Project Deep Dive Open.

User manual.

Abdón Crespo Álvarez.

This document will serve as a guide of how to access all the aspects of the project and the controls of every mode that the engine has to offer. It assumes that the project was configured successfully. In consequence, refer to the installation manual provided along with this one and found in the same directory.

First of all, in order to start the engine, if the user wishes to launch it with its Visual Studio project opened, it can be done by pressing the F5 key. That will compile the project and launch it with the possibility of adding breakpoints and other debug utilities offered by the IDE.
On the other hand, if the user desires to execute the engine without using Visual Studio, then the user must navigate (assuming that the address starts from the project's root directory) to /x64 (or the type of system that the project was compiled for by the user) and then, in the "Debug" directory, make sure that the directories "AIData", "records", Resources" and "saves" are all updated (as, for example, if the shader programs are missing, the engine will produce an error when attempting to load them). Finally, in order to start the engine, execute the "Deep dive open.exe" executable file.

Once the engine starts, by default, a command console will open featuring a menu with three possible options. **Enter 1** to start the example AI game and access its main menu, **2** to enter the level editor mode and **3** to stop the engine's execution.

If the user chooses to enter the example AI game, another menu will be printed to the console with 8 options, being the **8th one** to exit the example AI game and return to the previously mentioned menu. **The first two options** are for, respectively, initiating agent training simulations with the first one generating the AI agents with random weights (according to the parameters given to the AI game in the code) and the other one loading some AI agents from an correct ".aidata" file. Tampering with ".aidata" files is not supported and could lead to undefined behaviour.

The next two options are the same as the first two ones except they are for initiating testing simulations in which the AI agents are not trained and the only output is some statistics of how well they performed when completing their tasks

When training or testing AI agents, the engine will ask for a number of agents to create for the process. **In training mode,** said number must be even due to the genetic operators used during the simulation. **When loading previously generated AI data,** the engine will ask instead the path to its corresponding file and how many agents to load from the file (if the user declares to load “n” agents, the engine will load the first “n” agents found in the “.aidata” file).

Once the number of AI agents is specified (and they are loaded into the engine by using previously generated AI data), for training and testing modes, the engine will ask for a number of epochs for the simulation. An epoch is one iteration of the game or one game “match” in which the agents do their task. If there was an AI game of football, an epoch would be considered a single match.

The 5th and 6th options are for generating a record of one epoch with, respectively, AI agents with randomly generated weights or with agents that are loaded from a “.aidata” file. When selecting one of these options, the engine will first ask for the name of the “.rec” file that will be generated containing the record of the match.
The “.aidata” and “.rec” files are stored, respectively, in the “AIData” and “records” directories, found in the same directory where “Deep dive open.exe” is found. Inside those directories, a folder is created for each registered AI game in order not to mix files from diferent AI games as the definitions of AI data and AI actions may vary per AI games.

If the record’s name corresponds to one already existing in the AI game’s record file, then the engine will ask if the user desires to override the existing file or not. If not, then it will ask for another name.

The 7th option is for playing a previously generated record. The engine will ask the user the name of said record and, if found, it will start the engine’s graphical mode, load the level that is associated with said recording (inside the /saves/recordingsWorlds directory found in the same folder as where the “Deep dive open.exe” file is located).

Once the engine’s graphical part initialises in record playback mode, the user will have a free 3D camera to observe the level and the actions that the AI agents are doing. The controls for this mode are:

- W key for moving forward in the viewing direction.

- S key for moving backwards in the viewing direction.
- D key for moving to the right side in the viewing direction.
- A key for moving to the left side in the viewing direction.
- Move the mouse for looking around and change the viewing direction.
- X key for stopping the recording's playback.
- Down arrow key for pausing the recording's playback.
- Left arrow key for executing the record file's AI actions backwards.
- Right arrow key for executing the record fikle's AI action forward.

Finally, what remains to explain about **how to use this engine is the edit level mode,** activated by choosing the second option of the engine's initial menu.

Entering in this mode will activate the engine's graphical mode and display to the user a GUI (Graphical User Interface) that has three options: **"LOAD", "NEW" y "EXIT". The last one is** used for stopping the engine's graphical mode and return to its initial menu.

The "LOAD" option is used to load a previously existing level saved in one of the 5 level slots provided by the level. This load level will have 5 buttons (one for level slot) and another one for returning to the previous menu.
On the other hand, the "NEW" option is for randomly generating a new level and enter it.

Once the user enters a level in edit mode, it will have the same camera and controls as in record playback mode, with the exception that the X, left arrow, down arrow and right arrow are disabled as there is no record being played. However, the user has new controls, with the left mouse button, the user will destroy the block he is looking at if it is close enough. On the other side, with the right mouse button, the user will place a block where it is looking (only if it is looking at a non-null block first). Also, the user has 9 types of different blocks to place and to select them the user needs to press one of the 1 to 9 keys in the keyboard (the ones that are usually just below the function keys).

Additionally, the user can access an options menu by pressing the Escape key. It will have three options "LOAD", "SAVE" and "EXIT". The first one serves the same purpose as the one with the same name that was explained before, while the second is used to save the currently loaded level into one of the 5 slots (the menu for selecting said slots is equal to the load menu introduced earlier).
Finally, choosing the "EXIT" button WILL **NOT** automatically save the current level and the user will return to the graphical menu that was introduced early in this document with the "LOAD", "NEW" and "EXIT" options.