

COMBINED
SRS-SDD-Test
(CRDT)
Document

MasteRookie

abc123

Barış Çağlar Ayyıldız - 1941798

Burak Kahveci - 2114148

Bilgin Coşkun - 2035798

Furkan Buyrukoğlu - 2035756

Table of contents

1.Introduction

1.1.Purpose

1.2.Scope

1.3.Product Perspective

1.4 Initial and revised Project plan

2.System Requirements

2.1.Functional Requirements

3.System Design

3.1.Composition View

3.2.Module Structure

3.3.Source Code Structure

4.Testing

4.1.Details for System Test Plan

4.1.1.Features to be Tested

4.1.2.Features not to be Tested

4.2.Test Case Details

4.3.Detailed Test Results

4.4 Test code structure

5.References

6.Appendices

List of Figures

| | | |
|---|--------------------|----|
| 1 | Use-case diagram | 4 |
| 2 | Component diagram | 9 |
| 3 | Deployment diagram | 10 |
| 4 | Module structure | 11 |

1.Introduction

1.1.Purpose

The purpose of this project is to both provide a common platform or framework to develop, improve and evaluate different game-playing algorithms on 2D games and improving generalized, successful agents. Using this framework, user can try and learn different game-playing algorithms in an interactive way and get detailed results in the end. In addition we provide agents we developed using famous game-playing algorithms and extend them with configurations we added. Users can also use them as benchmark for their projects.

1.2.Scope

The framework allows the users to add the games they wanted using MasteRookie game API and add the agents they would like to test or develop using MasteRookie agent API. The system also allows the users to get detailed and interactive results about the performance of the agent and algorithm chosen and comparison with the algorithms developed and the agents we developed. The system is also open to any improvements like adding new graphs, metrics or saved checkpoints.

1.3.Product Perspective

The system is basically a desktop application and user can use it with a desktop computer necessary software packages installed. An interactive GUI is provided to users for both interaction and visualizing results.

Another interface user should interact with is common APIs. There are basically two APIs that user should use to benefit from the system with full capacity. Game API is used for implementing a game to use in the system, and Agent API is used to implement new agents for the system.

1.4 Initial and revised Project plan

In the first phase our purpose was working on developing comprehensive agents and then we added also extra software engineering properties to our project and developed a framework that both we and other users benefit to develop and evaluate game-playing agents. Now, we have both agents we developed and a framework for testing,evaluating and comparing them.

2. System Requirements

In this section, all the functional and nonfunctional requirements of the system will be provided with supporting diagrams to give good understanding of the system.

2.1. Functional Requirements

Here is the use-case diagram that shows the interaction of the users with the system and the explanation tables for these use-cases.

