

●タイトル：Amida

●ジャンル：3D あみだくじ

●制作期間：4 か月

●開発環境

- ・ハードウェア：Windows(i7、メモリ 32GB)
- ・開発用OS：Windows10
- ・統合開発環境：Microsoft Visual Studio 2022
- ・開発言語：C, C++, HLSL
- ・ライブラリ：directX11,FBXSDK

●操作方法

入力デバイス：マウス、キーボード

SPACE：ジャンプ

E：アイテム使用

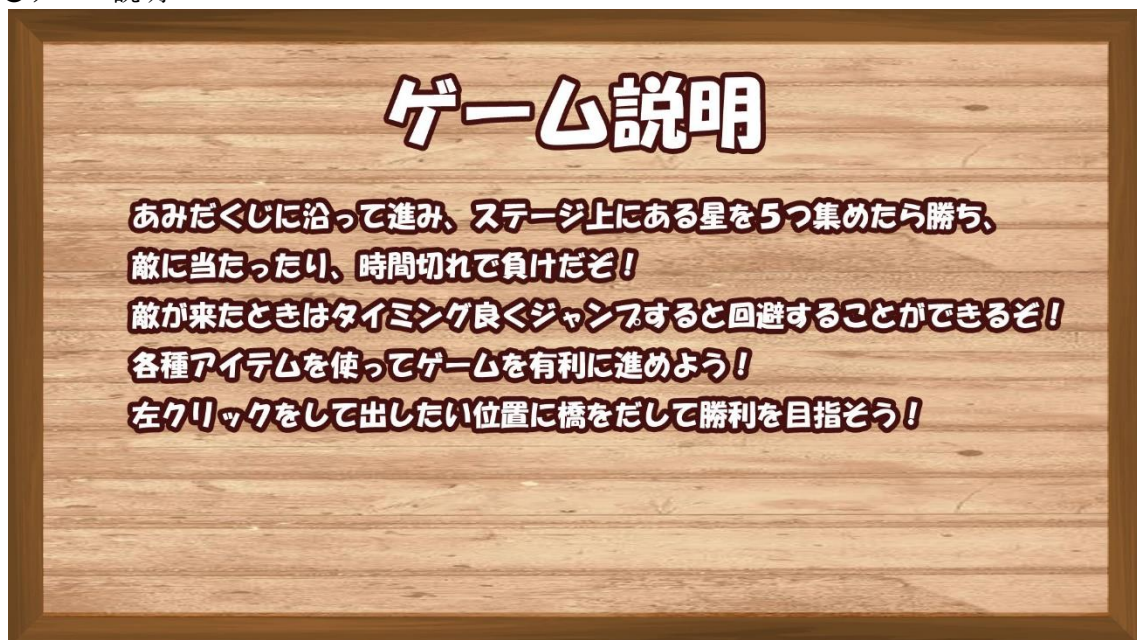
tab：カメラ切り替え

左クリック：木の板を出す

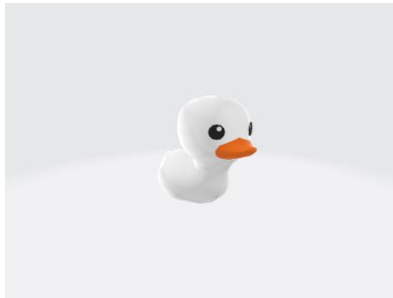
右クリック長押し：カメラ回転

ホイール：カメラ上下回転

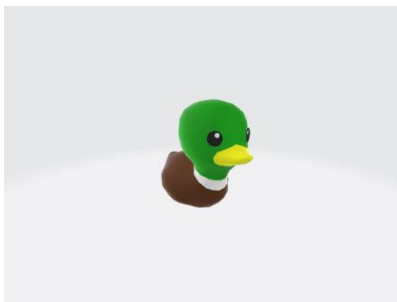
●ゲーム説明



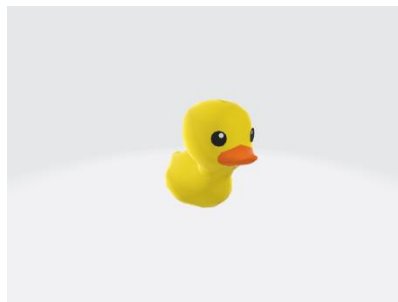
●敵



白：プレイヤーを追いかける



緑：ランダムなマスを目指す



黄：プレイヤーとの距離が 30 以下なら追いかける

●アイテムの種類



・プレイヤーの周りを回る玉(敵に当たると倒せる)



・爆弾(敵に当たると倒せる)



・羽（移動速度が一定時間速くなる）

● こだわり

- Engine/Astar.cpp

→Astar 法によってキャラクターを追う敵を実装した

- Enemy/EnemyBase.cpp

→EnemyBase を継承して EnemyRed,EnemyBlue クラスなどを実装した

→Enemy の移動に線形補間を使用してスムーズに動くようにした

- Assets/Shader/Normalmap.hlsl

→シェーダーでノーマルマップを実装した

- Assets/Shader/WaterShader.hlsl

→uv スクロールによって海の流れを表現した

- Assets/Shader/OutLine.hlsl

→トゥーンレンダリングを実装した

- Stage/stage.cpp

→橋を出す位置が分かりやすいように透明の橋を表示させた

- Engine/PolyLine.cpp

→polyline によってエフェクトを実装した

- Player/CharacterBase.cpp

→あみだくじに沿って進むキャラクターAI を実装した

- Engine/Global.h

→XMVECTOR から XMFLOAT3 の変換、距離の計算などのよく使用する関数を実装した

- Manager/GameManger.cpp

→GameManger でゲームの進行を管理した。

→星の位置のランダム生成をして規則性をなくした

- Observer/ResultObserver.cpp

→Observer パターンを Result に実装した

- Engine/BillBoard.cpp

→ビルボードを使ったエフェクトにこだわった

- Engine/Image.cpp

→画像クリックやフェード処理などのよく使う機能をクラスに追加した

- Engine/Model.cpp

→ステージに同じステージを何個も用意する為、フライウェイトパターンを使用した

- Engine/CsvReader.cpp

→ステージの情報を csv に保存し、コードの可読性を意識した

- 定数や変数などの命名規則を意識して組んだ

- モデル、UI の統一感にこだわった