

# Shadow Rpg - 作品紹介

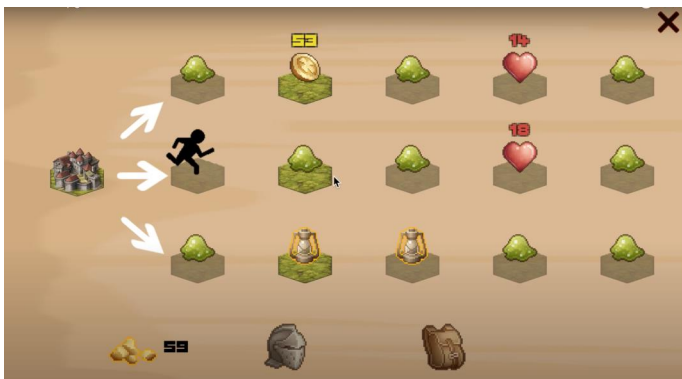


- 製作者 : 近畿大学工学部情報学科  
4年生 新卒 マ ドヒョン
- 作成期間 : 2022.5.13~2022.5.29
- ジャンル : ロールプレイングゲーム
- 目標プラットフォーム : パソコン
- 開発環境 : Unity
- 実行環境 : Mac
- 動画リンク : [https://www.youtube.com/watch?v=a\\_U2VKcoBZM](https://www.youtube.com/watch?v=a_U2VKcoBZM)
- ソースコード : <https://github.com/MaDohyun/Game/tree/master/ShadowRpg>
- ダウンロード :  
<https://3.gigafile.nu/0917-d58f946108ce8097daca8b8a50e792246>

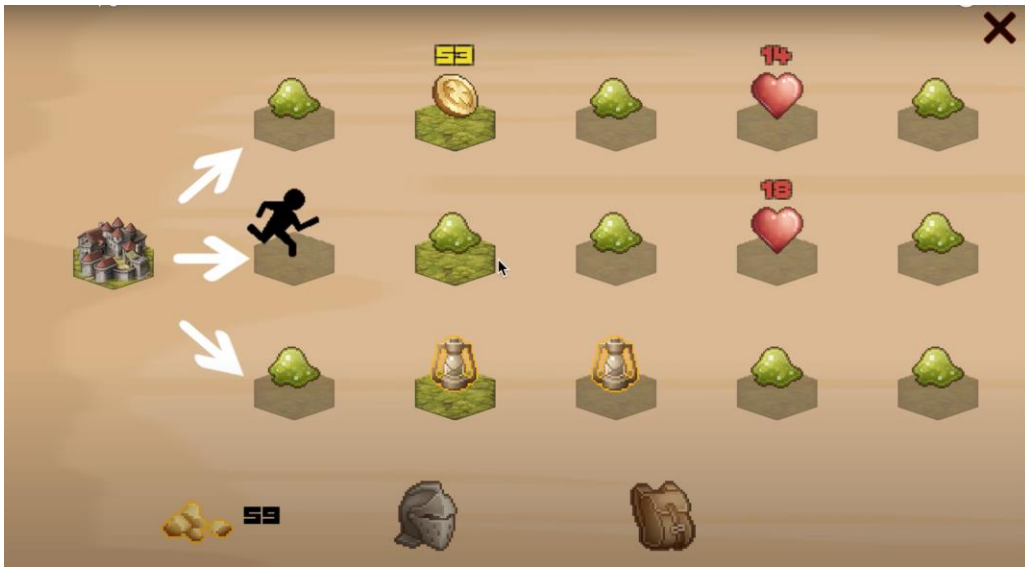
Shadow Rpgはさまざまなキャラクターを集めて敵を倒すゲームです。

# Scene

---



- Shadow Rpgは大きく二つのシーンで構成されています。
- Mapシーンではアイコンをタッチすることで目的地を選択することができます。
- MapシーンでモンスターのアイコンをタッチするとBattleシーンに移動し、モンスターと戦うことになります。



