

www.gamedevpbru.com

miniGameNo.3

Rev 2

ดณัຍ เวษฎาฐิตีกุล

จารุต บุศราทิว

DcG.IT@PBRU



การสร้างระบบต่อสู้แบบ Dragon Quest 1 (DQ1)
โดยใช้ Finite State Machine (FSM) เป็นแนวทาง
การจัดการลำดับการเล่น (Turn-based)



ศัตรูจะสู้กับผู้เล่นแบบ 1:1 ในแบบพลัดกันโจมตี และศัตรู
มีได้ 3 ประเภท คือ

1. นักรบ (แทนด้วย Cube สีแดง)
2. นักเวทย์ (แทนด้วย Cube สีนํ้าเงิน)
3. นักรบ (แทนด้วย Cube สีเขียว)



คำสั่งที่ใช้ในการต่อสู้ได้แก่

1. Attack
2. Defense
3. Heal (ใช้ MP)
4. Run

ฝั่งศัตรูจะเป็นการสุ่มรายการ 1,2 หรือ 3



จากต่อสู



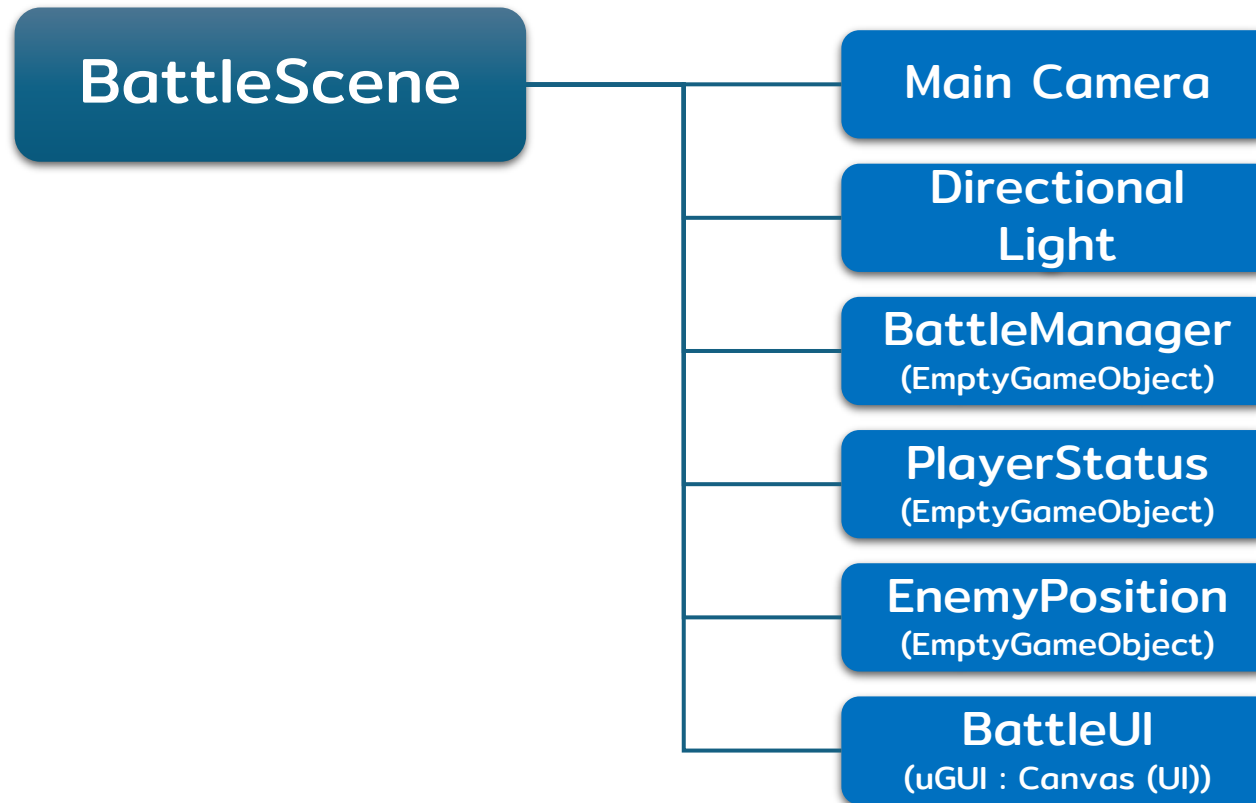
แนวคิดภาพรวมจากต่อสู้ (Battle Scene)

ลักษณะจาก

- จาก 3 มิติ (3D Scene)
- กล้องมุมมองบุคคลที่สามแบบนิ่ง (Fixed Camera)
- ผู้เล่นอยู่ด้านซ้าย / ศัตรูอยู่ด้านขวา
- ศัตรูแทนด้วย Cube สีต่าง ๆ
- เป็น Turn-Based Battle

