# Склад проєкту

Проєкт складається з 10 классів:

1. DiceThrow
2. DiceSideChecker
3. GetAbility
4. Damage
5. Heal
6. Poison
7. Shield
8. Unit
9. BattleSystem
10. BattleHUD

3 інтерфейсів:

1. IAbility
2. IDamageable
3. IReinforced

1 структури:

1. CubeSide

Та 1 enum

1. BattleState

\*сюда общую картинку\*

# DiceThrow Изображение выглядит как текст Автоматически созданное описание

Клас, який підкидує GameObject. Повинен бути компонентом цього GameObject.

Складається з:

1. приватного поля rigidbody, що зберігає компонент Rigidbody з GameObject
2. публічного інкапсульованого поля DiceVelocity, що зберігає поточну швидкість кубика
3. приватного методу Awake(), що визивається при створенні об’єкту. Використовується для отримання компоненту Rigidbody
4. приватного методу Update(), що визивається на кожному фреймі. Використовується для оновлення значення швидкості DiceVelocity
5. публічного методу ThrowDice(). Використовується іншими класами для підкидання GameObject

# DiceSideChecker

Клас, який розпізнає куб та його сторону. Повинен бути компонентом GameObject з трігером, що буде розпізнавати.

Складається з:

1. публічного поля cubeSide, що зберігає куби та їх сторони
2. приватного методу OnTriggerStay()
3. приватного методу OnTriggerExit()
4. приватного методу GetCube()

# GetAbility

Клас, який отримує здатності куба та їх рівні.

Складається з:

1. публічного методу GetCubeAbilities()
2. публічного методу GetAbilityLevel()

# Damage, Heal, Shield, Poison

Класи, що несуть певні здібності: нанесення шкоди, відновлення здоров’я, нанесення щиту, отруєння. Повинні бути компонентами сторін куба

Складаються з:

1. публічного методу CastAbility()

# Unit Изображение выглядит как текст Автоматически созданное описание

Клас, що визначає кожний «живий» об’єкт. Повинен бути компонентом цього GameObject

Складається з:

1. приватного поля currentHealth
2. приватного поля maxHealth
3. приватного поля shield
4. приватного поля unitName
5. публічного методу TakeDamage()
6. публічного методу Damage()
7. публічного методу Heal()
8. публічного методу Shield()
9. публічного методу GetName()
10. публічного методу GetHealth()
11. публічного методу GetMaxHealth()
12. публічного методу GetShield()

# BattleSystem Изображение выглядит как текст Автоматически созданное описание

Клас, що визначає бойову систему гри. Повинен бути компонентом GameController

Складається з:

1. приватного поля player
2. приватного поля enemy
3. приватного поля playerPosition
4. приватного поля enemyPosition
5. приватного поля playerHUD
6. приватного поля enemyHUD
7. приватного поля playerSideChecker
8. приватного поля enemySideChecker
9. приватного поля playerCubes
10. приватного поля enemyCubes
11. приватного поля \_playerInformation
12. приватного поля \_enemyInformation
13. приватного поля \_getAbility
14. публічного поля State
15. публічного поля OnStateChangeHandler()
16. публічного поля OnStateChange
17. приватного методу Start()
18. приватного методу SetupBattle()
19. приватного методу PlayerTurn()
20. приватного методу EnemyTurn()
21. приватного методу EndBattle()

# BattleHUD Изображение выглядит как текст Автоматически созданное описание

Клас, що визначає бойовий юай. Повинен бути компонентом GameController.

Складається з:

1. публічного поля nameText
2. публічного поля health
3. публічного поля shield
4. публічного методу SetHUD()
5. публічного методу SetHealth()
6. публічного методу SetShield()

# IAbility

Інтерфейс, що визначає кожну здібність. Кожна нова здібність повинна наслідуватися від нього

Складається з:

1. Публічного методу CastAbility()

# IDamageable

Інтерфейс, що визначає кожний об’єкт, що може наносити шкоду та бути пошкодженим. Кожна така сутність повинна наслідуватися від нього

Складається з:

1. Публічного методу TakeDamage()
2. Публічного методу Damage()

# IReinforced

Інтерфейс, що визначає кожний об’єкт, який може лікуватися/мати щит/будь-яке інше підсинення. Кожна така сутність повинна наслідуватися від нього

Складається з:

1. Публічного методу Heal()
2. Публічного методу Shield()

# CubeSide

Структура для зберігання GameObject куба та номер його сторони.

Складається з:

1. Публічного поля Cube
2. Публічного поля Side
3. Публічного конструктору CubeSide()

# BattleState Изображение выглядит как текст Автоматически созданное описание

Enum, що визначає основі ігрові стадії.

На виконання цієї частини пішло 15-18 годин(18, якщо врахувати розбір в асетах).

Тепер частина, яку я нажаль не встиг виконати.

Для того, щоб вороги далі змінювалися, потрібно лише додати їх префаби, додати у лист(у класі BattleSystem), та при закінченні їх хп у EndBattle видаляти перший елемент з листів enemy та enemyCubes. Далі знову визивати SetupBattle.

Для зміни кімнат я використав би новий клас, RoomManager. Можна використовувати префаби кімнат та створювати їх у рантаймі, проте мобільні пристрої, скоріше за все, не витримають такий метод. Тож, як на мене, краще одразу створити різні кімнати, та просто рухати їх. Наприклад, взяти довжину по осі(в мене по х) та рухати за допомогою Lerp(щоб була рівна анімація переходу).