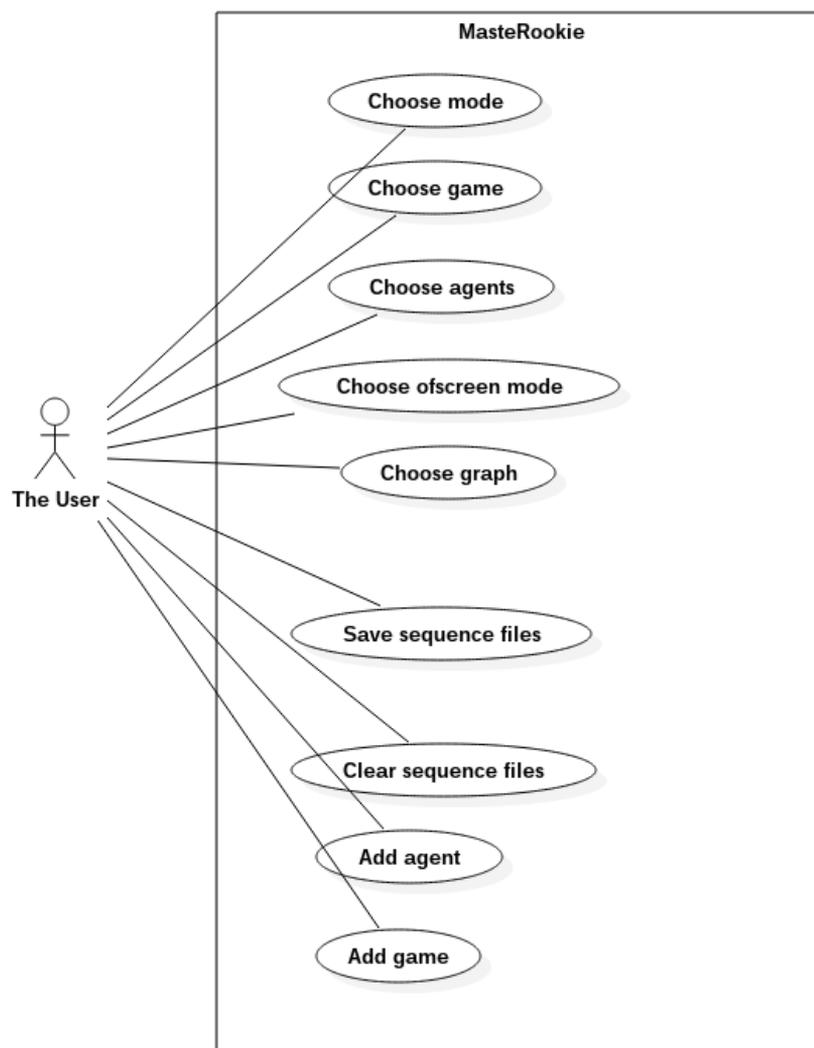


Figure 1 - Use-case Diagram

| | |
|--------------------------|--|
| Use Case Name | Choose mode. |
| Primary Actors | User. |
| Trigger | The user chooses the mode. |
| Precondition | Application is running and Setup tab is displayed. |
| Basic Path | 1 - The user clicks on "Toggle game mode" 2 - The user checks the current game mode from the second tab. |
| Alternative Paths | During the training or comparison process, from "Game and Stats" tab, the user may pass to "Setup" tab and continue from step 1. |
| Postcondition | The game mode is toggled. |
| Exception Paths | |

| | |
|--------------------------|---|
| Use Case Name | Choose game. |
| Primary Actors | User. |
| Trigger | The user chooses the game. |
| Precondition | Application is running and Setup tab is displayed. |
| Basic Path | 1 - The user clicks on "Game" drop down widget. 2 - The user clicks on any game to choose. |
| Alternative Paths | The same basic path can be followed either in "Single" mode or "Versus" mode. |
| Postcondition | The game wanted to be evaluated is chosen. |
| Exception Paths | |

| | |
|--------------------------|--|
| Use Case Name | Choose agent. |
| Primary Actors | User. |
| Trigger | The user chooses the agent (it is either ready or added by the user). |
| Precondition | Application is running and Setup tab is displayed. |
| Basic Path | 1 - The user clicks on “Agent 1” drop down widget. 2 - The user clicks on any agent to choose. |
| Alternative Paths | 1 – The user checks the game mode and ensure that the mode is “Versus” mode. 2 – The user clicks on “Agent 1” drop down widget. 3 - The user clicks on “Agent 2” drop down widget. |
| Postcondition | Agent 1 is chosen and if the mode is “Versus”, Agent 2 is also chosen. |
| Exception Paths | |

| | |
|--------------------------|---|
| Use Case Name | Choose off-screen mode. |
| Primary Actors | The user. |
| Trigger | The user chooses the off-screen mode for training. |
| Precondition | Application is running and Setup tab is displayed. |
| Basic Path | 1 - The user clicks on “Offscreen” check-box widget. |
| Alternative Paths | The same basic path can be followed either in “Single” mode or “Versus” mode. |
| Postcondition | The game screen is not displayed in “Game and Stats” tab while the game is running. |
| Exception Paths | |

| | |
|--------------------------|---|
| Use Case Name | Average-score vs. Epoch graph is added. |
| Primary Actors | The user. |
| Trigger | The user chooses the Average-score vs. Epoch choice. |
| Precondition | Application is running and Setup tab is displayed. |
| Basic Path | 1 - The user clicks on "Av. Score vs Epoch" check-box widget. |
| Alternative Paths | The same basic path can be followed either in "Single" mode or "Versus" mode. |
| Postcondition | During training or evaluation process, Average-score vs. Epoch graph is plotted lively. |
| Exception Paths | |

