


ポートフォリオシート

氏名	畑中 龍之介	所属	東京コミュニケーションアート専門学校
作品名	ぎゃ〜あ〜み〜	作品画像	
ジャンル	ミニゲーム		
プラットフォーム	WEB PC		
開発環境	Unity		
使用言語	C#		
制作期間	2022年8月〜現在		
チーム人数	プログラマー：4人 デザイナー：3人		

■概要
現在制作途中のプロジェクトになります。 1体の親と複数の子分からなる軍団、アーミーを操作してプレイします。 敵のファミリーの親を倒すと敵の子分が自分の子分になります。 制限時間内にどんどん敵を倒して、自分の軍団を大きくしていくゲームです。

■アピールポイント
・ディレクションを行っていました。学生最後の作品として力を入れています。 ・プログラマリーダーとして、ゲーム部分の設計を担当しています

■担当箇所
プログラム ・キャラの動きに関する処理 MoveCommonBase.cs MovePlayer1.cs MovePlayer2.cs ・キャラのオブジェクトの作成 CharacterBase.cs：キャラの基底クラス Henchman.cs：子分のクラス Parent.cs：親のクラス Factory.cs：キャラの生成をするクラス ・神クラスの設計 FamilyManager.cs：キャラのデータを保存する

■こだわりのコード
○キャラクターのクラス設計について ・クラスの説明 CharacterBase.csファイルのCharacterBaseクラスであたり判定を行います。 ・こだわりの関数：hitAllyParent,hitAllyHenchman,HitEnemyParent,HitEnemyHenchman,hitWildHenchman この関数は仮想関数で、あたり判定の際にCharacterBaseクラスで呼び出されます。 ・仮想関数にした訳 あたり判定の変更をしやすくする。 ・ワンポイント 処理内は「自分がどうなるのか」の処理のみすることで、オブジェクト同士の依存を極力減らしています。 ○神クラスの設計について ・クラスの説明 FamilyManager.csのFamilyManagerクラスは参照先にオブジェクトを渡すことが出来ます。 ・こだわりの変数：_henchman_list,_henchman_obj_list Dictionaryを使って親と子を紐づけました。 1つの親に対して子分が複数いる設計になるので、子分をキーにして親をコンテンツにしました。 ○親の動きの設計 ・クラスの説明 MoveCommonBase.csを継承しているクラスのMovePlayer1とMovePlayer2などを作成し親のオブジェクトにアタッチする仕組みにしています。 ・動きだけ分けた訳 親と子分のオブジェクトは自分に対して味方か敵かだけを見えています。そのため、動きを変えるだけでプレイヤーかどうか区別できます。 また、敵の動きのパターンを簡単に増やすことが出来ます。

■制作について
○目標：Steamでの配信 ・2D、3D両方で遊べるように 3Dと2Dで分けることで、低スペックPCでも動くようにし多くの人に触ってもらえるように工夫しています。 ・2Dから制作する理由 2Dデザイナーが多いので見た目をこだわれる2Dから制作し、後々新規チームメンバーを入れて制作する予定だからです。 ○設計について ・参照しやすい設計 接触時、誰に当たったか判断するために相手の情報を参照する必要がありました。そうするとGetComponentを多用する可能性あったのでFamilyManagerというクラスですべての親と子分の情報を持ちます。このクラスを参照することで、親と子分が気軽に双方のデータを参照できるようにしました。 ・3D化を見た設計 2Dから3Dに変えるにあたって座標が変わってしまうので座標を変更する処理は別クラスに分けています。 ○今後の目標 ・コードレビューを頂いた際に、Switch分の多用による可読性の低下を指摘された。 下記サイトを参考に条件分岐をへらす工夫をしていきたいと思います。 https://qiita.com/developer-kikikaikai/items/98c78 ○読みやすいコーディングルール 変数 ・変数の単語間は「_」で区切る。 ・グローバル変数とメンバ変数には頭に「_」をつける。 ・定数はすべて大文字（列挙の該当） 関数 ・一文字目は小文字 ・1関数1機能 ・1関数50行以内 ・始めの単語は動詞 ・Bool型を返す時は「is」から始める。 クラス、構造物 ・名前の一文字目は大文字 その他 ・一列80文字以内 ・()の内側に空白を入れる。例：if (true) ・ネストは3つまで

■今後の展望
・11月の学校のイベントで作品出展 ・Steamでの配信 ・3D化

■GitHubのURL	https://github.com/MoAI-199/LastProject
-------------	---