

BULLET★SHOOTER 企画書

バンタンゲームアカデミー
ゲームプログラマー専攻2年制2年次
井野 巧巳

基本情報

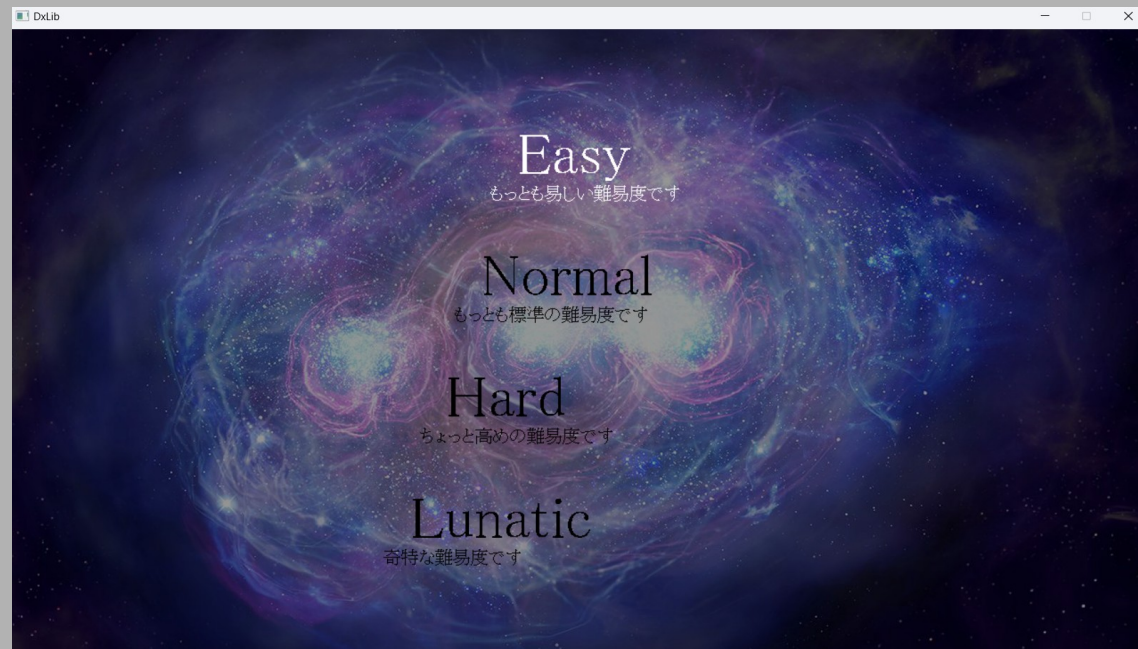
- タイトル : BULLET SHOOTER ★
- 制作者 : 井野巧巳 (一人)
 - 製作期間 : 2024 年 1 月 3 日 ~ 2024 年 7 月 1 日
 - 仕様 : Dx ライブラリ (C++) と講師提供のライブラリ
 - ジャンル : 3 D 弾幕シューティングゲーム
 - ターゲット層 : アクションゲームが好きな全ての人