| | |
|--------------------------|---|
| Use Case Name | Max-score vs. Epoch graph is added. |
| Primary Actors | The user. |
| Trigger | The user chooses the Max-score vs. Epoch choice. |
| Precondition | Application is running and Setup tab is displayed. |
| Basic Path | 1 - The user clicks on "Max. Score vs Epoch" check-box widget. |
| Alternative Paths | The same basic path can be followed either in "Single" mode or "Versus" mode. |
| Postcondition | During training or evaluation process, Max-score vs. Epoch graph is plotted lively. |
| Exception Paths | |

| | |
|--------------------------|--|
| Use Case Name | Choose saving Sequence files. |
| Primary Actors | The user. |
| Trigger | The user chooses to save sequence files. |
| Precondition | Application is running and Setup tab is displayed. |
| Basic Path | 1 - The user clicks on “Save Sequence Files” check-box widget. |
| Alternative Paths | The same basic path can be followed either in “Single” mode or “Versus” mode. |
| Postcondition | While games are running, the sequence files are added to the “seq” folder under game folder. |
| Exception Paths | |

| | |
|--------------------------|---|
| Use Case Name | Clearing Sequence files. |
| Primary Actors | The user. |
| Trigger | The user chooses to clear sequence files from previous session. |
| Precondition | Application is running and Setup tab is displayed. |
| Basic Path | 1 - The user clicks on “Clear Sequence Files” button. |
| Alternative Paths | The same basic path can be followed either in “Single” mode or “Versus” mode. |
| Postcondition | The sequence files in the “seq” folder under game folder are deleted. |
| Exception Paths | |

3.System Design

In this part, the key points, ideas and main structures we used in our design will be presented with the help of different diagrams. In the first section, component-wise design points will be explained, and in the second section modules will be presented. The last section will be about the "git" code structure related to our model structure.

3.1.Composition View

In this section, a top-level physical and logical components of the entire system will be presented with component-wise perspective. We will mainly describe the components, and logical and physical relationships, interactions, dependencies between them and the environment. While doing this, we will benefit from basic UML diagrams and notation.

Firstly, here is the component diagram that mainly shows basic system components and the logical relationships or interactions between them.

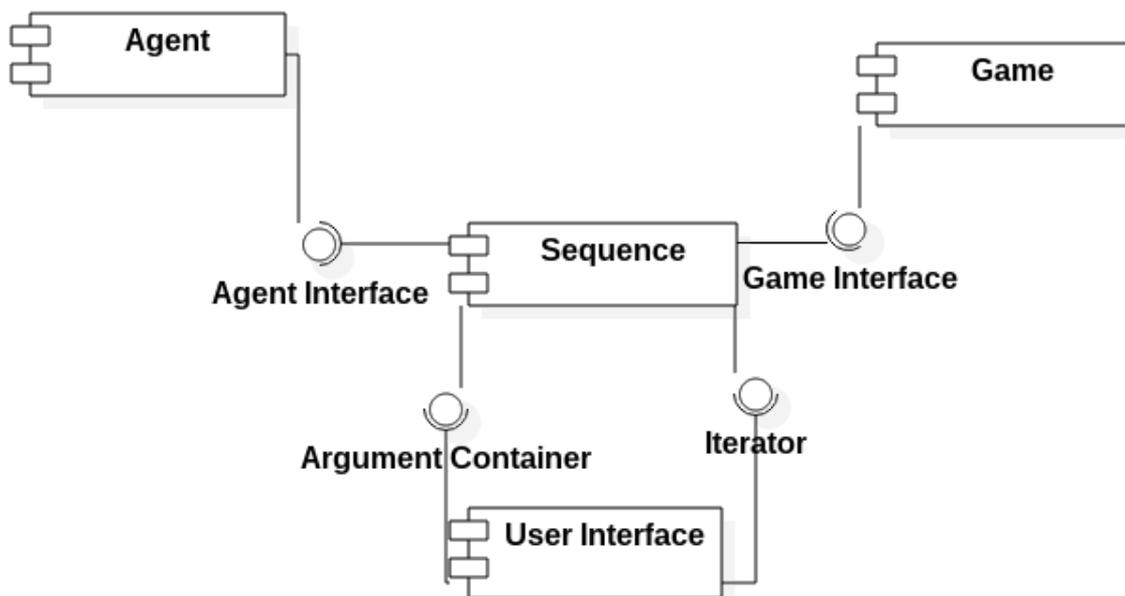


Figure 2 - Component Diagram

Here is the deployment diagram that shows the relationship between software, hardware components and the environment.

