

Monster Coliseum モンスターコロシウム

《モンスターコロシウム》説明資料

制作者：

何 兆祺（か ちょうき、“Jacky” Ho Siu Ki）

日本工学院八王子専門学校

ゲームクリエイター科四年制 ゲームプログラマーコース

2020 年 3 月 卒業見込み

作品概要

ジャンル：	3D アクション
使用プログラミング言語：	C++
開発環境：	DX ライブラリ (https://dxlib.xsrv.jp/index.html)
制作期間：	約 6 か月 (2019 年 1 月～6 月) (フレームワーク 1 か月 + ゲーム内容 5 か月)

制作動機

この作品を制作した理由は、私はゲームエンジンに頼らずに C++ で 3D アクションゲームの制作に挑戦したかったからです。

私は 3D アクションゲームが好きで、そのプレイヤーキャラクターの挙動制御に非常に興味を持っています。専門学校に在籍する 4 年間、3D ゲーム制作に必要なスキルを身につける目標を立ちましたが、最初の 2 年は C++ と数学の知識が足りなかったため、Unity の機能を利用して仕様を実装することに止まりました。

3 年の後期になってから、本格的に C++ を扱い始め、かつ Unity で 3D 計算の経験も積んでいるので、是非とも自作のフレームワークを組み上げて、それを利用して 3D ゲーム制作に挑戦しようと考えました。如何にゲームエンジンに頼らずに、自作のスク립トでプレイヤーキャラクターを制御できるのが今回の制作の課題となります。

心かけた部分

- プレイヤーキャラクターのアクション表現、特にモーション遷移の滑らかさ（各モーションの遷移タイミングは手作業で調整している）
- 複雑な処理を分割し、関数化することで、保守性の向上を目指す
- 各種変数、パラメータに適切な名前を付けることで、可読性の向上を目指す
- 勉強を兼ねてポストエフェクトシェーダー（ブルームシェーダー）を実装してみた

操作説明

ゲーム패드操作



Lスティック	キャラクター移動
Rスティック	カメラ回転
Y ボタン	攻撃（連打で連続攻撃）
Y + B ボタン	ジャンプ攻撃（攻撃後に Y 入力で連続攻撃に繋げる）
A ボタン	ステップ回避（普段は前方向のみ。攻撃後なら方向指定が可能）
RB ボタン	ガード（ガード中に Y ボタン：ガード攻撃）
Start ボタン	ポーズ
Back ボタン	ゲーム終了

キーボード操作

[W] [A] [S] [D]	キャラクター移動
[↑] [↓] [←] [→]	カメラ回転
[Space]	攻撃（連打で連続攻撃）
[B]	ジャンプ攻撃（攻撃後に Y 入力で連続攻撃に繋げる）
[N]	ステップ回避（普段は前方向のみ。攻撃後なら方向指定が可能）
[M]	ガード（ガード中に Y ボタン：ガード攻撃）
[Y]	ポーズ
[T]	雑魚戦をスキップ
[Esc]	ゲーム終了

