足りなくてこの図は駄目」というレビューをうけたことが思い出された。これはもちろん単なる私のミスだが、「情報量が足りなくてわからない」というレビューで頂いた言葉は、UML で表わすことのできないものの文書・図化も大切であると考えさせられた。特に今回は細かい仕様変更が起りやすいゲーム開発であり、UML のユースケース図に当たるのはゲームでいうとマニュアルだからそれを作ってみようなど様々なことをしたのが良い経験になった。

#### 4.3.2. ゲーム開発プロジェクト「さうんど おんりい」

##### 4.3.2.1. プロジェクトの流れとそれにたいする当時の感想

「やったこと」はほかの書類にあるので時系列に沿って当時の考えなどを書く。

##### 4.3.2.1.1. 4 月編

第 1 回目の講義ではメンバーと PM が集まり、自己紹介などや当時は決まっていなかったプロジェクト名を決めた。その時に「さうんど おんりい」か「sound only」と表記方法で議論が起こった。議論というほどのものではなく、要するに揉めたのだが、そこで PM がその単に揉めただけの事についてすら問題の解決方法として

- ・強制...その中で最大の権力者がある解決方法を選択すること
- ・撤退...自分の意見を放棄すること
- ・妥協...折衷案などお互い歩み寄ること
- ・対決...納得のいくまで話し合うこと

など用語があるという事を我々に教えてくれ、私はこれ以後毎回何か活動を行うたびに、その活動の概念を共有するための用語が既につけられていることに驚きながら活動することになった。また、初めての研究会である私は(当たり前のことであるのだろう)毎回ミートログをつけるという細かさにも驚いた。

第 2 回目の講義ではどんなゲームを作りたいかというのをそれぞれ発表しあった。一番おもしろそうだった堀さんの全方向ガンシューティングをハードの問題で試すことなくあっさり切ったのが残念であったが、どう考えてもソフトウェアから離れた問題であったので仕方がなかった。なお、今回初出の用語としては

- ・スコープ...プロジェクトで何をやるか。実質自分たちがやる範囲

・WBS...ワークブレイクダウンシステム。仕事を噛み砕いて構造化するシステムがあった。WBS 詳細、つまりこのプロジェクトで何をやるかは PM が来週までに作ってくるということであった。これが決まらなないと動けない私たちにとって最優先すべきのことであり、後に PM はこの時期(の作業量)は辛かった、と述べている。

#### 4.3.2.1.2. 5月編

第3回目の講義では今回立体音響を表現するための CRI・ミドルウェアの導入などが設定されていたが VS Express Edition ではコンパイルできず、最終的には研究会に掛け合って製品版をもらうことになった。この件に関して PM が述べた言葉は「(人件費が一番高いので)金で解決できることは金で解決する」であった。やはり働くということはそういうことなのだろうかと考えさせられた。また、その話の内容の中にそれに関して

・ステークホルダー...利権に絡む人

という用語が出た。「さうんど おんりい」プロジェクトであれば南雲氏や我々メンバーや PM や大岩先生など要するに関わっている人のことである。

第4回目の講義辺りでは正直毎回1コマ目にある他のプロジェクトの進捗報告及び自分たちのプロジェクトの進捗報告の時間がもったいないと思っていた。そんな時間があるのであれば、もっと自分たちのミーティングや設計実装に充てたいと考えていた。

第5回目の講義は体調不良で休んだ。PM もウチの両親が言うように体が資本、健康が第一だとメールやミーティングでもよく言っていた。やはり年上の方は生きていく中でなんらかの体を壊した経験があり、そのような結論に至るらしいとわかった。

#### 4.3.2.1.3. 6月編

第1反復の実装編であるが、最初の部分以外ほとんど実装にかかわっていない上に途中でその部分は変更され、この時期は講義そのものの以外の時間、アクションアイテム時間はかなり少なめであった。なお、進捗報告の発表時に突っ込まれたアクションアイテムとは何なのか、という質問に対して2コマ目のミーティング中に

・アクションアイテム...締め切りがはっきりしているやるべきタスク(作業)

であると定義された。そのアクションアイテムが割り当てられてはいたが、実時間は少なめであったので、ゲームに必要なだろう(無断で使用していい)フリーの音素材をゲームのストーリーに合わせて収集したりしていた。しかしこの時点で音声が必要であるとわかっていたが、人工音声にするのかいっそ人の声で録音するのかという点は曖昧なまま放置された。最初の WBS にない活動(後から必要だと判明したものなど)は優先度が下がったり、後手後手になってしまう事があると感じた。

この時期に自分が前に立って初めて進捗報告をして、毎週の進捗報告の重要性を感じた。毎週の自分たちの作業量のはっきりするし、どうしても出てくる内部では見過ごしてしまうような問題を外部から見てすぐさま解決に持っていける

というのが素晴らしい点であるといえる。欠点は時間がかかる事で、これは実際に発表する時間やその発表のための資料作りの時間である。解決方法としては、この発表のための資料はできるところまでテンプレート化してプロジェクトメンバーや研究会全体で共有すると時間の削減になるのではと思った。

#### 4.3.2.1.4. 7月編

第1反復で出来た成果物を第2反復で改良するという作業に入った。私は説明文を人工音声で作ったものやゲーム内の音(爆弾の音とか)をCRI Audioで使えるように.wav形式から.csb形式に変換し、実際に聞いてみて調整する作業がメインであった。この作業はCRI Audioを 版から正式版に変更し、CriAudioCraftという変換ソフトを利用した。当初 版で作成していたのだが、実装したプログラム正式版を利用しており、そのプログラム上ではないという事があったためである。ミート外でも作業を効率化するためのコミュニケーションが必須であると感じた。

音関連で苦労したのは、日本語で人工音声を作ってくれるWebページを探すことであった。最初は自分の声で作っていたが、男の声だと耳障りではないように音の波をいじるソフトでいじくった際にもノイズがのりやすいため断念した。また、人工音声は基本的に滅茶苦茶高いソフトを利用するのが主流であり、無料でサンプルを作成してくれるWebページは、そのサンプルを保存させない(しにくい)ようにしてあるのが普通だからだ。今回は運良く見つかったからよかったが、あらかじめ必要だとわかっているのに音声作り担当の私がそういう人脈作りもはやめにしておくべきであったと反省した。

#### 4.3.3. まとめ

はじめての「学習・ゲーム作り」のプロジェクトであったが、ユードー社に伺うなど、非常に楽しく行うことができた。今更だが、何度も出てきたプロジェクトとは

- ・プロジェクト...有期性と独自性を持つ活動

と私たちの中では定義されている。

有期性があるのは当然だが、私たちは特に独自性という点において胸を張れるプロジェクトであったといえる。

## 5. 添付資料

### 5.1. プロジェクト定義書

次ページからは、プロジェクト定義書である。



## 5.2. スコープ記述書

次ページからは、スコープ記述書である。

### 5.3. マスタースケジュール

次ページからは、今期のプロジェクトのマスタースケジュールである。

## 5.4. UML・試験手順書

### 5.4.1. 全体分析

図 3.2.1.3.1.3.1-1 ユースケース図

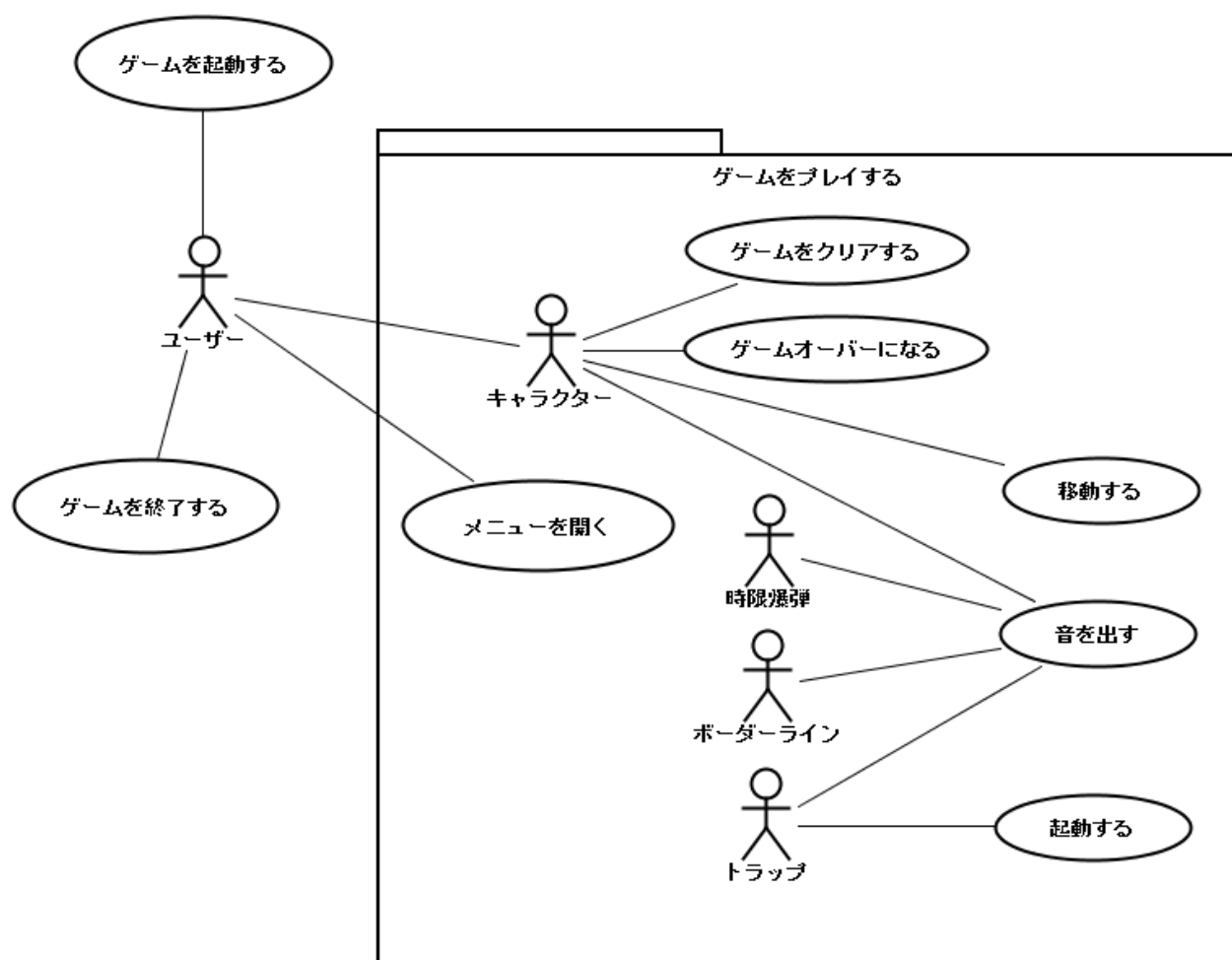


表 3.2.1.3.1.3.1-2 ユースケース記述

ユースケース番号	1
ユースケース名	ゲームを起動する
アクター	ユーザ
目的	ユーザはゲームを起動して遊びたい。
事前条件	4. ユーザはゲームを起動していない
基本系列	1 アクターは、アプリケーションを起動する 2 ゲームのシステムは立ち上がる
事後条件	ゲームのシステムは起動している
代替系列	基本系列 2 で、システムが既に立ち上がっている場合は警告をする
備考	無し
シナリオ	1 橋山はゲームがやりたいと思ったのでアプリケーションを起動させた 2 渡辺はゲームがやりたいと思ったのでアプリケーションを起動させようとしたが、最近痴呆気味のため既に起動させていたことを忘れていたため、既に起動済みであると警告をされた

