

### Functional requirements

| Name   | FR#1  | Catch pacman |
|--------|---|--------------|
| Resume | Stop or catch a pacman when you click on it   |              |
| Input  |   |              |
| Output | The Pacman is stopped but is still in the game zone and the rebounds that other pacmans have against him continue to be counted |              |

| Name   | FR#2   | Load game |
|--------|--|-----------|
| Resume | The configuration of a new game is loaded from a text file or from a pre-saved game      |           |
| Input  | The name of the file corresponding to a difficulty level or the name of a pre-saved game |           |
| Output | The game is loaded and ready to play   |           |

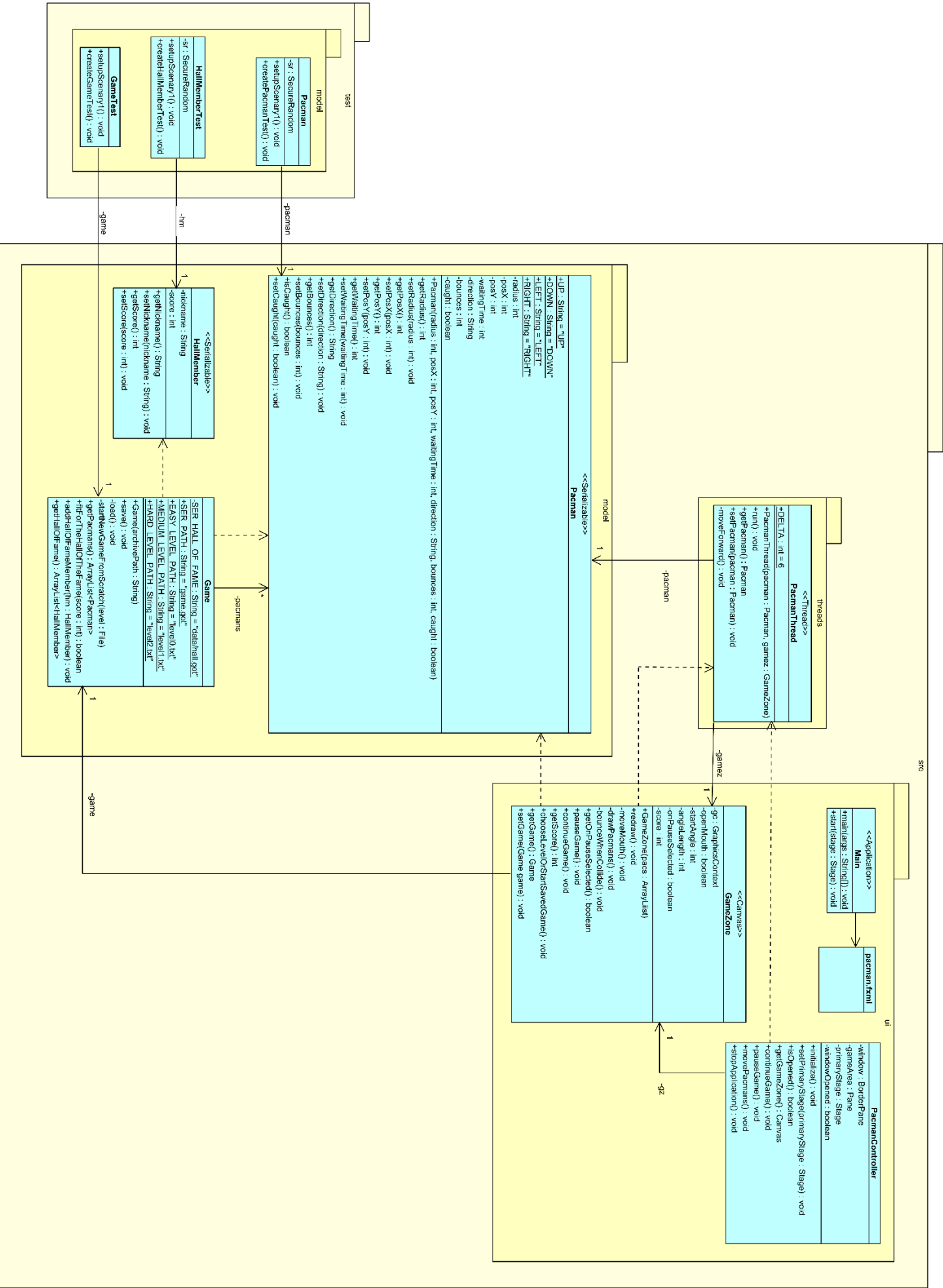
| Name   | FR#3  | Save game |
|--------|---|-----------|
| Resume | Save the current state of the game in a serialized file for later accessibility |           |
| Input  | The name of the file corresponding to the game                                  |           |
| Output | The game is saved in a serializable file  |           |

### Non-functional requirements

| ID    | Description   |
|-------|---|
| NFR#1 | Save the information of the last saved game and hall of fame, so that the program is persistent |
| NFR#2 | Create the interface using the javafx tool  |

### Traceability

| ID   | Functional requirement | Method   | Class                            |
|------|------------------------|--|----------------------------------|
| FR#1 | Catch pacman           | VerifyPacmanCatched.handle   | GameZone.<br>VerifyPacmanCatched |
| FR#2 | Load game              | load<br>Game<br>startNewGameFromScratch<br>chooseLevelOrStartSavedGame | Game<br><br>GameZone             |
| FR#3 | Save game              | save   | Game                             |



### Scenarios

| Name             | Class          | Scenario |
|------------------|----------------|----------|
| setupScenario1() | PacmanTest     | Empty    |
| setupScenario1() | GameTest       | Empty    |
| setupScenario1() | HallMemberTest | Empty    |

### Design of test cases

| Objective of the test: |            | Verify that all the parameters assigned to the POJOs(HallMember and Pacman) are correctly assigned |  |  |
|------------------------|------------|--|--|--|
| Class                  | Method     | Scenario   | Input  | Result   |
| Pacman                 | Pacman     | setupScenario1   | <ul style="list-style-type: none"> <li>radius = sr.nextInt();</li> <li>posX = sr.nextInt();</li> <li>posY = sr.nextInt();</li> <li>waitingTime = sr.nextInt();</li> <li>direction = Pacman.DOWN;</li> <li>bounces = sr.nextInt();</li> <li>boolean caught = sr.nextBoolean();</li> </ul> | A new Pacman has been successfully created. Each of the attributes of the new Pacman has correctly assigned the information passed by parameter.         |
| HallMember             | HallMember | setupScenario1   | <ul style="list-style-type: none"> <li>String nickname = "Rubiu5";</li> <li>int score = sr.nextInt();</li> </ul>   | A new HallMember has been successfully created. Each of the attributes of the new HallMember has correctly assigned the information passed by parameter. |

| Objective of the test: |        | Verify that the Game constructor can read a level file when the path that is passed as a parameter is valid and that it does not read it when it is not |   |  |
|------------------------|--------|---|---|--|
| Class                  | Method | Scenario  | Input   | Result   |
| Game                   | Game   | setupScenario1  | <ul style="list-style-type: none"> <li>path = "path";</li> </ul>              | The archive has not been found so the level has not been loaded              |
| Game                   | Game   | setupScenario1  | <ul style="list-style-type: none"> <li>path = Game.HARD_LEVEL_PATH</li> </ul> | The archive was found in the data directory and has been successfully loaded |