

# GhostBreaker 解説

創価大学 4 年生 堀田 祐輝

2025 年 7 月 8 日

## 1 制作動機

ある時、ゲームソフトウェア業界では Unity がよく使われていると聞き、Unity と付随する C# の上達をしたいと思います。

そこで、講義の一環として作成した簡単なお化け退治のゲームを作成していたことを思い出しました。

また、もう少しゲーム性を付けるという予定もあったため、本ゲームの作成を行いました。

## 2 使用ツール

本ゲームの作成に使用した開発環境は Unity です。

理由としては、ゲームプログラマー業界では Unity がよく使われていると聞き、上達することが近道であると考えたためです。

使用言語は C# で、こちらの理由は、以前の Unity 学習で C# を使っていて、下手に C++ に変えるよりは C# を上達する方が有効であると考えたためです。

テスト環境は Windows10 で、理由としては現在作者の手に持っているのが Windows10 のみだったためです。

## 3 ゲームの起動方法

ExcuteFolder（実行ファイル）にある Obake \_\_ Project.zip を全て展開後、Obake \_\_ Project フォルダ内にある GhostBreaker.exe をダブルクリックすると、ゲームを起動できます。

## 4 ゲーム内容説明

### 4.1 概要

プレイヤーが銃と直接攻撃を使って大きなゴーストを倒すことが目的のゲームです。画像 1 はゲーム画面です。

プレイヤーはゴーストたちの攻撃を受けるとゲーム画面下部にあるライフを減らし、なくなるとゲームオーバーです。

銃は弾数が決められていて、一定の数を撃つと弾薬を拾わない限り打てなくなります。残り弾数はゲーム画面で確認することが出来ます。

大きなゴーストの他に小さなゴーストも存在し、倒すと時々ライフ回復や弾薬のアイテムを落とします。

小さなゴーストは時間と共にランダムな位置で増えていきます。増える数は上限があり、25 体以上は増えないようになっています。



図 1 ゲーム画面

### 4.2 キャラクター説明

#### 4.2.1 プレイヤー

遊ぶ人が動かすことが出来るオブジェクトです。体力が存在し、5 つのハートがなくなるとゲームオーバーとなります。

Space キーを押すことで銃弾を撃つことが出来て、敵を攻撃することが出来ます。

大きなゴースト（敵）を倒すことが目標で、大きな

ゴーストを倒すとゲームクリアとなります。  
画像 2 が遊ぶ人が動かすことが出来るオブジェクトの画像です。



図 2 プレイヤー

#### 4.2.2 敵キャラ

敵キャラには 3 種類います。全ての小さな敵キャラはプレイヤーの銃弾が直接攻撃を行うことでダメージを与えることが出来、大きな敵キャラはプレイヤーの銃弾のみダメージを与えることが出来ます。

まずプレイヤーが倒す目標である「ボスゴースト」と、「突撃ゴースト」・「射撃ゴースト」がいます。画像 3 がボスゴーストであり、画像 1 の上部にある緑のバーがボスゴーストの体力です。ボスゴーストにぶつかってもライフは減少しませんし、特に攻撃することもないですが、倒さない限りゲームクリアにならないため、必ず倒す必要があります。

画像 4 の左側が突撃ゴーストで、画像 4 の右側が射撃ゴーストです。

突撃ゴーストはプレイヤーに向かって進んでいき、プレイヤーにぶつくとプレイヤーのライフを減少させます。

射撃ゴーストは動かずに、とげとげの弾をプレイヤーに向かって発射してきます。とげとげの弾が射撃ゴースト自体にプレイヤーがぶつくとプレイヤーのライフを減少させます。

また、突撃ゴーストと射撃ゴーストは倒すとライフを回復するアイテムや弾数を回復するアイテムを落とします。



図 3 ボスゴースト



図 4 突撃ゴーストと射撃ゴースト

#### 4.3 操作説明

使う操作キーは矢印キー(↑↓←→)/WASD キー、X キー、C キー、V キー、Space キーです。

矢印キー(↑↓←→)または、WASD キーはプレイヤーの移動に使います。上下キーで前後に進み、左右キーで横に回転します。

X キーはポーズ機能を使うためのキーです。押すとゲームが一時停止され、ポーズ画面が表示されます。ポーズ画面には音量調整、操作説明、タイトル画面に戻るボタンがあります。画像 5 はポーズ画面の画像です。

