

「3D卓球（3DTableTennis）」設計案

作成目的

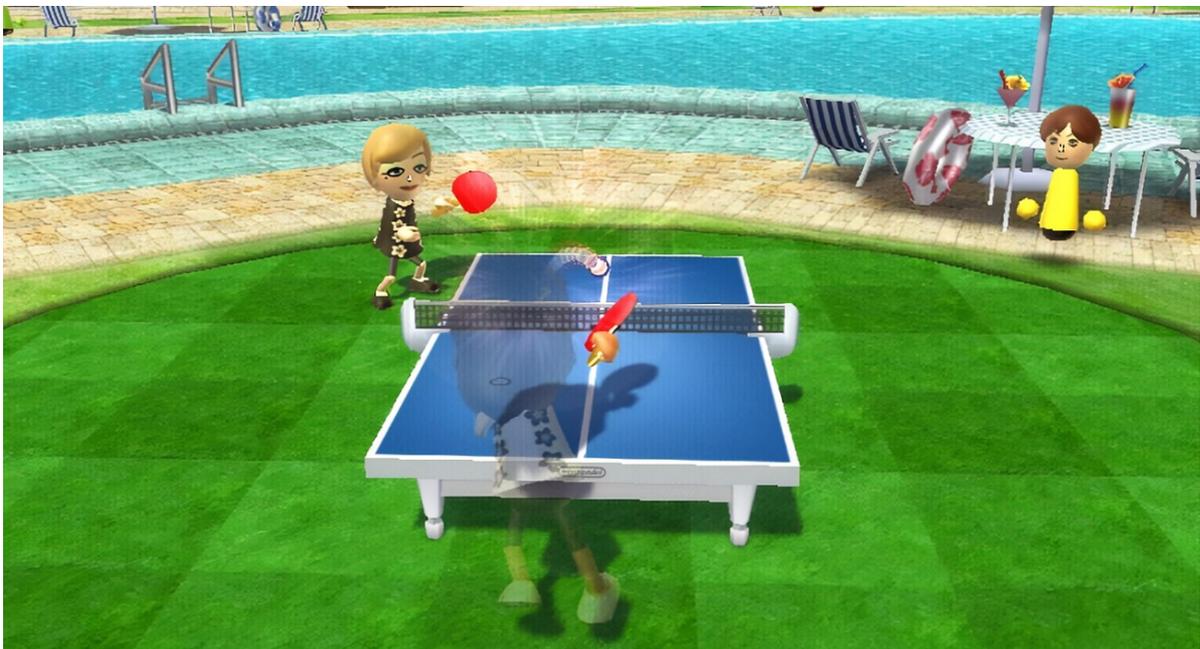
- 「大乱闘あおぎりブラザーズ」「千代浦蝶美」「3Dテトリス」のリファクタリングのアウトプットをするため（「各クラスの疎結合化」「クラスの継承」など）

概要

タイトル	3D卓球（3DTableTennis）
視点	3人称（FPS）
ジャンル	スポーツ
ゲームモデル	「Wii Sport Resort」の卓球
プレイ人数	1人
ゲームオーバー条件	エネミーが先に11点取ること
ゲームクリア条件	プレイヤーが先に11点取ること
製作期間（予定）	1週間以内
ルール	<ul style="list-style-type: none">11点先取1セットマッチ（デュースなし）コートチェンジなし最初のサーブは必ずプレイヤーからサーブは2バウンドではなく、1バウンド（レシーブと同じ）

<p>ゲームサイクル</p>	<ol style="list-style-type: none"> 1. タイトル画面 2. 「Game Start」ボタンを押す 3. 試合画面へ遷移し、試合開始 4. どちらかが勝ったら試合終了 5. 負けたらゲームオーバー画面へ遷移 6. 勝ったらゲームクリア画面へ遷移 7. 各画面で「Restart」ボタンを押す 8. 「1」へ遷移（以降ループ）
<p>備考</p>	<ul style="list-style-type: none"> • プレーヤーのキャラクターは右利き • プレイヤーもNPCも、ドライブしか打てない • 台上プレイという概念は存在しない • ボールの当たり判定は、ドライブ時のラケットにしかない • カメラの角度とプレーヤーのキャラクターの角度は常に同じ（Fortniteのように試合中は、プレーヤーは自分のキャラクターの顔を見れない） • バックハンドドライブでもフォアハンドドライブでも、ボールは必ずプレーヤーのキャラクターの向いている方向に飛んでいく

