作品ドキュメント

●マグネット

・作品概要

動作環境：

PC(2台)、インターネットのつながる環境、コントローラー必須。

プレイ人数：

２人。

ゲーム内容：

磁力を駆使して、マップ上部にあるゴールを目指す、協力型ゲーム。

・ゲームの説明

ゲームスタートまで：

本ゲームは、インターネットを利用した協力型のゲームになるので

２プレイヤーの準備が整うまで、ゲームシーンに遷移しません。

PC２台、両方でゲームを起動することでゲームシーンに遷移します。

操作説明

ゲーム起動後、ゲームパッドの「スタートボタン」で

操作一覧が表示されます。

・開発

開発人数：

３人。

開発期間：

２０１９年１２月～２０２０年９月。

使用ツール：

VisualStudio2017、DirectX11、Github、3dsMax2020、effekseer、photoncloud。

使用言語：

C++、HLSL。

・技術的にチャレンジした点

①ネットワークの完全同期

　Photon Realtimeを使って、通信を作成しました。

　１フレーム内で、自分が操作しているパッドの情報をバッファリング

して送信し、相手のパッドのバッファリングデータが受信されるまで、

ネットワークの更新のみを行って、バッファリングデータの受信をまつ

ことでパッドの情報の足並みをそろえています。

そのバッファリングデータを使ってゲームを進行させることで、

完全同期を実装しています。

　また、PC間でPlayer1とPlayer2の更新関数を呼び出す順番が

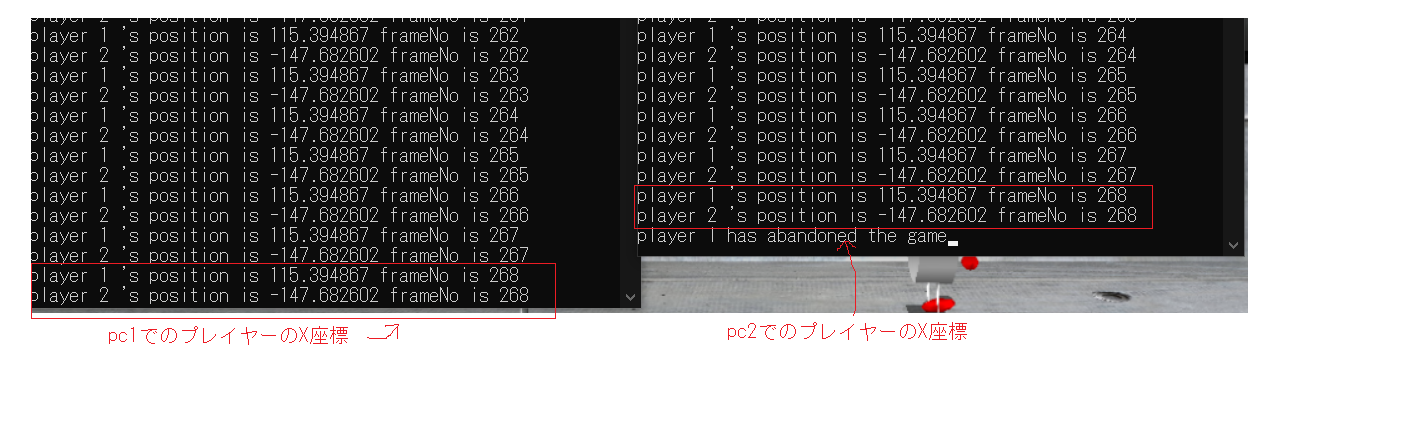
異なると、結果が変わってしまうのでUpdateの順番をPlayer1

から必ず呼び出されるようにしました。

　同期ずれの問題点の発見方法として、printf関数でコンソールに

プレイヤーの位置、パッドの入力状態、Update順番等を出力させてデバッグを行いました。

コード→U22/GameTamplate/Game/TwoP\_Pad.cpp::Update()



②エフェクトの再利用

　エフェクトの再利用を可能にするため、リソースマネージャーを作成

しました。

エフェクト再生時に、エフェクトのハッシュでリソースマネージャーに検索をかけて、リソースマネージャ上にハッシュが登録されていれば、

そのエフェクトを戻り値として返します。

登録されていなければ、そのエフェクトにハッシュを付けて登録し、

作成することで再利用を可能にしました。

　再利用することにより、パフォーマンスが向上しました。

コード→ U22/GameTamplate/Game/Effect/Effect.cpp 27~46

U22/GameTamplate/Game/ResourseManager.h

③キャラコンの改造

　横と下の判定ができていたキャラコンに、上の当たり判定を追加。

上に当たっても、上から当たってきても違和感のないコリジョン解決を実装。

さらに、カプセル状のキャラコンしかなかったので新しくボックス状のキャラコンも追加。

これも、同様に全方位から衝突しても違和感のないコリジョン解決をします。 ④磁石の挙動

このゲームの肝となる磁石の動きを現実のようにかつ、ゲームとして楽しめるように、再現しました。

近づけば近づくほど強く、遠ければ遠いほど弱く磁力が働きます。

数が増えれば影響が大きくなり、間にいる磁石は２つの磁力の影響を受けた結果をきちんと反映します。