# Programming assignment:

# A video game with a database-backed leaderboard

|  |  |
| --- | --- |
| Now comes the time for Java code. Every time a user plays the game, when the game is over, a new record must be added to the database that includes the player’s name, the score, and what hero they used. Implementing this in Java and SQL is the goal of Task 3. | Space Invaders screenshot |

**INVADERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **player** | **hero** | **gametime** | **score** |
| 1 | crono | 1 | 2017-11-14 12:17:01 | 50 |
| 2 | crono | 2 | 2017-11-14 12:17:01 | 470 |
| 3 | marle | 3 | 2017-11-14 12:17:01 | 320 |
| 4 | lucca | 2 | 2017-11-14 12:17:01 | 5630 |
| 5 | crono | 1 | 2017-11-14 12:17:01 | 480 |

## Task 3: Front-end app for insert

1. Open the NetBeans project from Canvas. It requires JavaFX and JDBC to be set up correctly as libraries, so get help setting this up.
2. Play the game once. When the game is over, a message is printed to the console. (You may need to close the game window to see it.)
3. That message is printed in the updateDatabase method in the ScoreUpdater class in the edu.sxu.databases.invaders.backend package. Open it up and find the spot.
4. Notice that the JdbcTest class is also in the package. It is the code from Lab 22, and it is there for your reference. Copy the code from that class that opens the connection to the database. You will have to change the database from “hafh” to your net ID. Hard code your username and password. Test the program at this point to make sure there are no errors.
5. Finally, you need to run an INSERT command using a prepared statement. The code is like this:

//Record new score in database

String command = "INSERT INTO invaders (player, hero, score) " +

"VALUES (?, ?, ?)";

PreparedStatement stmt;

stmt = conn.prepareStatement(command);

//set parameters (student code)

stmt.executeUpdate();

System.out.println("Score inserted into database.");

1. The comment line above that says “student code” needs to be filled in. You need to call setString and setInt on the stmt object to fill in the values for the question marks. We did this in the lab, and there is one example in the JdbcTest example code. Please ask for help if you are stuck on this part.
2. Play the game a few times, using the same screenname and different screennames. Try different heroes. Then check the database table using PhpMiniAdmin. Make sure that in all cases, the database is updated correctly.
3. Upload the ScoreUpdater.java file to Canvas by the posted due date.