



# 『5びょうごのきみと』 プログラム説明資料

東京工芸大学 芸術学部  
ゲーム学科 プログラム分野 3年  
松島宗平

# 《目次》

【1】 概要

【2】 担当箇所・仕様

【3】 本制作での目標

【4】 ソースコードの担当箇所

【5】 工夫・こだわった点





# 【1】 概要

[作品実演動画.youtube](#)

## ■ ジャンル

\* パズルアクションゲーム

## ■ 開発目的

\* GFF AWARD 2023への出場

## ■ 開発期間 (約8ヶ月)

\* 2022/5/12 ~ 2023/1/6

## ■ チーム構成 (計8名)

\* 企画2名 / デザイン3名 / プログラム3名

## ■ 開発環境

\* Unity2021.3.0f (URP)

\* VisualStudio2019、C#

\* GitHub、Visual Studio Code

## ■ 企画書URL

\* [GitHub企画書.pdf](#)

## ■ GitHubURL

\* [プロジェクトファイル](#)



## 【2】 担当箇所・仕様

[担当箇所の一部.youtube](#)

[GitHub担当箇所.md](#)

- プレイヤー（兄）
  - 幽霊の弟（残像）
- 全般

- 一部ギミック
  - \* シーソー
  - \* ヒント機能
  - \* ゴール地点を示す青い鳥

- その他
  - \* プレイヤー・カメラ間の、障害物透過機能

- インゲーム時のUI
  - \* アクションボタン
  - \* ヒントボタン
  - \* ステージクリア時のUI

- 演出
  - \* ページめくりエフェクト
  - \* エフェクト生成処理
  - \* ポストエフェクト導入
  - \* ゲームクリア
    - 特設ステージ





### 【3】 本制作での目標

- 拡張・変更しやすいコードを書く。  
→デザインパターンの使用（シングルトン、ステートパターン）
- ギミック担当者と作業の連携を取りやすいようにする。  
→プレイヤーの情報を取得・変更するためのインターフェイスの実装
- 自分もチームメイトにも読みやすいコードを書く。
- URPでポストエフェクトの導入。 ([参考サイト](#))



# 【4】 ソースコードの担当箇所

[GitHub担当コード.md](#)

- (1) Cameraフォルダ
- (2) Characterフォルダ
- (3) Controllerフォルダ
- (4) Editorフォルダ
- (5) Effectフォルダ (Scripts/)
- (6) Gimmickフォルダ
- (7) UIフォルダ
- (8) Utilityフォルダ
- (9) Effectフォルダ (Resources/)



# (1) Cameraフォルダ

スクリプトファイル	概要
<a href="#">MyCinemachineDollyCart.cs</a>	鳥をゴールへ向かってレールに沿って動かす際、不自然な傾きにならないよう調節する。
<a href="#">OccludeeBehaviour.cs</a>	3Dオブジェクトを（段階的に）半透明にする機能を提供する。
<a href="#">OccluderBehaviour.cs</a>	トリガーに接触したオブジェクトが <a href="#">OccludeeBehaviour</a> を持っていたら、その機能呼んで半透明にする。 [工夫] 単一オブジェクト、複数オブジェクト一括のどちらにも対応させた。
<a href="#">PivotBehaviour.cs</a>	A地点と B地点 を繋ぐようなコライダーを作る機能を提供する。 プレイヤーとカメラ間に障害物があるかどうか判定するための当たり判定。