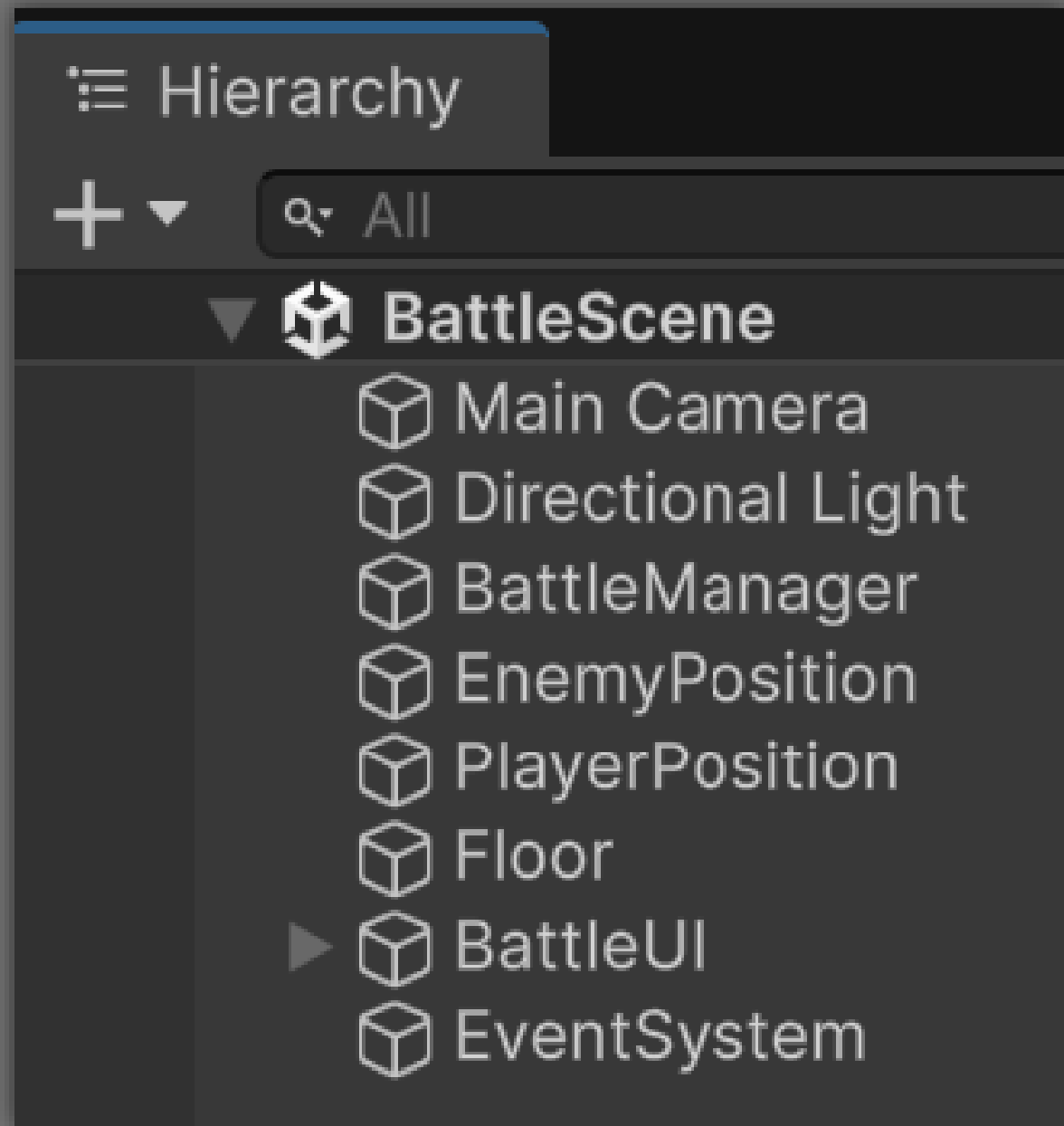


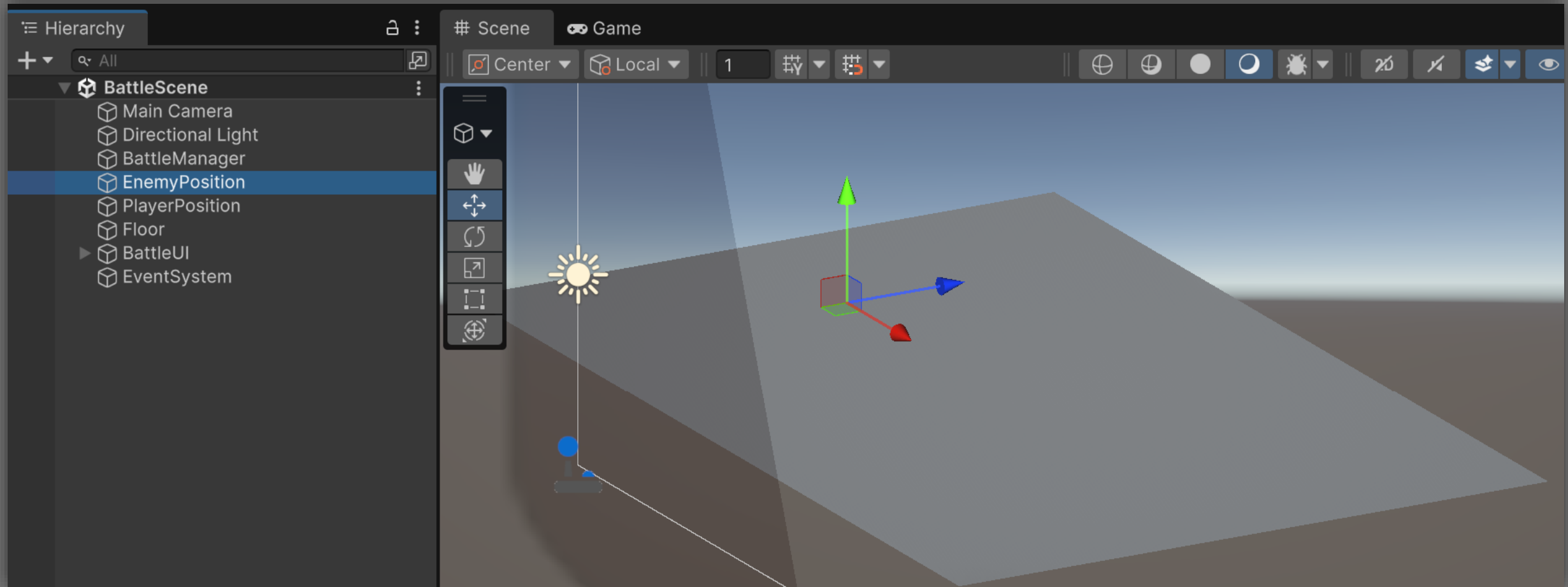
โครงสร้าง Scene

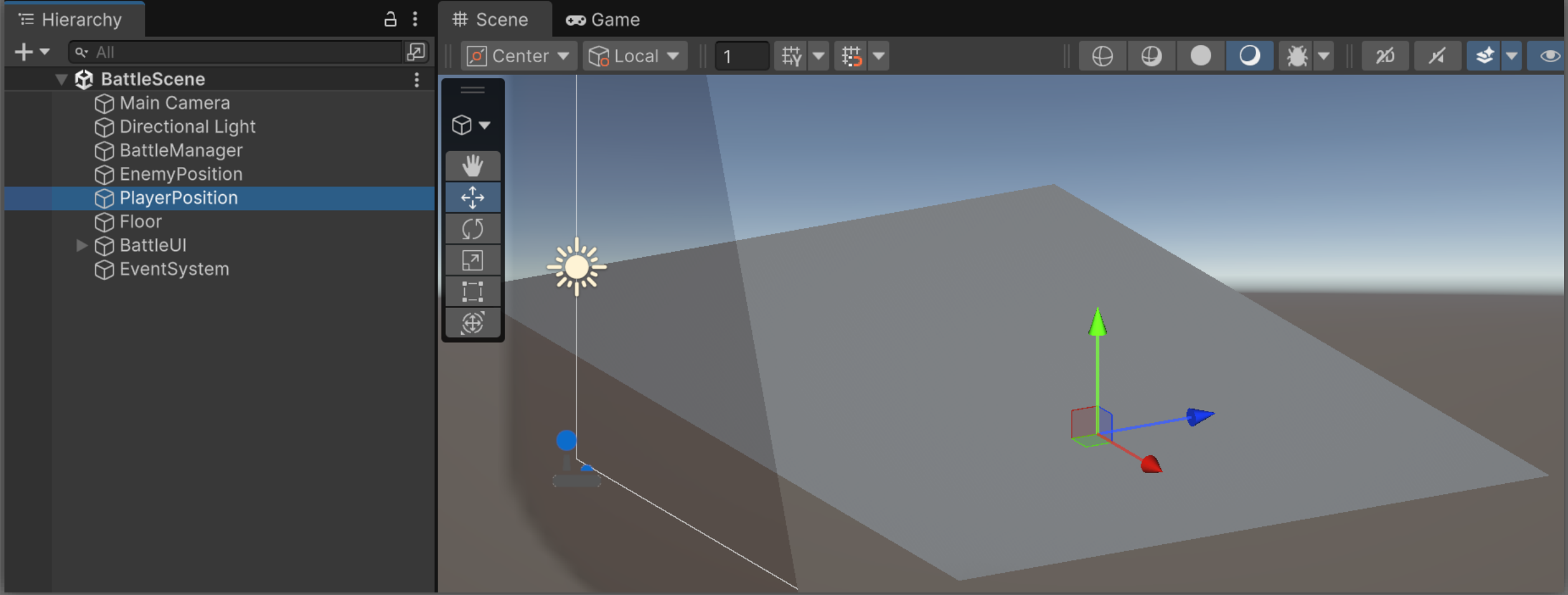




สร้างฉากต่อสู้ชื่อ BattleScene







การตั้งค่ามุมมองกล้อง

กล้องนิ่ง และไม่มี Camera Shake มาก

```
Camera.main.transform.position = new Vector3(0, 4.5f, -4f);
```

```
Camera.main.transform.rotation = Quaternion.Euler(30f, 0f, 0f);
```





CameraControl

```
1 using UnityEngine;
```

```
2
```

```
   Unity Script (1 asset reference) | 0 references
```

```
3
```

```
public class CameraControl : MonoBehaviour
```

```
4
```

```
{
```

```
   Unity Message | 0 references
```

```
5
```

```
void Start()
```

```
6
```

```
{
```

```
7
```

```
    Camera.main.transform.position = new Vector3(0, 4.5f, -4f);
```

```
8
```

```
    Camera.main.transform.rotation = Quaternion.Euler(30f, 0f, 0f);
```

```
9
```


```
}
```

```
10
```


```
}
```





Inspector 🔒 ⋮


 **Main Camera** Static ▼


Tag **MainCamera** ▼ Layer **Default** ▼

▶  **Transform** ? ⌵ ⋮

▶  **Camera** ? ⌵ ⋮

 **Audio Listener** ? ⌵ ⋮

▼  **Universal Additional Came** ? ⌵ ⋮

▼  **Camera Control (Script)** ? ⌵ ⋮

Script 📄 **CameraControl** ⊙



UI

กล่องข้อความด้านล่างสำหรับแสดงข้อความเป็นลำดับ เช่น

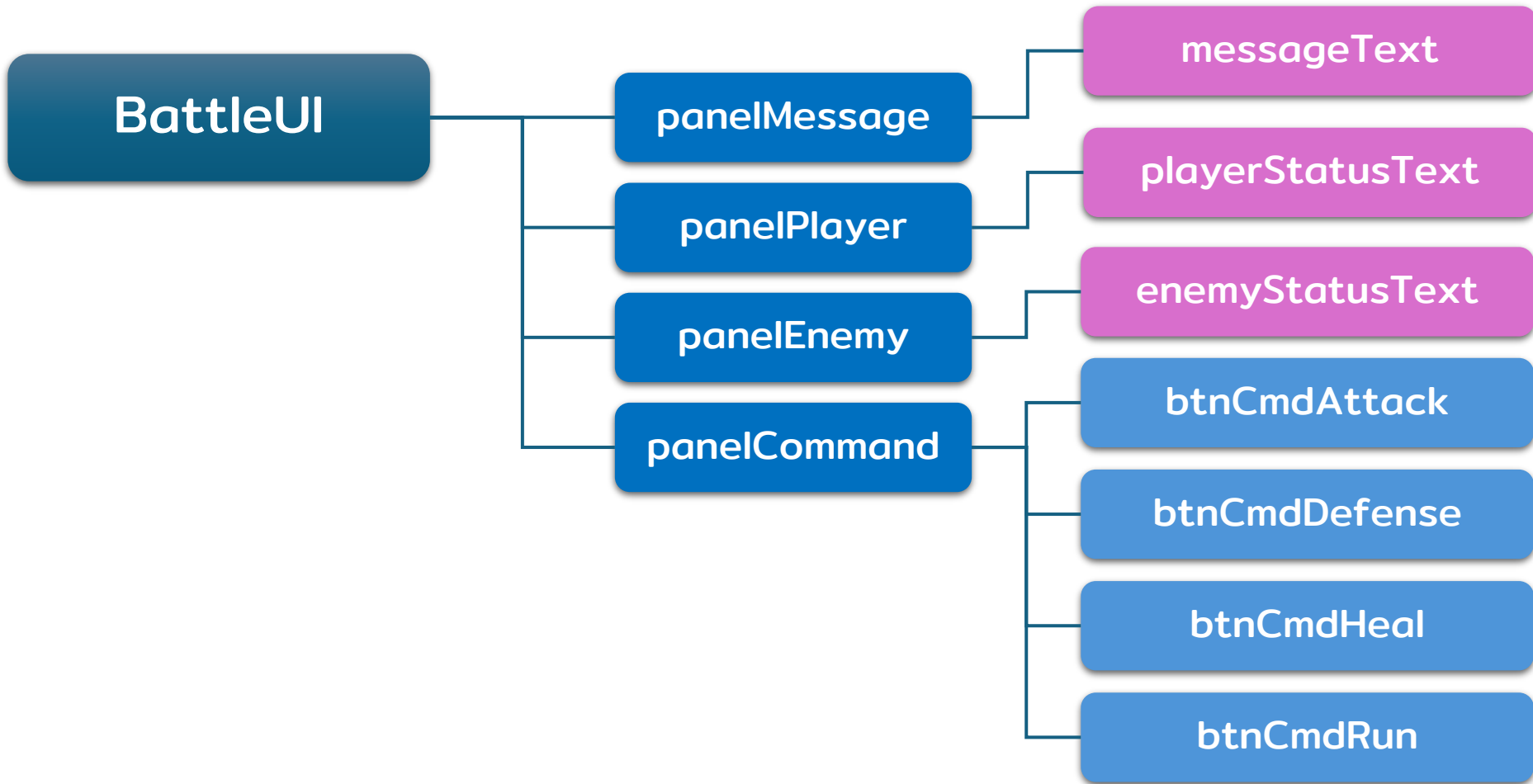
“Slime attacks!”

“Player takes 5 damage!”

โดยใช้ Coroutine เพื่อแสดงข้อความทีละบรรทัด



โครงสร้าง UI



▼ BattleUI

▼ panelMessage

messageText

▼ panelPlayer

playerStatusText

▼ panelEnemy

enemyStatusText

▼ panelCommand

▶ btnCmdAttack

▶ btnCmdDefense

▶ btnCmdHeal

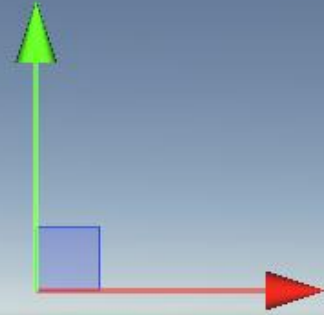
▶ btnCmdRun

EventSystem



Player (Job) : HP xx MP xx

Enemy (Job) : HP xx MP xx



New Text

Attack

Defense

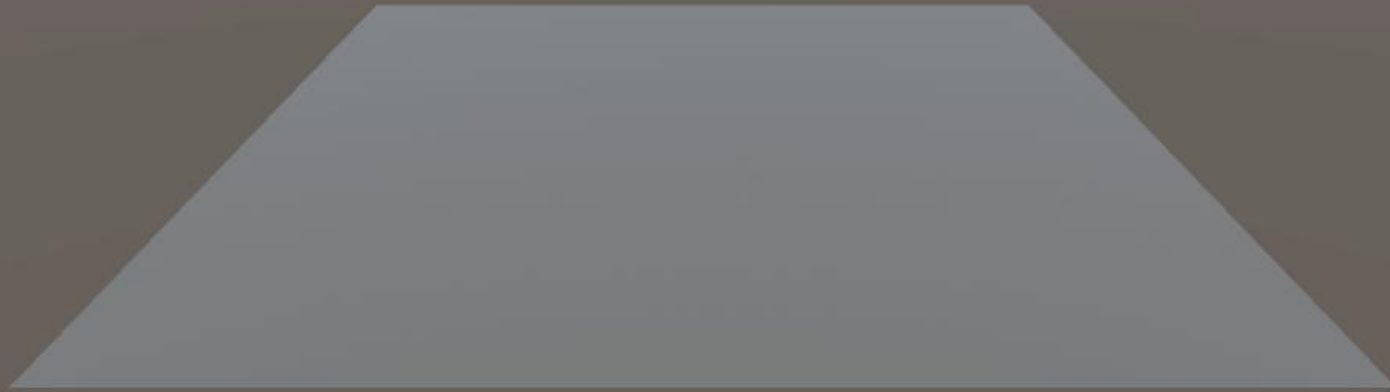
Heal

Run



Player (Job) : HP xx MP xx

Enemy (Job) : HP xx MP xx



New Text

Attack

Defense

Heal

Run



แนวคิดการคำนวณ Damage



แนวคิด Damage แบบ JRPG

หลักสำคัญมี 5 ส่วน

1. พลังโจมตี (STR / INT)
2. พลังป้องกัน (Defense)
3. เลเวล (Level Scaling)
4. ความสุ่มเล็กน้อย (Random)
5. ตัวคูณพิเศษ (Critical / Guard / Type Bonus)