ゲームの流れ



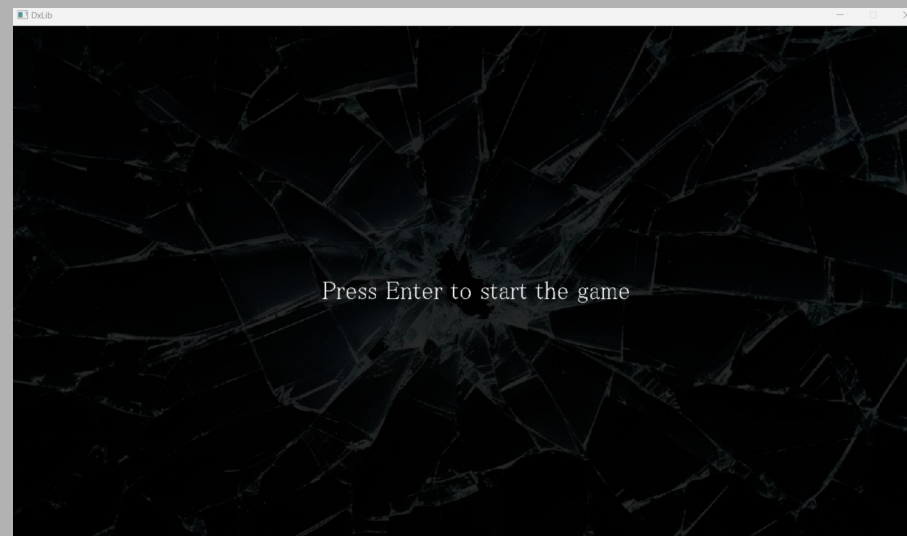
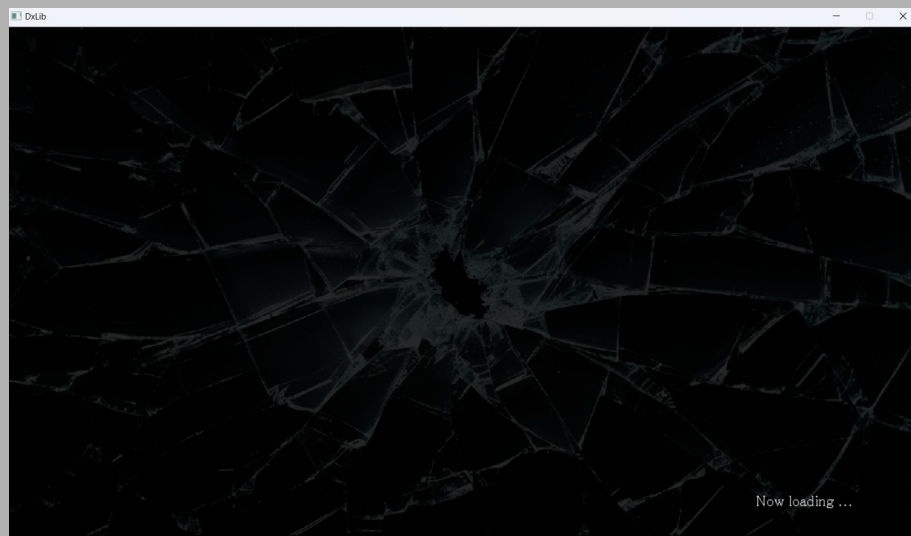
① タイトル

