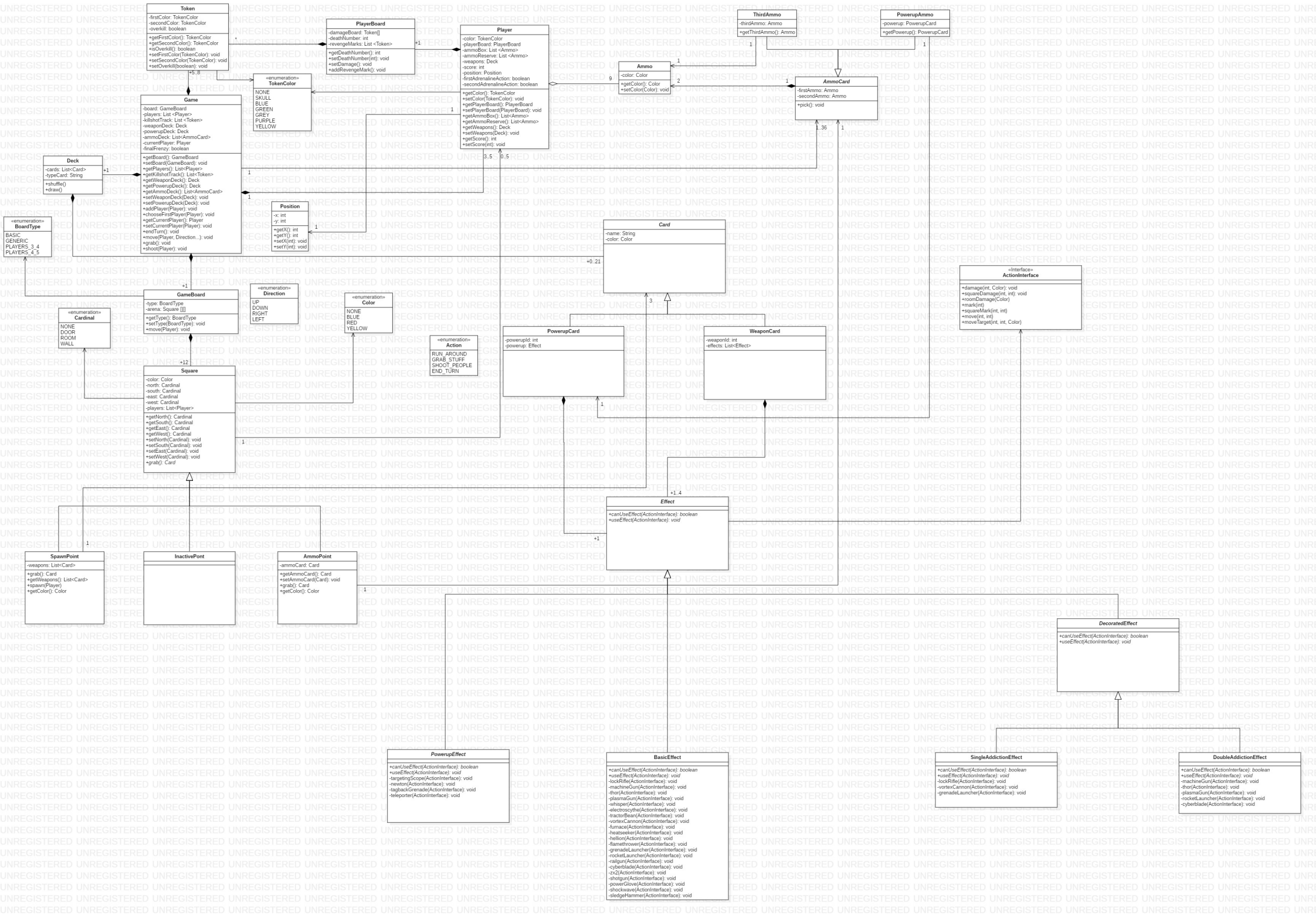
Token
-firstColor: TokenColor
-secondColor: TokenColor
-overkill: boolean
+getFirstColor(): TokenColor
+getSecondColor(): TokenColor
+isOverkill(): boolean
+setFirstColor(TokenColor): void
+setSecondColor(TokenColor): void
+setOverkill(boolean): void
PlayerBoard
-damageBoard: Token[]
-deathNumber: int
-revengeMarks: List <Token>
+getDeathNumber(): int
+setDeathNumber(int): void
+setDamage(): void
+addRevengeMark(): void
Player
-color: TokenColor
-playerBoard: PlayerBoard
-ammoBox: List <Ammo>
-ammoReserve: List <Ammo>
-weapons: Deck
-score: int
-position: Position
-firstAdrenalineAction: boolean
-secondAdrenalineAction: boolean
+getColor(): TokenColor
+setColor(TokenColor): void
+getPlayerBoard(): PlayerBoard
+setPlayerBoard(PlayerBoard): void
+getAmmoBox(): List<Ammo>
+getAmmoReserve(): List<Ammo>
+getWeapons(): Deck
+setWeapons(Deck): void
+getScore(): int
+setScore(int): void
ThirdAmmo
-thirdAmmo: Ammo
+getThirdAmmo(): Ammo
PowerupAmmo
-powerup: PowerupCard
+getPowerup(): PowerupCard
Ammo
-color: Color
+getColor(): Color
+setColor(Color): void
AmmoCard
-firstAmmo: Ammo
-secondAmmo: Ammo
+pick(): void
«enumeration»
TokenColor
NONE
SKULL
BLUE
GREEN
GREY
PURPLE
YELLOW
Game
-board: GameBoard
-players: List <Player>
-killshotTrack: List <Token>
-weaponDeck: Deck
-powerupDeck: Deck
-ammoDeck: List<AmmoCard>
-currentPlayer: Player
-finalFrenzy: boolean
+getBoard(): GameBoard
+setBoard(GameBoard): void
+getPlayers(): List<Player>
+getKillshotTrack(): List<Token>
+getWeaponDeck(): Deck
+getPowerupDeck(): Deck
+getAmmoDeck(): List<AmmoCard>
+setWeaponDeck(Deck): void
+setPowerupDeck(Deck): void
+addPlayer(Player): void
+chooseFirstPlayer(Player): void
+getCurrentPlayer(): Player
+setCurrentPlayer(Player): void
+endTurn(): void
+move(Player, Direction...): void
+grab(): void
+shoot(Player): void
Deck
-cards: List<Card>
-typeCard: String
+shuffle()
+draw()
Position
-x: int
-y: int
+getX(): int
+getY(): int
+setX(int): void
+setY(int): void
«enumeration»
BoardType
BASIC
GENERIC
PLAYERS_3_4
PLAYERS_4_5
Card
-name: String
-color: Color
«Interface»
ActionInterface
+damage(int, Color): void
+squareDamage(int, int): void
+roomDamage(Color)
+mark(int)
+squareMark(int, int)
+move(int, int)
+moveTarget(int, int, Color)
GameBoard
-type: BoardType
-arena: Square[][]
+getType(): BoardType
+setType(BoardType): void
+move(Player): void
«enumeration»
Direction
UP
DOWN
RIGHT
LEFT
«enumeration»
