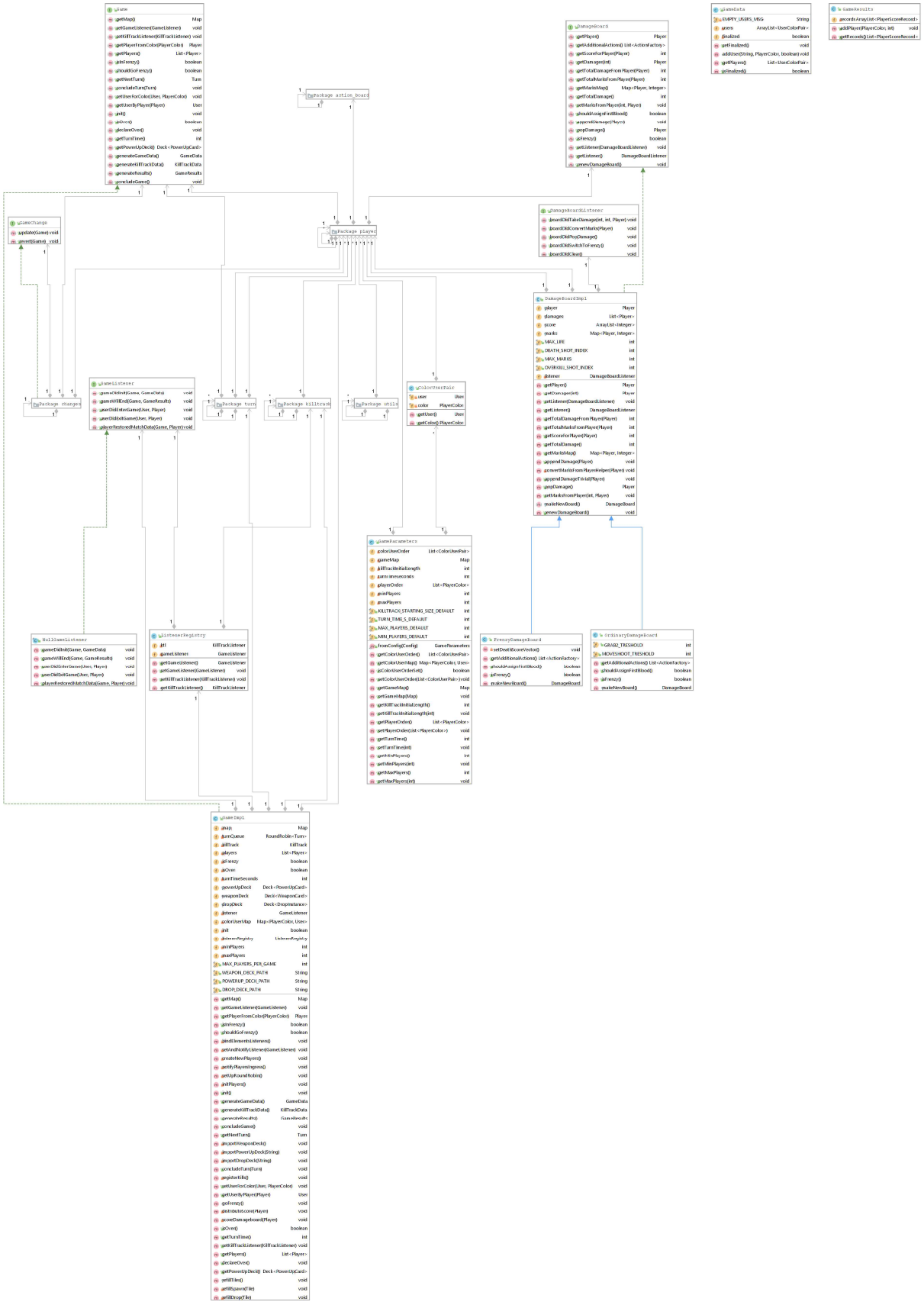
Game
getMap() Map
getGameListener(GameListener) void
getKillTrackListener(KillTrackListener) void
getPlayerFromColor(PlayerColor) Player
getPlayers() List<Player>
isInFrenzy() boolean
shouldGoFrenzy() boolean
getNextTurn() Turn
concludeTurn(Turn) void
getUserForColor(User, PlayerColor) void
getUserByPlayer(Player) User
init() void
isOver() boolean
declareOver() void
getTurnTime() int
getPowerUpDeck() Deck<PowerUpCard>
generateGameData() GameData
generateKillTrackData() KillTrackData
generateResults() GameResults
concludeGame() void
Package action_board
DamageBoard
getPlayer() Player
getAdditionalActions() List<ActionFactory>
getScoreForPlayer(Player) int
getDamage(int) Player
getTotalDamageFromPlayer(Player) int
getTotalMarksFromPlayer(Player) int
getMarksMap() Map<Player, Integer>
getTotalDamage() int
getMarksFromPlayer(int, Player) void
shouldAssignFirstBlood() boolean
appendDamage(Player) void
popDamage() Player
isFrenzy() boolean
getListener(DamageBoardListener) void
getListener() DamageBoardListener
genewDamageBoard() void
GameData
EMPTY_USERS_MSG String
users Array<List<UserColorPair>
finalized boolean
getFinalized() void
addUser(String, PlayerColor, boolean) void
getPlayers() List<UserColorPair>
isFinalized() boolean
GameResults
records Array<List<PlayerScoreRecord>
addPlayer(PlayerColor, int) void
getRecords() List<PlayerScoreRecord>
GameChange
update(Game) void
gameEnd(Game) void
Package player
DamageBoardListener
boardDidTakeDamage(int, int, Player) void
boardDidConvertMarks(Player) void
boardDidPopDamage() void
boardDidSwitchToFrenzy() void
boardDidClear() void
DamageBoardImpl
player Player
damages List<Player>
score ArrayList<Integer>
marks Map<Player, Integer>
MAX_LIFE int
DEATH_SHOT_INDEX int
MAX_MARKS int
OVERKILL_SHOT_INDEX int
listener DamageBoardListener
getPlayer() Player
getDamage(int) Player
getListener(DamageBoardListener) void
getListener() DamageBoardListener
getTotalDamageFromPlayer(Player) int
getTotalMarksFromPlayer(Player) int
getScoreForPlayer(Player) int
getTotalDamage() int
getMarksMap() Map<Player, Integer>
appendDamage(Player) void
convertMarksFromPlayerHelper(Player) void
appendDamageToTotal(Player) void
popDamage() Player
getMarksFromPlayer(int, Player) void
makeNewBoard() DamageBoard
genewDamageBoard() void
GameListener
gameDidInit(Game, GameData) void
gameWillEnd(Game, GameResults) void
userDidEnterGame(User, Player) void
userDidExitGame(User, Player) void
playersRestoredMatchData(Game, Player) void
Package changes
Package turn
Package killtrack
Package utils
ColorUserPair
user User
color PlayerColor
getUser() User
getColor() PlayerColor
GameParameters
colorUserOrder List<ColorUserPair>
gameMap Map
killTrackInitialLength int
turnsInmeseconds int
playerOrder List<PlayerColor>