ユースケース番号	2
ユースケース名	ゲームをプレイする
アクター	ユーザー
目的	ユーザーはゲームで遊びたい
事前条件	・ ゲームが起動して、タイトルへ移動している
基本系列	1 アクターは Enter を押してゲームを開始する 2 システムはゲームの説明をアナウンスする 3 システムはゲーム中の状態になる
事後条件	ゲームがプレイ中の状態になっている
代替系列	基本系列 2 で、ゲーム中の状態でない場合は何も起こさない
備考	・ ゲームはテンキーとマウスを用いて遊ばれる
シナリオ	(7) 佐藤はゲームを起動したあとに操作音声 flowed ので、Enter を押した。ゲームが開始され、「説明を飛ばす場合は Enter を押してください」という音声 flowed。その後、ストーリー紹介が始まったが面倒なので、Enter キーを押すとキャラクターの操作ができるようになった。

ユースケース番号	3
ユースケース名	ゲームを終了する
アクター	ユーザー
目的	ユーザーはゲームをやめたい。
事前条件	<ul style="list-style-type: none"> <li>・ システムは起動している</li> <li>・ システムはゲーム中の状態ではない</li> </ul>
基本系列	1 アクターは、0 キーを押してゲームを終了しようとする 2 システムは、ゲームをやめるがいいかと聞く 3 アクターは、Enter を押してゲームを終了する 4 システムは終了する
事後条件	システムは終了している
代替系列	基本系列 3 で、再度 0 キーが押された場合は元の状態に戻る
備考	無し
シナリオ	佐藤はゲームに飽きたので、終了したいと思った。 おもむろに 0 キーを押すと、「ゲームを終了します。終了する場合は Enter を押してください。キャンセルしたい場合は 0 キーを押してください」と尋ねられた。 佐藤はゲームを終了したいと思ったので Enter を押し、ゲームを終了させた。

ユースケース番号	4
ユースケース名	ゲームをクリアする
アクター	キャラクター
目的	ゲームの目的を達成したことをプレイヤーに知らせる
事前条件	ゲームがプレイ中である
基本系列	1 アクターは、実行可能な動作を用いて時限爆弾に接触する 2 システムは、プレイヤーと時限爆弾が触れたか判定する。 触れていた場合、時限爆弾は解除されクリアとなる 3 システムは、触れたことをアナウンスし、クリアまでにかかった時間を知らせる。 4 システムはタイトルに戻る。
事後条件	ゲームはタイトル画面へ移動している
代替系列	2 で、触れていなかった場合はそのままゲームを続行する
備考	無し
シナリオ	渡辺は時限爆弾の音が聞こえたので、そちらへ移動した。 時限爆弾の音が大きくなったところでゲームクリアの音声が流れてファンファーレが鳴った。3 分 40 秒でクリアしたことが流れて、タイトルへ戻ることが音声で伝えられた。その後すぐに、タイトルにおける操作音声が流れ始めた。

ユースケース番号	5
ユースケース名	ゲームオーバーになる
アクター	キャラクター
目的	ユーザーにゲームの目的が達成できなかったことを知らせる
事前条件	ゲームがプレイ中である。
基本系列	<p>1 アクターは、</p> <ul style="list-style-type: none"> <li>・ 即死トラップにひっかかる</li> <li>・ ボーダーラインを超える</li> <li>・ 制限時間がなくなる</li> </ul> <p>のいずれかの条件を満たす</p> <p>2 システムは、ゲームオーバーになったことをアナウンスする</p> <p>3 システムは、ゲームをタイトル画面に戻す</p>
事後条件	ゲームはタイトル画面へ移動している
代替系列	無し
備考	無し
シナリオ	<p>橋山は左から時限爆弾の音がしていると思い左へ向かった．</p> <p>途中でボーダーラインの警告音が鳴ったが，時限爆弾があると信じて左へ向かい続けると警告音とともにゲームオーバーになった．</p> <p>タイトルへ戻ることが音声で伝えられて，その後操作音声 flowed のでタイトルへ戻ったことが分かった．</p>

ユースケース番号	6
ユースケース名	メニューを開く
アクター	ユーザー
目的	操作説明の確認や，ゲームの途中終了を行う
事前条件	ゲームがプレイ中である
基本系列	<p>1 アクターは、Enter を押す</p> <p>2 システムは、「チュートリアルの場合は 1 を，タイトルに戻る場合は 2 を，メニューを閉じる場合は Enter を押してください」という音声を流す</p> <p>3 アクターは、思うままにキーを押す</p> <p>4 システムは、押されたキーに応じて反応を起こす</p>
事後条件	メニュー項目ごとのアクションが行われ
代替系列	無し
備考	<ul style="list-style-type: none"> <li>・ 1 の場合、システムはゲーム開始時に流れたチュートリアルを流す。</li> <li>・ 2 の場合、システムはゲームをタイトル画面に移動する</li> <li>・ Enter の場合、システムはメニューを閉じ、ゲームを続行する</li> </ul>
シナリオ	<p>佐藤はワープトラップと即死トラップの音の違いを忘れてしまい，Esc を押してメニューを開いた．操作音声 flowed ので，1 を押してチュートリアルを選択して，その中でトラップの音の違いを覚えた．その後メニューに戻り，3 を押してメニューを閉じた．</p>

ユースケース番号	7
ユースケース名	移動する
アクター	キャラクター
目的	トラップを避け、時限爆弾にたどり着きたい
事前条件	ゲーム中である
基本系列	1 ユーザーは、テンキーやマウスで移動したい方向を入力する。 2 システムは、入力通りにプレイヤーを移動させる
事後条件	キャラクターが移動している
代替系列	無し
備考	<ul style="list-style-type: none"> <li>・ 移動時には足音が鳴る</li> <li>・ 移動はテンキーで行い、それぞれ5が前、2が後ろ、1が左、3が右に対応している。</li> <li>・ 方向はマウスで変えられ、マウスを動かすとキャラクターの向きをその方向へ回すことができる</li> </ul>
シナリオ	渡辺は正面からトラップの音が聞こえてきたので、キャラクターを迂回させようと を押した。すると、キャラクターの足音とともにトラップの音が右方向へずれていった。

ユースケース番号	8
ユースケース名	起動する
アクター	トラップ
目的	キャラクターのゲームクリアを邪魔する
事前条件	<ul style="list-style-type: none"> <li>・ ゲームがプレイ中である</li> <li>・ 対象のトラップが起動前である</li> </ul>
基本系列	1 アクターは、キャラクターがアクターの座標上に移動すると起動する 2 アクターは警告音を出し、トラップの効果を発動させる
事後条件	トラップが起動して、トラップごとのアクションが行われる
代替系列	音が聞ける状態でない場合（罠にはまっている等） 音は聞こえない
備考	<ul style="list-style-type: none"> <li>・ トラップは2種類ある</li> <li>・ ワープトラップ：キャラクターをフィールド内のどこかに強制的に移動する。場所は毎回ランダムで決定する。</li> <li>・ 即死トラップ：ただちにゲームオーバーになる</li> <li>・ ワープトラップが起動した場合はトラップは起動済みとなり、以降同じトラップにひっかかることはない。また、起動済みのトラップは音を発しなくなる。</li> </ul>
シナリオ	堀はトラップの音と時限爆弾の音が同じ正面方向から聞こえてくるので、一か八かそちらへ向かってみた。トラップの音が若干大きかったように聞こえたのはトラップが時限爆弾の真正面にあったからで、トラップにひっかってしまった。警告音声とともにワープトラップが起動して、時限爆弾の音はまったく聞こえなくなってしまった。

ユースケース番号	9
ユースケース名	音を出す
アクター	キャラクター，時限爆弾，トラップ，ボーダーライン
目的	ユーザーに存在を知らせる
事前条件	<ul style="list-style-type: none"> <li>・ 全体：ゲームがプレイ中である</li> <li>・ キャラクター：移動中である</li> <li>・ トラップ：起動前である</li> <li>・ 時限爆弾・トラップ・ボーダーライン：一定距離内に存在する</li> </ul>
基本系列	1 アクターは音を鳴らす必要がある場合、音を発する
事後条件	アクタごとにそれぞれの音が鳴る
代替系列	無し
備考	<ul style="list-style-type: none"> <li>・ キャラクター：移動すると足音になる</li> <li>・ 時限爆弾：毎秒，時計が動く音が鳴る（残り時間が近づくと，カウントダウンを始める）</li> <li>・ トラップ：断続的に電子音が鳴る</li> <li>・ ボーダーライン：警告音が鳴る</li> <li>・ 時限爆弾とトラップとボーダーラインは，キャラクターとの距離に反比例して発する音が大きくなる．</li> </ul>
シナリオ	中川はゲームを始めたものの，どこに時限爆弾があるか分からず，とりあえずうろうろしていた．すると，足音に混じって右から時計が動くような音が聞こえてきたので，そちらへ移動すると時限爆弾を発見することができた．