C キーはカメラの切り替えを行うキーです。C キーを一度押すことでカメラの追従を一度解除して、プレイヤーの背面を映す方向へと移動します。再び C キーを押すことでカメラが追従するようになります。

V キーは直接攻撃を行うキーです。V キーを押すと、プレイヤーが正面に直接攻撃を行い、近くの敵を攻撃します。攻撃を受けた小さな敵キャラは倒れます。

Space キーは銃から銃弾を発射するキーです。銃弾は敵キャラにぶつくとぶつかった敵キャラにダメージを与えることが出来ます。しかし、銃弾は一度に撃てる回数が決まっており、弾数を回復するアイテムを取らない限り撃てなくなってしまいます。



図 5 ポーズ画面



図 7 重要度によって色を変える



図 8 ボタンが反応できるか色を変える

## 5 アピールポイント (工夫した点)

この作品は、UI を意識して作成しました。意識した理由としては、これより前に作成したゲームでのフィードバックに、ゲームが分かりにくく、難しく感じるという意見がありました。そのため、本ゲームでは様々な可能性を考えて可能な範囲でユーザーに寄り添った設計を意識しました。

### 5.1 表記において

表記上で意識した UI は、三つあります。一つ目は、ゲーム画面に文字を極力なくすことです。画像 6 は画像 1 の下部を切り抜いたものです。銃弾やライフを文字ではなく絵を交えて直感的に理解できるようにしました。また、操作方法についても、落ちて確認できるポーズ画面に移動させ、最小限にとどめました。



図 6 ゲーム画面の UI

二つ目は画像 7 のように重要度によって色を濃くして、重要なものに意識が向くようにしました。また、極力派手な色や組み合わせを避けて目が疲れにくくしました。

三つ目は画像 8 のようにボタンの上にマウスカーソルが重なると色が緑に変わるようにして、ボタンが押せるかどうか分かりやすくしました。

### 5.2 設定において

画像 9 はタイトル画面で遷移できる設定画面です。ポーズ画面の一ページ目とほとんど一緒です。設定画面ではゲームの難易度に直接影響しないが、人によっては調整したいと考えるものを調節できるようにしました。回転速度については、遅いとプレイヤーの正面以外の敵を狙いにくくなり、速いと敵に攻撃を当てにくくなり、一長一短であるため難易度には大きく影響しないと判断しました。

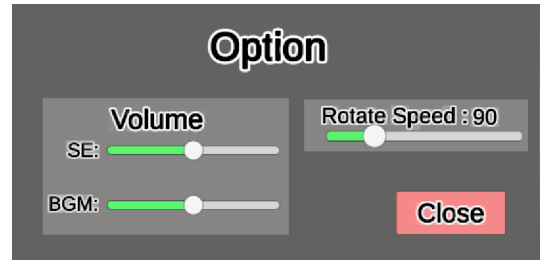


図 9 設定画面

### 5.3 そのほか意識した点

そのほか意識した点とした点として、フォームの入力においてアルファベットの他に平仮名や漢字が入力して文字を打つときに障害が起きないようにしました。画像 10 は実際にフォームに入力したものです。

また、銃の中に銃弾が残っているときと銃弾がないときで音を変えて弾切れかどうか把握しやすくなり、クリックしたときに音が出すことで反応しているかどうか理解することが出来るようにしました。



図 10 フォーム画面

## 6 苦戦した点

最も苦戦した箇所はカメラの設置及びカメラリセットです。

講義内で作成したときは、プレイヤーとなるモデルとカメラを親子関係にすることで追従させていました。

しかし、今回使用したモデルとアニメーションでは、緻密につくられていて、動くたびにカメラが揺れてしまい、思うように動けなかったり、遊んでいて非常に不快なゲームとなってしまいました。

