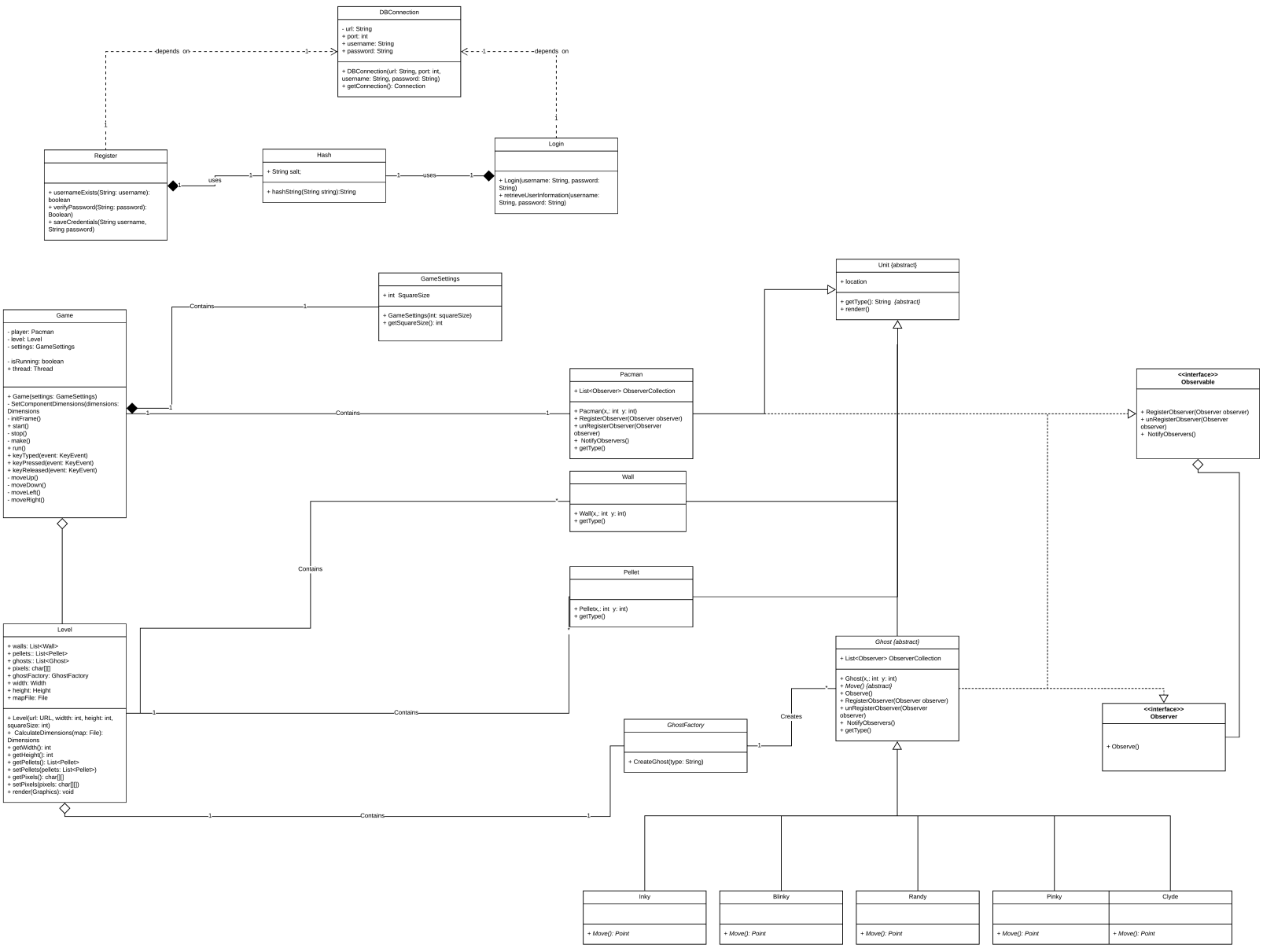
Class diagram

Explanation

In our class diagram we have 3 logical compontents, the authentication system, The game including the level and the different types of units.

authentication gets handled by a databasse and to simple login and register classes that make use of a hash class in order to create a hashed version of the password. Each game contains a level, a pacman and gameSettings. When a game gets initialized the level class draws the board, with walls, pellets and different type of ghosts through the ghostFactory. Inside the game class there is a call to the level.getPellets() method in order to see if the game should be ended or not.

Because each ghost has its own moving behaviour, Ghost is a abstract class that lets the subclasses implement their own moving behaviour. In order to move, some of the ghost need to be aware of both the position of pacman and the other ghosts. Therefor both Pacman and Ghost implement an Observable interface in able to broadcast their positions. Ghost implements the Observe interface as well in order to get updated on the positions of other units.


DBConnection
- url: String
+ port: int
+ username: String
+ password: String
+ DBConnection(url: String, port: int, username: String, password: String)
+ getConnection(): Connection
depends on
depends on
Register
+ usernameExists(String: username): boolean
+ verifyPassword(String: password): Boolean)
+ saveCredentials(String username, String password)
uses
Hash
+ String salt;
+ hashString(String string):String
uses
Login
+ Login(username: String, password: String)
+ retrieveUserInformation(username: String, password: String)
GameSettings
+ int SquareSize
+ GameSettings(int: squareSize)
+ getSquareSize(): int
Contains
Game
- player: Pacman
- level: Level
- settings: GameSettings
- isRunning: boolean
+ thread: Thread
+ Game(settings: GameSettings)
- SetComponentDimensions(dimensions: Dimensions
- initFrame()
+ start()
- stop()
- make()
+ run()
+ keyTyped(event: KeyEvent)
+ keyPressed(event: KeyEvent)
+ keyReleased(event: KeyEvent)
- moveUp()
- moveDown()
- moveLeft()
- moveRight()
Contains
Unit {abstract}
+ location
+ getType(): String {abstract}
+ renderr()
Pacman
+ List<Observer> ObserverCollection
+ Pacman(x,: int y: int)
+ RegisterObserver(Observer observer)
+ unRegisterObserver(Observer observer)
+ NotifyObservers()
+ getType()
<<interface>>
Observable
+ RegisterObserver(Observer observer)
+ unRegisterObserver(Observer observer)
+ NotifyObservers()
Wall
+ Wall(x,: int y: int)
+ getType()
Contains
Pellet
+ Pellet(x,: int y: int)
+ getType()
Level
+ walls: List<Wall>
+ pellets:: List<Pellet>
+ ghosts:: List<Ghost>
+ pixels: char[][]
+ ghostFactory: GhostFactory
+ width: Width
+ height: Height
+ mapFile: File
+ Level(url: URL, width: int, height: int, squareSize: int)
+ CalculateDimensions(map: File): Dimensions
+ getWidth(): int
+ getHeight(): int
+ getPellets(): List<Pellet>
+ setPellets(pellets: List<Pellet>)
+ getPixels(): char[][]
+ setPixels(pixels: char[][])
+ render(Graphics): void
Contains
Ghost {abstract}
+ List<Observer> ObserverCollection
+ Ghost(x,: int y: int)
+ Move() {abstract}
+ Observe()
+ RegisterObserver(Observer observer)
+ unRegisterObserver(Observer observer)
+ NotifyObservers()
+ getType()
Creates
GhostFactory
+ CreateGhost(type: String)
<<interface>>
Observer
+ Observe()
Contains
Inky
+ Move(): Point
Blinky
+ Move(): Point
Randy
+ Move(): Point
Pinky
+ Move(): Point
Clyde
+ Move(): Point

PacMan Sequence Diagram with 3 usecases

group 29