図 3.2.1.3.3.3-1 オブジェクト図

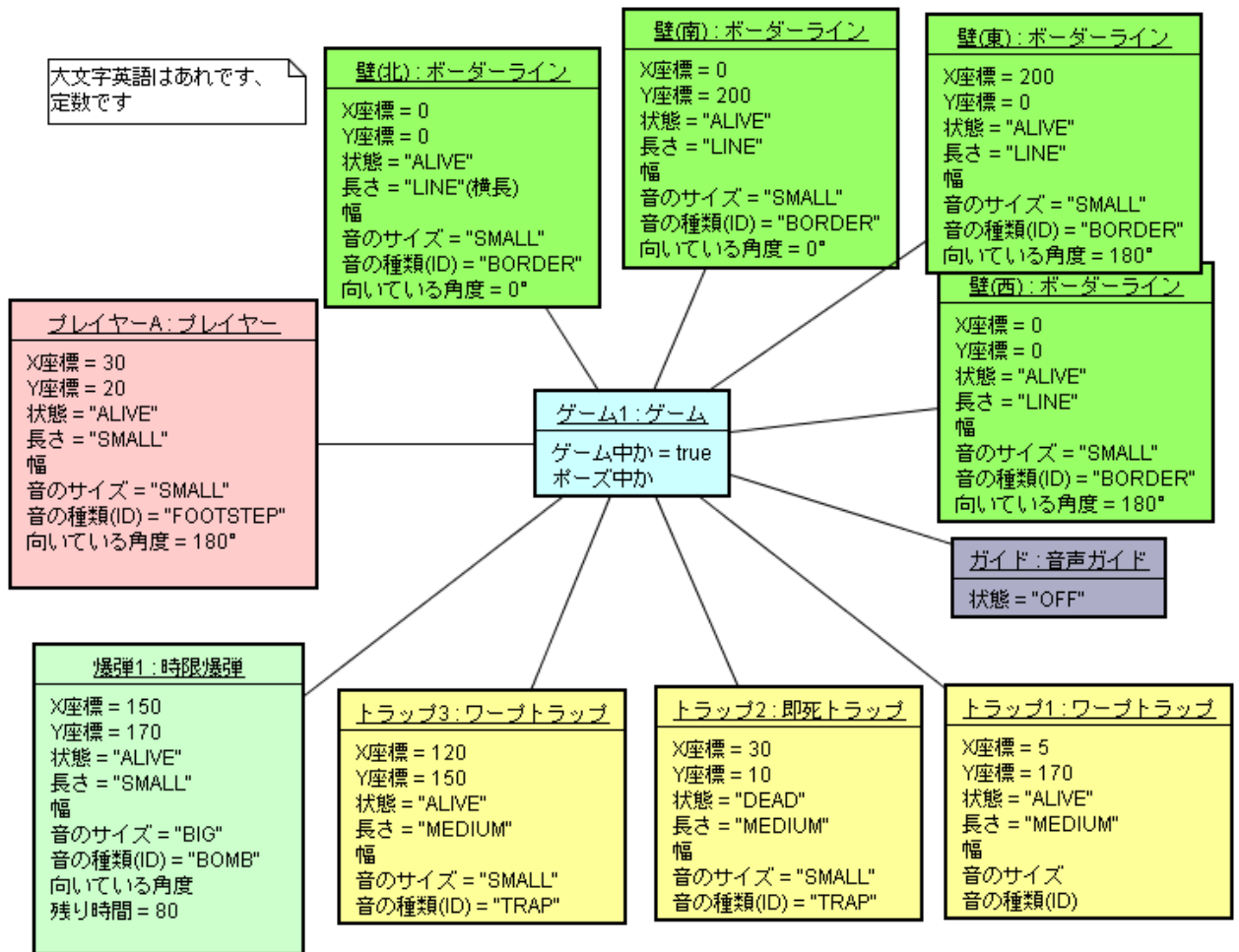


図 3.2.1.3.3.3-2 クラス図

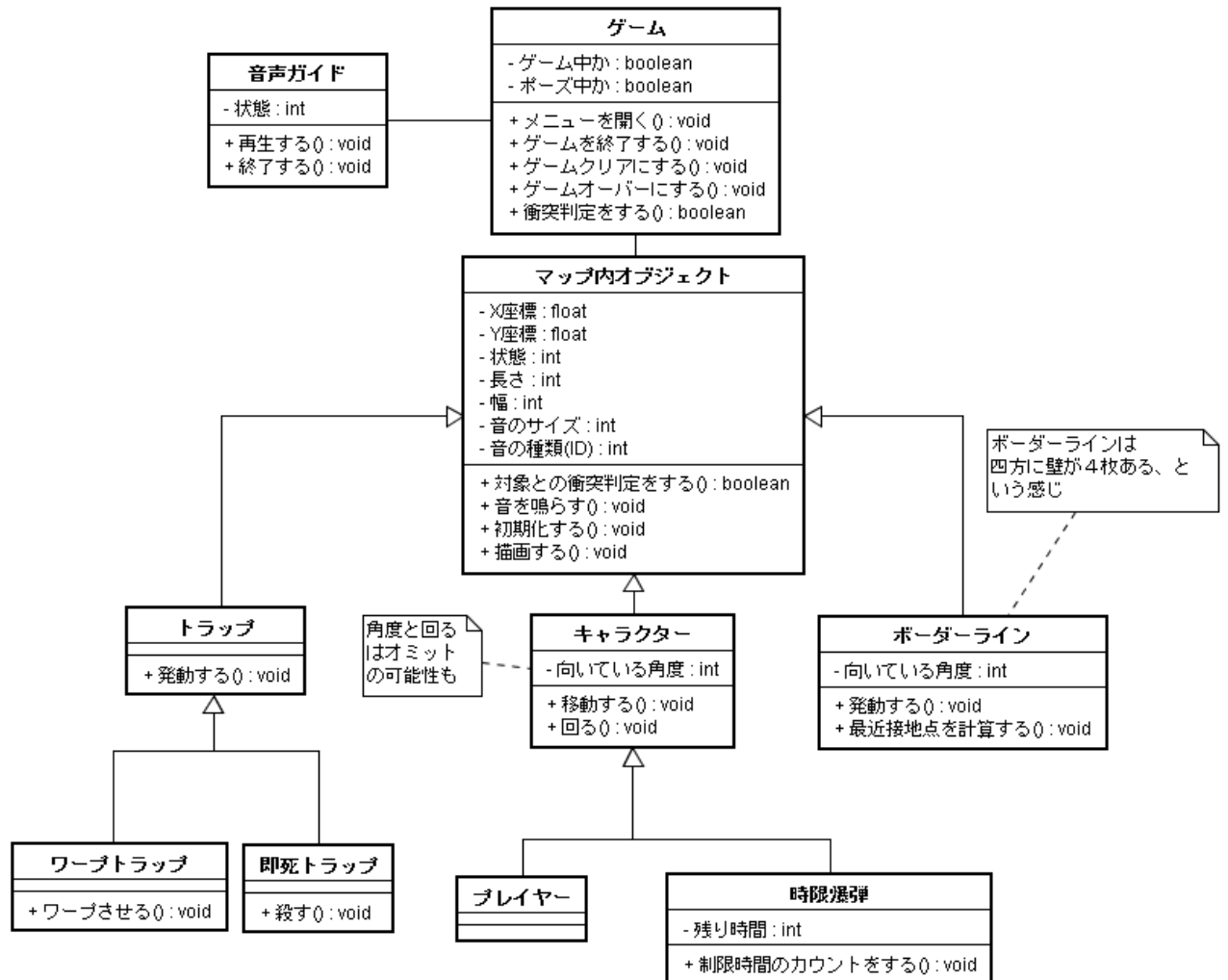


図 3.2.1.3.3.3-3 シーケンス図 1 処理の流れ

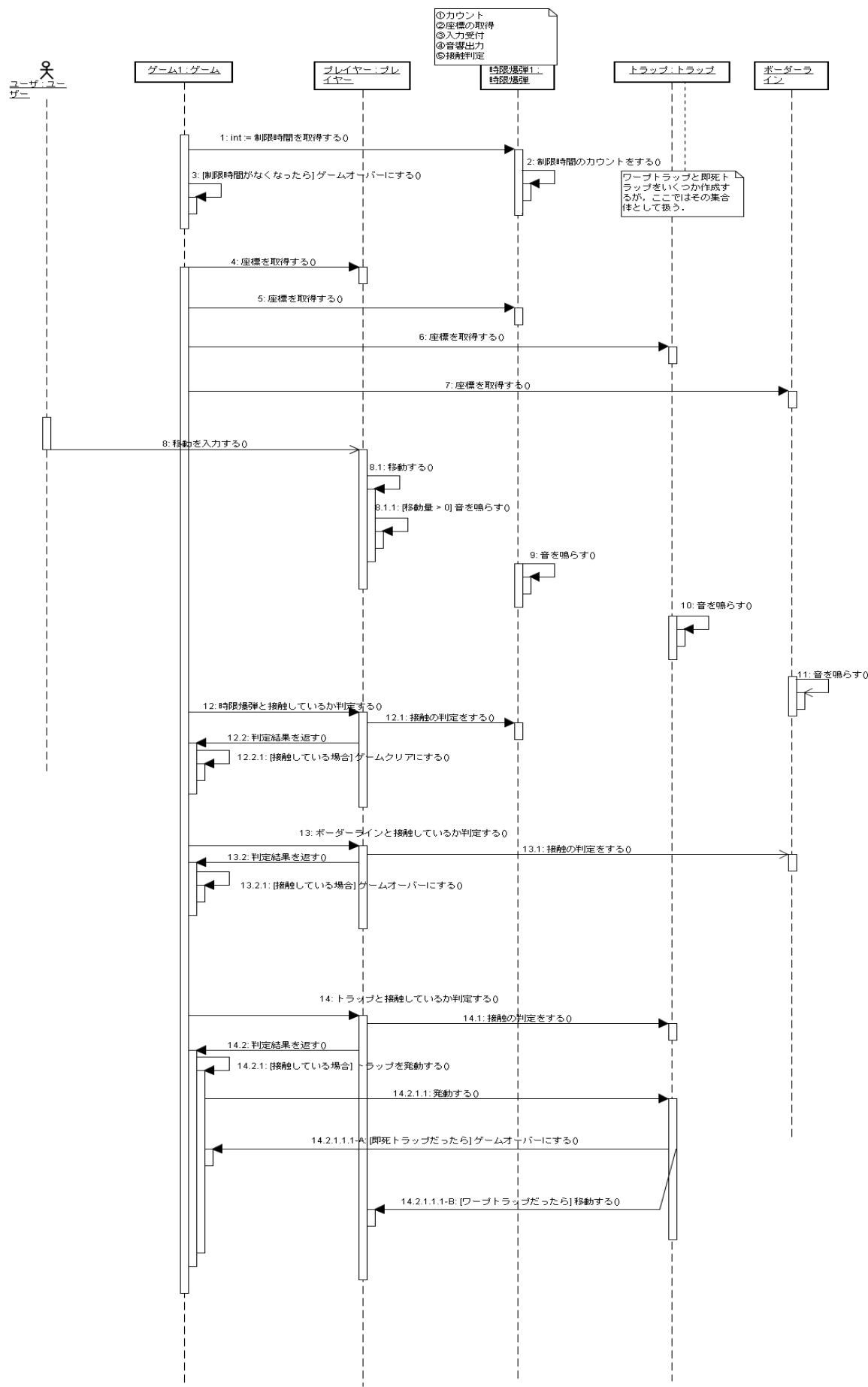


