Stage 2

▼ Psuedo Codes for Pre-Run & On-Run Strategy

    ▼ Enemy

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Enemy : MonoBehaviour
{
    private EnemyBase eBase;

    private float maxStamina;
    private float maxFatigue;
    private float currentStamina;
    private float currentFatigue;

    private Enemy(EnemyBase eBase)
    {
        this.eBase = eBase;
        CalculateMaxStamina();
        CalculateMaxFatigue();
    }

    private void CalculateMaxStamina()
    {
        var a = eBase.Consistency;
        var b = eBase.Nature;
        // Some calculation between a and b
    }

    private void CalculateMaxFatigue()
    {
        var a = eBase.Consistency;
        var b = eBase.Nature;
        // Some calculation between a and b
    }

    public void StaminaDepletion(float d)
    {
        d += 1; //Some calculation
        currentStamina -= d;
    }
}
```

    ▼ EnemyBase

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu(fileName = "EnemyName", menuName = "ScriptableObjects/Enemy", order = 1)]
public class EnemyBase : ScriptableObject
{
    [Header("Rough Attributes")]
    [SerializeField] private float offense;
    [SerializeField] private float defense;
    [SerializeField] private float patience;
    [SerializeField] private float courage;
    [SerializeField] private float consistency;
    [SerializeField] private enemyNature nature;
    [Range(0.0f, 5.0f)][SerializeField] private float control;

    public float Consistency { get { return consistency; } }
    public enemyNature Nature { get { return nature; } }
    public enum enemyNature
    {
        Calm,
        Anger,
        Etc
    }
}
```
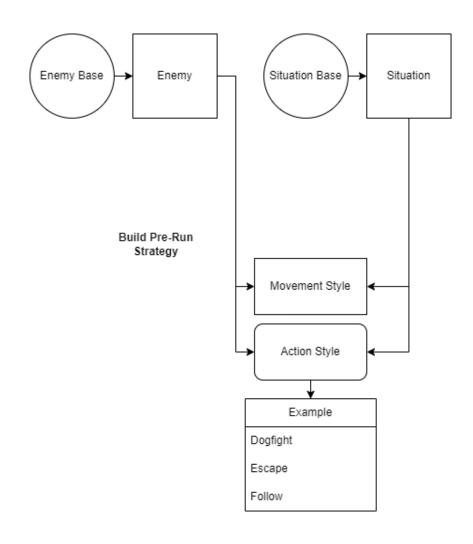
▼ Situation

   Similar to Enemy

▼ SituationBase

   Similar to Situation Base

▼ Movement Style

- Integration with Situation + Enemy ⇒ Base Movement Style

- Use pathfinding algorithms with variations depending on the Base Movement

- A* Pathfinding

- Algorithm Custom

▼ Action Style

- Integration with Situation + Enemy ⇒ Base Action STYLE

- Depends on Task ⇒ Dogfight, Escape, Sacrifice

- Algoritm Custom

Enemy Base → Enemy

Situation Base → Situation

**Build Pre-Run
Strategy**

Movement Style

Action Style

Example
Dogfight
Escape
Follow