

ブロック崩し

「メディア情報学プログラミング演習」

グループ3

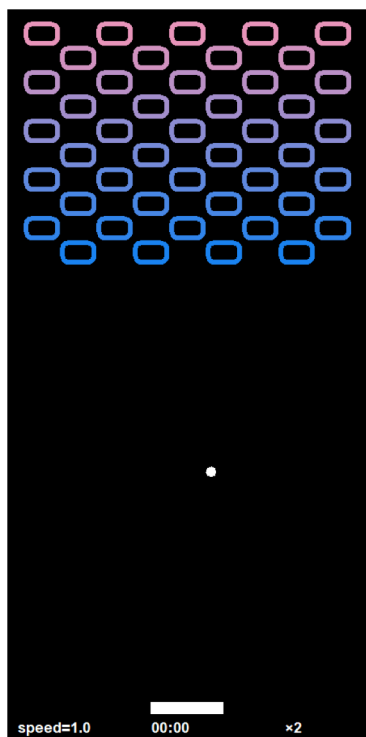
1810181 鴨田恭佑

1810220 木元颯人

1810277 酒井佑旗

作成プログラムの概要

ブロック崩し 特徴



- 1: シンプルな操作性
- 2: アイテムの取得による、エフェクト付与
- 3: ステージ作成による、難易度調整
- 4: 時間経過に伴うボールの速度上昇

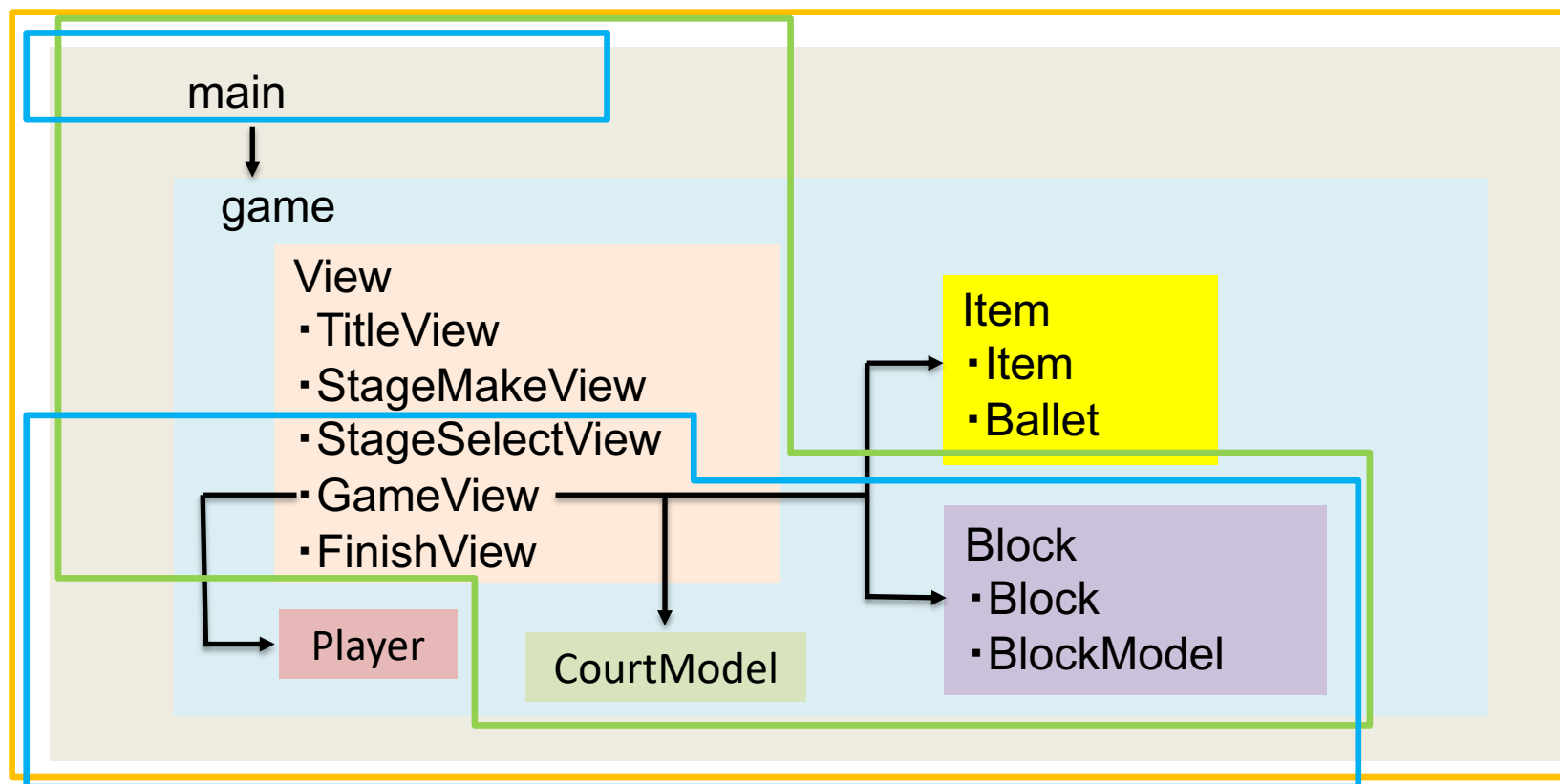
プログラムの構成

- ・MVCモデルで構築.

木元

鴨田

酒井



担当分担

鴨田

- Block,Viewクラスの設計・実装.
- レポートの全体構成.

木元

- CourtModelクラスの設計・音楽.
- プレゼンテーション作成.

酒井

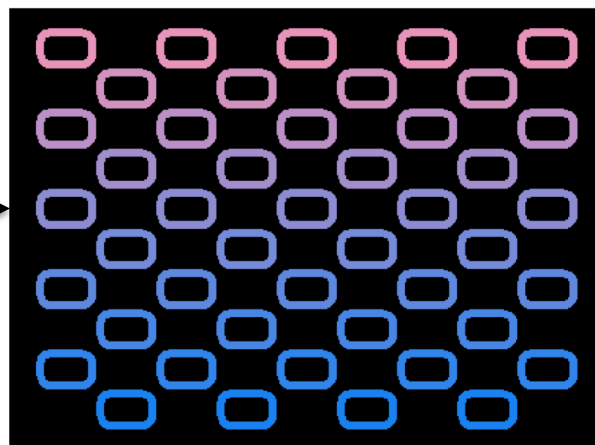
- View,Item,Playerクラスの設計・実装.
- 作成したクラスの統合. 最終調整.

鴨田担当分: Block, View クラス

工夫点

- ・テキストファイルから0,1の配列を読み取り、1の部分ブロックとして表示
- ・スタート時のボールの軌道やスピード、アイテムの制限時間、残基の表示の作成

```
101010101  
010101010  
101010101  
010101010  
101010101  
010101010  
101010101  
010101010  
101010101  
010101010
```