## < Map Scene >

- プレイヤーはマップシーンから目的地を選択することができます。  
アイコンの種類によってお金を得たり、新しいキャラクターを得たり、モンスターと戦うことができます。
- プレイヤーはモンスターを打ち破ったり、マップシーンでお金を得て商店で装備やキャラクターを手に入れることができます。



- プレイヤーは、装備ウィンドウやキャラクター編成ウィンドウで自分が使用する装備やキャラクターを選択することができます。
- キャラクターロットや装備ロットにマウスを近づけると情報を得ることができます。また、キャラクターは100ゴールドを出してアップグレードさせることが可能です。

# Map Sceneの主要クラス

MapPlayer
+destination:GameObject +map:GameObject moveSpeed:int mapGap:Vector2 mapXGap:float
-Update():void +SetDestination(destination:GameObject,Speed:int):void

MapPlayer クラス

MapPlayerクラスでは、マップのプレイヤーアイコンに関するクラスです。目的地を設定して、右に移動した距離だけマップオブジェクトを左に移動させてカメラに固定する役割を果たしています。

Land
landbutton:GameObject iconGenerator:IconGenerator map:GameObject moveSpeed:int player:GameObject iconVector3:Vector3 selectedLand:bool icon:MapIcon Level:int +landOn:bool
Start():void Update():void MoveComeHere():void DestroyButton():void

Land クラス

Landクラスでは、オブジェクトがクリックされる場合、プレイヤーの目的地をLandクラスのオブジェクトに設定します。

MapUIManager
shadowSelectionPanel:ShadowSelectionPanel equipmentPanel:EquipmentPanel currentMoneyText:Text shadowSelectionPanelOn:bool inventoryPanelOn:bool
Update():void ShadowPanelOn():void InventoryPanelOn():void CurrentMoneyTextOn():void

MapUIManager クラス

Landクラスでは、目的地をクリックする場合、プレイヤーの目的地をLandクラスのオブジェクトに設定します。

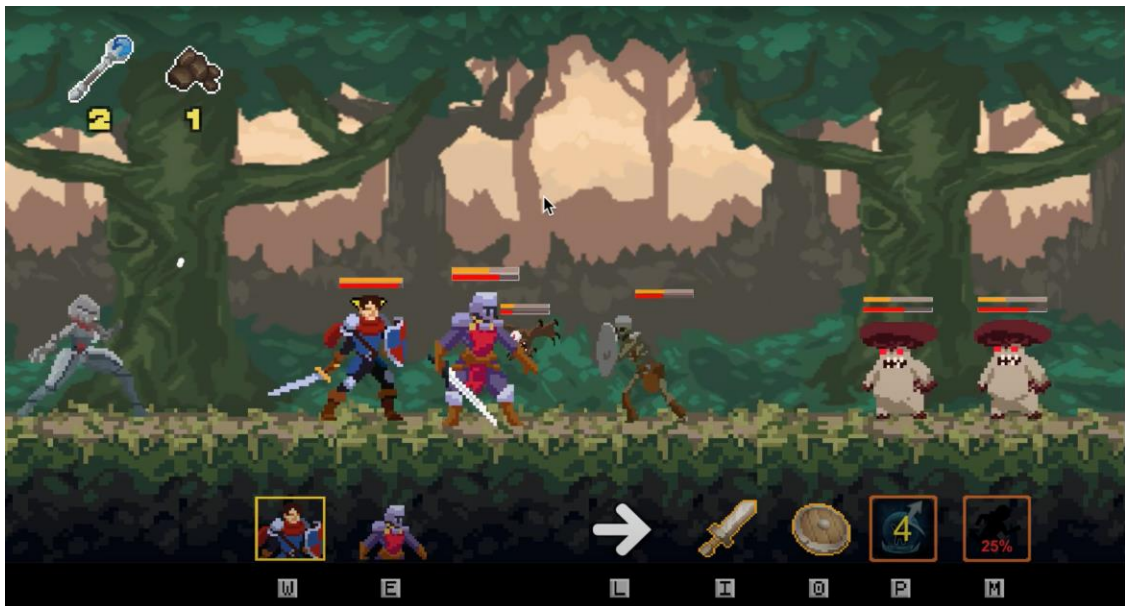
GameManager
mapState:MapState MapDestination:GameObject battleLevel:int
-Awake():void -Start():void Update():void +MapOn():void +ChangeMoveState():void +StageClear():void +ResetEquipment():void +ResetShadow():void +SwapShadow(battleShadowList:List<Shadow>,from:int,to:int):void +SwapShadow(battleEnemyList:List<Enemy>,from:int,to:int):void +GameOver():void +GameClear():void +GameReset():void

