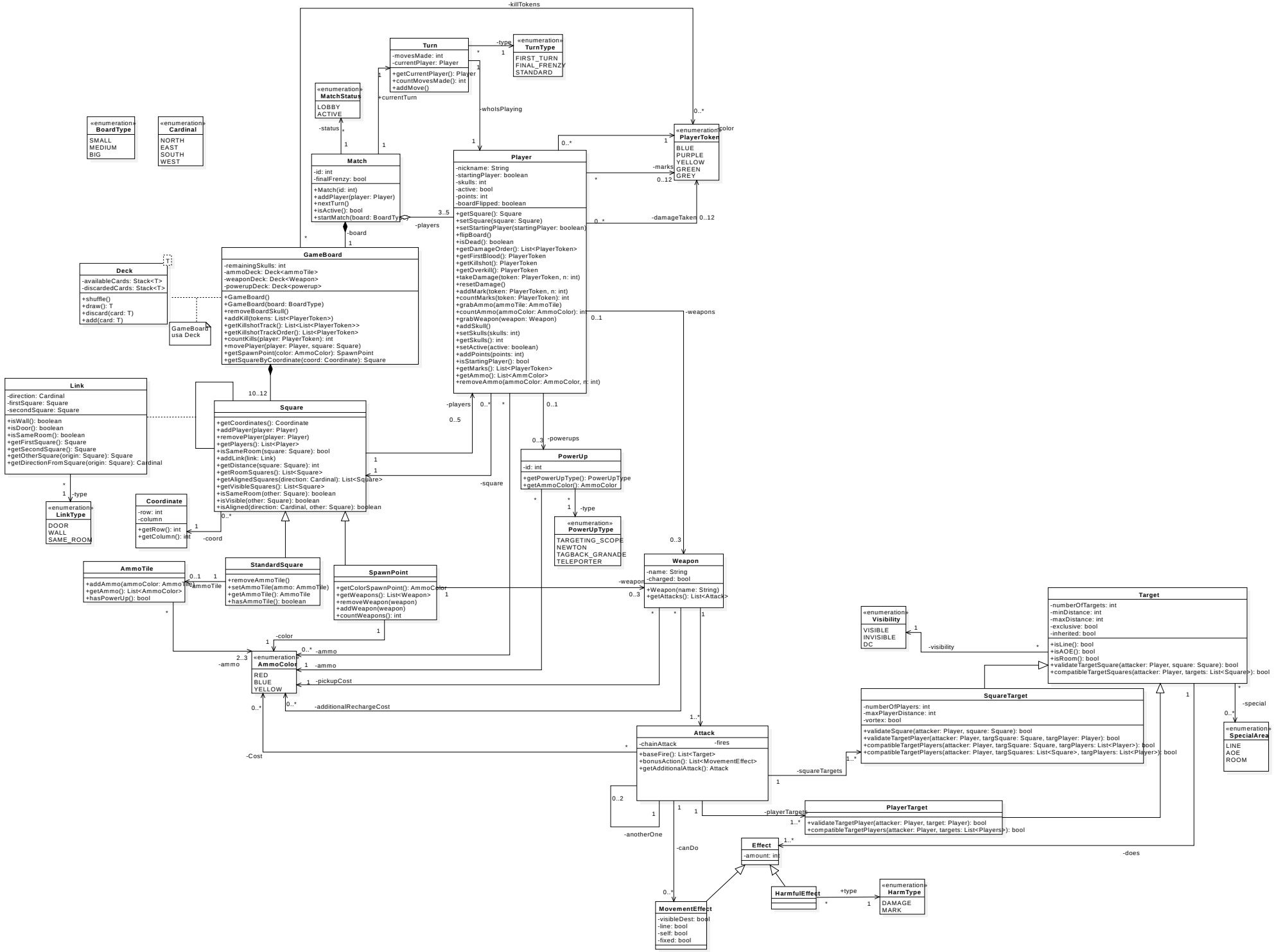
-killTokens
Turn
-movesMade: int
-currentPlayer: Player
+getCurrentPlayer(): Player
+countMovesMade(): int
+addMove()
-type
«enumeration»
TurnType
FIRST_TURN
FINAL_FRENZY
STANDARD
«enumeration»
MatchStatus
LOBBY
ACTIVE
-status
+currentTurn
-whoIsPlaying
«enumeration»
BoardType
SMALL
MEDIUM
BIG
«enumeration»
Cardinal
NORTH
EAST
SOUTH
WEST
Match
-id: int
-finalFrenzy: bool
+Match(id: int)
+addPlayer(player: Player)
+nextTurn()
+isActive(): bool
+startMatch(board: BoardType)
-board
-players
Player
-nickname: String
-startingPlayer: boolean
-skulls: int
-active: bool
-points: int
-boardFlipped: boolean
+getSquare(): Square
+setSquare(square: Square)
+setStartingPlayer(startingPlayer: boolean)
+flipBoard()
+isDead(): boolean
+getDamageOrder(): List<PlayerToken>
+getFirstBlood(): PlayerToken
+getKillshot(): PlayerToken
+getOverkill(): PlayerToken
+takeDamage(token: PlayerToken, n: int)
+resetDamage()
+addMark(token: PlayerToken, n: int)
+countMarks(token: PlayerToken): int
+grabAmmo(ammoTile: AmmoTile)
+countAmmo(ammoColor: AmmoColor): int
+grabWeapon(weapon: Weapon)
+addSkull()
+setSkulls(skulls: int)
+getSkulls(): int
+setActive(active: boolean)
+addPoints(points: int)
+isStartingPlayer(): bool
+getMarks(): List<PlayerToken>
+getAmmo(): List<AmmColor>
+removeAmmo(ammoColor: AmmoColor, n: int)
-color
«enumeration»
PlayerToken
BLUE
PURPLE
YELLOW
GREEN
GREY
-marks
-damageTaken
Deck
-availableCards: Stack<T>
-discardedCards: Stack<T>
+shuffle()
+draw(): T
+discard(card: T)
+add(card: T)
GameBoard usa Deck
GameBoard
-remainingSkulls: int
-ammoDeck: Deck<ammoTile>
-weaponDeck: Deck<Weapon>
-powerupDeck: Deck<powerup>
+GameBoard()
+GameBoard(board: BoardType)
+removeBoardSkull()
+addKill(tokens: List<PlayerToken>)
+getKillshotTrack(): List<List<PlayerToken>>
+getKillshotTrackOrder(): List<PlayerToken>
+countKills(player: PlayerToken): int
+movePlayer(player: Player, square: Square)
+getSpawnPoint(color: AmmoColor): SpawnPoint
+getSquareByCoordinate(coord: Coordinate): Square
-weapons
Link
-direction: Cardinal
-firstSquare: Square
-secondSquare: Square
+isWall(): boolean
+isDoor(): boolean
+isSameRoom(): boolean
+getFirstSquare(): Square
+getSecondSquare(): Square
+getOtherSquare(origin: Square): Square
+getDirectionFromSquare(origin: Square): Cardinal
-type
«enumeration»
LinkType
DOOR
WALL
SAME_ROOM
Square
+getCoordinates(): Coordinate
+addPlayer(player: Player)