ゲーム画面モデル



操作方法

移動

WASD

視点操作	マウス
バックハンドドライブ	左クリック
フォアハンドドライブ	右クリック

スクリプト

スクリプト名	処理内容	備考
GameData	<ul style="list-style-type: none"> ゲーム全体のデータの管理 	<ul style="list-style-type: none"> シングルトン
RacketController	<ul style="list-style-type: none"> フォアハンドドライブの動き バックハンドドライブの動き 基本の構え 	<ul style="list-style-type: none"> 各ラケットにアタッチ
ControllerBase	<ul style="list-style-type: none"> PlayerControllerとNPCControllerの共通処理 	<ul style="list-style-type: none"> PlayerControllerとNPCControllerの親クラス
PlayerController	<ul style="list-style-type: none"> 移動 バックハンドドライブ フォアハンドドライブ 	<ul style="list-style-type: none"> CharacterControllerを使用 ControllerBaseクラスを継承 プレイヤーのキャラクターにアタッチ
OwnerType	<ul style="list-style-type: none"> 所有者の種類 	<ul style="list-style-type: none"> enumのみ
BoundPoint	<ul style="list-style-type: none"> コート所有者の取得 仮想の跳ねる位置の取得 	<ul style="list-style-type: none"> 各コートにアタッチ
EnemyController	<ul style="list-style-type: none"> 移動 ドライブ 	<ul style="list-style-type: none"> ControllerBaseクラスを継承 王ネミーのキャラクターにアタッチ
BallController	<ul style="list-style-type: none"> 挙動制御 	<ul style="list-style-type: none"> ボールにアタッチ

ScoreManager	<ul style="list-style-type: none"> ● 得点の管理 ● 審判 	
UIManager	<ul style="list-style-type: none"> ● ゲームスタート演出 ● ゲームオーバー演出 ● ゲームクリア演出 ● 得点の表示・更新 ● メッセージ 	
GameManager	<ul style="list-style-type: none"> ● ゲームスタート処理 ● ゲームオーバー処理 ● ゲームクリア処理 ● 試合の進行 	
SoundDataSO	<ul style="list-style-type: none"> ● 音のデータを管理 	<ul style="list-style-type: none"> ● ScriptableObjectクラスを継承
SoundManager	<ul style="list-style-type: none"> ● 音の再生 ● 音の停止 	<ul style="list-style-type: none"> ● シングルトン

実装手順

1. GitHubとSourcetreeのリポジトリの作成 (9/24に完了)
2. アセットのインポートと、ステージ・ラケット・キャラクター等の作成 (9/24に完了)
3. GameDataを作成 (9/24に完了)
4. RacketControllerを作成 (9/24に完了)
5. PlayerControllerを作成 (9/25に完了)
6. Table.csを作成 (9/25に完了)
7. OwnerType.csを作成 (9/25に完了)
8. BoundPoint.csを作成 (9/25に完了)
9. BallControllerを作成 (9/26に完了)
10. EnemyControllerを作成 (9/26に完了)
11. ControllerBaseの作成 (9/26に完了)
12. ScoreManager.csを作成 (9/26に完了)
13. UIManagerを作成 (9/29に完了)
14. GameManagerを作成 (9/29に完了)
15. SoundDataSOを作成 (9/30に完了)
16. 音のインポートと設定 (9/30に完了)
17. SoundManagerを作成 (9/30に完了)

18. ~~音をゲーム上に配置~~ (9/30に完了)
19. ~~バグの修正~~ (9/30に完了)
20. ~~自主リファクタリング~~ (9/30に完了)