ゲームの流れ



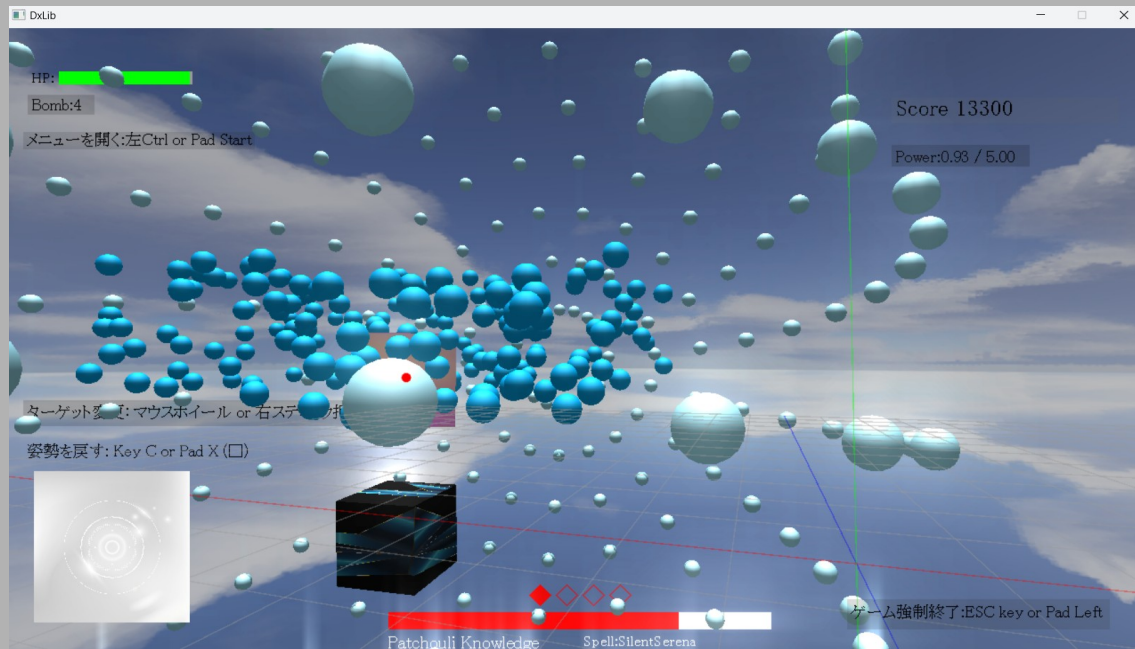
② 難易度選択

ゲームの流れ



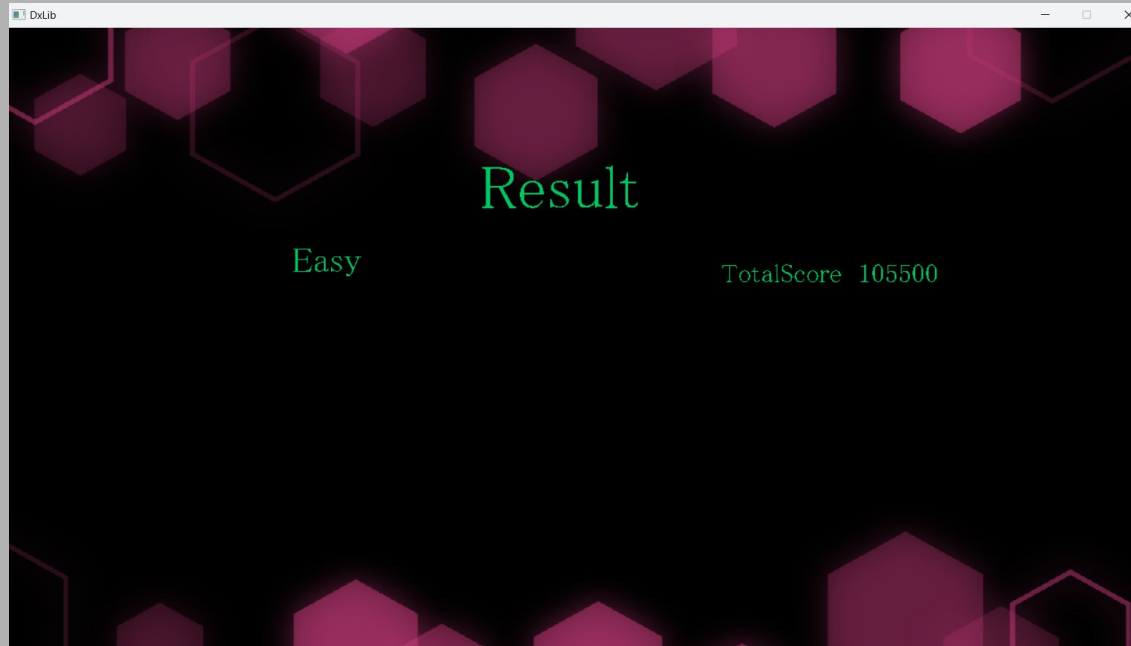
③ ロード

ゲームの流れ



③ プレイ

ゲームの流れ



④ リザルト

※ 弾幕の実装は全て「東方プロジェクト」というコンテンツを引用しています。それにより変数名、関数名には実際に引用した弾幕やキャラクターの名前を使用している部分があります。