«enumeration» MapState
Move StageClear StageClearNot

GameManager クラス

GameManagerは目的地のクリアなどを判断して、Mapシーンでプレイヤーのステータスを変更します。そして現在のシーンがMapシーンの場合、MapのオブジェクトをOnに変更し、他のシーンの場合はOff処理を行います。



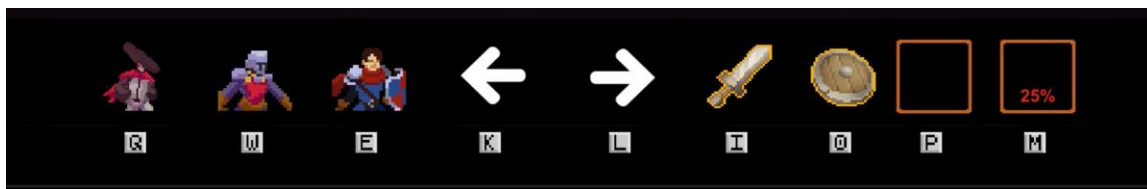


## < Battle Scene >

- バトルシーンに突入すると、プレイヤーはBattleManagerクラスを通じて自分が設定したbattleShadowListで最大3つのキャラクターを召喚することになります。
- バトルシーンに突入すると、敵はBattleManagerクラスを通じてGameManagerのbattleLevelに合わせて様々な種類のモンスターを召喚します。



- 敵は人工知能で自動的に戦闘を行いますが、プレイヤーはキーボードを通じてキャラクターの行動を決定して戦闘を進めることができます。
- キーボードでキャラクターを選択すると、時間の流れが遅くなり、アクションゲージが溜まったらアクションボタンを押して行動を選択することができます。



## < Action Key >

- Kボタン：キャラクターがPlayerクラスのbattleShadowListの配列順序で、自分の順序より後ろにいるキャラクターとSwapします。もし自分が一番後ろにいるとボタンは消えることになります。
- Lボタン：キャラクターがPlayerクラスのbattleShadowListの配列順序で、自分の順序より前にいるキャラクターとSwapします。もし自分が一番前にいるとボタンは消えることになります。
- Iボタン：キャラクターが攻撃行動をとります。
- Oボタン：キャラクターが防御行動をとります。
- Pボタン：スキルのあるキャラクターの場合にスキルを発動します。
- Mボタン：25%の確率で戦闘から離脱します。

# Battle Sceneの主要クラス

```
class BattleManager {
+battleEnemyPositions:Transform[]
+battleShadowPositions:Transform[] = new Transform[3]
enemyGenerator: EnemyGenerator
+battleEnemyList:List<Enemy>
+shadowPositionSummonEffects:ParticleSystem[]
+battleCamera:Camera
selectedShadow:Shadow
+backGrounds:GameObject[]
+resultImage:GameObject
+resultMoneyText:Text
isSummonEffect:bool
colorChange:float
random:int
resultMoney:int
Start():void
Update():void
+GivePositionEnemy():void
+GivePositionBattleShadow():void
+SetBattleShadow():void
+SetBattleShadowEffect():void
+SetBattleEquipment():void
+SetEnemy():void
+SetBackGround():void
+SetSelectedShadow():void
+SlowTimeSpeed():void
+ReturnTimeSpeed():void
+MoveCamera(cameraMoveXPosition:float):void
+MoveCameraSelectedShadow():void
+MoveCameraOff():void
+BattleVictory():void
+BattleLose():void
+StageClear():void
+LateUpdate():void
+GameClear():void
}
```