## (2) Characterフォルダ

スクリプトファイル	概要
<a href="#">AfterimageBehaviour.cs</a>	弟の幽霊（残像）の挙動を管理する。Recorderクラスに記録されたプレイヤーのデータを5秒遅れで自身にセットする。 [工夫] シーソー上部にいる際、自動的にシーソーに接地するようにした。
<a href="#">BirdBehaviour.cs</a>	ゲーム開始時にゴールへ向かって飛ぶ鳥の挙動を管理する。
<a href="#">PlayerBehaviour.cs</a>	プレイヤーの挙動を管理するパースナルクラス。Update関数はここで使用する。 [工夫] IPlayGimmickインターフェイスの継承。ギミックとの情報のやり取りをインターフェイスで行う。 [担当箇所] BarrierPlayer()関数以外。
<a href="#">Player_Move.cs</a>	PlayerBehaviourのパースナルクラス。プレイヤーの移動処理を担う関数を定義。
<a href="#">Player_State.cs</a>	PlayerBehaviourのパースナルクラス。プレイヤーのステートを管理する。 [工夫] ステートマシン(有限オートマトン)の <a href="#">StateMachine</a> ジェネリッククラスを使用。
<a href="#">Recorder.cs</a>	プレイヤーの位置、回転、入力、ステートの情報を記録する。
<a href="#">RideSencor.cs</a>	プレイヤーが弟の幽霊（残像）に乗ったか、及び、プレイヤーが乗り状態に移って良いか判定する。





### (3) Controllerフォルダ

スクリプトファイル	概要
<a href="#">StageController.cs</a>	ステージ内で、他のスクリプト等から参照され易いオブジェクトを一纏めにしておく。 [担当箇所] 43 ～ 64行目（BGMの実装）以外。
<a href="#">Stage09Controller.cs</a>	ゲームクリア時演出用の特設ステージであるStage09のステートを管理するクラス。プレイヤーのオート移動、カメラワーク、エフェクトの生成などを行う。 [工夫] 列挙体でステートを設定し、ステートごとに挙動を管理した。



## (4) Editorフォルダ

スクリプトファイル	概要
<a href="#">FindReferenceAsset.cs</a>	Editor拡張。オブジェクトの参照を確認する。オブジェクトを右クリックし、そこから「参照を探す」を選択。 0個であれば、そのオブジェクトをプロジェクトファイルから削除することを検討する。 書籍を参考に作成。



ばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム

## (5) Effectフォルダ (Scripts/)

スクリプトファイル	概要
<a href="#">EffectManager.cs</a>	瞬間移動エフェクト・浮遊エフェクト等の弟の幽霊（残像）周りのエフェクトや、ゲームクリア演出用のエフェクトを生成させるクラス。AfterimageBehaviourクラス、Stage09Controllerクラスで使用する。  [工夫] シングルトンパターンのジェネリッククラス ( <a href="#">SingletonMonoBehaviour</a> ) を継承し、どのクラスからもPlay関数で、エフェクトを生成できる。
<a href="#">FloatEffect.cs</a>	残像が乗り状態ではなくなった時に、浮遊エフェクトを終了させるクラス。





## (6) Gimmickフォルダ

スクリプトファイル	概要
<a href="#">HintGimmick.cs</a>	プレイヤーがヒントエリアに入ったか検知する。ヒント（A）ボタンを押した場合、ヒントを表示させる。5秒後に消滅。
<a href="#">LadderGimmick.cs</a>	IPlayGimmickインターフェイスの使用方法をギミック担当者に説明するため、雛型のみ作成。
<a href="#">MovableObject.cs</a>	同じく、インターフェイスの使用方法をギミック担当者に説明するためのもの。プレイヤーが押し出すことのできるオブジェクトの試作。
<a href="#">SeesawGimmick.cs</a>	SeesawSencorの判定を受け、シーソーを傾ける。 [工夫] シーソーの傾き具合を列挙体でステートとして設定し、ステートごとに挙動を管理した。
<a href="#">SeesawSencor.cs</a>	シーソーの片方に、プレイヤー及び弟の幽霊（残像）が乗ったかどうか判定する。



