


J1-8



Tomato Fest!

2210020 天野 誠司

2210113 大杉 優志

2210593 松浦 史明

概要 1

2

🍅 スペインのトマト祭りから着想

🍅 往年のフラッシュゲームである「艦砲射撃」からゲーム構成の着想

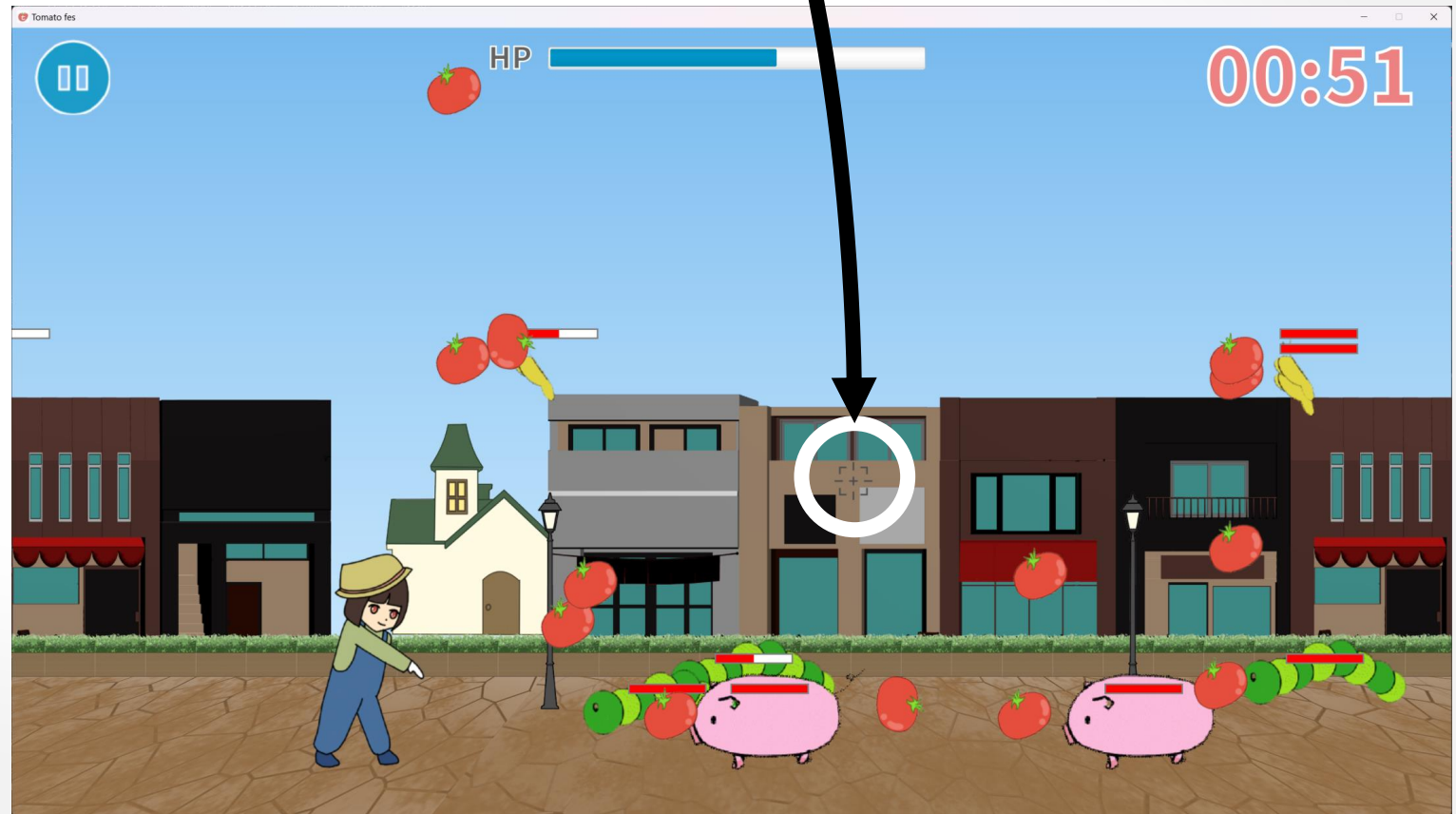


画像 : [トマティーナ - Wikipedia](#)

概要 1

3

- 🍅 迫り来る敵を果物（など）を当てて倒す
- 🍅 マウ斯卡ーソルで照準、投擲物選択は 1 ~ 5 キー
- 🍅 城を倒せば勝利
- 🍅 HPが0になるか時間オーバーで敗北