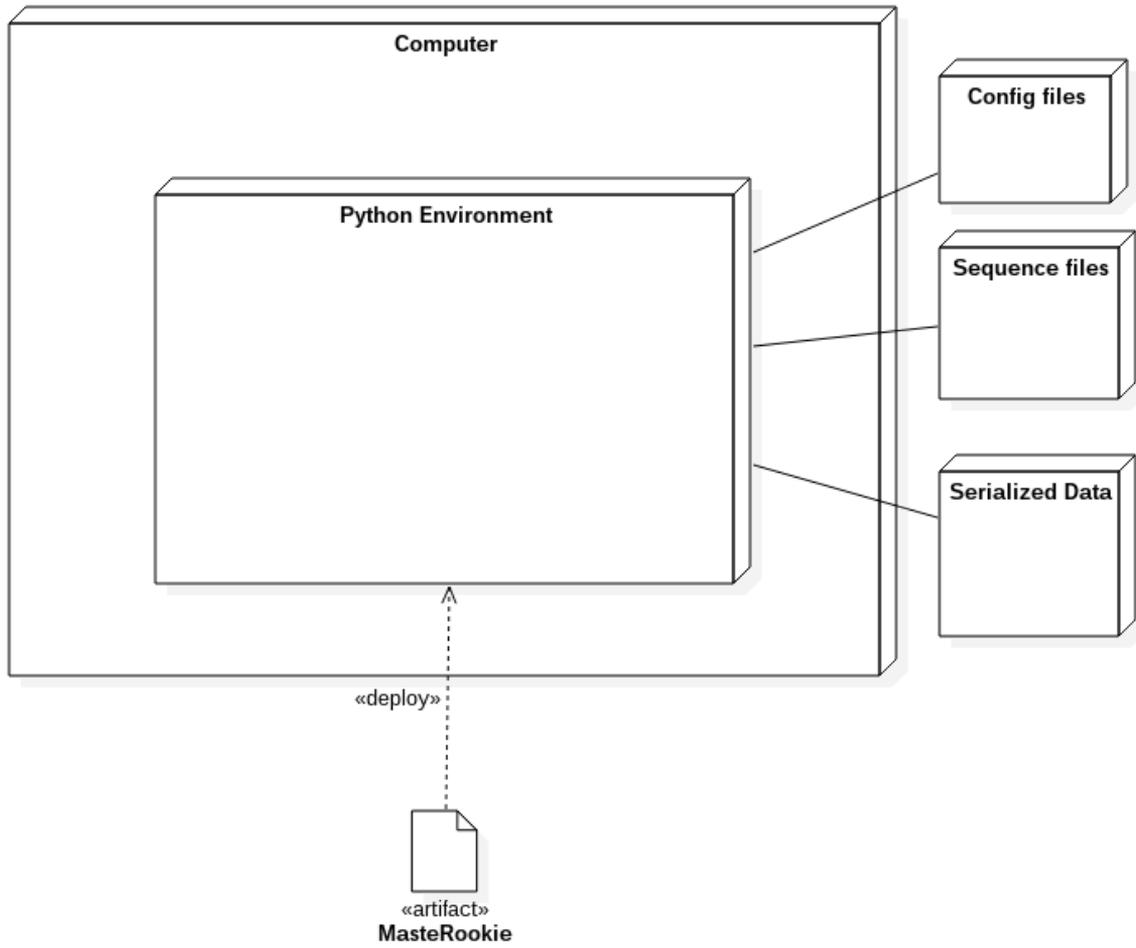


Figure 3 - Deployment Diagram

3.2. Module Structure

After an overview and component-wise presentation of our system, here is a detailed module-based diagram of the system. In the next section, the related “git” structure with these modules will be provided.