เป้าหมาย

เลเวลสูง = เก่งขึ้นชัดเจน

ค่าพลังมีผลจริง

ไม่ตีแรงหรือเบาจนเกินไป



ค่าสถานะที่ใช้ในสูตร : ฟังโจมตี

STR โจมตีกายภาพ

INT เวท / Heal

Level ตัวคูณความแรง



คำสถานะที่ใช้ในสูตร : ฟังก์ชันป้องกัน

Defense ลดความเสียหาย

AGI หาย / Critical



Physical Damage Formula

- **BaseDamage =**
 $(\text{Attacker.STR} \times 2 + \text{Attacker.Level})$
- **DefenseFactor =**
 $\text{Defender.Defense} + \text{Defender.Level}$
- **Damage =**
 $(\text{BaseDamage} - \text{DefenseFactor})$
 $\times \text{RandomFactor}$



RandomFactor และการป้องกันไม่ให้ติดลบ

- $\text{RandomFactor} = \text{Random}(0.9 \sim 1.1)$
- $\text{Damage} = \text{Max}(1, \text{Damage})$



ตัวอย่างคำนวณ

ผู้เล่น STR = 10, Level = 5

ศัตรู Defense = 6, Level = 4

จะได้ว่า

$$\text{BaseDamage} = (10 \times 2) + 5 = 25$$

$$\text{DefenseFactor} = 6 + 4 = 10$$

$$\text{Damage} = (25 - 10) \times 1.0 = 15$$



Magic Damage Formula

MagicDamage =

(Caster.INT × 2 + Caster.Level)

- (Target.INT + Target.Level)

MagicDamage *= Random(0.95 ~ 1.15)

MagicDamage = Max(1, MagicDamage)



Magic Defense Formula

MagicDef =

TargetINT×1.2 + Level×2



Heal Formula

HealAmount =

(INT * 3)

+ (Level * 2)

* Random(0.9 ~ 1.1)

จะจำกัดไม่เกิน MaxHP ?



Critical Hit (AGI) Formula

$$\text{CriticalChance} = \text{Attacker.AGI} \times 0.5 \%$$

เช่น กำหนดให้ AGI เป็น 10 จะมีโอกาสการเกิด Critical Hit 5%
ดังนั้น เมื่อเกิด Critical Hit จะทำให้

$$\text{Damage} \times = 1.5$$



Defense Command Formula

FinalDamage *= 0.5 // ลดลงครึ่งหนึ่งถ้าผู้เล่นเลือก Defense

หรือ

Defense += Level



Run (ကူ) Formula

RunChance =

(Player.AGI + Player.Level)

/ (Enemy.AGI + Enemy.Level) × 50%

ຈຳກັດໄມ່ເກີນ 90%



```
1 public static int CalculatePhysicalDamage(  
2     CharacterStats attacker,  
3     CharacterStats defender,  
4     bool isCritical,  
5     bool defenderGuard)  
6 {  
7     int baseDamage =  
8         attacker.STR * 2 + attacker.Level;  
9  
10    int defense =  
11        defender.DEF + defender.Level;  
12  
13    float random = Random.Range(0.9f, 1.1f);  
14
```



```
15     float damage =
16         (baseDamage - defense) * random;
17
18     if (isCritical)
19         damage *= 1.5f;
20
21     if (defenderGuard)
22         damage *= 0.5f;
23
24     return Mathf.Max(1, Mathf.RoundToInt(damage));
25 }
```



ความได้เปรียบตามประเภทศัตรู (Class Bonus)

ผู้โจมตี	เป้าหมาย	โบนัส
Warrior	Archer	x1.2
Archer	Mage	x1.2
Mage	Warrior	x1.2



แนวคิดการจัดสมดุล



ปัญหาที่พบบ่อย

1. สูตรเชิงเส้น จะทำให้ช่วงท้ายของเกมดาเมจแรงเกิน
2. $STR \times Level$ จะทำให้ได้ค่าตัวเลขที่สูงมาก
3. Defense ไม่ทันดาเมจ



วิธีแก้แบบ JRPG คลาสสิก

1. ใช้ Level เป็นตัวคูณย่อย
2. ใช้ รากที่สอง ($\sqrt{\quad}$) หรือ \log เพื่อลดการพุ่ง
3. ดาเมจโตช้าลงเรื่อย ๆ



Physical Damage Formula

- **AttackPower =**
 $(STR \times 1.5) + (Level \times 2) + \text{sqrt}(Level) \times 3$
- **DefensePower =**
 $DEF \times 1.2 + Level \times 1.5$
- **RawDamage =**
 $AttackPower - DefensePower$
- **FinalDamage =**
 $RawDamage \times \text{Random}(0.9 \sim 1.1)$
- **FinalDamage =**
 $\text{Max}(1, FinalDamage)$



Critical Hit Formula

- **CriticalChance =**
$$\text{AGI} \times 0.3 \%$$
- **CriticalDamage =**
$$\text{CriticalDamage} \times 1.5$$



Magic Damage Formula

- **MagicPower =**
 $(\text{INT} * 2) + (\text{Level} * 2.5) + \text{sqrt}(\text{Level}) * 4$
- **MagicDefense =**
 $(\text{Target.INT} * 1.2) + \text{Level} * 2$
- **MagicDamage =**
 $(\text{MagicPower} - \text{MagicDefense})$
 $\times \text{Random}(0.95 \sim 1.15)$
- **MagicDamage = Max(1, MagicDamage)**



Heal Formula

- HealAmount =
 $(\text{INT} * 3) + (\text{Level} * 4) + \text{sqrt}(\text{Level}) * 6$
- HealAmount =
 $\text{HealAmount} * \text{Random}(0.9 \sim 1.1)$



Defense Command Formula

$$\text{FinalDamage} = \text{FinalDamage} \times 0.5$$

หรือ

$$\text{DefensePower} = \text{DefensePower} \times 1.5$$



Run Formula

เลเวลสูงหนีง่ายขึ้น

RunChance =

$$\frac{(\text{Player.AGI} + \sqrt{\text{Player.Level}} * 5)}{(\text{Enemy.AGI} + \sqrt{\text{Enemy.Level}} * 5)} * 50$$

จำกัด Max 90% และ Min 10%



```
1 public static int PhysicalDamage(  
2     CharacterStats atk,  
3     CharacterStats def,  
4     bool critical,  
5     bool guard)  
6 {  
7     float attackPower =  
8         atk.STR * 1.5f  
9         + atk.Level * 2f  
10        + Mathf.Sqrt(atk.Level) * 3f;  
11  
12    float defensePower =  
13        def.DEF * 1.2f  
14        + def.Level * 1.5f;  
15
```



```
16     float damage =
17         (attackPower - defensePower)
18         * Random.Range(0.9f, 1.1f);
19
20     if (critical) damage *= 1.5f;
21     if (guard) damage *= 0.5f;
22
23     return Mathf.Max(1, Mathf.RoundToInt(damage));
24 }
```



โครงสร้างข้อมูลตัวละคร

(Player / Enemy ใช้ร่วมกัน)



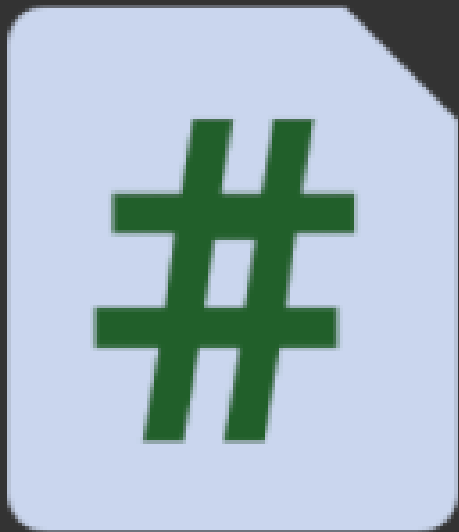
```
1 using UnityEngine;
2
3 [System.Serializable]
4 public class CharacterStat : MonoBehaviour
5 {
6     public int Level = 1;
7     public int HP, MaxHP;
8     public int MP, MaxMP;
9     public int STR;
10    public int DEF;
11    public int AGI;
12    public int INT;
13    public int EXP;
14    public int Money;
15 }
```

Unity Script | 2 references



CharacterStat



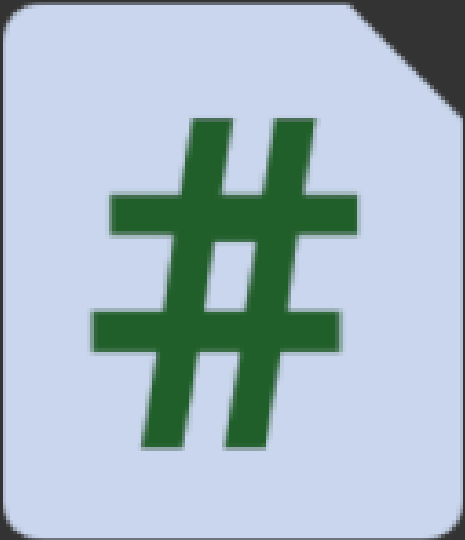


JobType

```
1 using UnityEngine;
2
3 [System.Serializable]
4 public enum JobType
5 {
6     Warrior,
7     Mage,
8     Archer
9 }
```