こだわった点

多様性

1. ノーマルタイプ・ボスタイプの2種類の敵

それぞれ挙動や攻撃パターンなどが異なる

2. 弾幕の豊富なパターン

全3ステージの各ボスはそれぞれ独自の弾幕でプレイヤーを攻撃

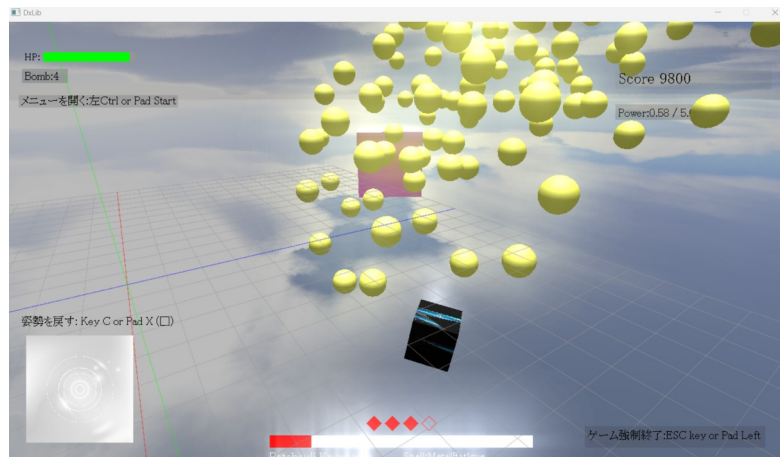
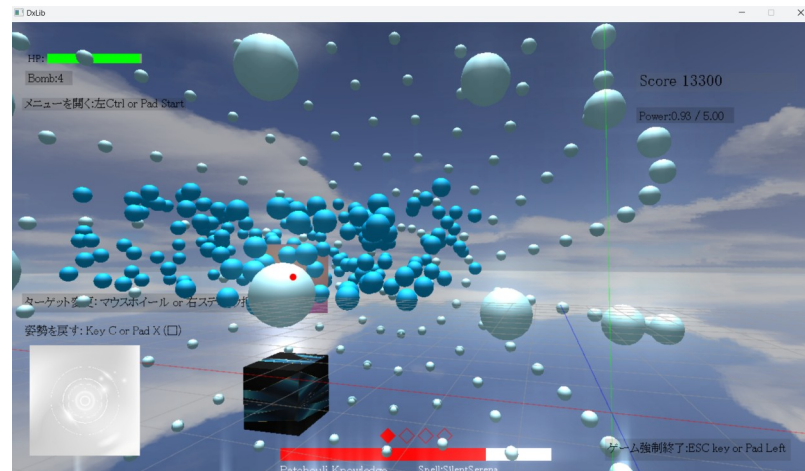
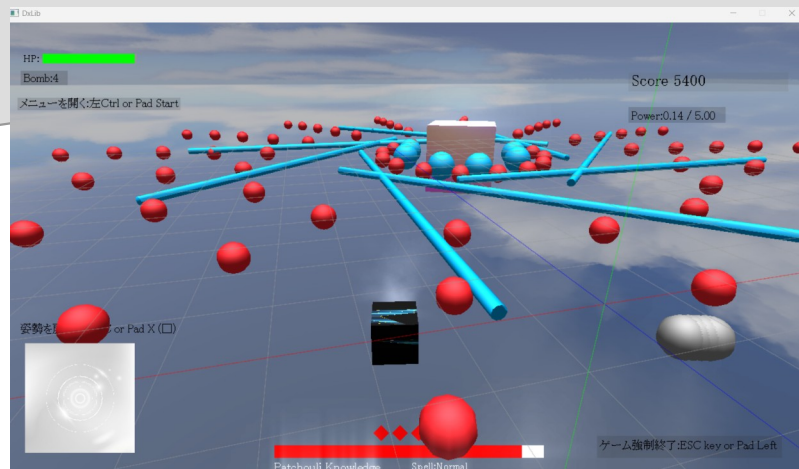
弹幕ロジックの定義