## (7) UIフォルダ 1/2

スクリプトファイル	概要
<a href="#">BookUI.cs</a>	ページをめくるエフェクト。ステージ開始・クリア時、ステージセレクト画面にて使用する。インターネットサイトを参考に作成。 [工夫] 列挙体を使用し、インスペクタウィンドウから、用途に合わせてモードチェンジできるようにした。企画者でも容易に変更できるよう意識した。
<a href="#">PageSelector.cs</a>	めくるページ一つ一つに付けるスクリプト。BookUIクラスと連動。
<a href="#">UITessellator.cs</a>	ページの折り目（中央線）近くにあるオブジェクトにアタッチする。ページめくりに際し折り目部分に見た目を対応させる。
<a href="#">StageUIScript.cs</a>	インゲーム時に表示するUIを統括するスクリプト。ゲームの状態によってのパネルの表示・非表示などを行う。 [工夫] <a href="#">Game_State.cs</a> のInGame列挙体での状態の設計を担当者に提案。このクラスの状態毎に、UI全体の挙動を管理した。 [担当箇所] ポーズ画面、BGM、SEの部分以外



## (7) UIフォルダ 2/2

スクリプトファイル	概要
<a href="#">UI_ActionButton.cs</a>	プレイヤーのとることのできるアクションを示すUI。プレイヤーのいる場所によって異なるアクションが表示される。 [工夫] アクション内容によっては、アクション(B)ボタンの長押しに、アニメーションを対応させた。
<a href="#">UI_HintButton.cs</a>	ヒント使用可能かどうかを表示する。使用可能になった・ボタンが押された場合に、アニメーションを再生する。
<a href="#">UI_RetryPanel.cs</a>	ゲームクリア後にステージリトライを選択した場合に、ページめくり演出を開始させ、終了後にシーンを再ロードするクラス。
<a href="#">UI_WarpEffect.cs</a>	ゲームクリア時のワープ風の渦巻きの様なエフェクトを再生するクラス。





## (8) Utilityフォルダ

スクリプトファイル	概要
<a href="#">ASMB Actions.cs</a>	プレイヤーアニメータコントローラ用のStateMachineBehaviourクラス。 各アクションのアニメーションが終了した際に、プレイヤーにアクションステートから遷移よう命令する。
<a href="#">Interfaces.cs</a>	各ギミックからプレイヤーにアクセスするためのインターフェイスや、UIボタン用のインターフェイスを集めた.csファイル。
<a href="#">ManagerScript.cs</a>	実行ファイル上でEscapeキーを押すと強制終了できる。
<a href="#">SingletonMonoBehaviour.cs</a>	シングルトンパターンのジェネリッククラス。 EffectManagerスラスで使用。
<a href="#">StateMachine.cs</a>	ステートマシン（有限オートマトン）を作成するジェネリッククラス。 PlayerBehaviourクラス（主にPlayer_State.cs）にて使用。



## (9) Effectフォルダ (Resources/)

シェーダファイル	概要
<a href="#">NextPage.shader</a>	見開きの紙を一気にめくるエフェクト。
<a href="#">BurningPaper.shader</a>	紙がじわじわと燃えるエフェクト。
<a href="#">PageUI.shader</a>	見開き状態の本のページをめくるエフェクト。 インターネットサイトを参考に作成。
<a href="#">AlphaCrunch-AlphaBlended.shader</a>	炎が揺らめきを表現するエフェクト。



ばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム

# 【5】工夫・こだわった点

※ピックアップしてご紹介致します。

- (1) ステートパターン活用
- (2) シングルトンパターンの活用
- (3) インターフェイスの活用
- (4) 幽霊の弟の、5秒遅れの動き
- (5) プレイヤーが謎解きに集中できるように
- (6) 絵本の世界をどのように表現するか





# (1) ステートパターンの活用

- 仕様の追加・変更柔軟に対応できるように

- \* プレイヤーや、ゲームクリア演出用の特設ステージ、インゲームUI等
- \* [学内展示](#)前日の準備日に頂いた指摘を、展示日当日までに改善できた。

- 可読性の高いコードに

- \* ステート毎に、遷移時や更新時の処理を分けて書くことが出来るため。

- ステートマシン（有限オートマトン）

- \* [StateMachine.cs](#) / [Player\\_State.cs](#)、[PlayerBehaviour.cs](#)