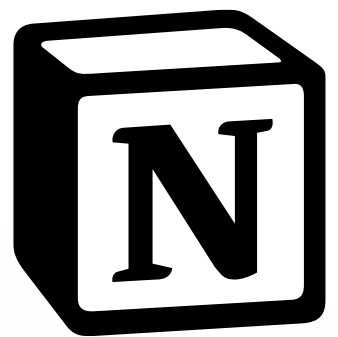


概要 2

4







Discord



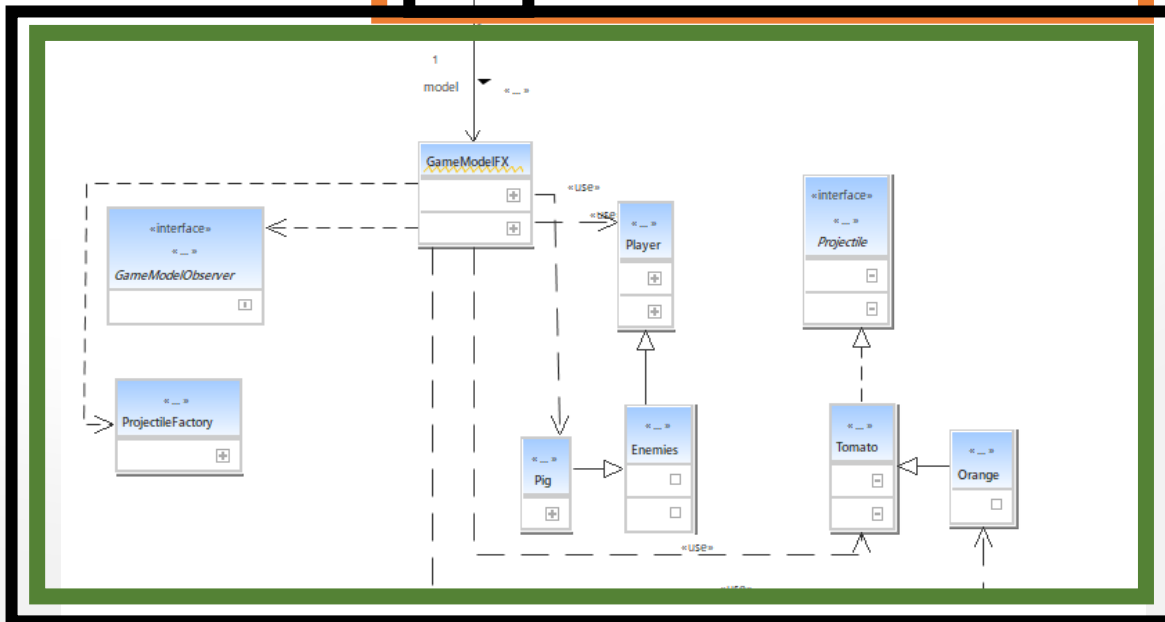
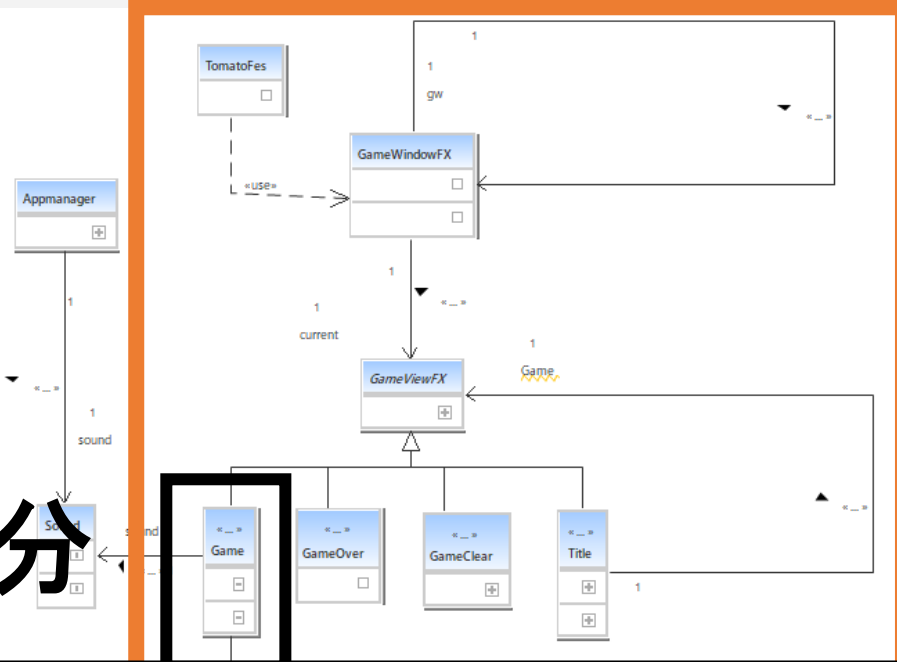
構成・担当箇所