図 3.2.1.3.3.3-4 シーケンス図 2 ゲームを開始する

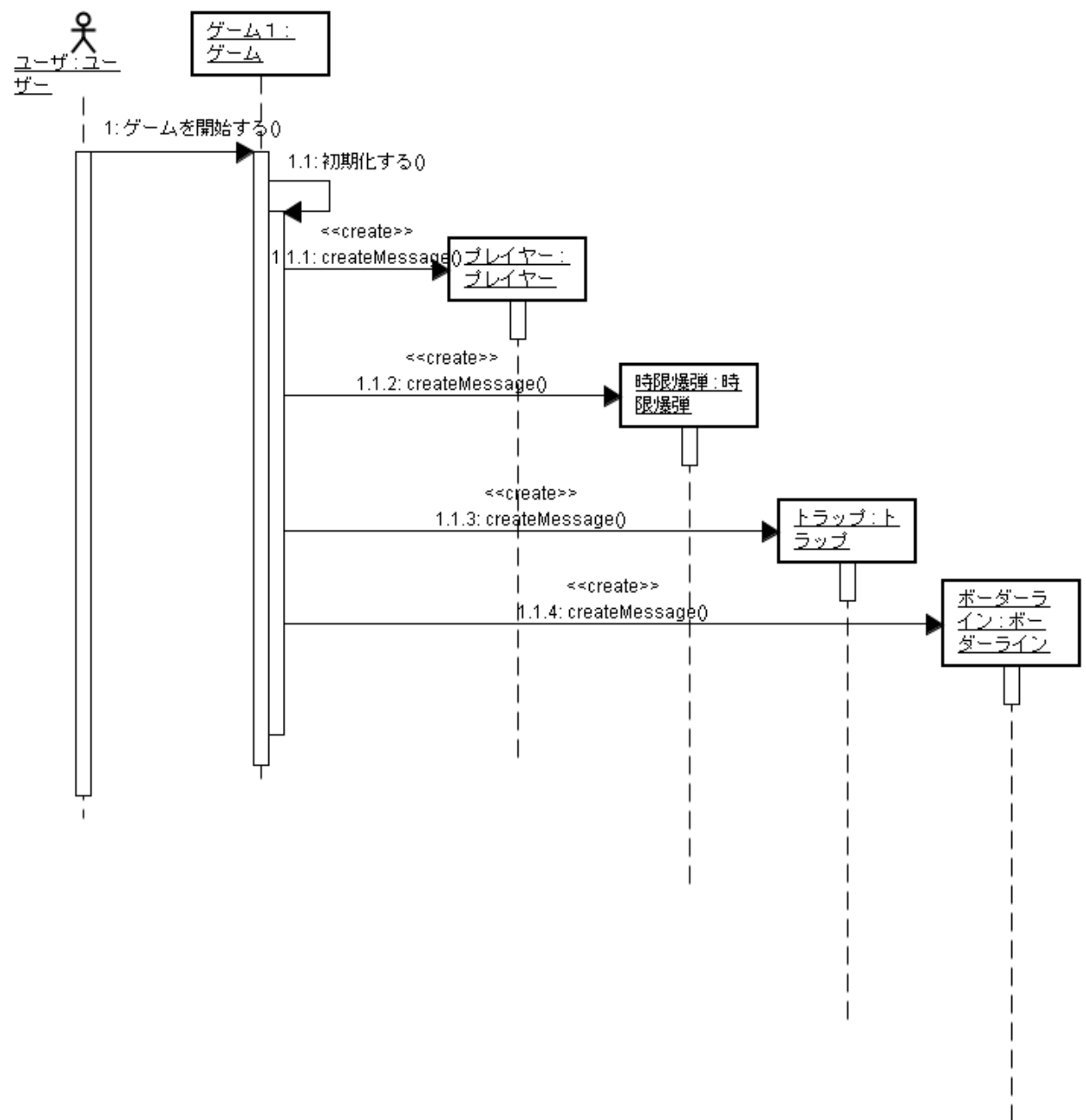


図 3.2.1.3.3.3-5 シーケンス図 3 ゲームをクリアする

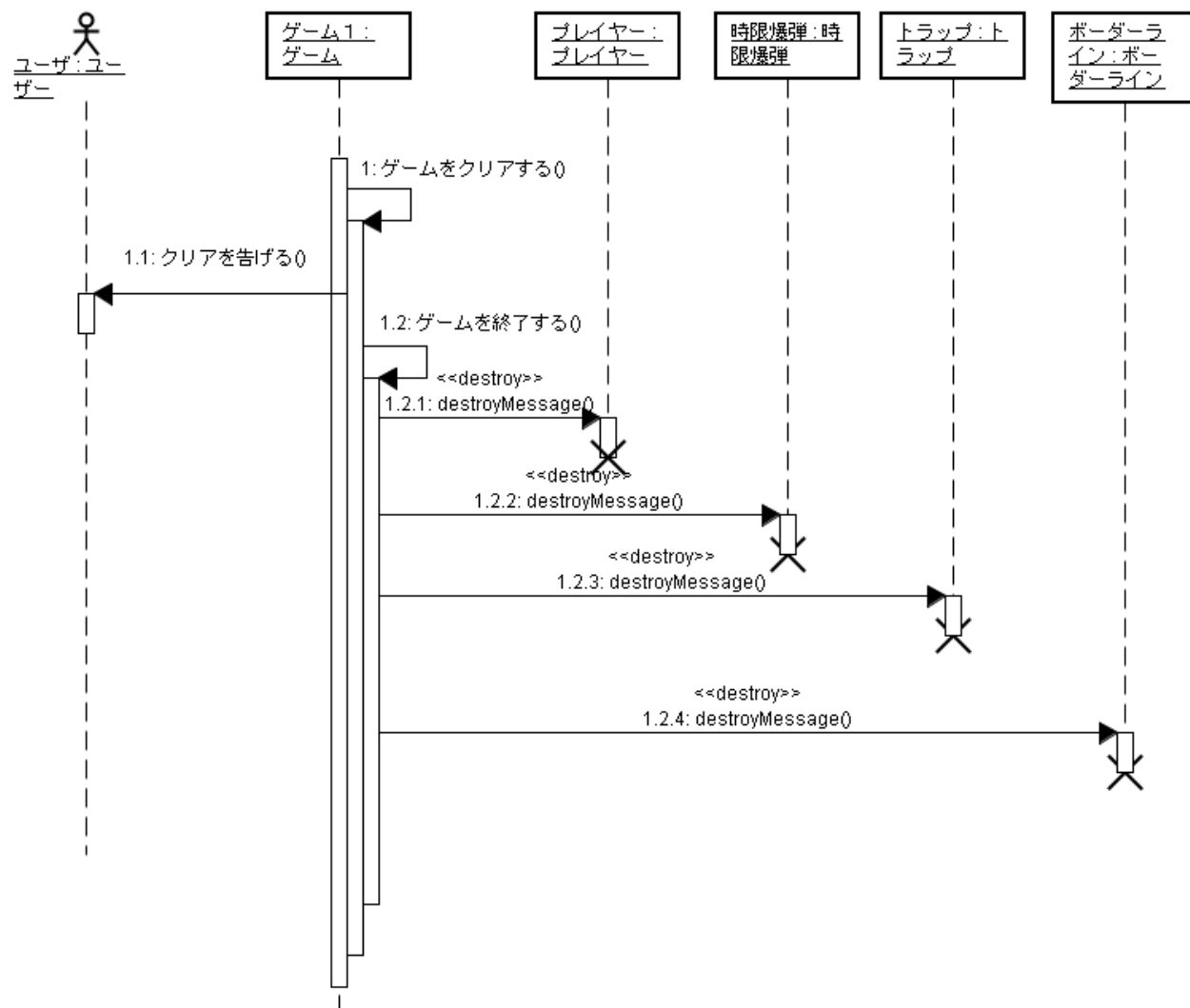
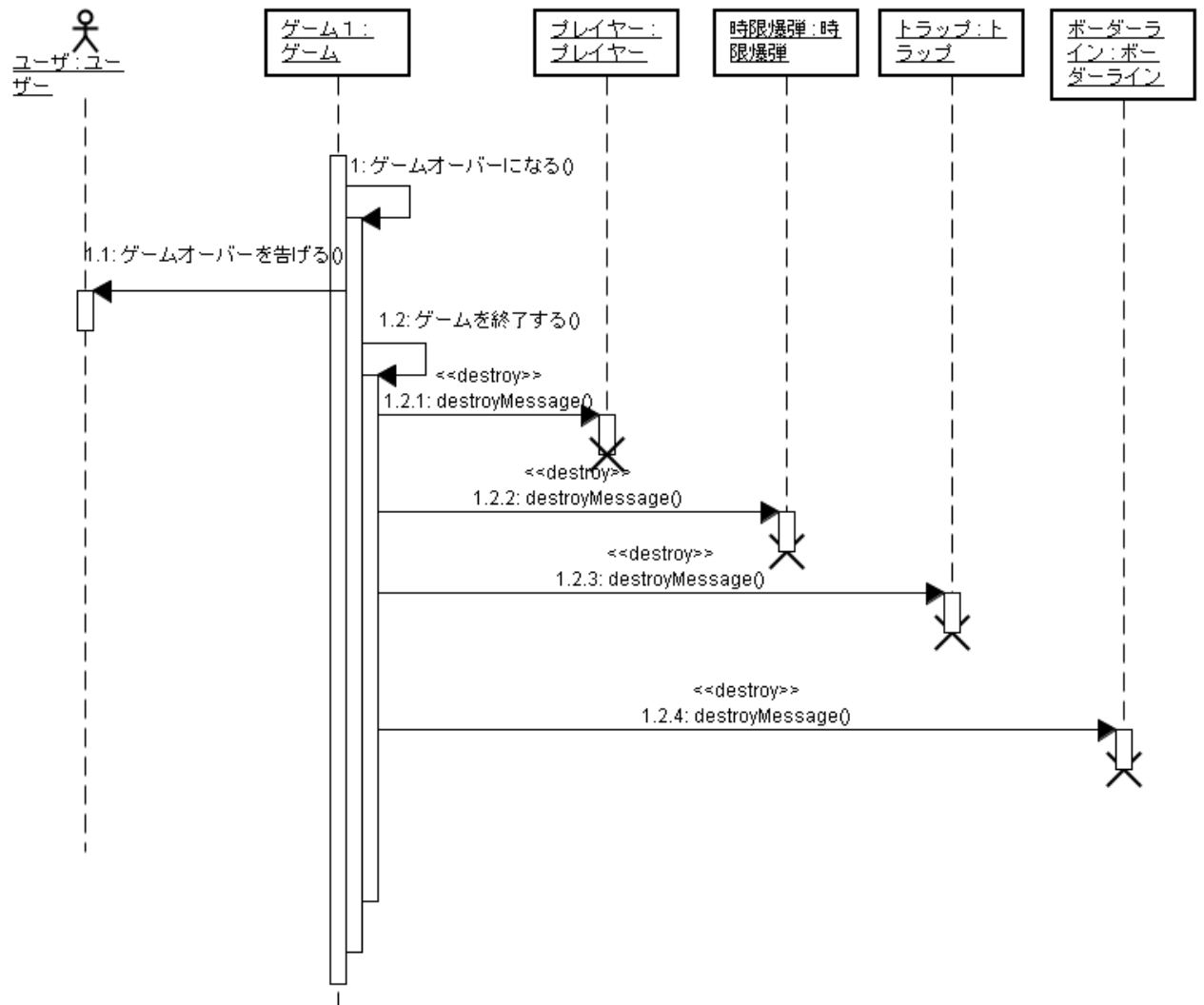


