

GameManager

```
public static GameState gameState // ゲームのステータス

public static bool[] doorsOenedState // ドアの開閉状況

public static int key1 // 鍵 1 の持ち数
public static int key2 // 鍵 2 の持ち数
public static int key3 // 鍵 3 の持ち数
public static bool[] keysPickedState // 鍵の取得状況

public static int bill // お札の持ち数
public static bool[] itemsPickedState // アイテムの取得況

public static bool hasSpotLight // スポットライトを持っているか
public static int playerHP // プレイヤーの HP
```

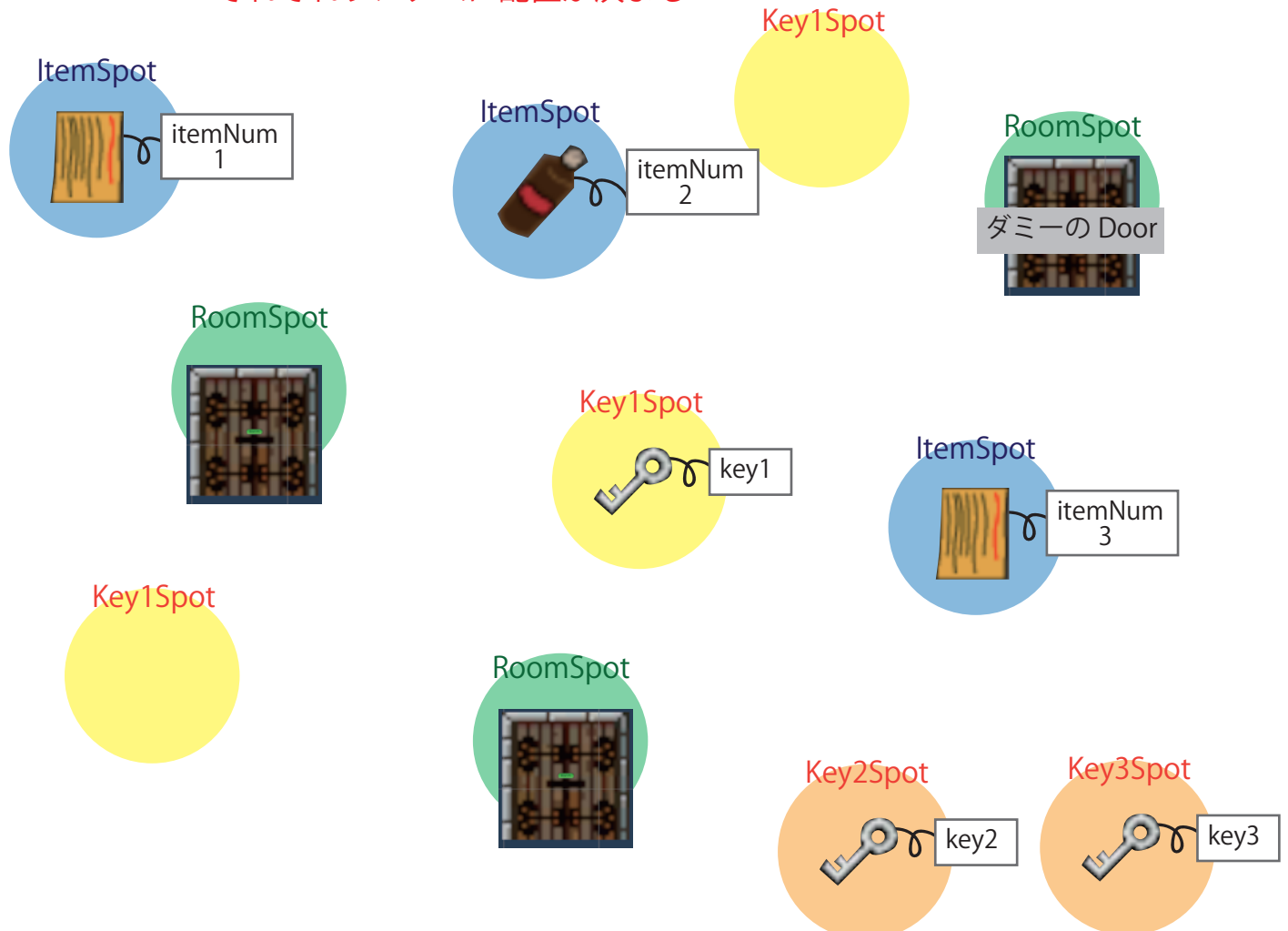
public enum GameState

- playing
- talk
- gameover
- gameclear
- stop
- ending

RoomManager の役割

ゲーム開始時に「Room、Item、Key」の配置を決めてそれぞれを初期化
シーン読み込み時に「Room、Item、Key」の配置と消化状況も再現

それぞれランダムに配置が決まる



RoomManager

```
public static int[] doorsPositionNumber // 各入口の配置番号
public static int key1PositionNumber // 鍵 1 の配置番号
public static int[] itemsPositionNumber // アイテムの配置番号

public GameObject[] items // 5 つのアイテムプレハブの内訳
public GameObject room // ドアのプレハブ
public GameObject dummyDoor // ダミードアのプレハブ
public GameObject key // キーのプレハブ

public static bool positioned // 初回配置が済みかどうか
public static string toRoomNumber // Player が配置されるべき位置
```

RoomManager の Start メソッドの構造

①static 変数 **positioned == false** → 初期配置決め

- StartDoorsPosition メソッド // ドア配置
- StartKeysPosition メソッド // キー配置
- StartItemsPosition メソッド // アイテム配置
- **positioned** を true に

positioned == true → 再配置

- LoadDoorsPosition メソッド // ドア配置
- LoadKeysPosition メソッド // キー配置
- LoadItemsPosition メソッド // アイテム配置

②いずれにしても Player の位置を決める • PlayerPosition メソッド

