PATÓN FACADE Y SINGLETON

GameController

- gameController: GameController
- calculator: Calculator
- seDefiende: Boolean
- manager: Maneger

- roundCounter: Integer
- turnCounter: Integer
- startTurn: Integer

- currentWeapon: Integer

- player: Player
- enemy: EnemyTransition

- GameController()
+ getInstance(): GameController
+ game()
+ startGame()
+ playersTurn() : Integer
+ changeWorld(AbstractEnemyFactory)
+ randomEnemy(): EnemyTransition
+ play()

Calculator

- calculator: Calculator

- Calculator()
+ getInstance(): Calculator
+ applyDamageToPlayer(Integer round, EnemyTransition attacker., Player victim)
+ applyDamageToEnemy(Integer round, Player attacker., EnemyTransition victim, Integer currentWeapon, Integer optionDefense)
- calculateDamage(Player victim, int intensity)

+ hasDecoratorOf(CharacterAction , CharacterActionComponentDecorator): CharacterAction
+ instanceofHasComponentDecorator(CharacterAction, CharacterActionComponentDecorator: CharacterActionComponentDecorator

PATRÓN STRATEGY Y TEMPLATE

<<Interface>>
ServiceStrategy
+ applyStrategy(): void

Extends

<<Interface>>
EnemyStrategy
+ damagePlayer(): void
+ applyDefense(): void
+ enemyHasDied(): void

<<Abstract class>>
AbstractEnemyStrategy
enemy: EnemyTransition
+ applyStrategy(): void
+ damagePlayer(): void
+ applyDefense(): void
+ enemyHasDied(): void

Extends: **DefensiveStrategy**, **OffensiveStrategy**, **DeadStrategy**, **DefaultStrategy**

PATÓN DECORATOR

<<Interface>>
CharacterAction
+ getDescription(): String

<<Enum>>
CharacterType
+ Player
+ Enemy

<<Interface>>
ActionBaseComponent
+ getCharacterType(): CharacterType

<<Interface>>
CharacterActionComponentDecorator
+ getCharacterAction(): CharacterAction
+ setCharacterACtion(CharacterAction characterAction)

PlayerActionBaseComponent
- characterType: CharacterType
- dtoCharacterProperties: DTOCharacterProperties
+ PlayerActionBaseComponent(DtoCharacterProperties)
+ getDescription(): String
+ getCharacterType(): CharacterType
+ toString(): String

EnemyActionBaseComponent
- characterType: CharacterType
- dtoCharacterProperties: DTOCharacterProperties
+ EnemyActionBaseComponent(DtoCharacterProperties)
+ getDescription(): String
+ getCharacterType(): CharacterType
+ toString(): String

<<Abstract class>>
AbstractCharacterActionComponentDecorator
characterAction: CharacterAction
+ getCharacterAction(): CharacterAction
+ setCharacterAction(CharacterAction)

Extends: **IntensityAttackComponent**, **PochoAttackComponent**, **SwordAttackComponent**, **MagicAttackComponent**, **LeekAttackComponent**

<<Interface>>
Character
+ getLife(): Integer
+ getStrength(): Integer
+ getResistence(): Integer
+ play()

DTOCharacterProperties
- name: String
- life: Integer
- strength: Integer
- resistence: Integer
+ CharacterProperties(String, Integer, Integer, Integer)

PATRÓN ABSTRACT FACTORY

<<Interface>>
Enemy
+ getStrategy(): EnemyStrategy

<<Interface>>
EnemyTransition
+ setLife(int i)

<<Interface>>
Warrior
+ getWeapon(): CharacterAction

<<Interface>>
Warlock
+ getMagicPower(): CharacterAction

<<Interface>>
Miku
+ getShiny(): Boolean
+ getLeek(): CharacterAction

<<Abstract class>>
AbstractWarrior
dtocharacterProperties: DTOCharacterProperties
enemyStrategy: EnemyStrategy
weapon: CharacterAction
+ getEnemeyStrategy(): EnemyStrategy
+ getWeapon(): CharacterAction
+ play()
getters y setters de los atributos DTO