図 3.2.1.3.3.3-6 シーケンス図 4 ゲームオーバーになる



#### 5.4.2. 反復作業 1 回目

図 3.2.2.3.2.3-1 ユースケース図

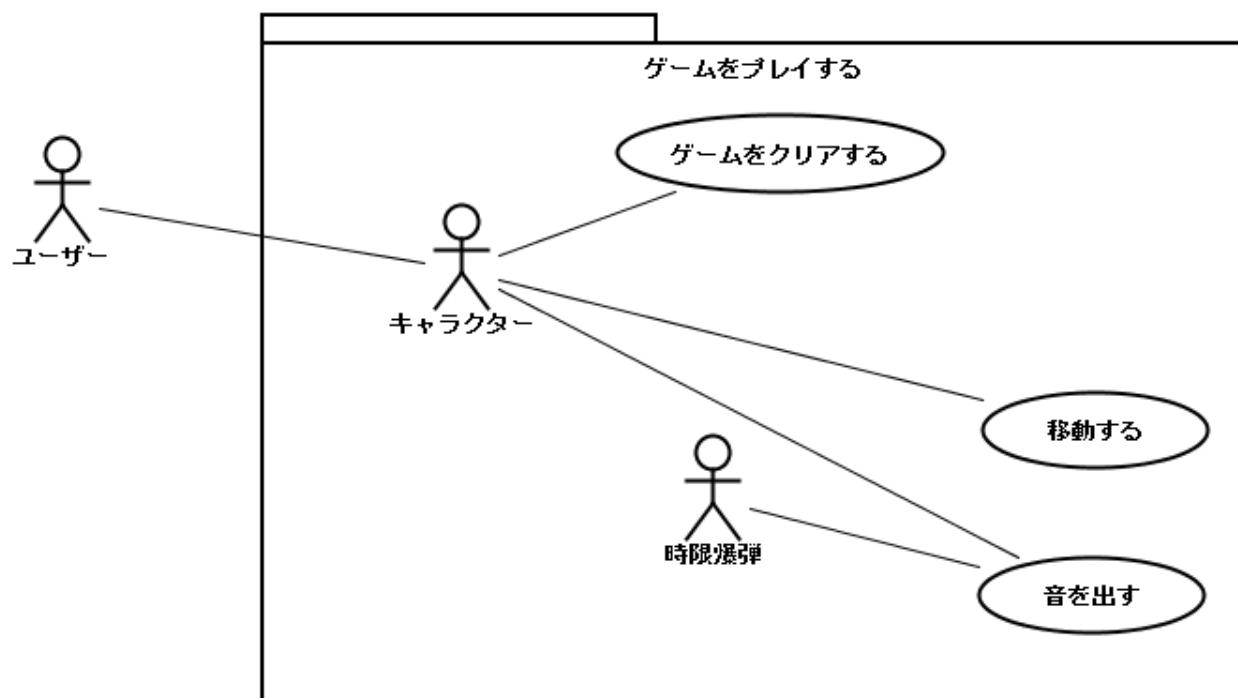


図 3.2.2.3.3.3-1 オブジェクト図

大文字英語はあれです、  
定数です

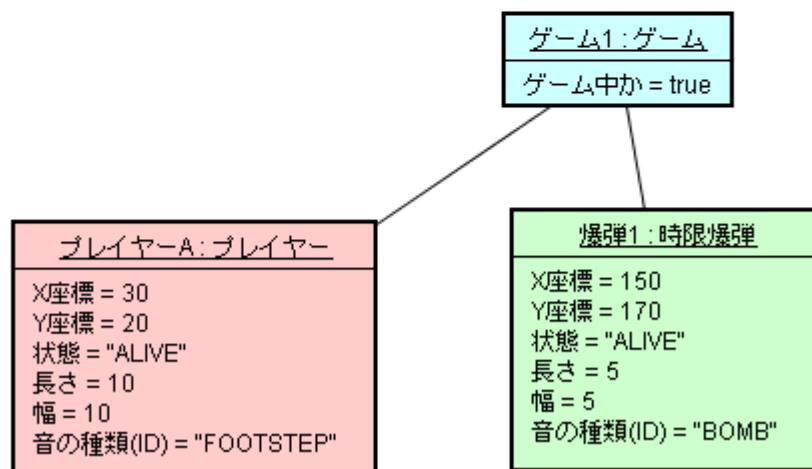
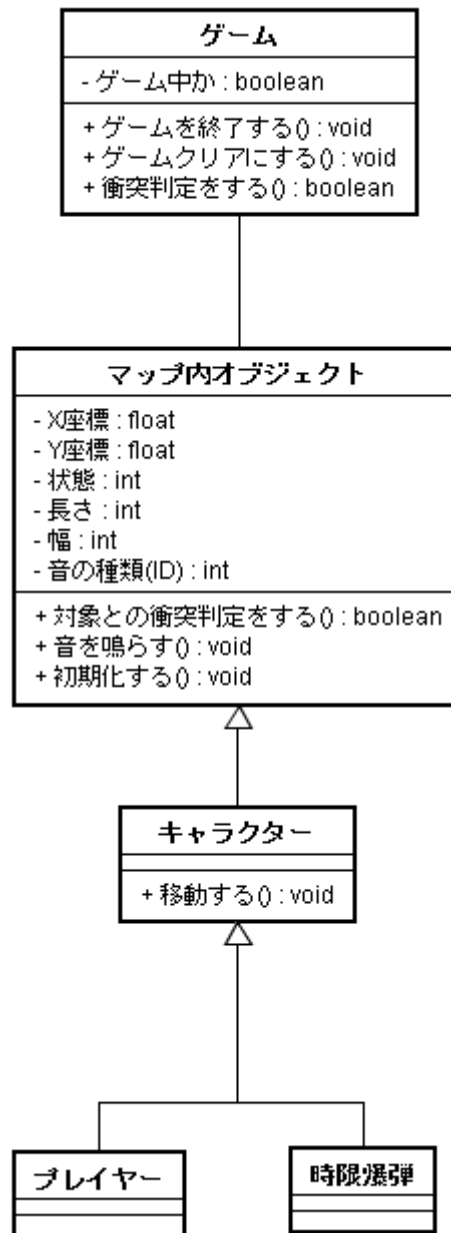


