・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とっていれば自由

-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