

## 1. グループ番号, グループ名

F

## 2. グループメンバー

1Y18F014 井上智裕

1Y18F055 國分智利

1y19f112 東山龍之介

1w172343 楊敏舜

1w182208 谷口優真

## 3. つくる作品の仕様

### \*1人称迷路ゲーム

パックマンみたいに敵モブがいる。

ーゴール制：特定のチェックポイント3つ回ったらゴール。時間を競う。

- ・決まった迷路を作る

- ・敵モブ

ー移動方式：つきあたりで判定

- ・通った所の壁にマーキングする機能

- ・回数限定でジャンプ（要検討）

- ・ミニマップ（余裕があれば）

## 4. どのようにプログラミングするか

- ・描画に必要なパラメータは？データ構造は？グローバル変数？構造体？

- ・ global maze

迷路のマップデータ。二次元配列。

一壁の座標：4bit指定

- ・ global チェックポイント座標

二次元配列。

- ・ 自キャラ座標(現在いるマス)

- ・ 敵キャラの座標管理

一移動アルゴリズム

- ・ 迷路の描画

- ・ キャラクターの描画

一自キャラ

一敵キャラ

- ・ 自キャラの操作システム

====

最低限、描画 & ゲームの制御の二つの部分の連携方法（具体的にどのようなデータを渡せば、描画できるのか）をまず決める。

一パラメータ

- ・ var Gameboard: Gameboard
- ・ var CheckPoint: [Point]
- ・ var StartPoint: Point

ーデータ構造の定義

- ・ Gameboard

- ・ Point

ー連携方法

未定

5. プログラムを機能(モジュール)に分割し、各モジュール作成の担当者の割当

担当者：未定。下記タスクを割り振る。

\* 制御

- ・ 自キャラ制御

ー座標管理

ー操作方式

ー移動制限

ーマーキングシステム

- ・ 敵キャラ制御

ー座標管理

ー突き当り条件分岐

ー自キャラとの当たり判定

- ・ タイマー機能

- ・ マップ作成

ー壁の座標

ーチェックポイントの座標

ースタート地点

\* 描画

- ・ 敵モブオブジェクト

- ・ 壁オブジェクト

ーマーク前

ーマーク後

- ・ チェックポイント

ーチェック前

ーチェック後

## 6. どのようにプログラムをまとめるか

Github で

## 7. どのようにプログラムのソースコードを共有するか

Github

<https://github.com/c-app-dev-waseda-group-f/waseda-c-application-development-2020-spring-group-work>

参考 : <https://desktop.github.com/>