図 3.2.2.3.3.3-2 クラス図





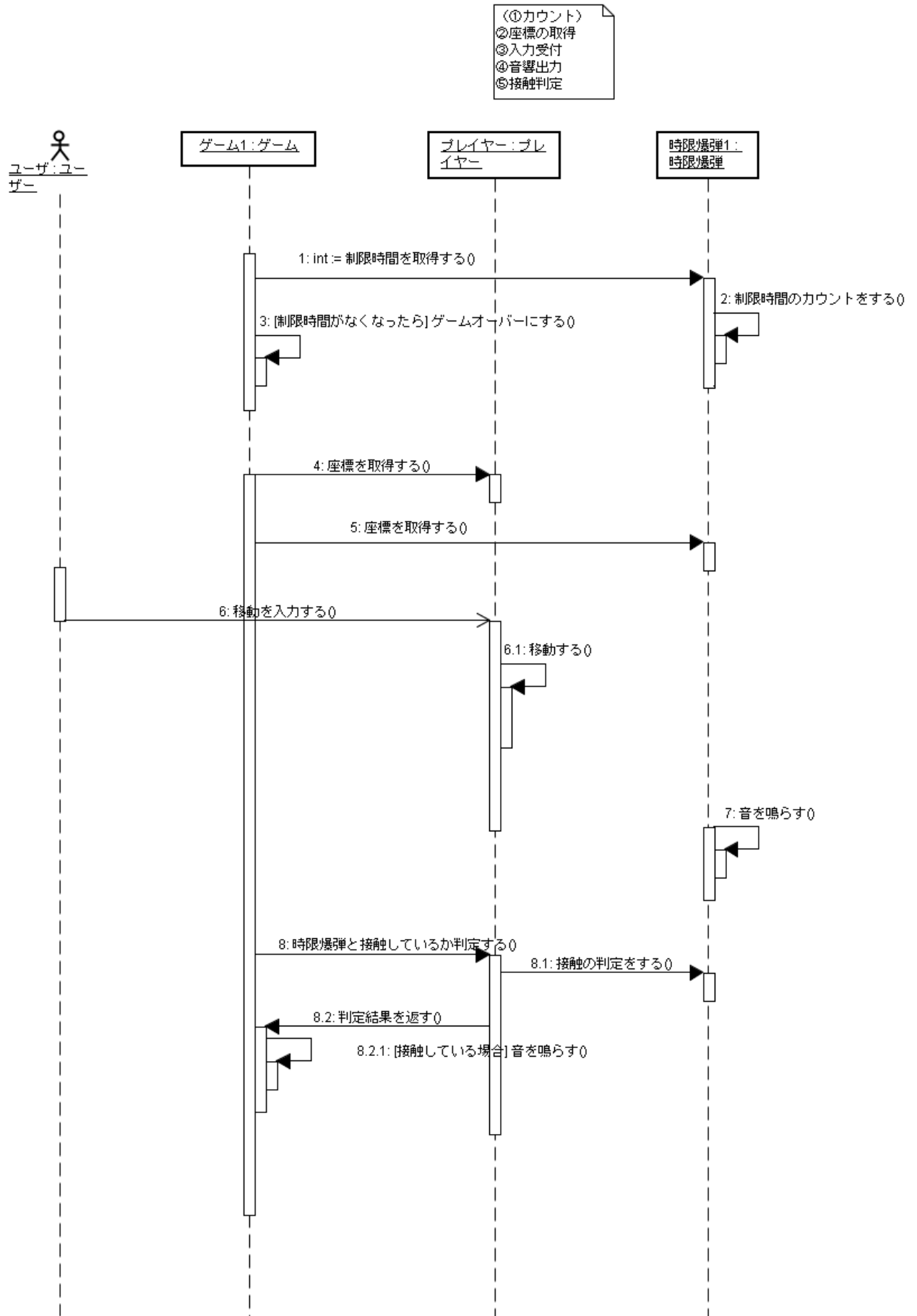


図 3.2.2.3.3.3. -3 シーケンス図 1 処理の流れ

図 3.2.2.3.3.3 -4 シーケンス図 2 ゲームを開始する

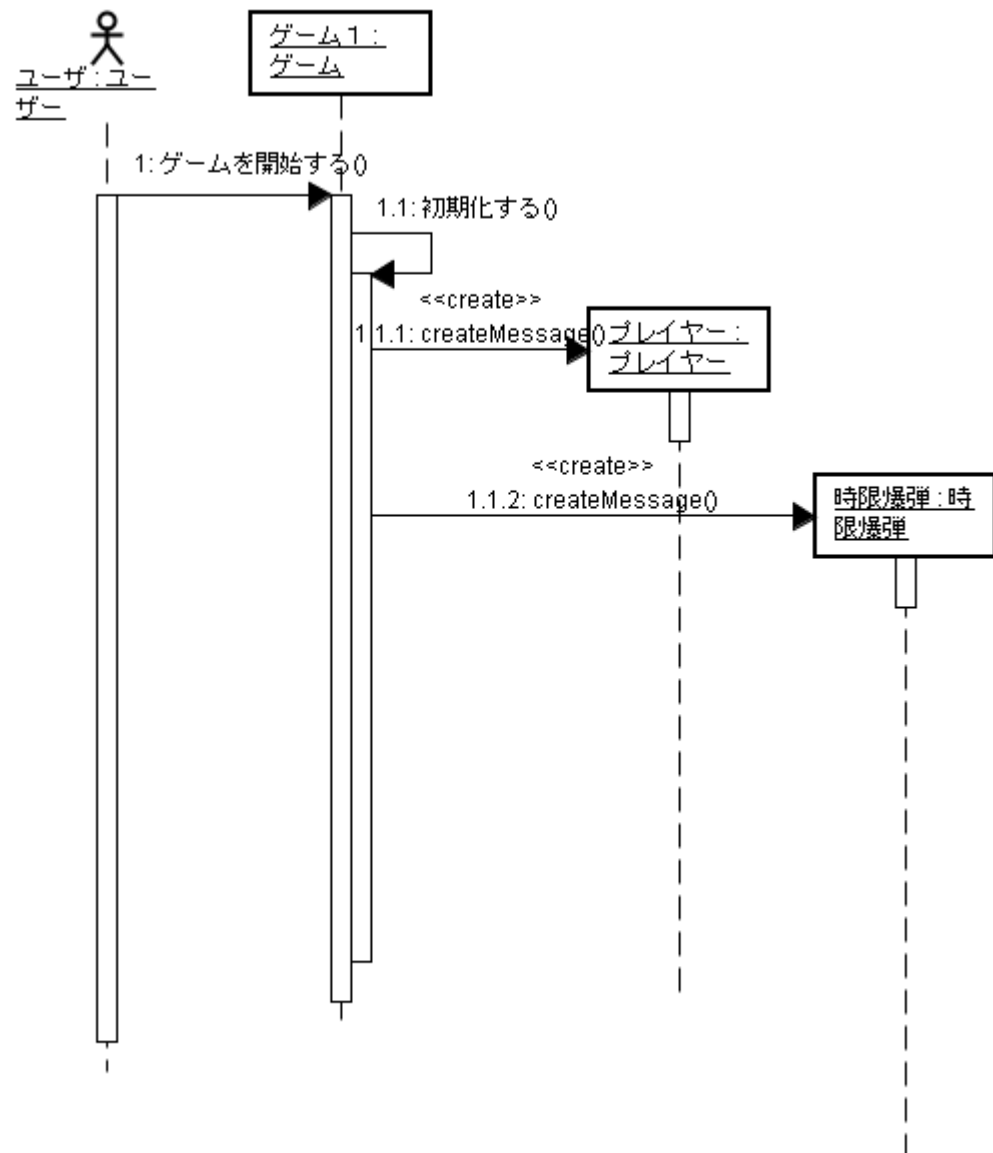
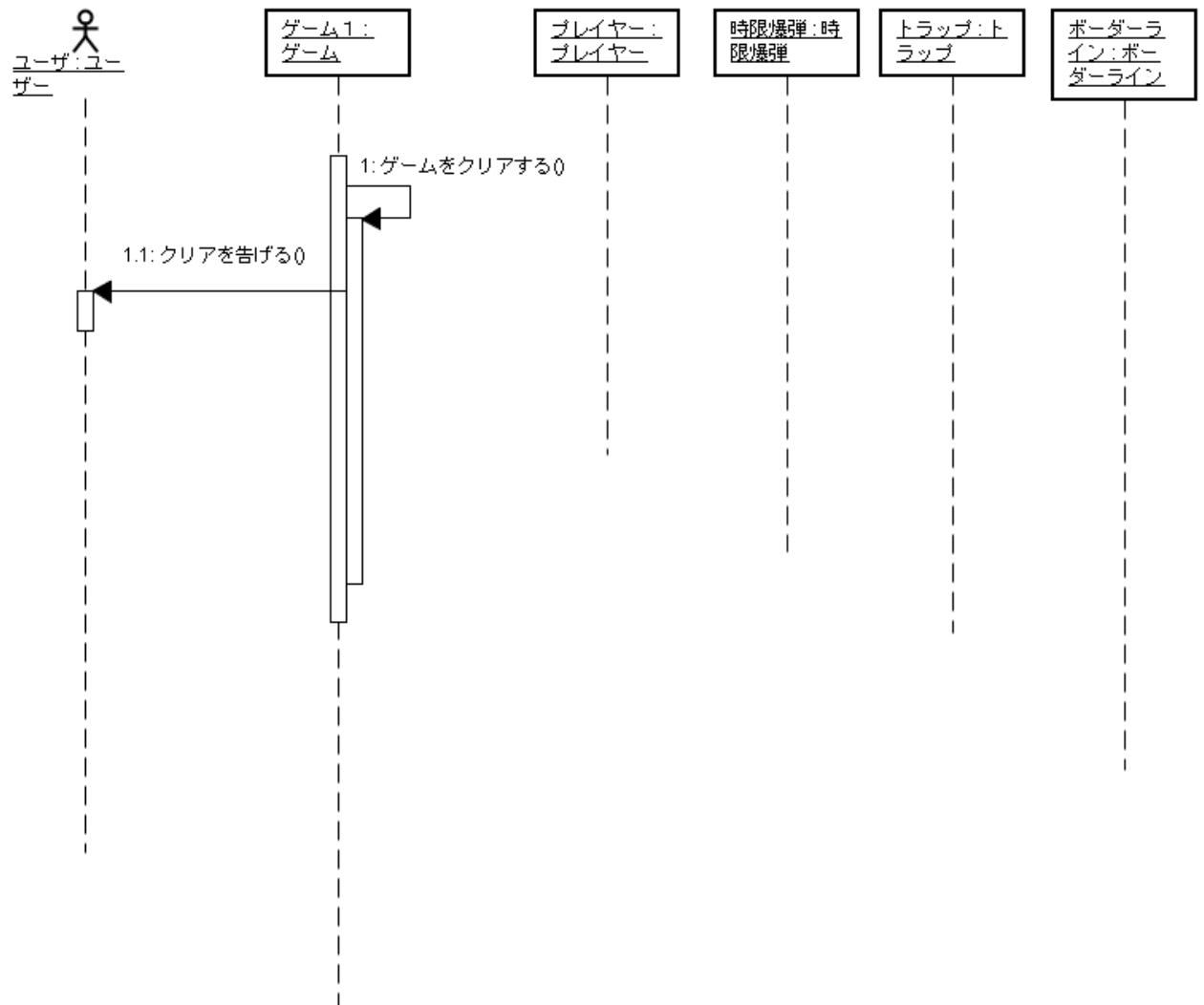


図 3.2.2.3.3.3 -5 シーケンス図 3 ゲームをクリアする



No	システム仕様			備考	
	分類	要求仕様との対応	要求事項		
1	全般	システム概要		・映像の無いゲームであること。 ・健康者と目の不自由な者の両方が同様に楽しむことのできるゲームであること。	
2		制約条件		・画面には何も表示されないこと。(ただし、製作中の段階に限ってデバッグ中の画面を表示してもよい) ・音だけでゲームの状態が理解できること。 ・サラウンド音響を用いて、ゲーム中の各オブジェクトの存在地点を音だけで把握できること。	
3		インタフェース	マニュアル「操作方法」	・入力にはマウスとテンキーによって行われる。 ・ゲーム中のキャラクターの操作は、キャラクターの移動はテンキーで行われること。 ・状態遷移の選択(ゲームの開始やメニューの表示他)はテンキーの入力によって行われること。 ・詳細な入力対応はマニュアル及び各操作時のユースケース記述を参考にすること。	
4	ゲーム制御	システムを起動する	ユースケース番号 1	・システムが起動前であること。 ・システムのアイコンをダブルクリックするとウィンドウが開き、システムは起動すること。 ・プレイヤーがシステムを起動すると、ゲームは起動した状態となること。この際、ゲームの状態はゲーム前(タイトル画面)とすること。 ・プレイヤーがシステムを起動すると、ゲームは起動した状態となること。この際、ゲームの状態はゲーム中(タイトル画面)とすること。	
5		ゲームを開始する	ユースケース番号 2	・ゲームの状態がゲーム前(タイトル画面)であること。 ・プレイヤーが入力(Enterキー)を行った場合、「ゲームを開始します」という音声と共にゲームの状態をゲーム中に遷移させること。 ・開始の際は、全てのオブジェクトの位置・状態を初期化すること。 ・ゲームを起動させた際、ゲームの状態は中断中(メニュー表示中)でありチュートリアルを流すこと。	
6		システムを終了する	ユースケース番号 3	・ゲームの状態がゲーム前(タイトル画面)であること。 ・プレイヤーが入力(0キー)を行った場合、「ゲームを終了します」という音声と共にシステムを終了した状態とすること。	
7		メニューを表示する	ユースケース番号 6	・ゲームの状態がゲーム中であること。 ・プレイヤーが入力(Enterキー)を行った場合、ゲームを中断中にし、メニューを表示させること。	
8		メニューを閉じる	ユースケース番号 6	・ゲームの状態がゲーム中で、中断中(メニュー表示中)であること。 ・プレイヤーが入力(Enterキー)を行った場合、「メニューを閉じます」という音声と共に中断を解くこと。	
9		チュートリアルを流す	ユースケース番号 6	・ゲームの状態がゲーム中で、中断中(メニュー表示中)であること。 ・プレイヤーが入力(1キー)を行った場合、ゲームの説明をするチュートリアルを流すこと。なお、ゲームを開始した際は自動的にチュートリアルを流す段階とすること。 ・チュートリアル中にプレイヤーが入力(Enterキー)を行うと、チュートリアルは終了され、中断は解除されること。	
10		ゲームを終了する	ユースケース番号 6	・ゲームの状態がゲーム中で、中断中(メニュー表示中)であること。 ・プレイヤーが入力(0キー)を行った場合、「ゲームを終了します」という音声と共にゲームの状態をゲーム前(タイトル画面)に遷移させること。 ・Ctrl+Alt+Del操作を行い、「タスクの終了」を選択するとゲームが終了すること。	
11		ゲームクリアにする	ユースケース番号 4、マニュアル「ゲームの流れ」	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと。 ・プレイヤーがゲームクリアの条件を満たした場合、「ゲームクリアです」という音声と共にゲームの状態をゲーム前(タイトル画面)に遷移させること。 ・プレイヤーがゲームクリアの条件を満たした場合、効果音を流すこと。	
12		ゲームオーバーにする	ユースケース番号 5、マニュアル「ゲームの流れ」	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと。 ・プレイヤーがゲームオーバーの条件を満たした場合、「ゲームオーバーです」という音声と共にゲームの状態をゲーム前(タイトル画面)に遷移させること。	
13		接触判定をする	ユースケース番号 4	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと ・キャラクターが移動を行い、座標が変わったこと。 ・キャラクターと他のオブジェクトが接触したかどうか、両方の座標とサイズを取得した上で計算を行ってを判定すること(ここで言うオブジェクトとは時限爆弾、トラップ、ボーダーラインを指す)。 ・判定の結果、接触が有効だと判断された場合、各オブジェクトに用意されたりアクションを起こさせること。	
14	キャラクター	音響出力する	ユースケース番号 9	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと ・キャラクターが移動を行い、一定距離動くとキャラクターの座標から「コッ、コッ」という足音を鳴らすこと。	
15		移動する	ユースケース番号 7、マニュアル「操作方法」	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと ・プレイヤーがマウスを用いて入力を行い、その入力に応じてキャラクターは動作すること。具体的な操作と行動の対応はユースケース・マニュアルを参照。	
16		描画する	-	・ゲームの状態がゲーム中であること ・キャラクターが存在する地点を確認するため、ウィンドウに位置を描画すること。なお、デバッグ用の機能である。	
17	時限爆弾	音響出力	ユースケース番号 9	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと ・時限爆弾が有効であること(解除もしくは爆破していない状態であること) ・時限爆弾が存在する地点から、キャラクターの存在する地点までの距離と角度(座標)を計算し、その地点から恒久的に「カチ、カチ」という音を発生させること。	
18		カウントする	マニュアル「ゲームの流れ」	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと ・時限爆弾が有効であること(解除もしくは爆破していない状態であること) ・設定された時間から経過した時間を刻むこと。	
19		爆発する	ユースケース番号 5	・ゲームの状態がゲーム中であること。 ・時限爆弾が有効であること(解除もしくは爆破していない状態であること) ・経過時間が設定した時間を超過した場合、爆発の効果音と共にゲームの状態をゲームオーバーとすること。(No12参照)	
20		解除する	ユースケース番号 4	・ゲームの状態がゲーム中であること。 ・時限爆弾が有効であること(解除もしくは爆破していない状態であること) ・設定した時間内に時限爆弾とキャラクターの接触判定が有効だと判断された場合、解除音と共にゲームをクリアすること。(No11参照)	
21		描画する	-	・ゲームの状態がゲーム中であること ・時限爆弾が有効であること(解除もしくは爆破していない状態であること) ・時限爆弾が存在する地点を確認するため、ウィンドウに位置を描画すること。なお、デバッグ用の機能である。	
22	トラップ	音響出力する	ユースケース番号 9	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと。 ・トラップが有効であること(起動前の状態であること)。 ・トラップが存在する地点から、キャラクターの存在する地点までの距離と角度(座標)を計算し、その地点から恒久的に「ピー」という警告音を発すること。	
23		起動する	ユースケース番号 8	・ゲームの状態がゲーム中であること。 ・トラップが有効であること(起動前の状態であること)。 ・トラップとキャラクターの接触判定が有効だと判断された場合、トラップ起動音と共に下記のトラップの効果音を発動させること。 ・トラップがワープトラップであった場合、キャラクターをフィールド内のランダムな地点に移動させること。 ・トラップが即死トラップであった場合、ゲームをゲームオーバーにすること。(No12参照)	
24		描画する	-	・ゲームの状態がゲーム中であること。 ・トラップが有効であること(解除もしくは爆破していない状態であること)。 ・トラップが存在する地点を確認するため、ウィンドウに位置を描画すること。なお、デバッグ用の機能である。	
25		音響出力する	ユースケース番号 9	・ゲームの状態がゲーム中で、中断中(メニュー表示中)ではないこと。 ・部屋の四辺の端(ボーダーラインが存在する場所)から、キャラクターの存在する地点までの最短距離と角度を計算し、その地点から恒久的に警告音を発すること。	

