|  |  |  |  |
| --- | --- | --- | --- |
| クラス番号 | 3E01SP | 氏名 | 井上颯騎 |
| 進捗報告：  プロジェクトの作成。Gitでの管理を開始。<https://github.com/KoreanGinseng/CreativeWork_M>  シーンマネージャーを作成。各シーンの管理、フェードの演出導入。  アセット管理の作成。  テクスチャや、音楽、フォントなどのアセットの管理をし、一度読み込めばどのファイルからでも呼び出し可能にした。  操作方法がマウスのみなので頻繁に使うことになるボタンクラスの作成。  ウィンドウのサイズ変更、最前面に表示/解除する、タイトルバーの表示/非表示処理の作成。  プロジェクトの変更。移行。  企画書の書き直し。  MidiInputの取得。MidiOutputの作成。  Midiファイルからデータを取得するクラスの作成。  Midiファイルの取得データからノーツ情報を生成。  ノーツ情報から譜面データを作成。  譜面データをゲームシーンにて描画。  落下速度を計算してサイズを変更後にノーツを落下。  ノーツの先頭が鍵盤の上部に達した時点での判定を作成。  各Hit情報を作成。リザルト画面で結果を描画。  SMFのトラックごとに譜面を作成し、プレイする譜面を選択できるようにした。  自動演奏の仕様をコンボカウントしないように変更した。  チャンネル別に音を再生する、音色を変更する部分を作成中。  ノーツの落下速度をBPMによって変更したいが計算式がわからず悩み中。  現在の落下速度はBPMには依存していないが、BPMによってノーツの長さは変わるようになっている。  各画面に遷移するためのキーの操作方法を描画。 | | | |
| MidiOutでチャンネルごとに音色を変更、音を出力できるように変更した。  トラックデータから追加で音色データを取得すようにした。  トラックデータごとにデータを保存するようにデータ構造を変更した。ここでの変更が大きかったので時間がかかってしまった。  前回悩んでいた計算は解決したがまだ実装していない。  鍵盤の範囲を設定して範囲外に振ってくるノーツは自動で演奏するように設定できるようにした。  演奏中以外のトラックは自動で演奏するようにした。  ボタンクラスの追加を行った。  アセット管理クラスにフォントを管理できるように追加した。  タイトル画面から選択画面へボタンを使って遷移できるようにした。  ボタンクラスの追加に伴い、マウスのみで画面操作ができるように  設定シーンと選択シーンにボタンを追加した。  トラック別音色変更のテスト用にクラシック曲のMidファイルを追加した。  TODOの整理。コメントの追加。  ピアノロールの色を変更していた部分を  キーボードの押されている情報から演奏中のトラックのなっている音の情報に変更。  これにより、AutoPlayでもピアノロールが光るように。  FallSpeedを曲ごとにjsonファイルで管理するように変更。  演奏できるトラック番号をjsonファイルで管理するようにデータを追加。  Title、Select、GameシーンにGUIを追加。  pngやmp3の追加によりResourceフォルダ内の整理。  ボタンクラスのテクスチャを設定した場合の描画方法の変更。  ボタンが押されたときに透明度64の白をかぶせて押したというアクションをみえるようにした。  テクスチャの追加により、アクセス用のキー名とファイル名を列挙した。  スコアの実装をした。判定結果別でスコアが変動するだけ。コンボでの加算などはなし。  トラック名を取得した場合にトラック名がない曲の場合エラーになっていた問題を修正した。 | | | |
| リザルト画面のUI作成  設定画面で必要のない項目を削除  設定画面のUIを作成  jsonファイルで音色を管理するように変更。  設定データの保存、読み込みを追加  読み込み中の画面を作成 | | | |