BattleManager クラス

BattleManagerクラスでは、キャラクターと敵の生成を実行し、カメラの移動をコントロールします。そしてステージのクリアを判断するクラスです。

```
class BattleSceneUIManager {
selectedShadowIcon:GameObject
shadowSelectedBorderLines:GameObject[]
shadowIcons:GameObject[]
shadowSkillIcon:GameObject
shadowSkillIconGray:GameObject
shadowSkillDelayText:Text
EquipmentIcons:GameObject[]
leftMoveIcon:GameObject
rightMoveIcon:GameObject
EquipmentTexts:Text[]
selectedIconGap:Vector3
selectedShadow:Shadow
-Start():void
-LateUpdate():void
+SetShadowIcons():void
+SetShadowSkillIcon():void
+SetSelectedBorderLine():void
+SetSelectedShadowIcon():void
+SetEquipmentIcon():void
+SetEquipmentText():void
+SetMoveIcon():void
+UIOff():void
+SetSelectedShadow():void
}
```

BattleSceneUIManager クラス

BattleSceneUIManagerクラスでは、キャラクターやスキルのアイコンなどbattleシーンのUIをコントロールします。

```
class BattleKeyController {
selectedShadow:Shadow
defenceShields:GameObject[]
defenceShieldGap:Vector3
cameraMoveXPosition:float
random:int
runImage:GameObject
runText:Text
runCoolTime:float
runDelay:float
isRunOn:bool
-Start():void
Update():void
+ShiledOff():void
+SetSelectedShadow():void
}
```

BattleKeyController クラス

BattleKeyControllerクラスでは、Battleシーンでキーボードの入力を受け、そのアクションを実行させる役割をします。

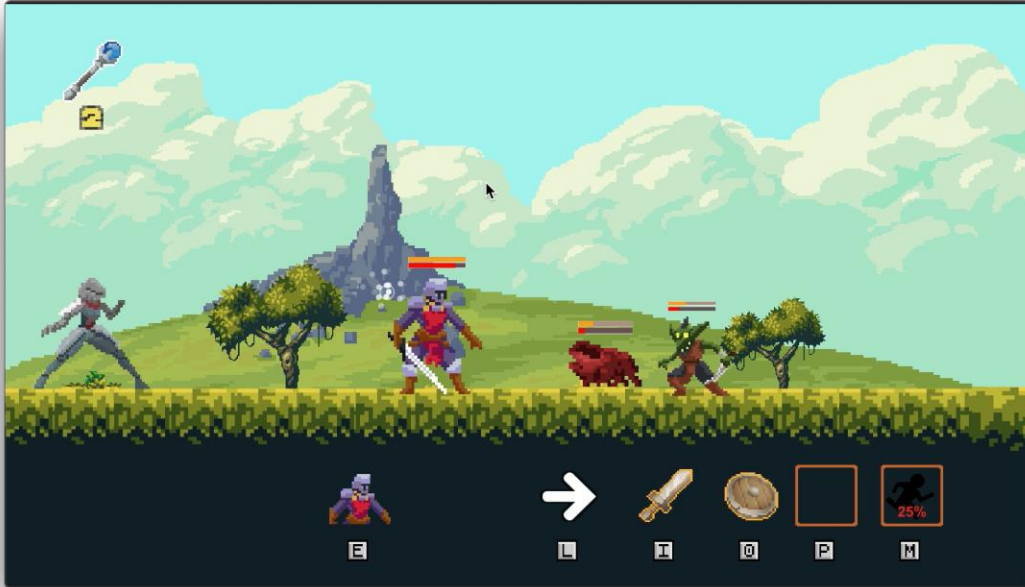
```
class GameManager {
mapState:MapState
MapDestination:GameObject
battleLevel:int
-Awake():void
-Start():void
Update():void
+MapOn():void
+ChangeMoveState():void
+StageClear():void
+ResetEquipment():void
+ResetShadow():void
+SwapShadow(battleShadowList:List<Shadow>,from:int,to:int):void
+SwapShadow(battleEnemyList:List<Enemy>,from:int,to:int):void
+GameOver():void
+GameClear():void
+GameReset():void
}
```

```
enum MapState {
Move,
StageClear,
StageClearNot
}
```