0 references





EnemyType

```
1 using UnityEngine;
2
3 0 references
4 public enum EnemyType
5 {
6     Warrior, // Red
7     Mage,    // Blue
8     Archer   // Green
9 }
```



ตัวอย่างการเปลี่ยนสีของศัตรู

```
void SetEnemyColor(EnemyType type)
{
    Color c = Color.white;
    if (type == EnemyType.Warrior) c = Color.red;
    if (type == EnemyType.Mage) c = Color.blue;
    if (type == EnemyType.Archer) c = Color.green;

    GetComponent<MeshRenderer>().material.color = c;
}
```



ค่าเริ่มต้นแต่ละอาชีพ

อาชีพ	STR	DEF	AGI	INT	HP	MP
Warrior นักรบ	✓✓✓✓	✓✓✓✓	✓✓	✓	✓✓✓✓	✓
Mage นักเวทย์	✓	✓✓	✓✓	✓✓✓✓	✓	✓✓✓✓
Archer นักรบ	✓✓✓	✓✓	✓✓✓✓	✓✓	✓✓	✓✓



ออกแบบ Stat Growth ต่อ Level (STR/INT/DEF)



เป้าหมาย

Level ต้น จะมีค่าสถานะขึ้นขีด

Level กลาง จะมีโตช้าลง

Level สูง จะมีไม่พุ่งจนสุดรพัง



วิธีที่ใช้

Linear + $\sqrt{\text{Level}}$

มีค่า Base (คลาสเป็นตัวกำหนด)

ใช้ตัวคูณต่างกันต่อคลาส



สูตรแกนของ Stat Growth (Core Formula)

StatValue =

BaseStat

+ (Level × LinearGrowth)

+ ($\sqrt{\text{Level}}$ × CurveGrowth)



Warrior (สายพลัง / จัด)

Stat	Base	Linear	Curve
STR	12	1.2	2.0
DEF	10	1.1	1.8
INT	4	0.3	0.5
AGI	6	0.6	1.0



Level	STR	DEF	INT	AGI
1	15	13	5	8
2	17	15	5	9
3	19	16	6	10
4	21	18	6	10
5	22	20	7	11
6	24	21	7	12
7	26	22	7	13
8	27	24	8	14
9	29	25	8	14
10	30	27	9	15
11	32	28	9	16
12	33	29	9	17
13	35	31	10	17
14	36	32	10	18
15	38	33	10	19
16	39	35	11	20
17	41	36	11	20
18	42	37	12	21
19	44	39	12	22
20	45	40	12	22



Mage (สายเวท)

Stat	Base	Linear	Curve
STR	4	0.3	0.5
DEF	5	0.4	0.7
INT	12	1.3	2.2
AGI	6	0.6	1.0



Level	STR	DEF	INT	AGI
1	5	6	16	8
2	5	7	18	9
3	6	7	20	10
4	6	8	22	10
5	7	9	23	11
6	7	9	25	12
7	7	10	27	13
8	8	10	29	14
9	8	11	30	14
10	9	11	32	15
11	9	12	34	16
12	9	12	35	17
13	10	13	37	17
14	10	13	38	18
15	10	14	40	19
16	11	14	42	20
17	11	15	43	20
18	12	15	45	21
19	12	16	46	22
20	12	16	48	22



Archer (สายเร็ว / ครีตคอส)

Stat	Base	Linear	Curve
STR	8	0.8	1.2
DEF	7	0.6	1.0
INT	6	0.5	0.8
AGI	12	1.2	2.0



Level	STR	DEF	INT	AGI
1	10	9	7	15
2	11	10	8	17
3	12	11	9	19
4	14	11	10	21
5	15	12	10	22
6	16	13	11	24
7	17	14	12	26
8	18	15	12	27
9	19	15	13	29
10	20	16	14	30
11	21	17	14	32
12	22	18	15	33
13	23	18	15	35
14	24	19	16	36
15	25	20	17	38
16	26	21	17	39
17	27	21	18	41
18	27	22	18	42
19	28	23	19	44
20	29	23	20	45



ความสัมพันธ์กับ Damage Formula

คลาส	จุดเด่น
Warrior	Physical Damage เสถียร / Tank
Mage	Magic Damage แรงสุด / Heal ดี
Archer	Critical บ่อย / Run ง่าย



```
1 public static int CalculateStat(  
2     int baseStat,  
3     float linear,  
4     float curve,  
5     int level)  
6 {  
7     float value =  
8         baseStat  
9         + level * linear  
10        + Mathf.Sqrt(level) * curve;  
11  
12    return Mathf.RoundToInt(value);  
13 }
```



ออกแบบ HP / MP Growth



แนวคิด

HP โตเร็วกว่า Damage เสมอ

MP โตตามบทบาท (Mage > Archer > Warrior)

ใช้สูตร $\text{Linear} + \sqrt{\text{Level}}$ เหมือน Stat เพื่อให้จัดสมดุลได้ง่าย



HP Formula

MaxHP =

BaseHP

+ (Level × HP_Linear)

+ ($\sqrt{\text{Level}}$ × HP_Curve)



MP Formula

MaxMP =

BaseMP

+ (Level × MP_Linear)

+ ($\sqrt{\text{Level}}$ × MP_Curve)



Warrior

Stat	Base	Linear	Curve
HP	120	25	35
MP	20	4	6



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Stat	Base	Linear	Curve		Level	STR	DEF	INT	AGI	HP	MP	
2	STR	12	1.2	2		1	15	13	5	8	180	30	
3	DEF	10	1.1	1.8		2	17	15	5	9	219	36	
4	INT	4	0.3	0.5		3	19	16	6	10	256	42	
5	AGI	6	0.6	1		4	21	18	6	10	290	48	
6	HP	120	25	35		5	22	20	7	11	323	53	
7	MP	20	4	6		6	24	21	7	12	356	59	
8						7	26	22	7	13	388	64	
9						8	27	24	8	14	419	69	
10						9	29	25	8	14	450	74	
11						10	30	27	9	15	481	79	
12						11	32	28	9	16	511	84	
13						12	33	29	9	17	541	89	
14						13	35	31	10	17	571	94	
15						14	36	32	10	18	601	98	
16						15	38	33	10	19	631	103	
17						16	39	35	11	20	660	108	
18						17	41	36	11	20	689	113	
19						18	42	37	12	21	718	117	
20						19	44	39	12	22	748	122	
21						20	45	40	12	22	777	127	
22						21	46	41	13	23	805	131	

Warrior

Mage

Archer

+



Mage

Stat	Base	Linear	Curve
HP	80	15	25
MP	50	18	30



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Stat	Base	Linear	Curve		Level	STR	DEF	INT	AGI	HP	MP	
2	STR	4	0.3	0.5		1	5	6	16	8	120	98	
3	DEF	5	0.4	0.7		2	5	7	18	9	145	128	
4	INT	12	1.3	2.2		3	6	7	20	10	168	156	
5	AGI	6	0.6	1		4	6	8	22	10	190	182	
6	HP	80	15	25		5	7	9	23	11	211	207	
7	MP	50	18	30		6	7	9	25	12	231	231	
8						7	7	10	27	13	251	255	
9						8	8	10	29	14	271	279	
10						9	8	11	30	14	290	302	
11						10	9	11	32	15	309	325	
12						11	9	12	34	16	328	347	
13						12	9	12	35	17	347	370	
14						13	10	13	37	17	365	392	
15						14	10	13	38	18	384	414	
16						15	10	14	40	19	402	436	
17						16	11	14	42	20	420	458	
18						17	11	15	43	20	438	480	
19						18	12	15	45	21	456	501	
20						19	12	16	46	22	474	523	
21						20	12	16	48	22	492	544	
22						21	13	17	49	23	510	565	



Archer

Stat	Base	Linear	Curve
HP	100	20	30
MP	35	10	15



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Stat	Base	Linear	Curve		Level	STR	DEF	INT	AGI	HP	MP	
2	STR	8	0.8	1.2		1	10	9	7	15	150	60	
3	DEF	7	0.6	1		2	11	10	8	17	182	76	
4	INT	6	0.5	0.8		3	12	11	9	19	212	91	
5	AGI	12	1.2	2		4	14	11	10	21	240	105	
6	HP	100	20	30		5	15	12	10	22	267	119	
7	MP	35	10	15		6	16	13	11	24	293	132	
8						7	17	14	12	26	319	145	
9						8	18	15	12	27	345	157	
10						9	19	15	13	29	370	170	
11						10	20	16	14	30	395	182	
12						11	21	17	14	32	419	195	
13						12	22	18	15	33	444	207	
14						13	23	18	15	35	468	219	
15						14	24	19	16	36	492	231	
16						15	25	20	17	38	516	243	
17						16	26	21	17	39	540	255	
18						17	27	21	18	41	564	267	
19						18	27	22	18	42	587	279	
20						19	28	23	19	44	611	290	
21						20	29	23	20	45	634	302	
22						21	30	24	20	46	657	314	



```
1 public static int CalculateResource(  
2     int baseValue,  
3     float linear,  
4     float curve,  
5     int level)  
6 {  
7     float value =  
8         baseValue  
9         + level * linear  
10        + Mathf.Sqrt(level) * curve;  
11  
12    return Mathf.RoundToInt(value);  
13 }
```



ออกแบบ EXP Curve (Level Up)



แนวคิด

ต้นเกมเลเวลอัปไว (รู้สึกสนุก)

กลางเกมเริ่มช้าลง (ต้องวางแผน)

ท้ายเกมต้องฟาร์มจริง (คุณค่าเลเวลสูง)



วิธีคำนวณ

Polynomial + $\sqrt{\text{Level}}$

ไม่ใช่ Exponential ตรง ๆ (ตัวเลขไม่ระเบิด)



EXP Required တံ့ Level

EXP_to_Next =

BaseEXP

+ (Level³ × A)

+ (Level² × B)

BaseEXP = 50

A = 0.5

B = 5



สูตร EXP Required ต่อ Level

EXP_to_Next =

BaseEXP

+ (Level³ × A)

+ (Level² × B)

BaseEXP = 50

A = 0.5

B = 5



สูตร EXP Required ต่อ Level

EXP_to_Next =

50

+ (Level × Level × Level × 0.5)

+ (Level × Level × 5)



สูตร EXP สะสมรวม (Total EXP Formula)

TotalEXP(Level) =

Σ EXP_to_Next (1 ถึง Level-1)



EXP Enemy Formula

EnemyEXP =

$$\text{EnemyLevel}^2 \times \text{EXP_Factor}$$

โดย

$$\text{EXP_Factor} = 6$$



แนวคิดการให้โบนัสตอบแทน

- ศัตรูแข็งแกร่ง ให้ค่า EXP +20-50%
- Boss ให้ EXP $\times 3$
- Elite ให้ EXP $\times 1.5$



```
1 public static int ExpToNextLevel(int level)
2 {
3     return Mathf.RoundToInt(
4         50f
5         + 0.5f * level * level * level
6         + 5f * level * level
7     );
8 }
```



```
1 public void GainExp(int exp)
2 {
3     stats.EXP += exp;
4
5     while (stats.EXP >= ExpToNextLevel(stats.Level))
6     {
7         stats.EXP -= ExpToNextLevel(stats.Level);
8         stats.Level++;
9         LevelUp();
10    }
11 }
```