7

View / Controller

ゲーム本体に関わる部分

Model



大杉

M

素材作成

天野

MC

物理演算

松浦

VC

JavaFXを用
いた、フロ
ントエンド

- ・ 果物・敵の変数設定

果物が与えるダメージや、敵のHP等を変数を一般的に設定した雛形を作成。

工夫点：雛形を用いることで複数出現させるときに簡略化できる。

- ・ 当たり判定

キャラクター、敵、果物のあたり部分を円と近似し、重なったときに1を返す関数を作成

- ・ キャラクター等素材作成

キャラクターのデザイン

キャラクターのアニメーションや背景、ホーム・クリア画面等の素材を作成



【Model】

敵や投擲物を配列管理

工夫＞敵がプレイヤーに向かう・CSVにより敵出現管理

【物理演算】

果物それぞれに時間変数 t を与え、 x 座標 y 座標毎に重力を考慮し計算

工夫＞掛け算はなるべく避け $+=$, $-=$ を用いてなるべく計算を軽くした。

【Controller】

mouseClickedで物理演算に必要な定数計算

```
70 //以下ファイル読み込み
71 filedata = new ArrayList<>(); //これにファイルデータ入れる
72 try {
73     File f2 = new File("stagedata/城.txt");
74     BufferedReader br2 = new BufferedReader(new FileReader(f2));
75
76     String line;
77     // 城の情報
78     if((line = br2.readLine()) != null) {
79         String[] data = line.split(",", 0);
80         double x = Double.parseDouble(data[0]); double y = Double.parseDouble(data[1]);
81         double hp = Double.parseDouble(data[2]);
82         //配列には追加しない
83         castle = new Castle(x,y, hp);
84     }br2.close();
85     // 1行ずつCSVファイルを読み込む
86     // 敵の情報
87     File f1 = new File("stagedata/stage.txt");
88     BufferedReader br1 = new BufferedReader(new FileReader(f1));
89     double x,y; String kind_enemy; //double time_r;
90     double enemy_t;
91     while ((line = br1.readLine()) != null) {
92         String[] data = line.split(",", 0); // 行をカンマ区切りで配列に変換
93         x = Double.parseDouble(data[0]); y = Double.parseDouble(data[1]);
94         kind_enemy = data[2];
95         enemy_t = Double.parseDouble(data[3]);
96         filedata.add(new Dataclass(x, y, kind_enemy, enemy_t)); //time_r,
97     }
98     br1.close();
99
100 } catch (IOException e) {
101     System.out.println(e);
102 }
```

```
372 double p2X = p2.getX();
373 p2X += (int)(p2.getV()*Math.cos(p2.getTheta())); // 投擲物を右に動かす
374 p2.setX(p2X);
375
376 double p2Y = p2.getY();
377 p2Y -= (int)(p2.getV()*Math.sin(p2.getTheta()) - (9.8/240)*p2.getT() - 0); // 投擲物を上
378 p2.setY(p2Y);
379 p2.setaddT(1.0);
```

【View】

▼工夫した点

- ・オブジェクトを描画するグループ全体のみを入れ替えることで、簡潔な構造でのシーン変化を実現
- ・横スクロールについて、単純にプレイヤーに追従させるのではなく遅延追従させることにより、画面酔いの防止
- ・z座標を変化させてオブジェクトを配置したことによる、半立体的な背景生成

【Controller】

- ・クリックした際のModelとの連携

【その他】

- ・コードの統合、全体的な修正



名前	更新日時	種類	サイズ
bin	2024/02/02 2:52	ファイル フォルダー	
enemy	2024/02/02 2:52	ファイル フォルダー	
fonts	2024/02/02 2:52	ファイル フォルダー	
img	2024/02/02 2:52	ファイル フォルダー	
img_game	2024/02/02 2:52	ファイル フォルダー	
img_pj	2024/02/02 2:52	ファイル フォルダー	
img_title	2024/02/02 2:52	ファイル フォルダー	
img_ui	2024/02/02 2:52	ファイル フォルダー	
jre	2024/02/02 2:53	ファイル フォルダー	
lib	2024/02/02 2:51	ファイル フォルダー	
SE	2024/02/02 2:52	ファイル フォルダー	
src	2024/02/02 2:52	ファイル フォルダー	
stagedata	2024/02/02 2:52	ファイル フォルダー	
TomatoFes_lib	2024/02/02 2:51	ファイル フォルダー	
TomatoFes.exe	2024/02/02 2:54	アプリケーション	143 KB
TomatoFes.jar	2024/02/02 2:51	Executable Jar File	40 KB

ご清聴ありがとうございました