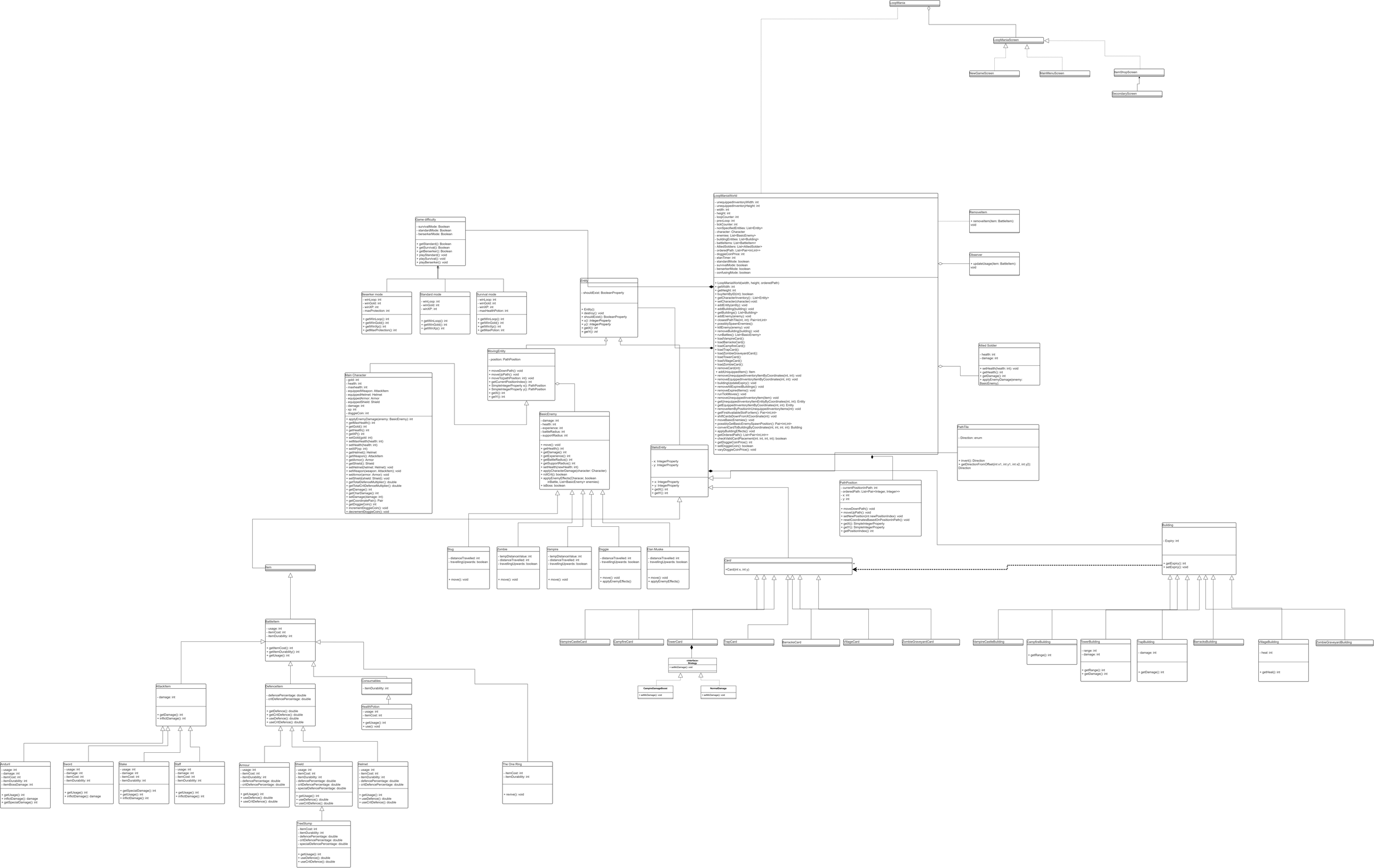
LoopMania
LoopManiaScreen
NewGameScreen
MainMenuScreen
ItemShopScreen
SecondaryScreen
LoopManiaWorld
- unequippedInventoryWidth: int
- unequippedInventoryHeight: int
- width: int
- height: int
- loopCounter: int
- prevLoop: int
- tickCounter: int
- nonSpecifiedEntities: List<Entity>
- character: Character
- enemies: List<BasicEnemy>
- buildingEntities: List<Building>
- battleItems: List<BattleItem>
- AlliedSoldiers: List<AlliedSoldier>
- orderedPath: List<Pair<int,int>>
- doggieCoinPrice: int
- elanTimer: int
- standardMode: boolean
- survivalMode: boolean
- berserkerMode: boolean
- confusingMode: boolean
+ LoopManiaWorld(width, height, orderedPath)
+ getWidth: int
+ getHeight: int
+ buyItemByID(int): boolean
+ getCharacterInventory(): List<Entity>
+ setCharacter(character) void:
+ addEntity(entity): void
+ addBuilding(building): void
+ getBuildings(): List<Building>
+ addEnemy(enemy): void
+ closestPathTile(int, int): Pair<int,int>
+ possiblySpawnEnemies():
+ killEnemy(enemy): void
+ removeBuilding(building): void
+ runBattles(): List<BasicEnemy>
+ loadVampireCard():
+ loadBarracksCard():
+ loadCampfireCard():
+ loadTrapCard():
+ loadZombieGraveyardCard():
+ loadTowerCard():
+ loadVillageCard():
+ loadZombieCard():
+ removeCard(int):
+ addUnequippedItem(): Item
+ removeUnequippedInventoryItemByCoordinates(int, int): void
+ removeEquippedInventoryItemByCoordinates(int, int): void
+ buildingUpdate(Expiry(): void
+ removeAllExpiredBuildings(): void
+ removeExpiredItems(): void
+ tickMovies(): void
+ removeUnequippedInventoryItem(item): void
+ getUnequippedInventoryItemEntityByCoordinates(int, int): Entity
+ getEquippedInventoryItemByCoordinates(int, int): Entity
+ removeItemByPositionInUnequippedInventoryItems(int): void
+ getFirstAvailableSlotForItem(): Pair<int,int>
+ shiftCardsDownFromXCoordinate(int): void
+ possiblyGetBasicEnemySpawnPosition(): Pair<int,int>
+ convertTileToBuildingByCoordinates(int, int, int, int): Building
+ applyBuildingEffects(): void
+ getOrderedPath(): List<Pair<int,int>>
+ checkValidCardPlacement(int, int, int, int): boolean
+ getDoggieCoinPrice(): int
+ sellDoggieCoin(): boolean
+ varyDoggieCoinPrice(): void
RemoveItem
+ removeItem(item: BattleItem): void
Observer
+ updateUsage(item: BattleItem): void
Game difficulty
- survivalMode: Boolean
- standardMode: Boolean
- berserkerMode: Boolean
+ getStandard(): Boolean
+ getSurvival(): Boolean
+ getBerserker(): Boolean
+ playStandard(): void
+ playSurvival(): void
+ playBerserker(): void
Berserker mode
- winLoop: int
- winGold: int
- winXP: int
- maxProtection: int
+ getWinLoop(): int
+ getWinGold(): int
+ getWinXp(): int
+ getMaxProtection(): int
Standard mode
- winLoop: int
- winGold: int
- winXP: int
+ getWinLoop(): int