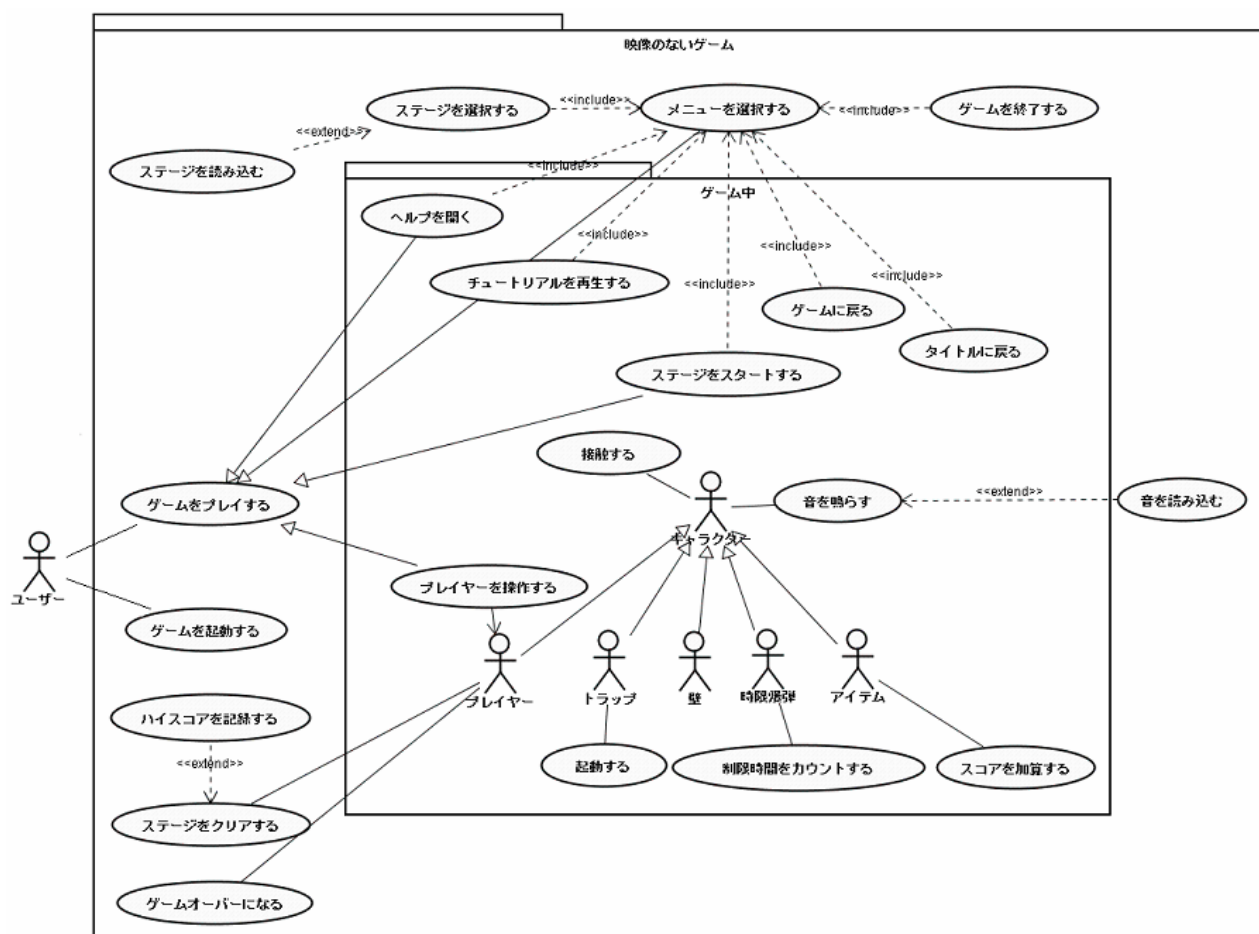
表 3.2.2.3.4.3 -1 システム仕様一覧

No	試験項目			試験手順	判定基準	試験結果	判定	備考	試験担当
	大区分 (システム仕様)	中区分	小区分						
1	ゲーム制御	システムを起動する		SoundOnlyFramework.exeを開くとシステムを起動し、ウインドウが開くことを確認する。	起動すればウインドウが開くかどうか	ウインドウが自動でひらいた	○	ウインドウ右上の×を押下することでも終了できる	渡辺
2		システムを終了する		ゲームが起動してある状態で、「タスクの終了」でシステムを終了し、「Ctrl+Alt+Del」操作を行い、「タスクの終了」を選択すると、システムは終了し、ウインドウが閉じることを確認する。	「タスクの終了」でシステムを終了し、ウインドウが閉じるかどうか	ウインドウが閉じた	○	移動する量が決まっておらず、そこまで細かい動きはできない(以下3つも同様)	渡辺
3	キャラクター	移動する	前に移動	ゲームが起動してある状態で、上キーを押すと前方(デバッグ画面上では上方)に移動するか確認する。	デバッグ画面上で上に動くか確認できること	上キーを押すと上に動いた	○		渡辺
4			右に移動	ゲームが起動してある状態で、右キーを押すと右方向(デバッグ画面上でも右方)に移動するか確認する。	デバッグ画面上で右に動くか確認できること	右キーを押すと右に動いた	○		渡辺
5			左に移動	ゲームが起動してある状態で、左キーを押すと左方向(デバッグ画面上でも左方)に移動するか確認する。	デバッグ画面上で左に動くか確認できること	左キーを押すと左に動いた	○		渡辺
6			後ろに移動	ゲームが起動してある状態で、下キーを押すと後ろ(デバッグ画面上では下方)に移動するか確認する。	デバッグ画面上で下に動くか確認できること	下キーを押すと下に動いた	○		渡辺
7		描画する		ゲームが起動してある状態のとき、キャラクターが位置する場所に四角形が描画されているか確認する。	デバッグ画面上でキャラクターの位置を把握できるものが描画されていること	四角形ではなく横線だけが描画された	△		渡辺
8			音の出力	ゲームが起動してある状態のとき、時間爆弾が絶えず音を発しているか確認する。	起動している間ヘッドフォンから音が聞こえていること	間隔をあけてではあるが常に音が鳴った	○		渡辺
9			音の変異	ゲームが起動してある状態のとき、プレイヤーが座席を変えると、時間爆弾から出ている音も変わっているか確認する。					渡辺
10	時間爆弾	音響出力	音の方向	ゲームが起動してある状態のとき、時間爆弾が位置する場所から音が出ているか確認する。例えば、爆弾が左にある時は左から音が出て、左前方にある場合は左前方から音が出ているか確認する。	デバッグ画面上でキャラクターが爆弾の周りを動いて8方向から音が聞こえていること	5方向のサウンドヘッドフォンを使用すると8方向から聞こえた	○	プレイヤーが爆弾に重なっている状態で左から聞こえる正面だと音が小さく聞こえる	渡辺
11			音の減衰処理	ゲームが起動してある状態のとき、時間爆弾が位置する距離から音が出てくる音によって例えば、近づけば近づくほど音が大きくなり、遠ざかれば音が小さくなる、というようになっているか確認する。	デバッグ画面上でキャラクターと爆弾の距離を変化させて音が変化していること	デバッグ画面上で爆弾とキャラクターの位置を離れていくと音が小さくなる	○	ある程度距離が離れると一切聞こえなくなる	渡辺
12		描画する		ゲームが起動してある状態のとき、時間爆弾が位置する場所に四角形が描画されているか確認する。	デバッグ画面上で爆弾の位置を把握できるものが描画されていること	四角形ではなく横線だけが描画された	△		渡辺

