Marking Criteria

	Very Poor	Poor	Acceptable	Good	Excellent
Criteria	0-19%	20-39%	40-59%	60-79%	80-100%
Report	Minimal report that is underdeveloped and incomplete	**Structure:** A functional report that includes the key sections. Some subsections are missing, which jeopardizes the consistence and completeness of the report. **Methods:** Methods are not structured, without description of how you fit them in your application. **Results:** Results are solely presented. Some errors are identified. **Language:** There are typos, grammar errors, etc. in the report. **Visualization:** Tables and figures are presented with poor quality. The lack of captions or legends that makes them difficult to understand.	**Structure:** A report that includes the key sections. Most subsections are considered, making the report consistent. **Methods:** Methods are structured (such as presented as an algorithm or a flow diagram), with description of fitting them in your application. **Results:** Results are presented with comparison. **Language:** Mostly free of typos, grammar errors, etc. in the report. **Visualization:** Tables and figures are presented with acceptable quality. Captions and legends are provided to understand tables and figures.	**Structure:** A well-developed report that covers all the sections/subsections. The report is consistent and complete. **Methods:** Methods are structured (such as presented as an algorithm or a flow diagram), with detailed description of a progressive strategy to fit them in your application. **Results:** Results are presented with comparison and analysis. **Language:** Free of typos, grammar errors, etc. in the report. Writing with professional language. **Visualization:** Tables and figures are presented with good quality. Captions and legends are provided to understand tables and figures. Colors, line styles, and markers are controlled to make figures easy to read.	**Structure:** An excellent report that covers all the sections/subsections. The report is consistent and complete. **Methods:** Methods are clearly and concisely presented in a self-designed form. A thorough and detailed description of progressive strategy to fit them in your application. **Results:** Results are presented with comparison, analysis, and insightful arguments. **Language:** Free of typos, grammar errors, etc. in the report. Writing with professional language. **Visualization:** Tables and figures are presented with good quality. Captions and legends are provided to understand tables and figures. Colors, line styles, and markers are controlled to make figures easy to read.

	Very Poor	Poor	Acceptable	Good	Excellent
Criteria	0-19%	20-39%	40-59%	60-79%	80-100%
Technical Implementation	**Novelty:** Minimal modification of the provided template environment.	**Novelty:** Some modification of the template environment. Signs of effort in	**Novelty:** Modification of the template environment, with additional functions to adapt the codes to your	**Novelty:** Adding new functions/classes/ modules, etc. to the template environment, with evidence showing	**Novelty:** A complete self-designed set of codes that works for your application. Your self-designed

	With no effort in designing action model and task environment. **Reproducibility:** Codes are not reproducible. **Clarity:** No comments for key functions/classes/ modules. No 'readme.txt' file to help run the codes. **Professionality:** No constancy in codes structure, form layout or naming convention.	designing an action model and a simple task environment. **Reproducibility:** Codes can be used for reproducing some results. Some functions/module s are not executable due to grammar errors, etc. **Clarity:** Minimal comments for key functions/classes/ modules. 'Readme.txt' file does not provide a step-by-step instruction to re-run the codes. The programming language, version, environment, etc. information is not clarified in the file. **Professionality:** Little constancy in codes structure, form layout or naming convention.	application. Evidence in designing a realistic action model and a task environment with certain complexity, i.e., borders, more obstacles, etc. **Reproducibility:** Codes can be used for reproducing the main results. Some functions/modules may not be executable, but it does not affect getting the main results. **Clarity:** Simple comments for key functions/classes/ modules., that help understand the codes. 'Readme.txt' file does provide a step-by-step instruction to re-run the codes. The programming language, platform, environment, etc. information is not clarified in the file. **Professionality:** Some constancy in codes structure, form layout or naming convention.	that the performance of the codes is improved in your application. Evidence in designing modules that automatically generates a realistic action model and a task environment with certain complexity, i.e., borders, more obstacles, etc. **Reproducibility:** Codes can be used for reproducing the results, with no errors. **Clarity:** Detailed comments for key functions/classes/ modules., that help understand the codes. 'Readme.txt' file provides a step-by-step instruction to re-run the codes. The programming language, platform, environment, etc. information is clarified in the file. With notifications of potential failures and solutions to resolve them. **Professionality:** Constancy in codes structure, form layout or naming convention.	codes show the same or better performance than the provided template environment. Automatic action model module. A module that generates environment with no restrictions on complexity of the environment. **Reproducibility:** Codes can be used for reproducing the results, with no errors. **Clarity:** Detailed comments for key functions/classes/ modules., that help understand the codes. Suggestions on improvements, etc. 'Readme.txt' file provide a step-by-step instruction to re-run the codes. The programming language, version, environment, etc. information is clarified in the file. With notification of potential failures and solutions to resolve them. **Professionality:** High level of constancy in codes structure, form layout or naming convention.