+ getWinGold(): int
+ getWinAx(): int
Survival mode
- winLoop: int
- winGold: int
- winXP: int
- maxHealthPotion: int
+ getWinLoop(): int
+ getWinGold(): int
+ getWinXp(): int
+ getMaxPotion: int
Entity
- shouldExist: BooleanProperty
+ Entity():
+ destroy(): void
+ shouldExist(): BooleanProperty
+ x(): IntegerProperty
+ y(): IntegerProperty
+ getX(): int
+ getY(): int
MovingEntity
- position: PathPosition
+ moveDownPath(): void
+ moveUpPath(): void
+ moveToPathPosition: int): void
+ getCurrentPositionIndex(): int
+ SimpleIntegerProperty x(): PathPosition
+ SimpleIntegerProperty y(): PathPosition
+ getX(): int
+ getY(): int
BasicEnemy
- damage: int
- health: int
- experience: int
- battleRadius: int
- supportRadius: int
+ move(): void
+ getHealth(): int
+ getDamage(): int
+ getExperience(): int
+ getBattleRadius(): int
+ getSupportRadius(): int
+ setHealth(newHealth: int)
+ applyCharacterDamage(character: Character)
+ rollCrit(): boolean
+ applyEnemyEffects(Character, boolean inBattle, List<BasicEnemy> enemies)
+ isBoss: boolean
StaticEntity
- x: IntegerProperty
- y: IntegerProperty
+ x: IntegerProperty
+ y: IntegerProperty
+ getX(): int
+ getY(): int
Allied Soldier
- health: int
- damage: int
+ setHealth(health: int): void
+ getHealth(): int
+ getDamage(): int
+ applyEnemyDamage(enemy: BasicEnemy):
PathTile
- Direction: enum
+ invert(): Direction
+ getDirectionFromOffset(int x1, int y1, int x2, int y2): Direction
PathPosition
- currentPositionInPath: int
- orderedPath: List<Pair<Integer, Integer>>
- x: int
- y: int
+ moveDownPath(): void
+ moveUpPath(): void
+ setNewPosition(int newPositionIndex): void
+ resetCoordinatesBasedOnPositionInPath(): void
+ getX(): SimpleIntegerProperty
+ getY(): SimpleIntegerProperty
+ getPositionIndex(): int
Main Character
- gold: int
- health: int
- maxhealth: int
- equippedWeapon: AttackItem
- equippedHelmet: Helmet
- equippedArmor: Armor
- equippedShield: Shield
- damage: int
- xp: int
- doggieCoin: int
+ applyEnemyDamage(enemy: BasicEnemy): int
+ getMaxHealth(): int
+ getGold(): int
+ getHealth(): int
+ getXP(): int
+ setGold(gold: int)
+ setMaxHealth(health: int)
+ setHealth(health: int)
+ setXP(xp: int)
+ getHelmet(): Helmet
+ getWeapon(): AttackItem
+ getArmor(): Armor
+ getShield(): Shield
+ setHelmet(helmet: Helmet): void
+ setWeapon(weapon: AttackItem): void
+ setArmor(armor: Armor): void
+ setShield(shield: Shield): void
+ getTotalDefenceMultiplier(): double
+ getTotalCritDefenceMultiplier(): double
+ getDamage(): int
+ getCharDamage(): int
+ setDamage(damage: int)
+ getCoordinatePair(): Pair
+ getDoggieCoin(): int
+ incrementDoggieCoin(): void
+ decrementDoggieCoin(): void
Slug
- distanceTravelled: int
- travellingUpwards: boolean
+ move(): void
Zombie
- tempDistanceValue: int
- distanceTravelled: int