<<Abstract class>>
AbstractWarlock
dtocharacterProperties: DTOCharacterProperties
enemyStrategy: EnemyStrategy
magicPower: CharacterAction
+ getEnemeyStrategy(): EnemyStrategy
+ getMagicPower(): CharacterAction
+ play()
getters y setters de los atributos DTO

<<Abstract class>>
AbstractMiku
dtocharacterProperties: DTOCharacterProperties
enemyStrategy: EnemyStrategy
leek: CharacterAction
shiny: Boolean
+ getShiny(): Boolean
+ isShiny(): Boolean
+ getEnemeyStrategy(): EnemyStrategy
+ getLeek(): CharacterAction
+ play()
getters y setters de los atributos DTO

ForestWarrior
+ ForestWarrior()
+ ForestWarrior(String name)

OceanWarrior
+ OceanWarrior()
+ OceanWarrior(String name)

DesertWarrior
+ DesertWarrior()
+ DesertWarrior(String name)

ForestWarlock
+ ForestWarlock()
+ ForestWarlock(String name)

OceanWarlock
+ OceanWarlock()
+ OceanWarlokck(String name)

DesertWarlock
+ DesertWarlock()
+ DesertWarlock(String name)

ForestMiku
+ ForestMiku()
+ ForestMiku(String name)

OceanMiku
+ OceanMiku()
+ OceanMiku(String name)

DesertMiku
+ DesertMiku()
+ DesertMiku(String name)

ForestEnemyFactory

OceanEnemyFactory

DesertEnemyFactory

<<Interface>>
AbstractEnemyFactory
+ createWarrior(): Warrior
+ createWarlock(): Warlock
+ createMiku(): Miku

Manager
- manager: Maneger
- abstractFactory: AbstractEnemyFactory
+ private Manager()
+ getInstance(): Manager
+ getAbstractEnemyFactory(): AbstractEnemyFactory
+ createWarrior(): Warrior

PATRÓN STATE

Context de State Pattern

Player
- dtocharacterProperties: DTOCharacterProperties
- inventory: List<CharacterActionComponentDecorator>
- currentState: PlayerState
- damagedState: PlayerState
- paralizedState: PlayerState
- activeState: PlayerState
- characterAction: CharacterAction
+ Player(String name, Integer life, Integer strength, Integer resistence)
+ addWeaponInventory(): List<CharacterComponentDecorator>
+ getCurrentState(): PlayerState
+ setCurrentState(PlayerState)
+ getDamagedState(): PlayerState
+ getParalizedState(): PlayerState
+ getActiveState(): PlayerState
+ getInventory(): List<CharacterComponentDecorator>
+ getCharacterAction(): CharacterAction
+ setCharacterAction(CharcterAction characterAction)
+ play()
getters y setter de DTO

<<Interface>>
PlayerState
+ process(): void

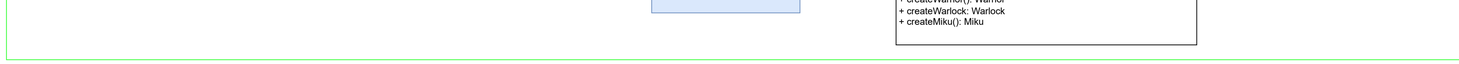
<<Interface>>
PlayerStateTransition
+ actionDamaged()
+ actionParalized()
+ actionActiveState()

<<Abstract class>>
AbstracPlayerState
player: Player
+ process()
+ actionDamagedState()
+ actionParalizedState()
+ actionActiveState()

DamagedState
+ DamagedState(Player)
+ process()
+ actionActiveState()

ParalizedState
+ ParalizedState(Player)
+ process()
+ actionActiveState()

ActiveState
+ ActiveState(Player)
+ process()
+ actionActiveState()


+ createWarrior(): Warrior
+ createWarlock: Warlock
+ createMiku(): Miku