Testing Strategies:

- Unit Testing: involves testing individual components or functions of the software in isolation. The primary goal is to verify that each unit performs as designed.

- Integration Testing: Integration testing focuses on verifying the interactions between different units. The goal is to ensure that combined parts of the application work together as intended.

For our testing Strategies we focused on the game physics and database and the interaction between them.

Test details:

➔ **For database unit test:**

we tested the following functions

- FileExists()
  Test done by creating a test file then check if it exist or not

- Register()

  Test done on two parts first by testing the ability of the function to only accept the correct format. Second by checking if a user file has been created or not.

- Login():

  Test done by creating test user and using multiple sets of usernames and passwords and check if it will accept only the correct information or not.

– saveUserData()

  Test done by creating test game record and check if it will be stored or not

- loadUserData
  Test done by creating a user file with some game record and check if it will be loaded correctly or not

➔ **For game physics unit test:**

we tested the following functions

- checkGameState()
  Test done by creating test board with some played moves on it then check if the function will return the game state correctly or not

- playerMove()
  Test done by inputting row and column and test if the move was played correctly on the board or not.

- AiMove()
  Test done by giving the function an incomplete board and check if the function returns the best possible move or not

➔ **For Integration test between game physics and database:**

The test done by creating a test board then using the functions in game physics to complete the game board then functions from database are used to store the game record after that we check if the stored data are correct or not

Test Results:

```
[==========] Running 10 tests from 3 test suites.
[----------] Global test environment set-up.

[----------] 1 test from IntegrationTest
[ RUN      ] IntegrationTest.integration_test
[       OK ] IntegrationTest.integration_test (3 ms)
[----------] 1 test from IntegrationTest (3 ms total)

[----------] 3 tests from GameTest
[ RUN      ] GameTest.checkGameState
[       OK ] GameTest.checkGameState (0 ms)
[ RUN      ] GameTest.playerMove
[       OK ] GameTest.playerMove (0 ms)
[ RUN      ] GameTest.AiMove
[       OK ] GameTest.AiMove (0 ms)
[----------] 3 tests from GameTest (2 ms total)

[----------] 6 tests from DatabaseTest
[ RUN      ] DatabaseTest.FileExists
[       OK ] DatabaseTest.FileExists (0 ms)
[ RUN      ] DatabaseTest.Register
[       OK ] DatabaseTest.Register (0 ms)
[ RUN      ] DatabaseTest.Login
[       OK ] DatabaseTest.Login (0 ms)
[ RUN      ] DatabaseTest.saveUserData
[       OK ] DatabaseTest.saveUserData (0 ms)
[ RUN      ] DatabaseTest.SaveLastGame
```

```
[       OK ] DatabaseTest.SaveLastGame (0 ms)
[ RUN      ] DatabaseTest.loadUserData
[       OK ] DatabaseTest.loadUserData (0 ms)
[----------] 6 tests from DatabaseTest (2 ms total)

[----------] Global test environment tear-down
[==========] 10 tests from 3 test suites ran. (8 ms total)
[  PASSED  ] 10 tests
```

Coverage Report:

GCC Code Coverage Report

Directory: src/
Date: 2024-06-26 03:33:17
Coverage: low: ≥ 0% medium: ≥ 75.0% high: ≥ 90.0%

	Exec	Total	Coverage
Lines:	351	800	43.9%
Functions:	19	65	29.2%
Branches:	224	1050	21.3%

List of functions

File	Lines		Functions		Branches	
Database.cpp	80.4%	246 / 306	81.2%	13 / 16	44.3%	133 / 300
entry_menu.cpp	0.0%	0 / 23	0.0%	0 / 7	0.0%	0 / 16
game_window.cpp	36.5%	105 / 288	27.3%	6 / 22	25.6%	91 / 356
login_window.cpp	0.0%	0 / 94	0.0%	0 / 8	0.0%	0 / 183
main.cpp	0.0%	0 / 6	0.0%	0 / 1	0.0%	0 / 8
profile_menu.cpp	0.0%	0 / 41	0.0%	0 / 7	0.0%	0 / 138
register_window.cpp	0.0%	0 / 42	0.0%	0 / 4	0.0%	0 / 49

Generated by: GCOVR (Version 7.2)

Figure 1: Coverage Report

In the coverage report Fig.1, all the testes were focused on the database and the game physics part in the game window.