

# Homework 1 (pacman): Function Diagram

| Keith Bates

## Player

### Player Struct:

- Overloaded operator== (Position): checks if two positions are the same.

### Player Class:

- (Player) Constructor: stores data such as position, lives, points, and other fields.
- Get\_name: gets the name of the player.
- Get\_points: gets the points of the player.
- Get\_position: gets the position of the player in the game.
- Is\_human: checks if the player is an AI or human.
- HasTreasure: checks if the player has found the treasure.
- IsDead: checks if the player has lost all their lives.
- GetLives: gets the number of lives the player has remaining in the game.
- ChangePoints: increase/decrease the points total of the player.
- SetPosition: sets the position of the player in the game.
- SetHasTreasure: sets whether or not the player has found the treasure.
- SetIsDead: sets the player's status to dead if they have lost all their lives.
- SetLives: sets the number of lives the player has remaining in the game.
- ToRelativePosition: calculates the relative position of the player in the game.
- Stringify: converts the player object into a string.

- GetMoves: to get the number of moves the player has made in the game.
- IncrementMoves: to increase the number of moves the player has made in the game.

## Game & Board

### Board Class:

- (Board) Constructor: constructs a board of preset dimensions and attributes
  - Calls Board::get\_rows() and Board::get\_cols() to determine size.
- Get\_rows: gets the number of rows in the game board.
  - preset value.
- Get\_cols: gets the number of columns in the game board.
  - preset value.
- Get\_square\_value: gets the value of the square at the given position.
- SetSquareValue: sets the value of the square at the given position.
- GetMoves: gets the possible Positions that a Player/Enemy could move to.
  - Calls get\_square\_value for adjacent squares to check if a player can move there
- MovePlayer: moves a player to a new position on the board and returns true if successful.
  - Maybe calls Board::getMoves() to check if a move is valid.
  - Calls Player::setPosition() to change the player position.
- MoveEnemy: moves an enemy to a new position on the board and returns true if successful.
  - Maybe calls Board::getMoves() to check if a move is valid.
  - Calls Player::setPosition() to change the player position.
- Stringify: converts the board object into a printable string.

## Game Class:

- (Game) Constructor: constructs the initial game attributes.
- NewGame: initializes a new game, given one human player and a number of enemies to generate.
  - Calls constructors for Board and Player
- TakeTurn: have the given Player take their turn.
  - Calls Board::movePlayer(), Board::getMoves() and Game::isGameOver().
- TakeTurnEnemy: have the enemy take a turn.
  - Calls Board::moveEnemy(), Board::getMoves() and Game::isGameOver().
- IsGameOver: checks if the game is over.
  - Calls Game::checkIfDotsOver() (I changed the name to camel Case), Player::get\_position() and compares to Player::get\_position() for the enemies.
- CheckifdotsOver: checks if all pellets have been collected.
- GenerateReport: creates a text based report of the game's results.
  - Calls Player::Stringify().
- Stringify: converts the game object into a string by overloading the ostream operator.