ศัตรู



แนวคิด Enemy Scaling

1. Enemy ไม่ใช่สูตรเดียวกับ Player
2. Enemy จะ “อ่อนกว่า Player เล็กน้อย” ใน 1:1
3. ความยากมาจาก
 1. จำนวน
 2. ประเภท
 3. Elite / Boss
4. Level สูง = อัดขึ้น + แรงขึ้น แต่ไม่ one-shot



Enemy Stat Modifier

EnemyStat =

(BaseStat

+ Level × Linear

+ $\sqrt{\text{Level}} \times \text{Curve}$)

× TypeModifier



Enemy Type Stat Modifier (TypeModifier)

Type	HP	STR	DEF	EXP
Normal	x1.0	x1.0	x1.0	x1.0
Elite	x1.5	x1.3	x1.2	x1.5
Boss	x3.0	x1.8	x1.6	x3.0



Enemy Type

Enemy	STR	INT	AGI	พสกักรรรม
Red (Warrior)	สูง	ต่ำ	กลาง	Attack / Guard
Blue (Mage)	ต่ำ	สูง	กลาง	Magic / Heal
Green (Archer)	กลาง	ต่ำ	สูง	Attack / Run



Red Cube – Warrior Enemy

Stat	Base	Linear	Curve
STR	10	1	1.5
DEF	9	0.9	1.4
INT	4	0.3	0.5
AGI	6	0.6	0.8
HP	100	22	30



	A	B	C	D	E	F	G	H	I	J	K
1	Stat	Base	Linear	Curve		Level	STR	DEF	INT	AGI	HP
2	STR	10	1	1.5		1	13	11	5	7	152
3	DEF	9	0.9	1.4		2	14	13	5	8	186
4	INT	4	0.3	0.5		3	16	14	6	9	218
5	AGI	6	0.6	0.8		4	17	15	6	10	248
6	HP	100	22	30		5	18	17	7	11	277
7						6	20	18	7	12	305
8						7	21	19	7	12	333
9						8	22	20	8	13	361
10						9	24	21	8	14	388
11						10	25	22	9	15	415
12						11	26	24	9	15	441
13						12	27	25	9	16	468
14						13	28	26	10	17	494
15						14	30	27	10	17	520
16						15	31	28	10	18	546
17						16	32	29	11	19	572
18						17	33	30	11	19	598
19						18	34	31	12	20	623
20						19	36	32	12	21	649
21						20	37	33	12	22	674
22						21	38	34	13	22	699



Blue Cube – Mage Enemy

Stat	Base	Linear	Curve
STR	4	0.3	0.5
DEF	5	0.5	0.8
INT	12	1.3	2
AGI	5	0.5	0.7
HP	70	15	22



	A	B	C	D	E	F	G	H	I	J	K
1	Stat	Base	Linear	Curve		Level	STR	DEF	INT	AGI	HP
2	STR	4	0.3	0.5		1	5	6	15	6	107
3	DEF	5	0.5	0.8		2	5	7	17	7	131
4	INT	12	1.3	2		3	6	8	19	8	153
5	AGI	5	0.5	0.7		4	6	9	21	8	174
6	HP	70	15	22		5	7	9	23	9	194
7						6	7	10	25	10	214
8						7	7	11	26	10	233
9						8	8	11	28	11	252
10						9	8	12	30	12	271
11						10	9	13	31	12	290
12						11	9	13	33	13	308
13						12	9	14	35	13	326
14						13	10	14	36	14	344
15						14	10	15	38	15	362
16						15	10	16	39	15	380
17						16	11	16	41	16	398
18						17	11	17	42	16	416
19						18	12	17	44	17	433
20						19	12	18	45	18	451
21						20	12	19	47	18	468
22						21	13	19	48	19	486



Green Cube – Archer Enemy

Stat	Base	Linear	Curve
STR	8	0.8	1.2
DEF	7	0.7	1
INT	6	0.5	0.7
AGI	12	1.1	1.8
HP	85	18	26



	A	B	C	D	E	F	G	H	I	J	K
1	Stat	Base	Linear	Curve		Level	STR	DEF	INT	AGI	HP
2	STR	8	0.8	1.2		1	10	9	7	15	129
3	DEF	7	0.7	1		2	11	10	8	17	158
4	INT	6	0.5	0.7		3	12	11	9	18	184
5	AGI	12	1.1	1.8		4	14	12	9	20	209
6	HP	85	18	26		5	15	13	10	22	233
7						6	16	14	11	23	257
8						7	17	15	11	24	280
9						8	18	15	12	26	303
10						9	19	16	13	27	325
11						10	20	17	13	29	347
12						11	21	18	14	30	369
13						12	22	19	14	31	391
14						13	23	20	15	33	413
15						14	24	21	16	34	434
16						15	25	21	16	35	456
17						16	26	22	17	37	477
18						17	27	23	17	38	498
19						18	27	24	18	39	519
20						19	28	25	19	41	540
21						20	29	25	19	42	561
22						21	30	26	20	43	582



Warrior

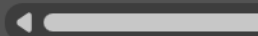
Mage

Archer

E-Warrior

E-Mage

E-Archer



Gold & Reward Formula

Gold =

EnemyLevel × Random(5-10)



Run Formula

Run =

$$\frac{(\text{PlayerAGI} + \sqrt{\text{Level} \times 5})}{(\text{EnemyAGI} + \sqrt{\text{Level} \times 5})} \times 50\%$$



Critical Formula

Chance = $AGI \times 0.3\%$

Damage = $\times 1.5$



Battle Balance Check (1:1)

สถานการณ์	ผลลัพธ์
Player vs Normal	ชนะสบาย
Player vs Elite	ต้องใช้ Heal
Player vs Boss	10-15 เกิร์น
Player Lv ต่ำกว่า	เสี่ยงตาย



```
1 public static int EnemyStat(  
2     int baseStat,  
3     float linear,  
4     float curve,  
5     int level,  
6     float modifier)  
7 {  
8     float value =  
9         baseStat  
10        + level * linear  
11        + Mathf.Sqrt(level) * curve;  
12  
13    return Mathf.RoundToInt(value * modifier);  
14 }
```



แนวคิด Boss Scaling

- ไม่ one-shot ผู้เล่น
- ไม่เป็นทุงเลือดอย่างเดียว
- มี “จังหวะการสู้”
- เลเวลสูง Boss จะฉลาดขึ้น ไม่ใช่แค่เลือดเยอะ



แนวคิด Boss Scaling

ดังนั้น Boss Scaling จะมี 4 แนว

- Stat Scaling
- Phase System
- Skill Unlock ตาม HP
- Enrage (กดดันช่วงท้าย)



Boss Level Formula

BossLevel = PlayerLevel + Offset



Boss Offset

ประเภทบอส	Offset
Mini Boss	+1
Area Boss	+2
Story Boss	+3
Final Boss	+5



Boss Stat Modifier

Stat	Modifier
HP	×3.0
STR	×1.8
DEF	×1.6
INT	×1.6
AGI	×1.3
EXP	×3.0
Gold	×2.5



Boss Phase System (หัวใจของบอส)

Phase	HP%	พฤติกรรม
Phase 1	100-70%	โจมตีปกติ
Phase 2	70-40%	ใช้สกิล / บัพ
Phase 3	40-20%	ดาเมจแรงขึ้น
Phase 4	20-0%	Enrage



Boss Phase System (หัวใจของบอส)

Phase	Damage	Defense	AI
Phase 1	$\times 1.0$	$\times 1.0$	Basic
Phase 2	$\times 1.2$	$\times 1.1$	Skill
Phase 3	$\times 1.4$	$\times 1.0$	Aggressive
Phase 4	$\times 1.6$	$\times 0.9$	All-in



Boss-Warrior Skill Unlock Table

HP%	Skill
≥70%	Heavy Slash
70%	Defense Up
40%	Power Strike
20%	Rage Mode



Boss-Mage Skill Unlock Table

HP%	Skill
≥70%	Fire
70%	Magic Shield
40%	Meteor
20%	Desperation Heal



Boss-Archer Skill Unlock Table

HP%	Skill
≥70%	?
70%	?
40%	?
20%	?



ตัวอย่าง Boss AI Decision Table

สถานการณ์	Action
$\geq 70\%$	Attack
70%	Buff
40%	Skill
20%	Enrage
MP ต่ำ	Basic Attack



Enrage System (ช่วงสุดท้าย เมื่อ HP \leq 20%)

Damage \times 1.6

Heal Effect \times 0.5

Guard Effect \times 0.7



จำกัด Damage ต่อเทิร์น

$$\text{MaxDamage} = \text{PlayerMaxHP} \times 0.35$$



จำกัด Heal ของ Boss

$$\text{BossHeal} \leq \text{MaxHP} \times 0.15$$



ตัวอย่าง Boss Scaling (Level 30)

Player Level 30

HP \approx 800

Damage \approx 50

Boss (Lv 33)

HP \approx 800 \times 3 = 2,400

Damage \approx 50 \times 1.8 = 90

Fight \approx 12–15 เท้สู



```
1 public void ApplyBossScaling(int playerLevel)
2 {
3     level = playerLevel + bossOffset;
4
5     stats.MaxHP = Mathf.RoundToInt(stats.MaxHP * 3f);
6     stats.STR = Mathf.RoundToInt(stats.STR * 1.8f);
7     stats.DEF = Mathf.RoundToInt(stats.DEF * 1.6f);
8     stats.INT = Mathf.RoundToInt(stats.INT * 1.6f);
9     stats.AGI = Mathf.RoundToInt(stats.AGI * 1.3f);
10 }
```