弾一つ一つに固有 I D を持たせ、独立した動きを実装

この ID の識別によって、

回転、位置、時間など、弾一つ一つに異なる情報を付与。

ステージ 1

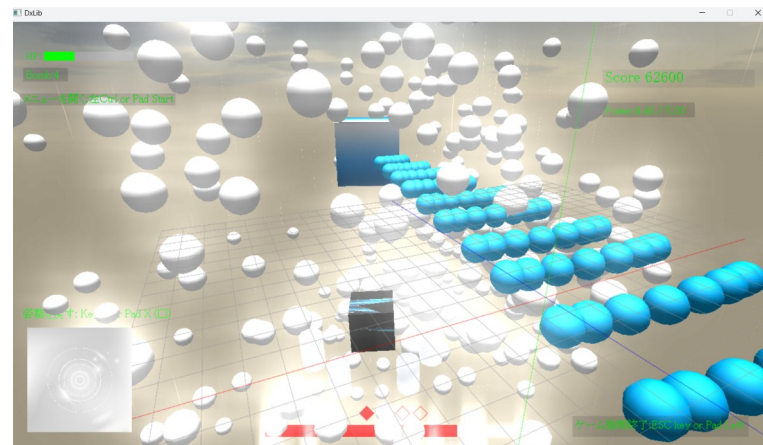
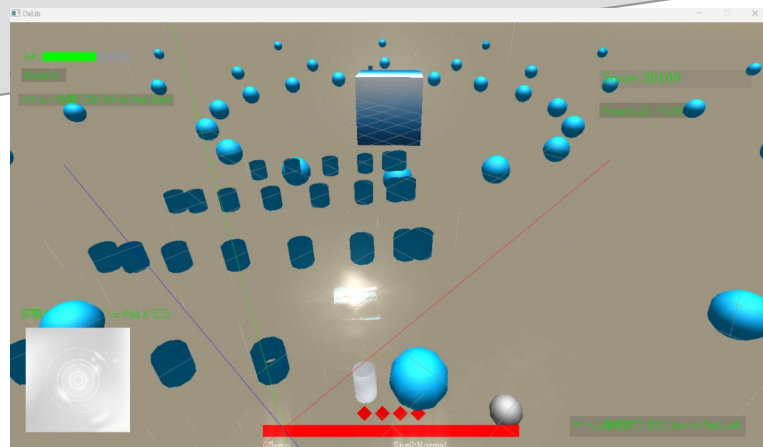