+removePlayer(player: Player)
+getPlayers(): List<Player>
+isSameRoom(square: Square): bool
+addLink(link: Link)
+getDistance(square: Square): int
+getRoomSquares(): List<Square>
+getAlignedSquares(direction: Cardinal): List<Square>
+getVisibleSquares(): List<Square>
+isSameRoom(other: Square): boolean
+isVisible(other: Square): boolean
+isAligned(direction: Cardinal, other: Square): boolean
-square
-coord
Coordinate
-row: int
-column
+getRow(): int
+getColumn(): int
-powerups
PowerUp
-id: int
+getPowerUpType(): PowerUpType
+getAmmoColor(): AmmoColor
-type
«enumeration»
PowerUpType
TARGETING_SCOPE
NEWTON
TAGBACK_GRANADE
TELEPORTER
AmmoTile
+addAmmo(ammoColor: AmmoColor)
+getAmmo(): List<AmmoColor>
+hasPowerUp(): bool
-ammoTile
StandardSquare
+removeAmmoTile()
+setAmmoTile(ammo: AmmoTile)
+getAmmoTile(): AmmoTile
+hasAmmoTile(): boolean
SpawnPoint
+getColorSpawnPoint(): AmmoColor
+getWeapons(): List<Weapon>
+removeWeapon(weapon)
+addWeapon(weapon)
+countWeapons(): int
-color
-ammo
«enumeration»
AmmoColor
RED
BLUE
YELLOW
-pickupCost
-additionalRechargeCost
-Cost
Weapon
-name: String
-charged: bool
+Weapon(name: String)
+getAttacks(): List<Attack>
-weapon
Attack
-chainAttack
-fires
+baseFire(): List<Target>
+bonusAction(): List<MovementEffect>
+getAdditionalAttack(): Attack
-anotherOne
-canDo
-squareTargets
-playerTargets
«enumeration»
Visibility
VISIBLE
INVISIBLE
DC
-visibility
Target
-numberOfTargets: int
-minDistance: int
-maxDistance: int
-exclusive: bool
-inherited: bool
+isLine(): bool
+isAOE(): bool
+isRoom(): bool
+validateTargetSquare(attacker: Player, square: Square): bool
+compatibleTargetSquares(attacker: Player, targets: List<Square>): bool
-special
«enumeration»
SpecialArea
LINE
AOE
ROOM
SquareTarget
-numberOfPlayers: int
-maxPlayerDistance: int
-vortex: bool
+validateSquare(attacker: Player, square: Square): bool
+validateTargetPlayer(attacker: Player, targSquare: Square, targPlayer: Player): bool
+compatibleTargetPlayers(attacker: Player, targSquare: Square, targPlayers: List<Player>): bool
+compatibleTargetPlayers(attacker: Player, targSquares: List<Square>, targPlayers: List<Player>): bool
PlayerTarget
+validateTargetPlayer(attacker: Player, target: Player): bool
+compatibleTargetPlayers(attacker: Player, targets: List<Players>): bool
-does
Effect
-amount: int
HarmfulEffect
+type
«enumeration»
HarmType
DAMAGE
MARK
MovementEffect
-visibleDest: bool
-line: bool
-self: bool
-fixed: bool