GameManager クラス

GameManagerは敗北や勝利などのメソッドを実行します。

## 制作の動機



ある程度Unityの基礎について勉強したから、実際にゲームでよく使う機能を勉強してみよう。

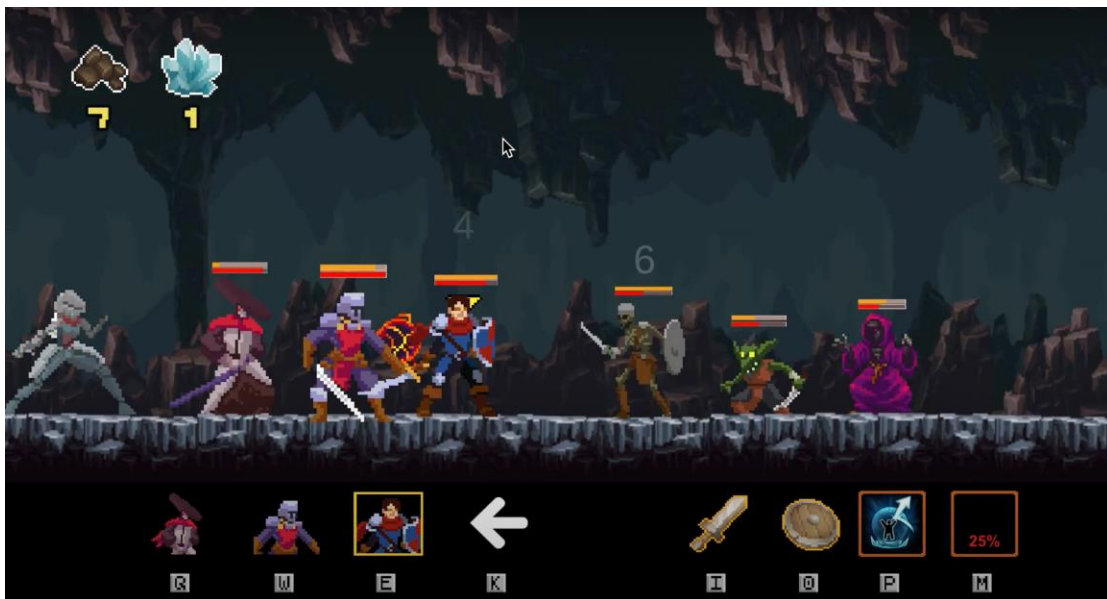


私が好きなロールプレイングゲームについて勉強しながらゲームを作成してみよう。



ロールプレイングゲームのショップ、インベントリ、キャラクターソルジャー装備アイテムなどを実装するゲームを作ってみよう。





## 作品のプログラム部分のアピールポイント



シャドウRPGは最近開発した作品で、以前作った作品の問題点を補完しました。

まずStaticやシングルトーンパターンを最小限に抑え、ゲーム進行がスムーズになるように作りました。

そしてキャラクターや敵を追加しやすくするために継承を積極的に活用しました。

また、キャラクターや装備のDBリストを作成し、Cloneを生成する際に楽に設定しました。

それにクラスやメソッド、変数の名前をできるだけ他の人にも理解しやすいようにしようと努力しました。

また、キャラクターなどはEnumを通じて状態をわかりやすいように表現しました。