FSM



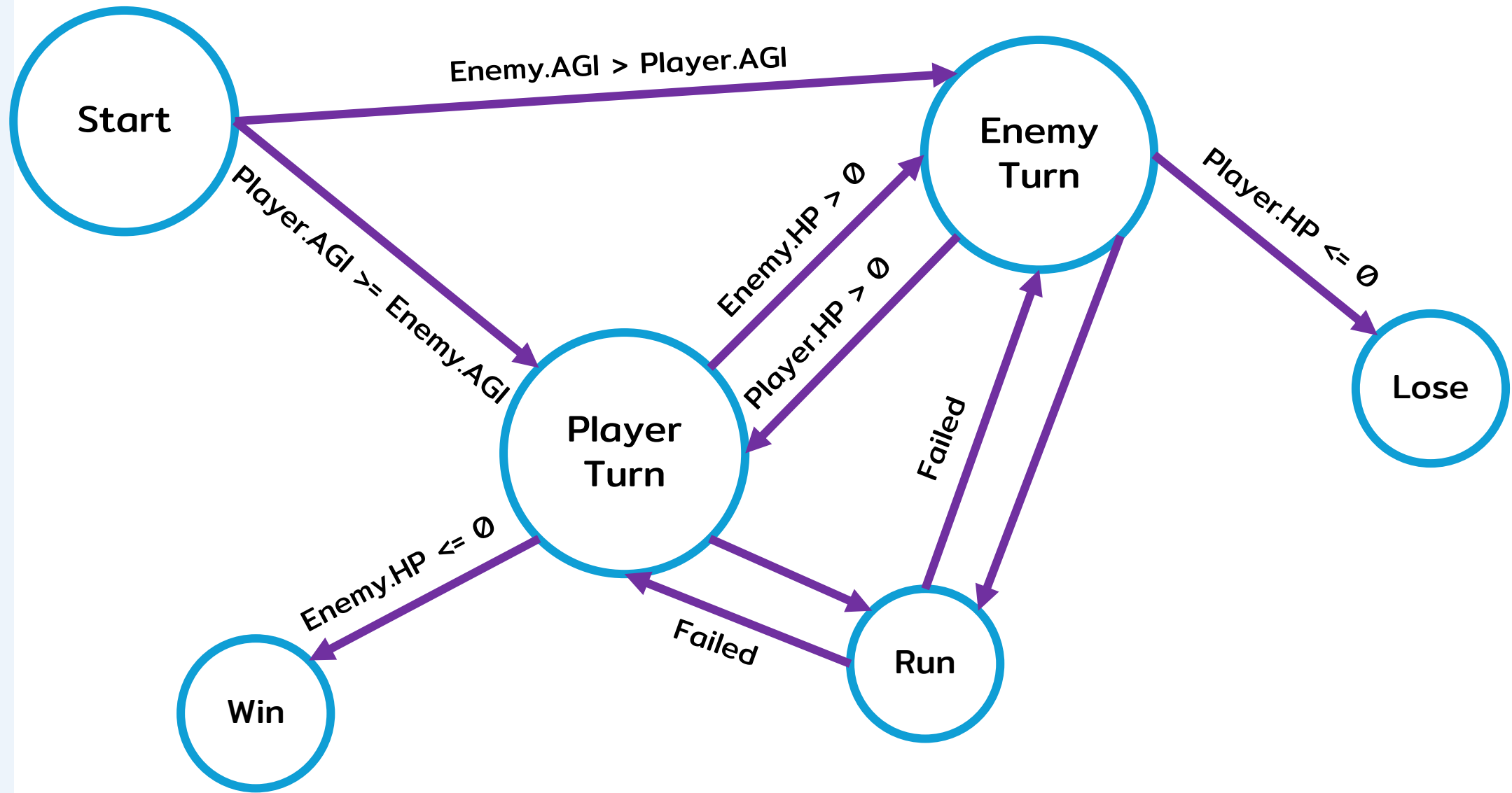
การออกแบบ Finite State Machine

สถานะ (State) ของการต่อสู้

1. START เตรียมข้อมูลผู้เล่น สุ่มสร้างศัตรู (Warrior, Mage, หรือ Archer)
 1. ถ้า $\text{Player.AGI} \geq \text{Enemy.AGI}$ เปลี่ยนสถานะเป็น Player Turn
 2. ถ้า $\text{Enemy.AGI} > \text{Player.AGI}$ เปลี่ยนสถานะเป็น Enemy Turn
2. Player Turn รอผู้เล่นกดปุ่มเลือกคำสั่ง (Attack, Defense, Heal, Run)
3. Enemy Turn คำนวณ AI ของศัตรู (สุ่ม 1-3)
4. Win สถานการณ์ชนะ ($\text{Enemy.HP} \leq 0$)
5. Lose สถานะการพ่ายแพ้ ($\text{Player.HP} \leq 0$)
6. Run สถานะการหนี



FSM





BattleState

```
1      using UnityEngine;
2
3      0 references
4      public enum BattleState
5      {
6          Start,
7          PlayerTurn,
8          EnemyTurn,
9          Win,
10         Lose,
11         Run
12     }
```





BattleCommand

```
1 using UnityEngine;
2
3 0 references
4 public enum BattleCommand
5 {
6     Attack,
7     Defense,
8     Heal,
9     Run
}
```



Attack

```
int Damage(int attackerSTR, int defenderDEF)
{
    int dmg = attackerSTR - defenderDEF / 2;
    return Mathf.Max(1, dmg + Random.Range(-2, 3));
}
```



Defense : ลดดาเมจรอบกั๊ตไป

```
bool isDefending;
```

```
if (isDefending)
```

```
    damage /= 2;
```



Heal : លើក HP ໂດຍໃຊ້ MP

```
int HealAmount()  
{  
    return INT * 2 + Random.Range(0, 5);  
}  
  
player.MP -= 5;  
  
player.HP = Mathf.Min(player.MaxHP, player.HP +  
HealAmount());
```



Run : кӯ

```
bool TryRun(int playerAGI, int enemyAGI)
{
    int chance = 50 + (playerAGI - enemyAGI) * 5;
    return Random.Range(0, 100) < chance;
}
```



SEUU EXP / Level Up

```
int NeedEXP(int level)
{
    return level * level * 10;
}
if (player.EXP >= NeedEXP(player.Level))
{
    player.Level++;
    player.MaxHP += 5;
    player.STR += 2;
    player.DEF += 1;
}
```



AI ศัตรู : สุ่มคำสั่ง

```
BattleCommand EnemyDecision()
{
    int r = Random.Range(0, 100);
    if (enemy.MP >= 5 && r < 30)
        return BattleCommand.Heal;
    if (r < 80)
        return BattleCommand.Attack;
    return BattleCommand.Defense;
}
```



AI ศัตรู : แนวคิดแบบง่าย (Priority + Random)

ใช้เงื่อนไขก่อนแล้วค่อยสุ่ม เช่น

1. ถ้า HP ต่ำจะต้องพิจารณา Heal
2. ถ้า MP หมดจะต้อง ห้าม Heal
3. จากนั้นสุ่มตาม “นิสัยอาชีพ”



AI ศัตรู : นิสัยอาชีพนักรบ

บุคลิก

1. ชอบโจมตีตรง ๆ
2. DEF สูง โดยมีการใช้ Defense บ้าง
3. แทบไม่หนี



พฤติกรรม

1. Attack บ่อยที่สุด
2. Defense เมื่อ HP ต่ำ
3. Heal บ่อยมาก



```
BattleCommand WarriorAI(CharacterStat enemy)
```

```
{
```

```
    if (enemy.HP < enemy.MaxHP * 0.3f && Random.Range(0,100) < 40)
```

```
        return BattleCommand.Defense;
```

```
    int r = Random.Range(0, 100);
```

```
    if (r < 70) return BattleCommand.Attack;
```

```
    if (r < 90) return BattleCommand.Defense;
```

```
    return BattleCommand.Attack;
```

```
}
```



AI ศัตรู : นิสัยอาชีพนักเวทย์

บุคลิก

1. ชอบใช้ Heal
2. ตัวบางชอบหนีบ่อย
3. โจมตีแรง แต่ไม่สม่ำเสมอ



พฤติกรรม

1. Heal เมื่อ HP ต่ำ
2. ถ้า MP หมดจะ Attack
3. HP ต่ำ และ MP ต่ำจะ Run



```
BattleCommand MageAI(CharacterStat enemy)
{
    if (enemy.HP < enemy.MaxHP * 0.4f && enemy.MP >= 5)
        return BattleCommand.Heal;
    if (enemy.HP < enemy.MaxHP * 0.25f && enemy.MP < 5)
        return BattleCommand.Run;
    int r = Random.Range(0, 100);
    if (r < 60) return BattleCommand.Attack;
    if (r < 80) return BattleCommand.Defense;

    return BattleCommand.Attack;
}
```



AI ศัตรู : นิสัยอาชีพนักธนู

บุคลิก

1. ว่องไว
2. ชอบโจมตี แต่ระวังตัว
3. หนีได้ดี



พฤติกรรม

1. Attack เป็นหลัก
2. Defense เมื่อ HP ต่ำ
3. Run ถ้าเสียเปรียบ



```
BattleCommand ArcherAI(CharacterStat enemy)
{
    if (enemy.HP < enemy.MaxHP * 0.3f && Random.Range(0,100) < 30)
        return BattleCommand.Run;

    int r = Random.Range(0, 100);

    if (r < 60) return BattleCommand.Attack;
    if (r < 85) return BattleCommand.Defense;

    return BattleCommand.Attack;
}
```



AI ศัตรู : ตัวเลือก AI (Dispatcher)

```
BattleCommand DecideEnemyAction(EnemyType type, CharacterStat enemy)
{
    switch (type)
    {
        case EnemyType.Warrior: return WarriorAI(enemy);
        case EnemyType.Mage: return MageAI(enemy);
        case EnemyType.Archer: return ArcherAI(enemy);
    }
    return BattleCommand.Attack;
}
```



ทำให้ AI ฉลาดขึ้น

1. เช็ค HP ผู้เล่น

```
if (player.HP < 10)  
    return BattleCommand.Attack;
```

2. เช็ค Turn ที่แล้ว

```
BattleCommand lastPlayerCommand;  
if (lastPlayerCommand == BattleCommand.Defense)  
    return BattleCommand.Attack;
```

