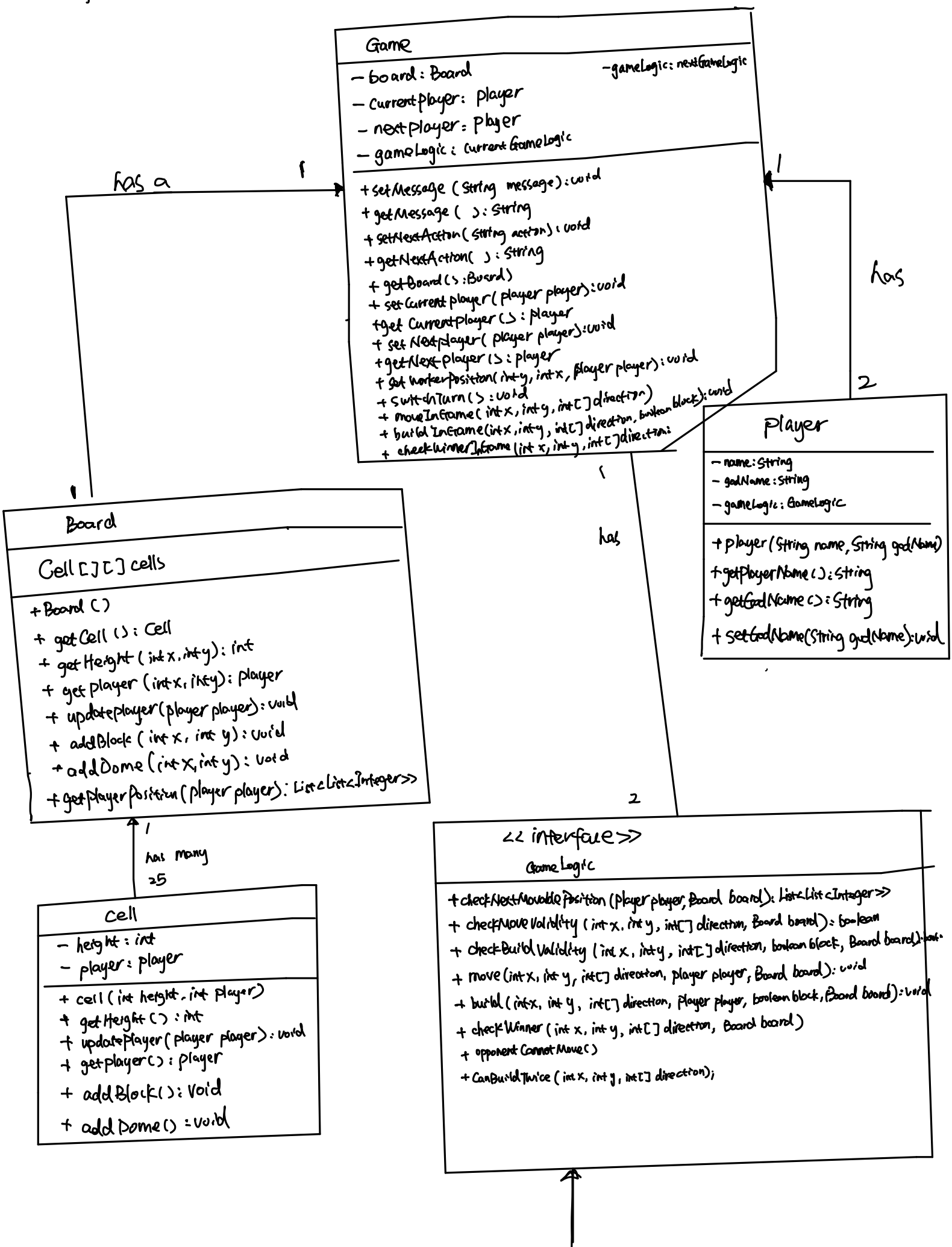
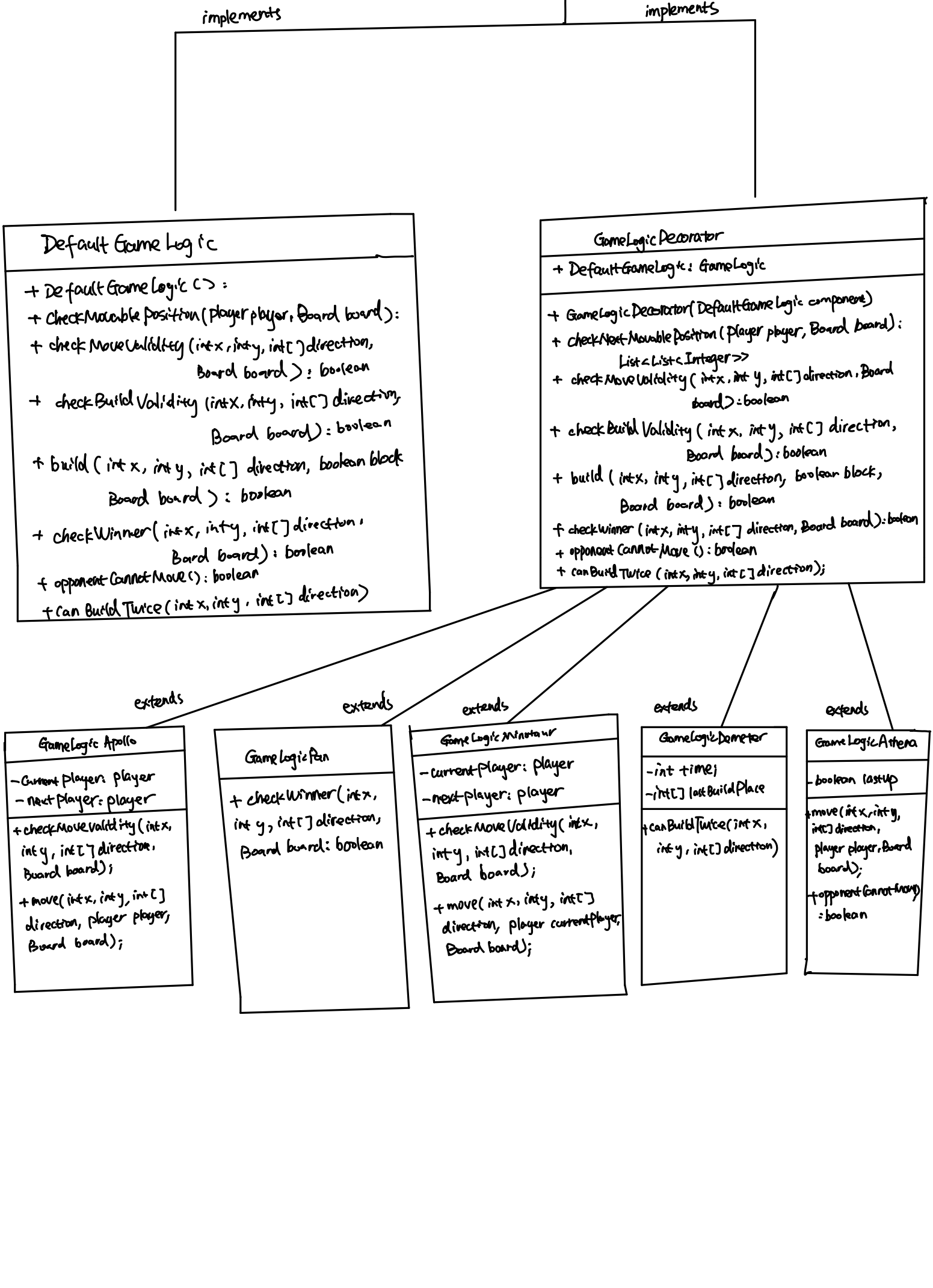
Object Model




implements
implements
DefaultGameLogic
+ DefaultGameLogic():
+ CheckMovable position(player player, Board board):
+ checkMoveValidity(int x, int y, int[] direction, Board board): boolean
+ checkBuildValidity(int x, int y, int[] direction, Board board): boolean
+ build(int x, int y, int[] direction, boolean block Board board): boolean
+ checkWinner(int x, int y, int[] direction, Board board): boolean
+ opponentCannotMove(): boolean
+ canBuildTwice(int x, int y, int[] direction)
GameLogicDecorator
+ DefaultGameLogic: GameLogic
+ GameLogicDecorator(DefaultGameLogic component)
+ checkNextMovablePosition(Player player, Board board): List<List<Integer>>
+ checkMoveValidity(int x, int y, int[] direction, Board board): boolean
+ checkBuildValidity(int x, int y, int[] direction, Board board): boolean
+ build(int x, int y, int[] direction, boolean block, Board board): boolean
+ checkWinner(int x, int y, int[] direction, Board board): boolean
+ opponentCannotMove(): boolean
+ canBuildTwice(int x, int y, int[] direction);
extends
extends
extends
extends
extends
GameLogicApollo
- Current player: player
- next player: player
+ checkMoveValidity(int x, int y, int[] direction, Board board);
+ move(int x, int y, int[] direction, player player, Board board);
GameLogicPan
+ checkWinner(int x, int y, int[] direction, Board board: boolean
GameLogicMinotaur
- current player: player
- next player: player
+ checkMoveValidity(int x, int y, int[] direction, Board board);
+ move(int x, int y, int[] direction, player currentPlayer, Board board);
GameLogicDemeter
- int time;
- int[] lastBuildPlace
+ canBuildTwice(int x, int y, int[] direction)
GameLogicAthena
- boolean lastUp
+ move(int x, int y, int[] direction, player player, Board board);
+ opponentCannotMove() :boolean