- 列挙体によるステート管理

- \* [Stage09Controller.cs](#)、[HintGimmick.cs](#)、[StageUIScript.cs](#)



## (2) シングルトンパターンの活用

### ■ エフェクトを生成するクラスにて使用

- \* シングルトンのジェネリッククラスを継承させた。
- \* どこからでも簡単に生成関数（Play関数）を呼ぶことができる。
- \* ID（列挙体）と座標を渡すことで、任意の種類と生成位置を設定できる。
- \* 誰でも利用できるように、使いやすさを意識して実装した。

### ■ 該当ソースコード（エフェクト生成関係）

- \* [ジェネリッククラス] [SingletonMonoBehaviour.cs](#)
- \* [ 継承先のクラス ] [EffectManager.cs](#)
- \* [ 関数使用先クラス ] [AfterimageBehaviour.cs](#)、[Stage09Controller.cs](#)





### (3) インターフェイスの活用

- [Interfaces.cs](#)のIPlayGimmick

- \* 各ギミックからプレイヤーの情報を取得・更新するためのもの。
- \* プレイヤーが何のアクションを取ることができるか等の、情報の更新を行う。
- \* ギミック担当者との作業の連携の効率化を図る。



- GitHubでのコンフリクトを防ぐことができた。

- [Interfaces.cs](#)のIButtonComponent

- \* [UI ActionButton](#)、[UI HintButton](#)共に必要になる機能をまとめた。





## (4) 幽霊の弟の、5秒遅れの動き

### ■ 実装方法

- ① [Recorderクラス](#)で、0.005秒おきにプレイヤーの情報を記録。
- ② [AfterimageBehaviourクラス](#)で、ゲーム開始5秒後からRecorderクラスのデータを、0.005秒おきに自身にセットする。

### ■ 兄の動きを正確に再現するために (コンセプトに当たる部分)

- ① プレイヤーのステートを、列挙体としてRecorderクラスに記録。
- ② [AfterimageBehaviourクラス](#)にもステートを反映させ、アニメーションステートマシンの遷移を制御する。

→ アニメーションの遷移を再現する事で、全く同じ動きに見せる。



## (5) プレイヤーが謎解きに集中できるように

### ■ プレイヤーカメラ間の障害物透過機能

\* ギミック攻略時、視界を遮るものを一部透過

\* [OccludeeBehaviour.cs](#)、[OccluderBehaviour.cs](#)、[PivotBehaviour.cs](#)

### ■ 兄と幽霊の弟が重なった場合、弟を徐々に透過

\* 画面の煩雑さを軽減 → ストレスなく考え込める

\* [AfterimageBehaviour.cs](#)

### ■ インゲームUI (アクションボタン、ヒントボタン)

\* 今、何ができるかを表示。 / 長押しにアニメを対応させた。

\* 仕様の提案から実装まで行った。

\* [UI\\_ActionButton.cs](#)、[UI\\_HintButton.cs](#)





## (6) 絵本の世界をどのように表現するか

### ■ ステージ開始時

- \* ページめくりエフェクト → ページフェードアウト
- \* 絵本世界にのめり込んでいく感覚を表現

[ページめくり演出]

[BookUI.cs](#)

[PageSelector.cs](#)

[UITessellator.cs](#)

[PageUI.shader](#)

### ■ ステージクリア時

- \* ページフェードイン → ページめくりエフェクト
- \* 一章分読み終えて現実へ引き戻され、余韻に浸る感覚を表現

[演出管理]

[StageUIScript.cs](#)

[UI\\_RetryPanel.cs](#)

[UI\\_WarpEffect.cs](#)

### ■ ステージ選択画面 (シンプルなページめくり演出)

### ■ 演出方法について自ら提案し、実装しました。





最後までご覧頂き、ありがとうございました。



ばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム SPAMすばむスパム