3. Critical Behavior

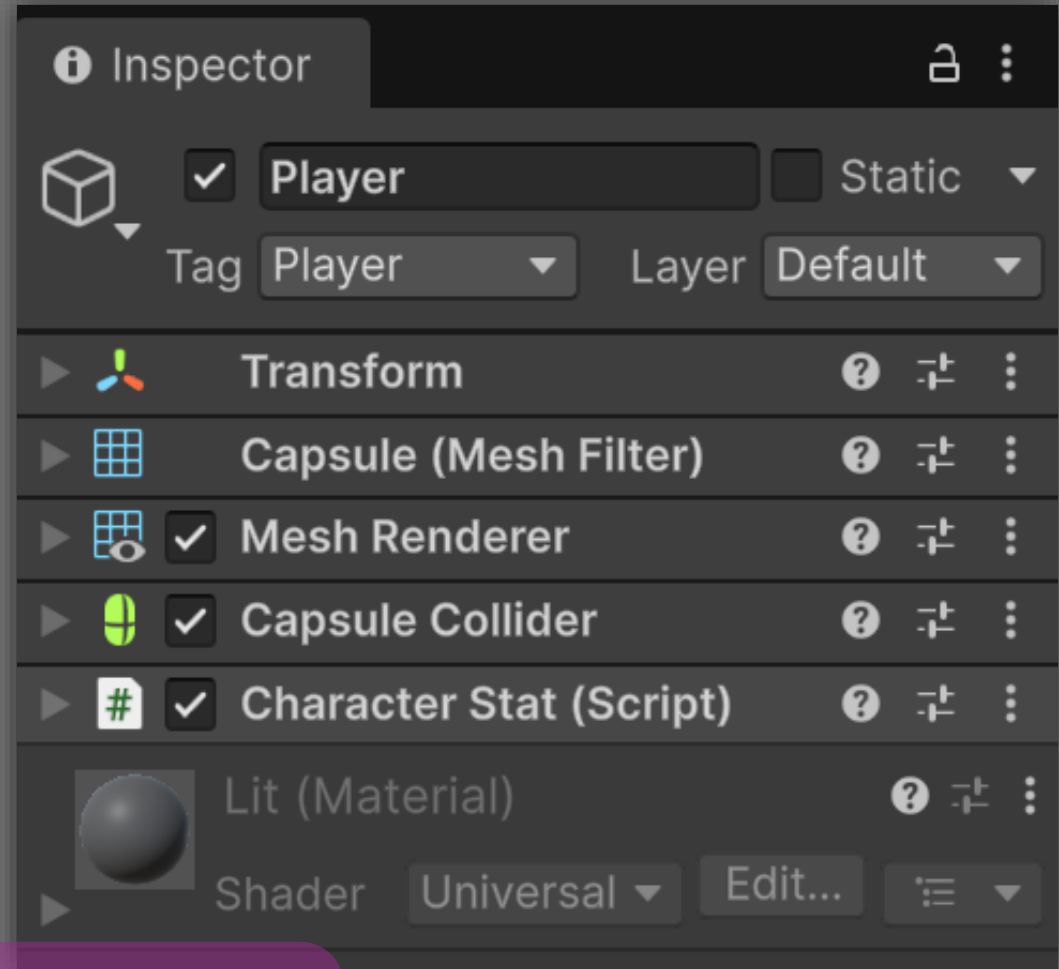
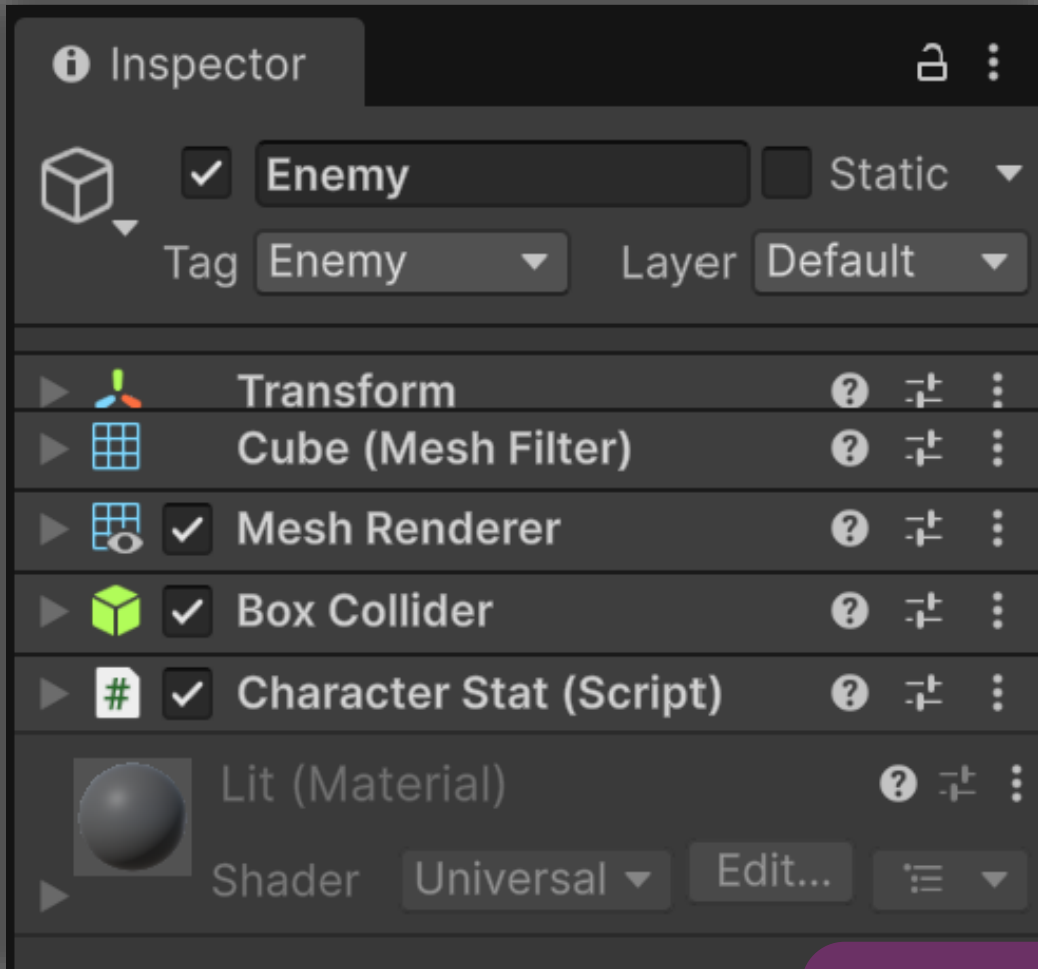
```
if (enemy.AGI > player.AGI && Random.Range(0,100) < 20)  
    return BattleCommand.Attack;
```



ตารางสรุป AI

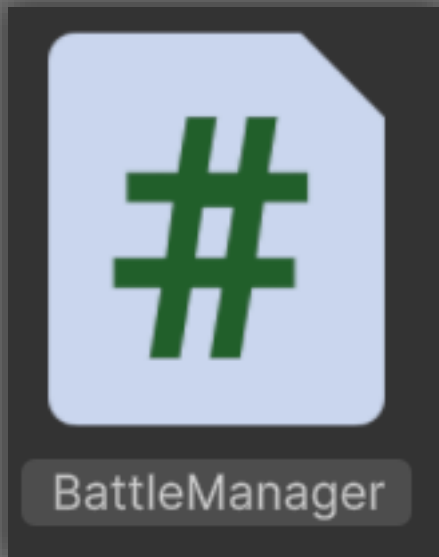
ศัตรู	Attack	Defense	Heal	Run
นักรบ	✓✓✓✓	✓✓✓	✓	✗
นักเวทย์	✓✓✓	✓✓	✓✓✓✓	✓✓
นักธนู	✓✓✓✓	✓✓✓	✗	✓✓✓





เตรียม Prefab ของ
Enemy และ Player





พบเจอกับ BattleManager

▼ # ✓ Battle Manager (Script) ? ⚙ ⋮

Script # BattleManager ⦿

Prefabs

Player Prefab 3D Player ⦿

Player Job Warrior ▼

Enemy Prefab 3D Enemy ⦿

SpawnPosition

Player Spawn 📍 PlayerPosition (Tr) ⦿

Enemy Spawn 📍 EnemyPosition (T) ⦿

UI

Message Text [T] messageText (Te) ⦿

Player Status Text [T] playerStatusText ⦿

Enemy Status Text [T] enemyStatusText ⦿

Battle

State Start ▼



```
1  ✓ using UnityEngine;
2  |   using System.Collections;
3  |   using TMPro;
4  |
5  |   ▾ Unity Script (1 asset reference) | 0 references
6  |   ✓ public class BattleManager : MonoBehaviour
7  |   |   {
8  |   |       [Header("Prefabs")]
9  |   |       public GameObject playerPrefab;
10 |   |       public JobType playerJob;
11 |   |       public GameObject enemyPrefab;
12 |   |
13 |   |       [Header("SpawnPosition")]
14 |   |       public Transform playerSpawn;
15 |   |       public Transform enemySpawn;
```



```
16 [Header("UI")]
17 public TextMeshProUGUI messageText;
18 public TextMeshProUGUI playerStatusText;
19 public TextMeshProUGUI enemyStatusText;
20
21 [Header("Battle")]
22 public BattleState state;
23
24 bool playerDefending;
25 bool enemyDefending;
26
27 private Unity.Mathematics.Random rnd;
28
29 private GameObject player;
30 private CharacterStat playerStat;
31 private GameObject enemy;
32 private CharacterStat enemyStat;
33 private EnemyType enemyType;
34
```



```
35 private void Awake()  
36 {  
37     uint seed = (uint)System.DateTime.Now.Ticks;  
38     rnd = new Unity.Mathematics.Random(seed == 0 ? 1 : seed);  
39 }  
40
```



```
41 void Start()
42 {
43     // Spawn Player
44     player = Instantiate(playerPrefab, playerSpawn.position, playerSpawn.rotation);
45     playerStat = player.GetComponent<CharacterStat>();
46
47     // Spawn Enemy
48     enemy = Instantiate(enemyPrefab, enemySpawn.position, enemySpawn.rotation);
49     enemyStat = enemy.GetComponent<CharacterStat>();
50
51     SetupBattle();
52 }
53
```



```
54  void SetupBattle()
55  {
56      state = BattleState.Start;
57
58      SetupPlayer();
59
60      // สุ่มอาชีพของศัตรู
61      enemyType = (EnemyType)rnd.NextInt(0, 3);
62      SetupEnemy(enemyType);
63
64      UpdateStatusUI();
65
66      StartCoroutine(StartBattle());
67  }
68
```



```
69  ✓ void SetupPlayer()
70  {
71  ✓   switch (playerJob)
72  {
73  ✓     case JobType.Warrior:
74      playerStat.MaxHP = 180;
75      playerStat.HP = 180;
76      playerStat.MaxMP = 30;
77      playerStat.MP = 30;
78      playerStat.STR = 15;
79      playerStat.DEF = 13;
80      playerStat.AGI = 8;
81      playerStat.INT = 5;
82      break;
```



```
83  ✓
84
85
86
87
88
89
90
91
92  case JobType.Mage:
    playerStat.MaxHP = 120;
    playerStat.HP = 120;
    playerStat.MaxMP = 98;
    playerStat.MP = 98;
    playerStat.STR = 5;
    playerStat.DEF = 6;
    playerStat.AGI = 8;
    playerStat.INT = 16;
    break;
```



```
93     case JobType.Archer:
94         playerStat.MaxHP = 150;
95         playerStat.HP = 150;
96         playerStat.MaxMP = 60;
97         playerStat.MP = 60;
98         playerStat.STR = 10;
99         playerStat.DEF = 9;
100        playerStat.AGI = 15;
101        playerStat.INT = 7;
102        break;
103    }
104    SetPlayerColor(playerJob);
105 }
```