木元担当分: CourtModelクラス

- checkHitBlock, checkHitメソッド
 - ー ブロックと壁のあたり判定に対応した値を返す。
- moveBallメソッド
 - ー ボールの動きを制御するメソッド。
 - ー 壁やブロックに当たるとボールの動きが変化するので、8つの変化が存在する。

```
public int checkHit(double x, double y, double playerX){  
    if (y < 568){  
        if (x <= ball_radius && ball_moving_x <= 0)  
            return 3;  
        if (x >= court_size_x - ball_radius &&  
            ball_moving_x >= 0) return 4;  
        if (y <= ball_radius && ball_moving_y <= 0)  
            return 1;  
    }
```

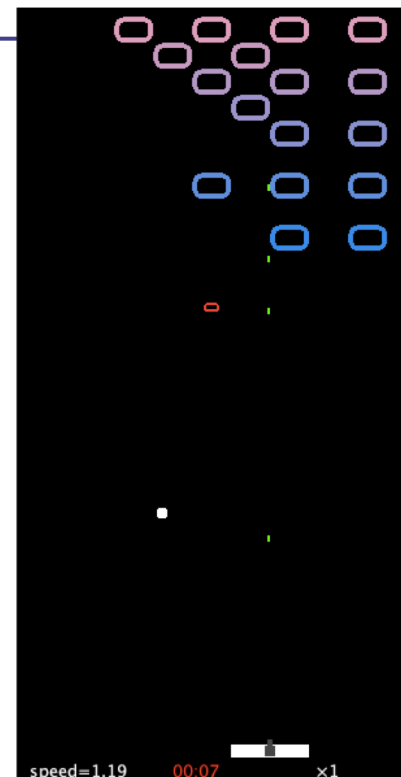
酒井担当分： 複数クラスの設計・実装・統合

- Item,ballet
 - 複製しやすいように柔軟さ意識した
 - アイテムの種類を容易に増やすことができる
- StagemakeView
 - デザインをシンプルにするためにアイコンを自作した



- 統合作業

- Model, View, Playerを統合. Dropboxでコードを共有しスムーズに統合できた



デモと質疑応答 の時間.