表 3.2.2.3.6.3-1 試験手順書・成績書

#### 5.4.1.1. 反復作業第二段階のユースケース図

図 3.2.4.3.2.3.-1 ユースケース図



#### 5.4.1.2. 反復作業第2回目分析のクラス図・状態遷移図

図 3.2.4.3.3.2.3.-1 クラス図

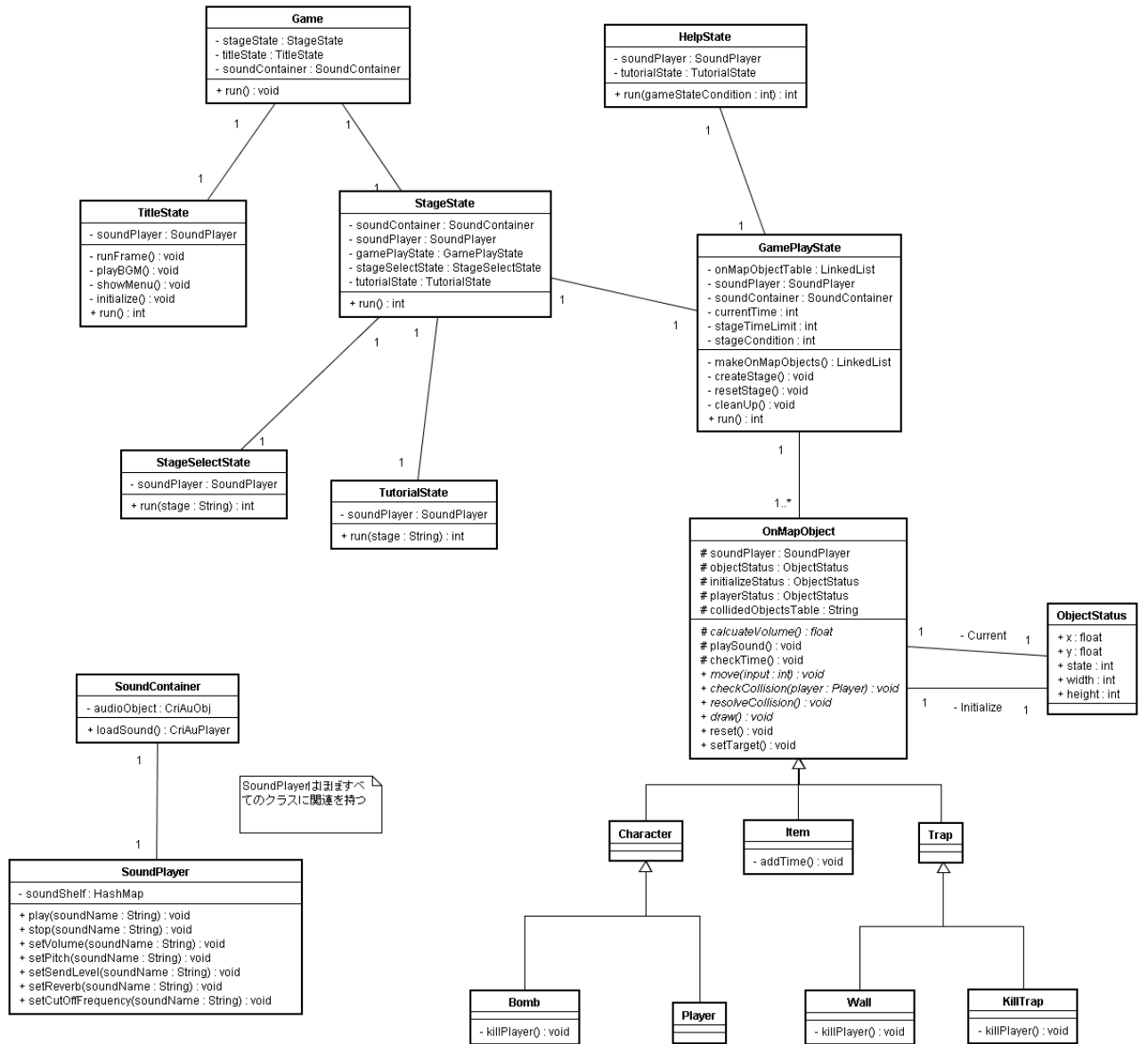
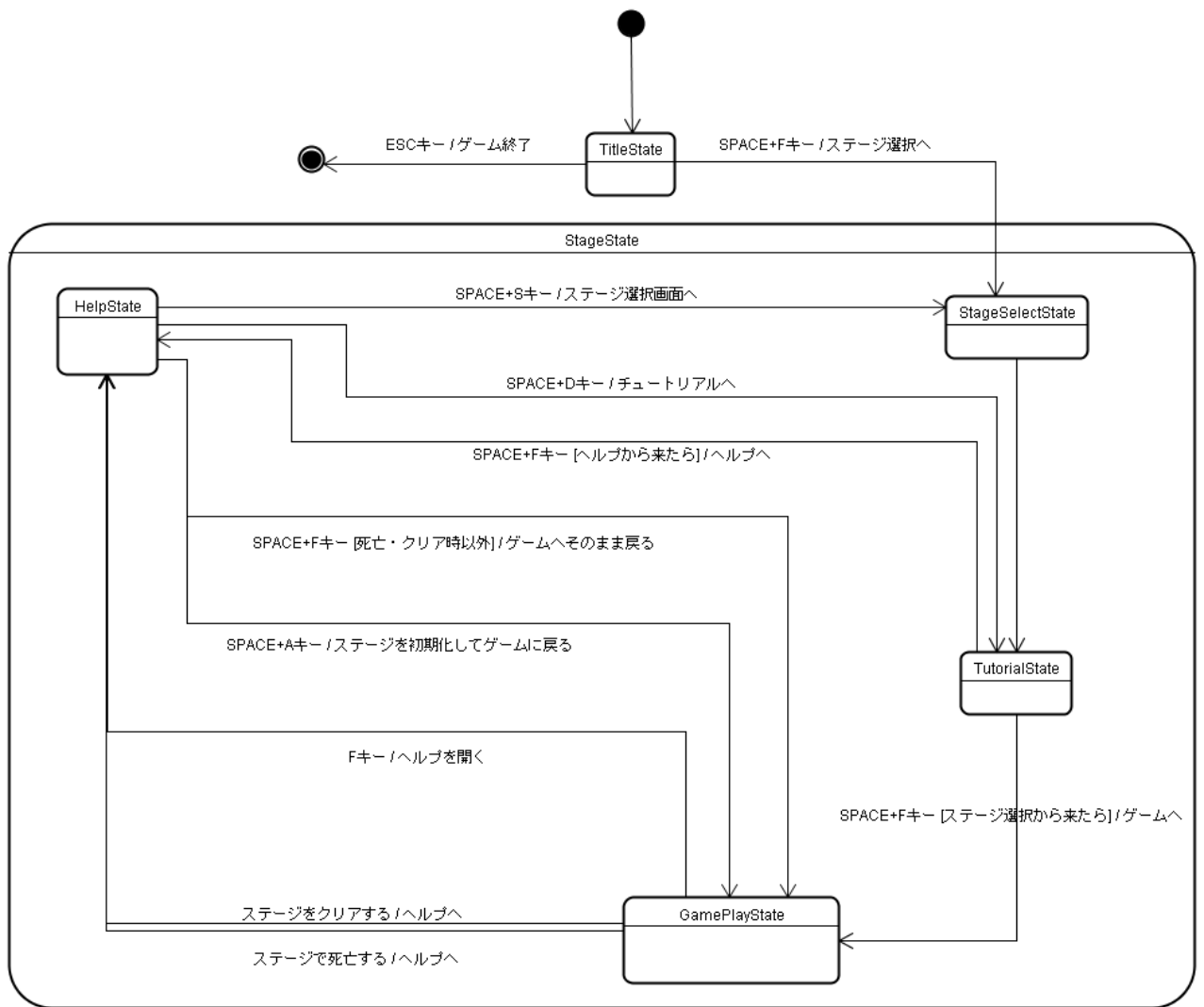


図 3.2.4.3.3.2.3.-2 反復作業第 2 回目分析の状態遷移図





## 5.5. ユーザーマニュアル

### ストーリー

時は 2XXX 年、惑星間での交流が発展している時代。

人類は地球外に文明に触れ、様々な技術を手に入れ、生活も豊かになった。

しかし、いいことばかりではなかった。

2XXX 年 月 308 日、一つのニュースが地球に伝えられた。予てから地球を敵対視する火星のテロリスト達が月を荒らしつくし、更には月内部の空洞に大量の罠と超強力な時限爆弾を仕掛けたという。

月内部は、電気系統が破壊されたため真っ暗だ。その上、超電磁波の影響で灯りを灯すことができない。爆弾処理チームは、センサーの音だけを頼りに爆弾を解除することになる。無謀かも知れないが、失敗すれば地球は滅亡だ。なんとかしなければならない。

なんで懐中電灯がダメでセンサーが大丈夫なのかは未来の技術者に聞いてください。

### ゲーム目的

あなたは爆弾処理班として爆弾を解除することを目的とします。

フィールド上にある罠を避けつつ、時間内に音だけを頼りに爆弾を探してください。

無事に爆弾まで辿りつけば爆弾は解除できます。

### ゲームの流れ

まず、ゲームを起動すると、「さうんど おんりい」というアナウンスが流れます。ステージ選択画面では、ステージナンバーを再生します。SPACE キーを押しながら A キーや S キーで好きなステージを選択してください。

ゲームが始まると、刻一刻と制限時間が近づきます。落ち着きたい場合や、チュートリアルを確かめたい場合、ゲームを中断したい場合は F キーを押すとメニューが開きます。メニューを開いている間は時間が流れないので、じっくり英気を養ってください。

カチッカチッと冷酷に時間を刻んでいる爆弾に辿り着くとクリアですが、制限時間内にたどりつけなかったり、トラップや壁に接触した場合はゲームオーバーとなります。

## 操作方法

### タイトル画面

Space を押しながら F を押して離す：ステージ選択画面へ

Esc：ゲーム終了

### ステージ選択画面

Space を押しながら S を押して離す：次のステージナンバーを選択

Space を押しながら A を押して離す：前のステージナンバーを選択

Space を押しながら F を押して離す：チュートリアルへ

### チュートリアル画面

F を押して離す：チュートリアルを流す

Space を押しながら F を押して離す：ゲーム開始

### ゲーム中

F：ヘルプを開く

：前に進む

：後ろに進む

：左に進む

：右に進む

### ヘルプ中

Space を押しながら A を押して離す：ステージリセット

Space を押しながら S を押して離す：ステージセレクトに戻る

Space を押しながら D を押して離す：ヘルプ内のチュートリアルへ

Space を押しながら F を押して離す：ゲームに戻る

### ヘルプ内のチュートリアル中

Space を押しながら F を押して離す：ヘルプに戻る

### ゲーム終了時（クリア・ゲームオーバー）

Space を押しながら A を押して離す：同じステージを始める

Space を押しながら S を押して離す：ステージセレクトに戻る

### トラップの種類

キュンキュンなる怪しい音のトラップに接触すると罠が発動します。

- ・即死トラップ...接触するとゲームオーバーになる

### アイテム

アイテムに触れると、ピンポンとなって制限時間が増えます。クリア時の残りタイムがスコアとなるのでできるだけ多くのアイテムを集めてみましょう。

### ボーダーライン

シューシューなる部屋の壁(ボーダーライン)に触れると、ゲームオーバーになります。壁も近づくと警告音を出しますので、注意してください。

### ワンポイント・アドバイス

トラップやボーダーラインは、近づくと感知器がその位置を察知して、罠のある辺りから音が聞こえるようになります。カチカチなる時限爆弾よりも小さい範囲でしか音が聞こえないので、注意して進みましょう。