106

1 reference

107

`void SetupEnemy(EnemyType type)`

108

`{`

109

`enemyStat.Level = playerStat.Level;`

110

`enemyStat.MaxHP = type switch`

111

`{`

112

`EnemyType.Warrior => 180,`

113

`EnemyType.Mage => 120,`

114

`EnemyType.Archer => 150,`

115

`_ => 0`

116

`};`

117

`enemyStat.HP = enemyStat.MaxHP;`

118



```
119  ✓
120
121  {
122      EnemyType.Warrior => 30,
123      EnemyType.Mage => 98,
124      EnemyType.Archer => 60,
125      _ => 0
126  };
127
128  enemyStat.MP = enemyStat.MaxMP;
129
130  ✓
131  enemyStat.STR = type switch
132  {
133      EnemyType.Warrior => 15,
134      EnemyType.Mage => 5,
135      EnemyType.Archer => 10,
136      _ => 5
137  };
```




```
152     enemyStat.INT = type switch
153     {
154         EnemyType.Warrior => 5,
155         EnemyType.Mage => 16,
156         EnemyType.Archer => 7,
157         _ => 5
158     };
159
160     SetEnemyColor(type);
161 }
162
```



```
163 void SetEnemyColor(EnemyType type)
164 {
165     Color c = Color.white;
166     if (type == EnemyType.Warrior) c = Color.red;
167     if (type == EnemyType.Mage) c = Color.blue;
168     if (type == EnemyType.Archer) c = Color.green;
169
170     enemy.GetComponent<MeshRenderer>().material.color = c;
171 }
172
```



```
173 void SetPlayerColor(JobType type)
174 {
175     Color c = Color.white;
176     if (type == JobType.Warrior) c = Color.red;
177     if (type == JobType.Mage) c = Color.blue;
178     if (type == JobType.Archer) c = Color.green;
179
180     player.GetComponent<MeshRender>().material.color = c;
181 }
182
```



```
183  ✓  IEnumerator StartBattle()
184  {
185  |   messageText.text = "The enemy has appeared!";
186  |   yield return new WaitForSeconds(1.5f);
187  |
188  |   // เลือกว่าใครโจมตีก่อนจากค่า AGI
189  |   ✓  if (playerStat.AGI >= enemyStat.AGI)
190  |   {
191  |       state = BattleState.PlayerTurn;
192  |       PlayerTurn();
193  |   }
194  |   ✓  else
195  |   {
196  |       state = BattleState.EnemyTurn;
197  |       StartCoroutine(EnemyTurn());
198  |   }
199  }
200
```



```
201      ✓
202      |
203      |█
204      |
205      |█
206      |

void PlayerTurn()
{
    messageText.text = "Select a command.";
    playerDefending = false;
}
```



```
207  ✓  IEnumerator EnemyTurn()
208      {
209          state = BattleState.EnemyTurn;
210          enemyDefending = false;
211
212          yield return new WaitForSeconds(1f);
213
214          BattleCommand cmd = DecideEnemyAction();
215
216          ExecuteEnemyCommand(cmd);
217          UpdateStatusUI();
218
```



```
219     if (playerStat.HP <= 0)
220     {
221         state = BattleState.Lose;
222         messageText.text = "You've lost...";
223         Object.Destroy(player);
224         yield break;
225     }
226
227     yield return new WaitForSeconds(1f);
228
229     state = BattleState.PlayerTurn;
230     PlayerTurn();
231 }
232
```



```
233     public void OnAttack()
234     {
235         if (state != BattleState.PlayerTurn) return;
236
237         int dmg = CalculateDamage(playerStat.STR, enemyStat.DEF, enemyDefending);
238         enemyStat.HP -= dmg;
239         messageText.text = $"You attacked the enemy, causing {dmg} HP loss.";
240         UpdateStatusUI();
241         if (enemyStat.HP <= 0)
242         {
243             state = BattleState.Win;
244             messageText.text = "You win!";
245             Object.Destroy(enemy);
246             return;
247         }
248         StartCoroutine(EnemyTurn());
249     }
250
```



```
251     public void OnDefense()  
252     {  
253         if (state != BattleState.PlayerTurn) return;  
254  
255         playerDefending = true;  
256         messageText.text = "You are defending.";  
257  
258         StartCoroutine(EnemyTurn());  
259     }  
260
```



```
261 public void OnHeal()  
262 {  
263     if (state != BattleState.PlayerTurn || playerStat.MP < 5) return;  
264  
265     playerStat.MP -= 5;  
266     int heal = CalculateHeal(playerStat.INT);  
267     playerStat.HP = Mathf.Min(playerStat.MaxHP, playerStat.HP + heal);  
268  
269     messageText.text = $"You use Heal to restore {heal} HP.";  
270     UpdateStatusUI();  
271  
272     StartCoroutine(EnemyTurn());  
273 }  
274
```



```
275     public void OnRun()
276     {
277         if (state != BattleState.PlayerTurn) return;
278
279         if (TryRun(playerStat.AGI, enemyStat.AGI))
280         {
281             state = BattleState.Run;
282             messageText.text = "You escaped successfully!";
283         }
284         else
285         {
286             messageText.text = "Escape failed!";
287             StartCoroutine(EnemyTurn());
288         }
289     }
290
```



```
291  ✓ BattleCommand DecideEnemyAction()
292  {
293  ✓   switch (enemyType)
294  {
295  ✓     case EnemyType.Warrior:
296  |       return WarriorAI();
297
298  ✓     case EnemyType.Mage:
299  |       return MageAI();
300
301  ✓     case EnemyType.Archer:
302  |       return ArcherAI();
303  }
304  return BattleCommand.Attack;
305  }
306
```



```
307 BattleCommand WarriorAI()
308 {
309     if (enemyStat.HP < enemyStat.MaxHP * 0.3f && rnd.NextFloat() < 0.4f)
310         return BattleCommand.Defense;
311
312     return BattleCommand.Attack;
313 }
314
```



```
315  ✓ BattleCommand MageAI()
316  {
317  |   if (enemyStat.HP < enemyStat.MaxHP * 0.4f && enemyStat.MP >= 5)
318  |       return BattleCommand.Heal;
319
320  |   if (enemyStat.HP < enemyStat.MaxHP * 0.25f && enemyStat.MP < 5)
321  |       return BattleCommand.Run;
322
323  |   return BattleCommand.Attack;
324  | }
325
```



```
326  BattleCommand ArcherAI()
327  {
328      if (enemyStat.HP < enemyStat.MaxHP * 0.3f && rnd.NextFloat() < 0.3f)
329          return BattleCommand.Run;
330
331      if (rnd.NextFloat() < 0.2f)
332          return BattleCommand.Defense;
333
334      return BattleCommand.Attack;
335  }
336
```



```
337     void ExecuteEnemyCommand(BattleCommand cmd)
338     {
339         switch (cmd)
340         {
341             case BattleCommand.Attack:
342                 int dmg = CalculateDamage(enemyStat.STR, playerStat.DEF, playerDefending);
343                 playerStat.HP -= dmg;
344                 messageText.text = $"Enemy attack. You lose {dmg} HP.";
345                 break;
346
347             case BattleCommand.Defense:
348                 enemyDefending = true;
349                 messageText.text = "The enemy is defending.";
350                 break;
351         }
```



```
352     case BattleCommand.Heal:
353         enemyStat.MP -= 5;
354         int heal = CalculateHeal(enemyStat.INT);
355         enemyStat.HP = Mathf.Min(enemyStat.MaxHP, enemyStat.HP + heal);
356         messageText.text = $"The enemy uses Heal to restore {heal} HP.";
357         break;
358
359     case BattleCommand.Run:
360         messageText.text = "The enemy tried to escape but failed.";
361         break;
362     }
363 }
364
```



```
365  ✓
366
367  |
368  | int CalculateDamage(int atk, int def, bool defending)
369  | {
370  |     int dmg = atk - def / 2;
371  |     dmg += rnd.NextInt(-2, 3);
372  |
373  |     if (defending)
374  |         dmg /= 2;
375  |
376  |     return Mathf.Max(1, dmg);
377  | }
378  |
```



376



377

378



379

380

```
int CalculateHeal(int intel)
{
    return intel * 2 + rnd.NextInt(0, 5);
}
```



```
381     ✓
382     {
383         int chance = 50 + (agi1 - agi2) * 5;
384         return rnd.NextInt(0, 100) < chance;
385     }
386
```



```
387  void UpdateStatusUI()
388  {
389      playerStatusText.text =
390          $"[P] HP {playerStat.HP}/{playerStat.MaxHP} MP {playerStat.MP}/{playerStat.MaxMP}";
391
392      enemyStatusText.text =
393          $"[E] HP {enemyStat.HP}/{enemyStat.MaxHP}";
394  }
395  }
396
```



ปุ่ม BattleUI

1. btnCmdAttack เชื่อมกับ OnAttack()
2. btnCmdDefense เชื่อมกับ OnDefense()
3. btnCmdHeal เชื่อมกับ OnHeal()
4. btnCmdRun เชื่อมกับ OnRun()



btnCmdAttack Static

Tag Untagged Layer UI

▶ Rect Transform

▶ Canvas Renderer

▶ Image

▼ Button

Interactable

Transition Color Tint

Target Graphic btnCmdAttack

Normal Color

Highlighted Color

Pressed Color

Selected Color

Disabled Color

Color Multiplier 1

Fade Duration 0.1

Navigation Automatic

Visualize

On Click ()

Runtime BattleManager.OnAtta

Battle

btnCmdDefense Static

Tag Untagged Layer UI

▶ Rect Transform

▶ Canvas Renderer

▶ Image

▼ Button

Interactable

Transition Color Tint

Target Graphic btnCmdDefense

Normal Color

Highlighted Color

Pressed Color

Selected Color

Disabled Color

Color Multiplier 1

Fade Duration 0.1

Navigation Automatic

Visualize

On Click ()

Runtime BattleManager.OnDefe

Battle



btnCmdHeal Static

Tag Untagged Layer UI

▶ Rect Transform

▶ Canvas Renderer

▶ Image

▼ Button

Interactable

Transition Color Tint

Target Graphic btnCmdHeal (In)

Normal Color

Highlighted Color

Pressed Color

Selected Color

Disabled Color

Color Multiplier 1

Fade Duration 0.1

Navigation Automatic

Visualize

On Click ()

Runtime BattleManager.OnHeal

Battle

btnCmdRun Static

Tag Untagged Layer UI

▶ Rect Transform

▶ Canvas Renderer

▶ Image

▼ Button

Interactable

Transition Color Tint

Target Graphic btnCmdRun (Im)

Normal Color

Highlighted Color

Pressed Color

Selected Color

Disabled Color

Color Multiplier 1

Fade Duration 0.1

Navigation Automatic

Visualize

On Click ()

Runtime BattleManager.OnRun

Battle



Q & A