弹幕ロジック

12
10/23/24

こだわった点

ステージ 2



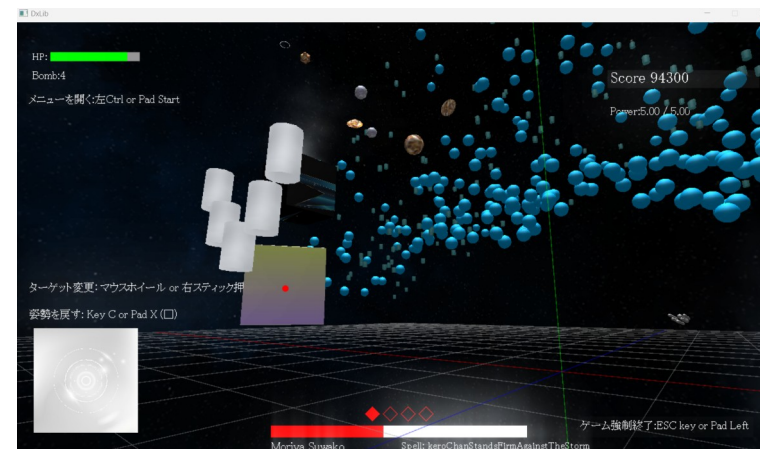
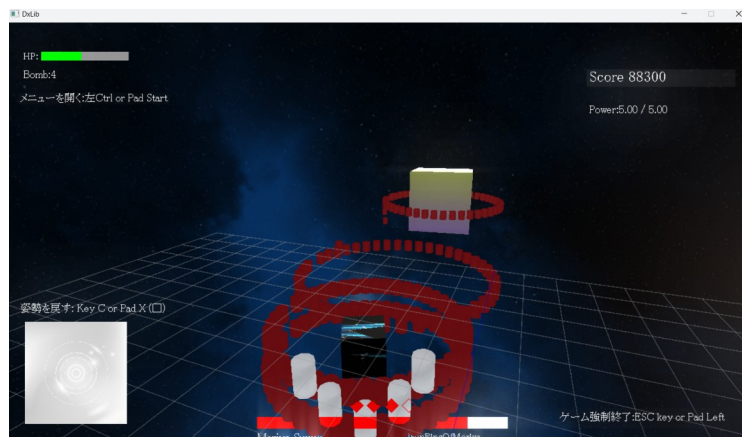
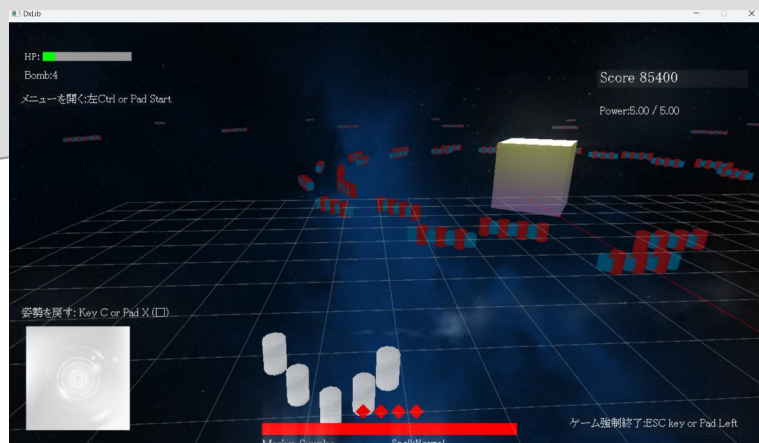
弾幕ロジック

13

10/23/24

こだわった点

ステージ 3



弹幕ロジック

14
10/23/24

こだわった点

苦労した点

1. 弾幕ロジックの実装

2. 固定カメラの実装

3. ノーマルエネミーの挙動制御

1. 弹幕ロジックの実装

こだわった点で紹介した「弹幕」に関する実装は、同時に苦勞した点でもある。

弹幕処理の実装には、三角関数を使用した複雑な計算が必要であったが、その知識が全くなかった私は何度も試行錯誤を繰り返した。

これに伴い膨大となったコードも処理の省略、共通機能の関数化などによって可読性の改善も行った。

2. 固定カメラの実装

敵座標の中央位置に表示されている赤い点が、固定カメラの焦点である。

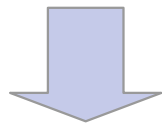


- カメラ固定中のプレイヤーの挙動

ターゲットを中心として位置回転を行う。この時のプレイヤーの移動や姿勢の制御は苦戦したポイントである。

- 例外処理対応

敵の座標はスマートポインタを使用した敵オブジェクトから取得。
しかし敵を倒してしまうと無効な参照により例外が発生。



別の敵へ焦点を動的に上書きすることで対処。

3. ノーマルエネミーの挙動制御

Moving (移動)

