

# 28班 企画書

織田武瑠, 矢野大暉, 山口力也

2019/10/28 提出

## 1 ゲーム内容

### 1.1 ゲームのタイトル

エージェント 3

### 1.2 ジャンル

脱出ゲーム

### 1.3 概略

ゲームの設定は、「犯罪組織に盗まれた金塊を組織に潜入して、複数のプレイヤーと協力して、組織的に取り返す」といった設定である。本ゲームは、同一のサーバに接続した複数のプレイヤーが協力する脱出ゲームである。プレイヤーは出入り口からスタートし、敵キャラ (NPC)、監視カメラに見つからないように、ステージ上に設置された金塊をゲットし再び敵、監視カメラに見つからないように出入り口に帰ってくるゲームである。ステージは複数用意されており、ステージが上がるごとに敵や監視カメラを増やすことで難易度を向上させる。

プレイヤーは敵キャラや監視カメラに対して妨害を行うことで、他プレイヤーが金塊を取るためのアシストを行うことができる。プレイヤーが行うことができる妨害として以下が挙げられる。

- 敵キャラに話しかける      敵の視界の固定
- 監視カメラのハッキング      監視が一時的に停止
- 敵キャラに催涙スプレー      敵の視界が無効化

## 1.4 プレイ人数

3人

## 1.5 操作方法

本ゲームはジョイパッドでの操作を行う。

移動：ジョイパッドのアナログスティック

妨害アクション：3 ボタン

金塊を取得：4 ボタン

## 1.6 画面構成

ゲームのイメージ図を図1に示す。

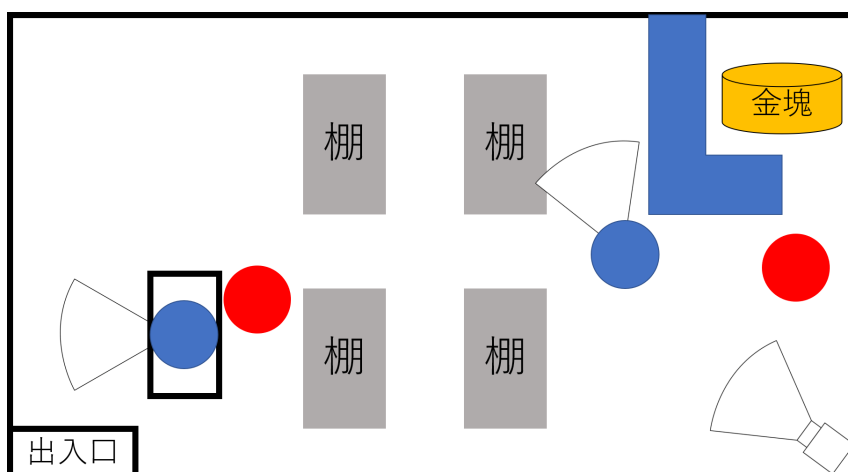


図 1: ゲームウィンドウのイメージ

赤で示しているのがプレイヤー，青で示しているのが敵キャラ (NPC)，右下隅の物体が監視カメラである。敵キャラ，監視カメラの扇形の物体は視界を表しており，この中にプレイヤーがいるとプレイヤーが「見つかった」判定となる。

## 1.7 勝敗，クリア，終了条件

クリア条件は，プレイヤーが金塊を取って出入り口へ向かうことができればゲームクリアである。ゲームオーバー条件はプレイヤーが敵キャラまたは

監視カメラに見つかってしまうことである。

## 1.8 その他の特徴

敵キャラの動きはルールベース型 AI を採用している。進捗状況によっては、強化学習なども用いる。敵キャラの AI の具体例としては以下が挙げられる。

- 敵がプレイヤーに近づく      敵がプレイヤーの方向へ向く
- 監視カメラにプレイヤーが近づく      カメラの首振りのスピードがはやくなる

また、ゲームに使用する素材はオリジナルのものを作成する。

## 2 実装方法:データの構造

### 2.1 ゲームに必要な変数，構造体

共通の構造		
データ型	変数名	内容
SDL_Rect	chararect	キャラクターの座標，高さ，幅
int	speed	キャラクターの速度
SDL_Texture*	imgaddress	キャラクターの参照する画像のアドレス

プレイヤーの構造		
データ型	変数名	内容
SDL_Bool	isgetgold	プレイヤーが金塊を持っているかどうか

敵キャラの構造		
データ型	変数名	内容
SDL_Bool	isfoundplayer	敵キャラがプレイヤーを発見したかどうか

監視カメラの構造		
データ型	変数名	内容
int	speed	監視カメラの首振りのスピード

### 3 実装方法:モジュール

#### 3.1 ゲームに必要な関数等の説明

void StartUp(void)	
関数名	StartUp
機能	SDL などの初期化処理を行う関数
引数	なし
返り値	なし

void Destroy(void)	
関数名	Destroy
機能	SDL などの終了処理を行う関数
引数	なし
返り値	なし

void PlayerSystem(void)	
関数名	PlayerSystem
機能	プレイヤーの移動などの処理を行う
引数	なし
返り値	なし

void EnemySystem(void)	
関数名	EnemySystem
機能	敵キャラの移動などの処理を行う
引数	なし
返り値	なし

void CameraSystem(void)	
関数名	CameraSystem
機能	カメラの座標などの変更を行う
引数	なし
返り値	なし

void Collision(void)	
関数名	Collision
機能	衝突判定を行う関数
引数	なし
返り値	なし

void RenderWindow(void)	
関数名	RenderWindow
機能	画面の描画を行う関数
引数	なし
返り値	なし

## 4 ガントチャート

作成したガントチャートを図 2 に示す。

	10.28~11.06 (1 週目)	11.07~11.10 (2 週目)	11.11~11.17 (3 週目)	11.18~11.24 (4 週目)
織田武瑠	ゲーム設計	ゲーム設計 <u>UI</u>	ゲーム設計 <u>UI</u>	サーバ設計
矢野大暉	ゲーム設計	ゲーム設計 <u>UI</u>	ゲーム設計 <u>UI</u>	サーバ設計
山口力也	ゲーム設計	ゲーム設計 <u>UI</u>	ゲーム設計 <u>UI</u>	サーバ設計

	11.25~12.01 (5 週目)	12.02~12.08 (6 週目)	12.09~12.15 (7 週目)	12.16~12.22 (8 週目)
織田武瑠	サーバ設計	デバッグ	デバッグ	<u>NPC</u>
矢野大暉	サーバ設計	デバッグ	デバッグ	アルゴリズムの 改良
山口力也	サーバ設計	デバッグ	デバッグ	

	12.23~01.19 (9 週目)	01.20~01.26 (10 週目)
織田武瑠	<u>NPC</u>	デバッグ
矢野大暉	アルゴリズムの 改良	デバッグ
山口力也		デバッグ

図 2: ガントチャート