minPlayers int
maxPlayers int
KILLTRACK_STARTING_SIZE_DEFAULT int
TURN_TIME_S_DEFAULT int
MAX_PLAYERS_DEFAULT int
MIN_PLAYERS_DEFAULT int
fromConfig(Config) GameParameters
getColorUserOrder() List<ColorUserPair>
getColorUserMap() Map<PlayerColor, User>
isColorUserOrderSet() boolean
getColorUserOrder(List<ColorUserPair>) void
getGameMap() Map
setGameMap(Map) void
getKillTrackInitialLength() int
setKillTrackInitialLength(int) void
getPlayerOrder() List<PlayerColor>
getPlayerOrder(List<PlayerColor>) void
getTurnTime() int
getTurnTime(int) void
getMinPlayers() int
getMinPlayers(int) void
getMaxPlayers() int
getMaxPlayers(int) void
NullGameListener
gameDidInit(Game, GameData) void
gameWillEnd(Game, GameResults) void
userDidEnterGame(User, Player) void
userDidExitGame(User, Player) void
playersRestoredMatchData(Game, Player) void
ListenerRegistry
all KillTrackListener
gameListener GameListener
getGameListener() GameListener
getGameListener(GameListener) void
getKillTrackListener(KillTrackListener) void
getKillTrackListener() KillTrackListener
FrenzyDamageBoard
setDeathScore(vector) void
getAdditionalActions() List<ActionFactory>
shouldAssignFirstBlood() boolean
isFrenzy() boolean
makeNewBoard() DamageBoard
OrdinaryDamageBoard
GRAB2_TRESHOLD int
MOVESHOOT_TRESHOLD int
getAdditionalActions() List<ActionFactory>
shouldAssignFirstBlood() boolean
isFrenzy() boolean
makeNewBoard() DamageBoard
GameImpl
map Map
turnQueue RoundRobin<Turn>
killTrack KillTrack
players List<Player>
isFrenzy boolean
isOver boolean
turnTimeSeconds int
powerUpDeck Deck<PowerUpCard>
weaponDeck Deck<WeaponCard>
dropDeck Deck<DropInstance>
listener GameListener
colorUserMap Map<PlayerColor, User>
init boolean
listenerRegistry ListenerRegistry
minPlayers int
maxPlayers int
MAX_PLAYERS_PER_GAME int
WEAPON_DECK_PATH String
POWERUP_DECK_PATH String
DROP_DECK_PATH String
getMap() Map
getGameListener(GameListener) void
getPlayerFromColor(PlayerColor) Player
isInFrenzy() boolean
shouldGoFrenzy() boolean
placeAllElementsInterness() void
getAndNotifyListener(GameListener) void
generateNewPlayers() void
notifyPlayersingenesis() void
getUpRoundRobin() void
initPlayers() void
init() void
generateGameData() GameData
generateKillTrackData() KillTrackData
generateResults() GameResults
concludeGame() void
getNextTurn() Turn
importWeaponDeck() void
importPowerUpDeck(String) void
importDropDeck(String) void
concludeTurn(Turn) void
registerKills() void
getUserForColor(User, PlayerColor) void
getUserByPlayer(Player) User
goFrenzy() void
distributeScore() void
scoreDamageboard(Player) void
isOver() boolean
getTurnTime() int
getKillTrackListener(KillTrackListener) void
getPlayers() List<Player>
declareOver() void
getPowerUpDeck() Deck<PowerUpCard>
refillTiles() void
refillSpawn(Tile) void
refillDrop(Tile) void