作品ドキュメント

河原電子ビジネス専門学校

ゲームクリエイター科2年

菅原　龍之

**作品Run Run Running(個人製作)**

**〇操作方法**

\* Xobx360のコントローラーのみ動作確認済み

・左スティックでPlayer移動。

・Aボタンでジャンプ。

・Bボタンでダッシュ。

**〇作成期間**

・2016年10月～2017年2月末

**〇アピールポイント**

・マリオのBダッシュ後の慣性を再現するため、スーパーマリオブラザーズ２を何度もプレイしました。

・前作でゲーム部分を担当したので今回はエンジン部分から作成しました。グラフィックプログラムに挑戦し、デプスシャドー、ブルーム、キューブマップ、リムライトを実装しました。

影＋ブルーム＋リムライト　　影＋ブルーム　　　　　　　影のみ



・前作ではメモリが足りずにゲームが途中で終了してしまう部分がありましたが、今作ではメモリを確保した部分の解放を徹底しました。また、同じモデルデータの使いまわしなどをしてメモリ使用量を削減しました。1.6GB使用していたのですが、0.6GBまで削減できました。また、もともとは手動でデリートを呼んでいたのですが、デリートの呼び忘れを防ぐため、スマートポインタを使用するようにしました。

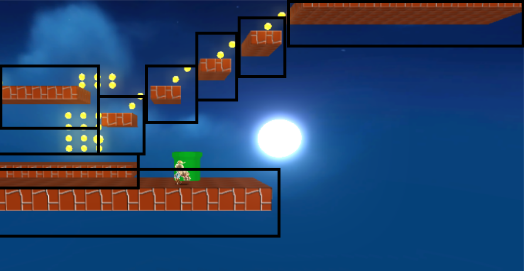
・Unityのエディタ拡張でステージのオブジェクトを設置、あたり判定を作成できるようにしました。

前作での操作が難しい点をふまえ今作では簡単に操作できるようにしました。実際に製作途中のゲームを色々な人にプレイしてもらい改善点などを見つけ初心者から楽しめるゲームを心がけて作成しました。

・モーフィングに挑戦して、フェイシャルアニメーションを実装しました。それによりボスを撃破した時プレイヤーが笑顔になるようにしました。

**〇今後の予定**

・プレイヤーの死亡は落下中の秒数で判定しています。そのためステージ２の高低差のあるステージで不具合がでているので、ステージに箱のあたり判定を複数追加し、その判定からはずれ数秒経過すると死亡するようにしたい。



←例の画像。

黒線の四角があたり判定

そして死亡した地点からのリトライを作成したい。

・ボックスを壊した時のエフェクトを物理エンジンを使用したものに改善したい。

・CPUでのモーフィングの実装を現在しているので、GPUで実装したい。

**〇担当ソースコード**

この作品は個人で作成したものなので大部分のコードを自分で書いています。

授業で使ったサンプルを参考に作成したものもありますので、下記に列挙します。

・Animation.cpp

・EffectManager.cpp

・Pad.cpp

・Physics.cpp

・SphereCollider.cpp

・SoundEngine.cpp

・SoundSource.cpp

・WaveFile.cpp

・CharacterController.cpp

\*CharacterController.cppはサンプルを元に作成し、自力で天井のあたり判定を追加。