- ・ プレイヤー追跡
プレイヤーとの距離が一定以内の時
- ・ ランダム移動
プレイヤーに気付いていない状態の時

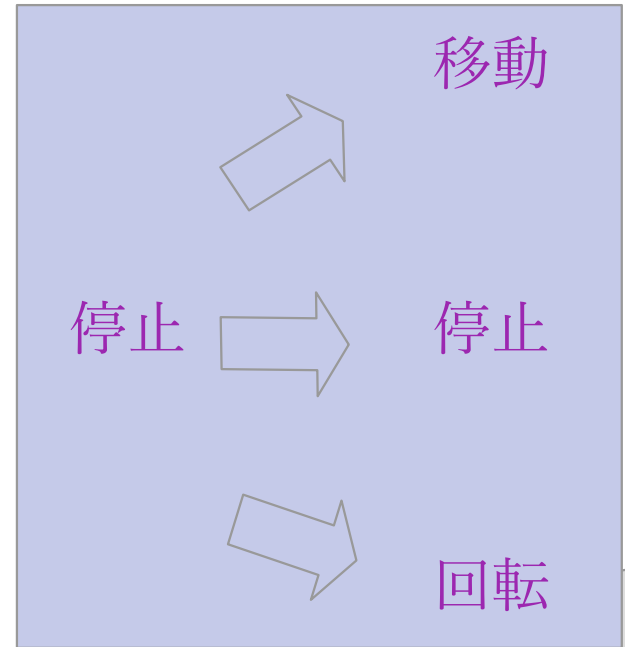
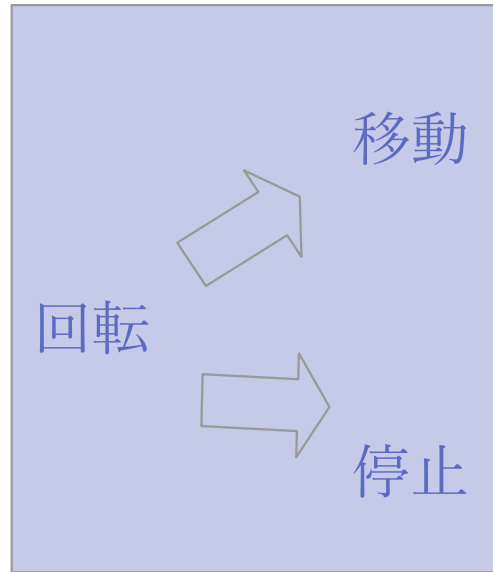
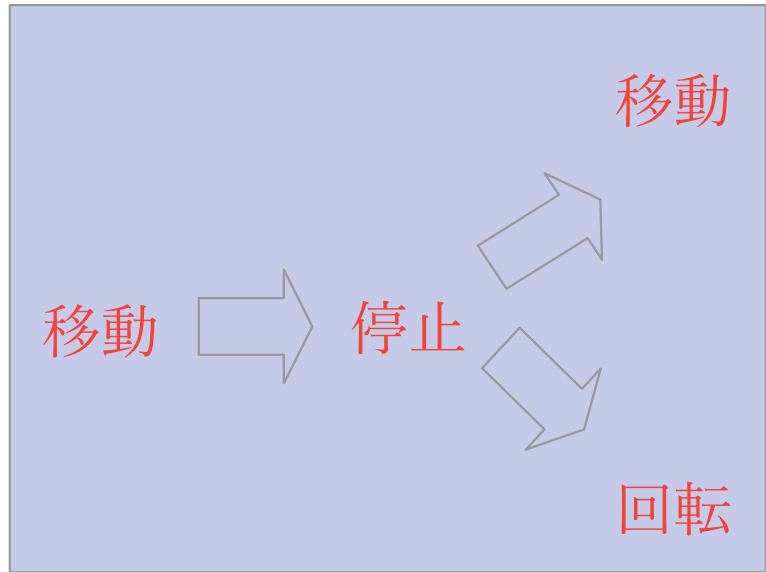
Stop (待機)

プレイヤーに気付いていない状態の時

Turn (回転)

プレイヤーに気付いていない状態の時

プレイヤーに気付いている場合の挙動は単純だが、そうでない場合



一定時間経過後
次の挙動がランダムに分岐

苦勞した点

操作方法

- ・ 移動 キーボード WASD キー、ゲームパッド十字キー
- ・ 前進 キーボード Space キー、ゲームパッド L1 、 LB
- ・ スロー移動 キーボード Shift キー、ゲームパッド△、 Y
- ・ 弾発射 マウス左、ゲームパッド○、 B

・ カメラ固定 マウス右、ゲームパッド R1 、 RB

・ ターゲット変更 マウスホイール

・ 視点操作 マウス、ゲームパッド左スティック

・ ボム マウス中央

・ ポーズ画面 キーボード左 Alt キー