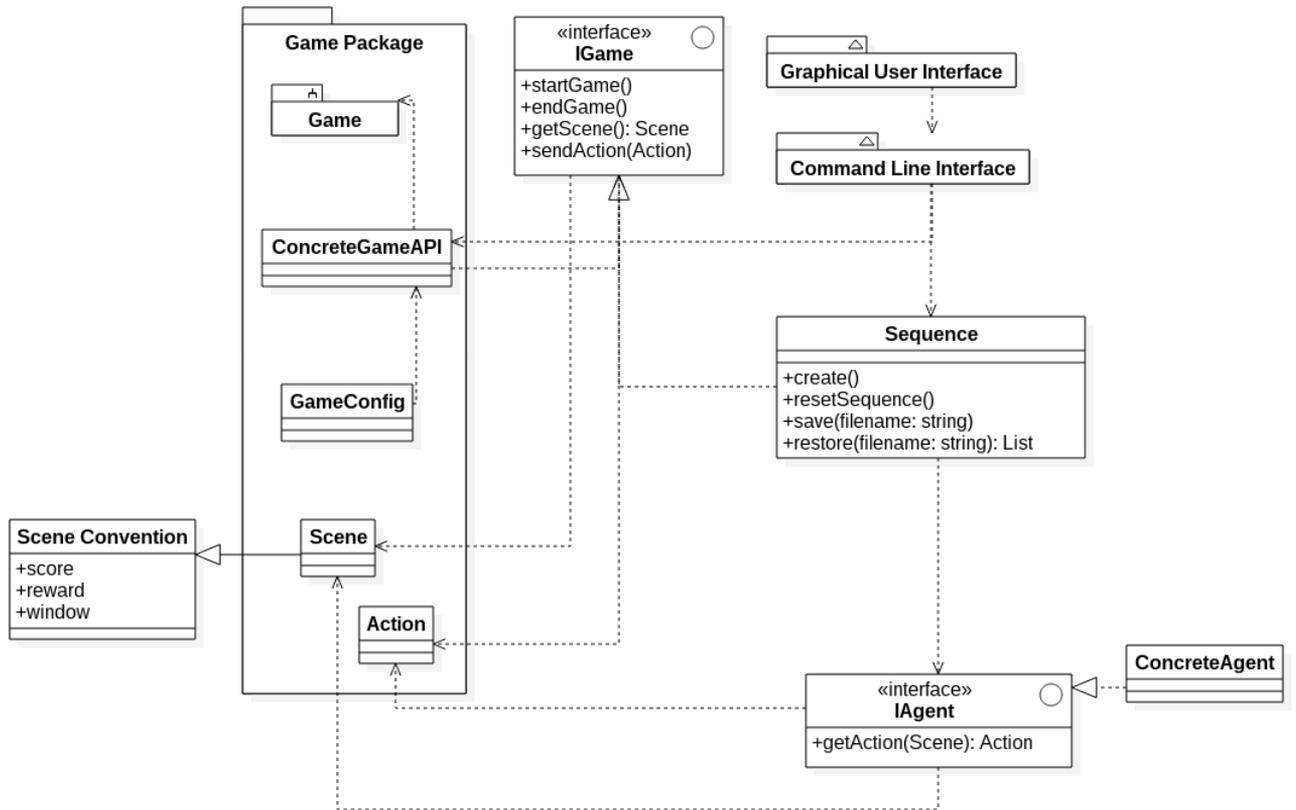


Figure 4 -Module Structure

3.3. Source Code Structure

GUI - gui.py

CLI - cli.py

Sequence - sequence.py

IAgent - iagent.py

IGame - igrade.py

ConcreteAgent - Each file under agents/

Game Package - Each folder under games/. Each one should include a file named <gamefoldername>config.py. Examples can be examined for format. And games can include a <gamefoldername>api.py with a class implementing IGame for communication between agent and system.

4. Testing

This part of the document is for testing the functions and features of “MasteRookie” and can be used as a reference to test and evaluate the system completely. In the next section, a detailed test report and results of it are presented in a systematic way. The test cases basically determine if the system meets the requirements written in System Requirements part.

4.1. Details for System Test Plan

In this section, all the functionalities and features to be tested or not to be tested will be explained in different test cases and evaluation criterias to pass or fail for them will also be explained. In addition, traceability matrix that links the items to be tested with the related requirements will also be provided.

