

- ・ とりあえずゲームシーンがあって一曲遊べる
- ・ とりあえず前作の曲を流用する BPM は 130
- ・ BPM の情報はファイルからロードする
- ・ BPM はコンポーネントの引数で決める感じで
- ・ キャラクタやノーツのアニメーションは基本的に横田さんが作成したものを使う
- ・ UI の挙動や UI の特殊なアニメーションは文さんが行う

作業リスト

・ 清水

<ノーツの制御、ノーツエンティティ作成>

- 譜面の読み込みを行う。ファイルから、ジャストの判定、飛んでくる向き、画像の種類、ノーツの音価の長さ(4 分音符とか 8 分音符とか)BPM が読み込めて出現できるシステムを作る。譜面の書き方の仕様はこれにのっっていけば自由
- ノーツの挙動を作成

・ 文

<UI 系エンティティの作成>

- 曲の終了時間や進行度を表すタイマーを作成
- マーカーUI を作成
- スコアゲージ UI の挙動を作成
- スコアのパーセンテージ UI 作成
- UI の挙動を作る
- ポーズ用 UI の作成
- 画像リソースの作成主に(彩色)

・ 横田

<アニメーション系コンポーネントの作成>

- BPM から小節やノーツの長さを割り出す汎用クラスを制作。
- これは画像のアニメーションでも使用できるようにしてほしい

参考 URL: <https://qiita.com/tyanmahou/items/ba127b022d2781cc3e36>

- BPM に応じてアニメーションを行うコンポーネントを作成
- 普通にアニメーションするコンポーネントの作成
- 画像リソースの作成主に(線画)

・ 佐藤

<音系と管理>

- 追加効果音(主にリザルト、新ノーツ用)
- p 版以降分の作曲(ゲーム BGM, タイトル画面, メニュー画面)
- 作業管理(常時)
- コードレビューとマージ