Color
NONE
BLUE
RED
YELLOW
«enumeration»
Cardinal
NONE
DOOR
ROOM
WALL
«enumeration»
Action
RUN_AROUND
GRAB_STUFF
SHOOT_PEOPLE
END_TURN
PowerupCard
-poweruplD: int
-powerup: Effect
WeaponCard
-weaponID: int
-effects: List<Effect>
Square
-color: Color
-north: Cardinal
-south: Cardinal
-east: Cardinal
-west: Cardinal
-players: List<Player>
+getNorth(): Cardinal
+getSouth(): Cardinal
+getEast(): Cardinal
+getWest(): Cardinal
+setNorth(Cardinal): void
+setSouth(Cardinal): void
+setEast(Cardinal): void
+setWest(Cardinal): void
+grab(): Card
Effect
+canUseEffect(ActionInterface): boolean
+useEffect(ActionInterface): void
SpawnPoint
-weapons: List<Card>
+grab(): Card
+getWeapons(): List<Card>
+spawn(Player)
+getColor(): Color
InactivePont
AmmoPoint
-ammoCard: Card
+getAmmoCard(): Card
+setAmmoCard(Card): void
+grab(): Card
+getColor(): Color
DecoratedEffect
+canUseEffect(ActionInterface): boolean
+useEffect(ActionInterface): void
PowerupEffect
+canUseEffect(ActionInterface): boolean
+useEffect(ActionInterface): void
-targetingScope(ActionInterface): void
-newton(ActionInterface): void
-tagbackGrenade(ActionInterface): void
-teleporter(ActionInterface): void
BasicEffect
+canUseEffect(ActionInterface): boolean
+useEffect(ActionInterface): void
-lockRifle(ActionInterface): void
-machineGun(ActionInterface): void
-thor(ActionInterface): void
-plasmaGun(ActionInterface): void
-whisper(ActionInterface): void
-electroscythe(ActionInterface): void
-tractorBeam(ActionInterface): void
-vortexCannon(ActionInterface): void
-furnace(ActionInterface): void
-heatseeker(ActionInterface): void
-hellion(ActionInterface): void
-flamethrower(ActionInterface): void
-grenadeLauncher(ActionInterface): void
-rocketLauncher(ActionInterface): void
-railgun(ActionInterface): void
-cyberblade(ActionInterface): void
-zx2(ActionInterface): void
-shotgun(ActionInterface): void
-powerGlove(ActionInterface): void
-shockwave(ActionInterface): void
-sledgeHammer(ActionInterface): void
SingleAddictionEffect
+canUseEffect(ActionInterface): boolean
+useEffect(ActionInterface): void
-lockRifle(ActionInterface): void
-vortexCannon(ActionInterface): void
-grenadeLauncher(ActionInterface): void
DoubleAddictionEffect
+canUseEffect(ActionInterface): boolean
+useEffect(ActionInterface): void
-machineGun(ActionInterface): void
-thor(ActionInterface): void
-plasmaGun(ActionInterface): void
-rocketLauncher(ActionInterface): void
-cyberblade(ActionInterface): void