4.1.1. Features to be Tested

The main functionalities of the Graphical User Interface and internal modules explained in System Design section and their interaction will be tested with the test cases that will be presented in the next section.

4.1.2. Features not to be Tested

Since it could last more than a couple of hours to train, try and evaluate, the agents will not be tested specifically.

4.2. Test Case Details

| | |
|--------------------------------|--|
| Test case identifier | T1 |
| Objective | Initializing Sequence object. |
| Inputs | 4 "None" type Python objects as constructor arguments. |
| Outcome(s) | Empty Sequence object will be created. |
| Environmental needs | Any device that can run Python codes. |
| Procedural Requirements | - |
| Intercase Dependencies | - |

| | |
|--------------------------------|---|
| Test case identifier | T2 |
| Objective | Creating Empty Sequence. |
| Inputs | A time interval ,game instance,agent instance,and a sequence name as constructor arguments. |
| Outcome(s) | A Sequence object will be created. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | - |
| Intercase Dependencies | - |

| | |
|--------------------------------|---|
| Test case identifier | T3 |
| Objective | Checking if the method order run by Sequence objects is true. |
| Inputs | A time interval ,game instance,agent instance,and a sequence name as constructor arguments and start a dummy session. |
| Outcome(s) | The true method order called is 1- Start game 2- Get scene 3- Get action 4- Send action |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | - |
| Intercase Dependencies | - |

| | |
|--------------------------------|--|
| Test case identifier | T4 |
| Objective | Checking if Sequence objects saved succesfully. |
| Inputs | A time interval ,game instance,agent instance,and a sequence name as constructor arguments and save sequence object. |
| Outcome(s) | A sequence file with the name "given_name.seq" will be created in the "seq" folder of the given name. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | - |
| Intercase Dependencies | - |

| | |
|--------------------------------|---|
| Test case identifier | T5 |
| Objective | Choosing single mode. |
| Inputs | “Toggle Game Mode” button is pressed and released until the single agent dropdown menu appears. Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | “Games and Stats” tab is chosen and only single game screen window is appeared. |
| Environmental needs | Any device that can run Python codes with required libraries. |
| Procedural Requirements | Setup tab is displayed. |
| Intercase Dependencies | T1 -T2 |

| | |
|--------------------------------|--|
| Test case identifier | T6 |
| Objective | Choosing versus mode. |
| Inputs | “Toggle Game Mode” button is pressed and released until two agent dropdown menus appear. Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | “Games and Stats” tab is chosen, Two game screen windows (one is on the left side and the other one is on the right side) are appeared. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed. |
| Intercase Dependencies | T1 -T2 |

| | |
|--------------------------------|---|
| Test case identifier | T7 |
| Objective | Choosing a specific game. |
| Inputs | From the "Game" dropdown menu, any game wanted can be chosen. Then a sample session is started with "Start Game" button pressed and released. |
| Outcome(s) | "Games and Stats" tab is chosen, the chosen game is appeared and played by the default agent chosen in the window. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed. |
| Intercase Dependencies | T1 -T2 - (T5 or T6) |

| | |
|--------------------------------|---|
| Test case identifier | T8 |
| Objective | Choosing a single agent. |
| Inputs | From the "Agent 1" dropdown menu, any agent wanted can be chosen. Then a sample session is started with "Start Game" button pressed and released. |
| Outcome(s) | "Games and Stats" tab is chosen, the default game is played by chosen agent and that agent's name is written in a text box under the screen. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed and single mode is chosen. |
| Intercase Dependencies | T1 -T2 - T5 |

| | |
|--------------------------------|--|
| Test case identifier | T9 |
| Objective | Choosing two agents in versus mode. |
| Inputs | From the “Agent 1” dropdown menu, any agent wanted can be chosen for the first agent, from the “Agent 2” dropdown menu, any agent wanted can be chosen for the second agent .Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | “Games and Stats” tab is chosen, the default game is played by chosen agents (one on the left and the other one on the right) and that agent’s names are written in a text box under screens. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed and versus mode is chosen. |
| Intercase Dependencies | T1 -T2 - T6 |