そこで、ネットに散らばっている情報や chatGPT を用いて解決策を探し、cinemachine という機能を見つけました。

詳細な設定を学習することで、カメラを揺れなくなったり、カメラの追従を一時解除してカメラをワープさせるという方式で問題を解決しました。

この経験より、やりたいことや問題点は先に Unity を使っていた人たちの情報や経験で解決することができることを実感し、今までためらっていた機能の拡張や追加に挑戦できるようになりました。

## 7 課題点と今後の展望

現在抱えている課題点として、二つあります。

一つ目はタイトル画面の設定で音量や回転速度を変更しても、ポーズ画面のスライドの位置が反映されない点です。シングルトンの数値で管理しているため、音量や回転速度は設定画面での数値になっています。

そのため、シングルトン化した数値を反映すれば解決しますが、繁栄方法が分からず、手詰まりとなっています。

二つ目はポーズ画面のページ移動を矢印キーでも動かすことができるようにするとゲームプレイ時の移

動に反応してポーズ画面のページが動いてしまう点です。ポーズ画面ではゲーム時間を 0 にしているため、逆にポーズ画面で矢印キーを押してもプレイヤーが動くことはないためゲームには大きく影響しません。したがって、ページ移動を矢印キーで反応しなくすれば解決しますが、同時に一々クリックしないとページ移動しなくなるなり不便になるため、より良い方法が見つかるまで保留にしています。

## 8 本作品にて学習できた点

本プロジェクトを作成して、分かりやすいように改善した UI がユーザーにとってまだわかりにくかったり、意図して考えた機能が想定外の使われ方をされてしまったりして、UI を意識するということが非常に難しいことであると理解しました。

相手を意識するという意味で、アルバイトのサービスを似ていることを改めて実感しました。実際に、フィードバックを受けるときに過程をアピールしたとき、あまり反応されず、結果や成果の説明ばかり求められました。これはサービス業においてお客様が早く注文の品が来ることしか求めていることと同じであり、アルバイトの経験をもっと生かすことが出来るのではないかと考えました。したがって今後ゲームの作成するときはアルバイトの経験を思い出しつつ作成することにします。

また、Unity の基本的な機能と C# プログラムの練習になりました。具体的には、コンポーネントの効果と意味や、TextMeshPro のフォント変更、別オブジェクトの関数の起動方法、.find の使用を避ける理由などです。

## 参考文献

- [1] OArielG(2019.1.7), 「Simple Heart Health System」,  
<https://assetstore.unity.com/ja-JP/search?q=heart>
- [2] Ada\_King(2017.12.5), Free Trees,  
<https://assetstore.unity.com/packages/3d/vegetation/trees/free-trees-103208>
- [3] SR Studios Kerala(2023.12.13) 「Little Ghost lowpoly(FREE)」,  
<https://assetstore.unity.com/packages/3d/characters/lowpoly/free-little-ghost-103208>

com/packages/3d/characters/  
little-ghost-lowpoly-free-271926#  
description

- [4] IGGProduction(2024.5.21) 「FREE Guy with a Gun - Voxel Character」,  
<https://assetstore.unity.com/packages/3d/characters/humanoids/free-guy-with-a-gun-voxel-character-283468>
- [5] Google(2007.2.26) 「Noto Sans Japanese」,  
<https://fonts.google.com/noto/specimen/Noto+Sans+JP/license?subset=japanese>
- [6] Kogiso(2019) 「日本語文字コード範囲指定 (ascii・ひらがな・カタカナ・第一水準および第二水準 (JIS-X0208-1997) に含まれる漢字)」,  
[https://gist.github.com/kgsi/ed2f1c5696a2211c1fd1e1e198c96ee4#file-japanese\\_full-txt](https://gist.github.com/kgsi/ed2f1c5696a2211c1fd1e1e198c96ee4#file-japanese_full-txt)
- [7] 効果音ラボ (2013) 「演出・アニメ」,  
<https://soundeffect-lab.info/sound/anime/>
- [8] 効果音ラボ (2013) 「ボタン・システム音」,  
<https://soundeffect-lab.info/sound/button/>
- [9] 魔王魂 (2017.1.1) 「ファンタジー 11」,  
[https://maou.audio/bgm\\_fantasy11/](https://maou.audio/bgm_fantasy11/)