- travellingUpwards: boolean
+ move(): void
Vampire
- tempDistanceValue: int
- distanceTravelled: int
- travellingUpwards: boolean
+ move(): void
Doggie
- distanceTravelled: int
- travellingUpwards: boolean
+ move(): void
+ applyEnemyEffects()
Elan Muske
- distanceTravelled: int
- travellingUpwards: boolean
+ move(): void
+ applyEnemyEffects()
Item
BattleItem
- usage: int
- itemCost: int
- itemDurability: int
+ getItemCost(): int
+ getItemDurability(): int
+ getUsage(): int
AttackItem
- damage: int
+ getDamage(): int
+ inflictDamage(): int
DefenceItem
- defencePercentage: double
- critDefencePercentage: double
+ getDefence(): double
+ getCritDefence(): double
+ useDefence(): double
+ useCritDefence(): double
Consumables
- itemDurability: int
HealthPotion
- usage: int
- itemCost: int
+ getUsage(): int
+ use(): void
Anduril
- usage: int
- damage: int
- itemCost: int
- itemDurability: int
- itemBossDamage: int
+ getUsage(): int
+ inflictDamage(): damage
+ getSpecialDamage(): int
Sword
- usage: int
- damage: int
- itemCost: int
- itemDurability: int
+ getUsage(): int
+ inflictDamage(): damage
Stake
- usage: int
- damage: int
- itemCost: int
- itemDurability: int
+ getSpecialDamage(): int
+ getUsage(): int
+ inflictDamage(): int
Staff
- usage: int
- damage: int
- itemCost: int
- itemDurability: int
+ getUsage(): int
+ inflictDamage(): int
Armour
- usage: int
- itemCost: int
- itemDurability: int
- defencePercentage: double
- critDefencePercentage: double
+ getUsage(): int
+ useDefence(): double
+ useCritDefence(): double
Shield
- usage: int
- itemCost: int
- itemDurability: int
- defencePercentage: double
- critDefencePercentage: double
- specialDefencePercentage: double
+ getUsage(): int
+ useDefence(): double
+ useCritDefence(): double
Helmet
- usage: int
- itemCost: int
- itemDurability: int
- defencePercentage: double
- critDefencePercentage: double
+ getUsage(): int
+ useDefence(): double
+ useCritDefence(): double
TreeStump
- itemCost: int
- itemDurability: int
- defencePercentage: double
- critDefencePercentage: double
- specialDefencePercentage: double
+ getUsage(): int
+ useDefence(): double
+ useCritDefence(): double
The One Ring
- itemCost: int
- itemDurability: int
+ revive(): void
Card
+Card(int x, int y)
VampireCastleCard
CampfireCard
TowerCard
TrapCard
BarracksCard
VillageCard
ZombieGraveyardCard
«interface»
Strategy
+ setAttDamage(): void
VampireDamageBoost
+ setAttDamage(): void
NormalDamage
+ setAttDamage(): void
Building
- Expiry: int
+ getExpiry(): int
+ setExpiry(): void
VampireCastleBuilding
CampfireBuilding
+ getRange(): int
TowerBuilding
- range: int
- damage: int
+ getRange(): int
+ getDamage(): int
TrapBuilding
- damage: int
+ getDamage(): int
BarracksBuilding
VillageBuilding
- heal: int
+ getHeal(): int
ZombieGraveyardBuilding