ゲームの流れ

1. ゲーム開始
2. ステージに出現する雑魚敵 3 体を倒す
3. 雑魚敵が全滅すると、ボスが登場する
4. ボスを倒すと、ゲームクリアとなる

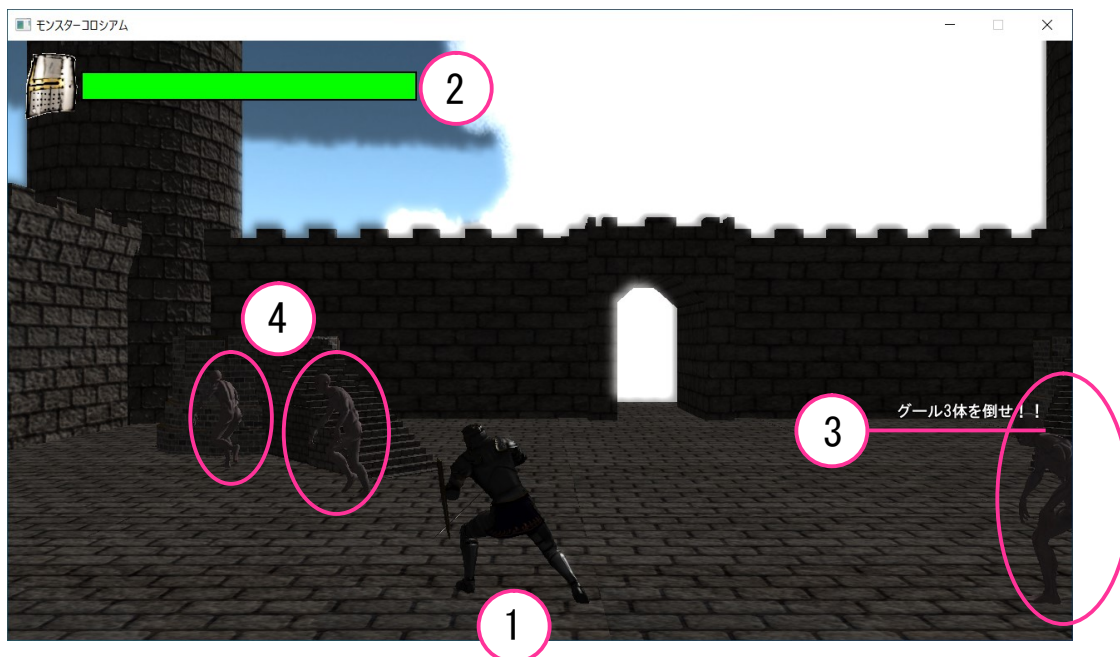
画面説明

タイトル画面



- ゲームパッドの Start ボタン、またはキーボードの Space キーを押すと、ゲーム開始

プレイ画面



1. プレイヤーキャラクター
 2. 体力ゲージ（力尽きるとゲームオーバーとなる。攻撃を受けなくて一定間隔に自動回復する）
 3. 現在の目的
 4. 敵キャラクター
- ゲームパッドの Start ボタン、またはキーボードの Y キーを押すと、ゲームが一時停止する
 - ボス戦の時、画面の下側にボスの体力が表示される
 - キーボードの T キーを押すと、雑魚戦をスキップできる

良かったと思うところ

- プレイヤーキャラクターのモーションは滑らかで、市販のゲーム顔負けの水準になっていると思う
- できるだけコードを分かりやすく書いたのと、コメントも沢山書いてあるので、仕様の追加、変更やデバッグはやりやすいと思う
- ゲームの進行に影響を及ぼすバグもなく、アプリケーションとしては安定していると思う
- 製作の間に修得した技術を作品に取り込んでいき、ステップアップしていることを実感できた
- ゲームエンジンに頼らずに作品を完成できて、いい経験になると思う

反省点

- 現状ボス敵のコライダーは胴体のみで、頭部、尻尾にもコライダーを付けたい
- プレイヤーの攻撃も、できればモデルのボーンの方に当たり判定を付けたかった
- イメージに合った回避モーションを見つからなかった。また攻撃命中のエフェクトも欲しかった
- カメラの回転処理の仕様上（ヨー、ピッチで回転している）、カメラリセットに相性が悪いらしく、カメラリセットを実装できなかったのが残念
- 敵の行動パターンのバリエーション、行動ルーチンがもっと凝った方がいいかと
- 進行状況管理の仕方、およびプログラムの設計には改善の余地がある。本格的にプログラム設計の勉強をするつもりだ（デザインパターンなど）

感想

今回の制作に通じて、自分のプログラミングスキルの上達と、ゲームエンジンの有難みを実感出来ました。

今までの作品は主にUnityで制作していたのと、C++自体はC#より不親切な部分があるので、制作はやや大変でした。特にベクトルや変換行列の計算など、Unityを使う時にあまり気にしなかった処理を自力で作るとき、不具合は避けられませんでした。例えば、キャラクターの回転処理を実装するとき、仕組みに合ったにもかかわらず、補間処理にバグがあって、担任の先生に相談してもなかなか解決できませんでした。3日間悩んだ結果、ようやく該当のクラスにコードミスを見つけて、何とも言えない気持ちになりました。こうして制作中にどんどん新しい課題や問題に直面し、解決していくうちに、スキルの上達と作品の完成度が上がっていくと実感出来ました。

今回の制作で一番楽しかったのは、自由にゲームの内容を決められることです。学校内のチーム制作はゲームコンテストに参加する都合上、テーマが限られがちからです。普段の授業とチーム制作のスケジュールは結構厳しくて、どうしてもこちらの制作に手を回せない時もありますが、作品を仕上げるため、休日の時間まで利用して制作を進めました。また、学校に行われた作品講評会にも参加し、企業側から作品の改善点や就活のアドバイスを頂いて、とても勉強になりました。この制作に得た経験は、これからの仕事に活かしたいと思います。

最後に、いつも指導してくれる担任の先生と、テストプレイをして貴重なコメントをしてくれた友人たち、そして作品講評会に貴重なご意見を下さった企業の方々に、感謝の気持ちを申し上げます。