| | |
|--------------------------------|---|
| Test case identifier | T10 |
| Objective | Choosing offscreen mode. |
| Inputs | “offscreen” checkbox is chosen .Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | “Games and Stats” tab is chosen, the screen window appears as empty but the game is being played internally. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed. |
| Intercase Dependencies | T1 -T2 |

| | |
|--------------------------------|---|
| Test case identifier | T11 |
| Objective | Average-score vs. Epoch graph choice is chosen in single mode. |
| Inputs | “Av. Score vs Epoch” checkbox is chosen .Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | “Games and Stats” tab is chosen, while the game is playing, Average-score vs. Epoch graph is plotted lively after a predefined number of epoches. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed and single mode is chosen. |
| Intercase Dependencies | T1 -T2 - T5 |

| | |
|--------------------------------|---|
| Test case identifier | T12 |
| Objective | Max-score vs. Epoch graph choice is chosen in single mode. |
| Inputs | “Max Score vs Epoch” checkbox is chosen .Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | “Games and Stats” tab is chosen, while the game is playing, Max-score vs. Epoch graph is plotted lively after a predefined number of epoches. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed and single mode is chosen. |
| Intercase Dependencies | T1 -T2 - T5 |

| | |
|--------------------------------|---|
| Test case identifier | T13 |
| Objective | Average-score vs. Epoch graph choice is chosen in versus mode. |
| Inputs | “Av. Score vs Epoch” checkbox is chosen .Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | “Games and Stats” tab is chosen, while the game is playing, Average-score vs. Epoch graph is plotted lively with two lines which represent two agents with different colors after a predefined number of epoches. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed and versus mode is chosen. |
| Intercase Dependencies | T1 -T2 -T6 |

| | |
|--------------------------------|---|
| Test case identifier | T14 |
| Objective | Max-score vs. Epoch graph choice is chosen in versus mode. |
| Inputs | “Max Score vs Epoch” checkbox is chosen .Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | “Games and Stats” tab is chosen, while the game is playing, Max-score vs. Epoch graph is plotted lively with two lines which represent two agents with different colors after a predefined number of epoches. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed and versus mode is chosen. |
| Intercase Dependencies | T1 -T2 - T6 |

| | |
|--------------------------------|--|
| Test case identifier | T15 |
| Objective | Saving sequence files. |
| Inputs | “Save Sequence Files” checkbox is chosen .Then a sample session is started with “Start Game” button pressed and released. |
| Outcome(s) | While the game is playing, sequence files for selected agent (with the following format: “timestamp-game-agent-seq”) is added to the game folder of the chosen game. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed and single mode is chosen. |
| Intercase Dependencies | T1 -T2 -T4 -T6 |

| | |
|--------------------------------|---|
| Test case identifier | T16 |
| Objective | Delete Sequence Files. |
| Inputs | “Delete Sequence Files” button is pressed and released. |
| Outcome(s) | All the sequence files for selected game are deleted from the game folder. |
| Environmental needs | Any device that can run Python codes |
| Procedural Requirements | Setup tab is displayed. |
| Intercase Dependencies | - |

4.3.Detailed Test Results

Here are the tested results of the test cases written in the previous section according to the related outcome or criteria given in each test case.

| TEST CASE | RESULT |
|------------------|---------------|
| T1 | Passed |
| T2 | Passed |
| T3 | Passed |
| T4 | Passed |
| T5 | Passed |
| T6 | Passed |
| T7 | Passed |
| T8 | Passed |
| T9 | Passed |
| T10 | Passed |
| T11 | Passed |
| T12 | Passed |
| T13 | Passed |
| T14 | Passed |
| T15 | Passed |
| T16 | Passed |

4.4 Test code structure

GUI Tests- guitest.py under Tests folder

Sequence Tests - sequencetest.py under Tests folder

5.References

- ISO-IEC-IEEE 29148 2011 - Systems and Software Engineering
- IEEE 1016-2009
- <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5167255>
- <http://docs.staruml.io/en/latest/index.html>
- IEEE. IEEE Std 1016 R - 2009.IEEE Standard for Information Technology-Systems Design- Software Design . Software & Systems Engineering Standards Committee of the IEEE Computer Society, 2009.

6.Appendices

There is no extra appendices apart from the figures and diagrams in the document. The figures are presented in the beginning part of the document.