

Maintenance Document

Contents

OVERVIEW	1
CORRECTIVE MAINTENANCE	1
ADAPTIVE MAINTENANCE	2
COMMENTARY	3
REFERENCES	4

Overview

This document explains how maintenance to the project should be performed. There are two types of maintenance to consider:

- 1) Corrective maintenance - Ensuring any bugs discovered after the game has been released are fixed. This is done by releasing patches and new versions of the game.
- 2) Adaptive maintenance - Ensuring the game runs correctly on new versions of operating systems or hardware.

The "Commentary" section gives the standard used for commenting source code.

Corrective Maintenance

Corrective maintenance - Ensuring any bugs discovered after the game has been released are fixed. This is done by releasing patches and new versions of the game.

The game was created using the development environment Game Maker [1]. Irrespective of the environment used it is almost impossible to create a game without errors and bugs that are only detected after it was released. Users will often report these errors to the game developer. Changes must be made to the game software in order to eliminate the error. If a user did report an error, it is important that they give the developer a copy of the **game_errors** file in the folder Hunger was installed. This will contain a record of the text message that was generated by the error. This message will give the section of code that caused the error. All code used in the game software is contained in individual objects or in scripts. Objects have been labelled with the prefix **obj_** and scripts with the prefix **scr_** so they can be easily identified. The section contained in the error message will contain one of these labels.

Once the section of code has been identified it is often useful to insert **show_message** function calls into the program. These stop the game running until the OK button is clicked.

Identifying which messages do display and which do not can be a good indication of exactly where the error occurs. Note that a line number recorded in the error message will only indicate where the error was detected and not necessarily where the error occurred.

By running the game in debug mode the values in global and instance variables can be determined. The times they change value can also be seen. This is done by entering the name of the required variable in the watch list. Using the debug feature simple expressions can be evaluated during run-time and their result displayed. For example, entering the expression **instance_exists(obj_rot)** will display 1 if there are any Rot objects present in that level and 0 if there are none. For more information about how to use debug mode see reference [1].

Once the error has been fixed the software can be re-released as a patch [3]. The easiest way to do this would be to make sure the new version of the game will overwrite the previous version when it is installed. The new version is distributed and replaces the previous version.

Adaptive Maintenance

Adaptive maintenance - Ensuring the game runs correctly on new versions of operating systems or hardware.

The game was created using the development environment Game Maker [1]. This is documented to produce games that are portable to any hardware and any version of the Microsoft Windows operating system. However, it is possible that some types of hardware or versions of operating system may stop the game executing as expected. The problem here is that there are so many different types of hardware it is almost impossible to ensure the game runs correctly on each one. The most feasible way to test whether the game works on a particular set of hardware is to release it to users. The same applies for operating systems.

If users find any errors with the game they will hopefully report these errors to the game developer. The developer can repeat the procedure leading to the error on different hardware and on the operating system the user was running. If this produced an error the problem is likely a bug in the games software (see "Corrective Maintenance"). If, however, no error was produced the problem is caused by the different hardware or operating system.

Assuming the reported error was a problem caused by different hardware or a different operating system there are three ways the problem might be solved.

1. The most drastic option is to access the Game Maker website [1] and check the latest version of the Game Maker software. If the latest version is the same as the version used to create the game this option will not work. If the latest version of Game Maker is different, download that version. The majority of versions of Game Maker are backwards compatible with games made in older versions. Update the game, or part of it which may be causing the problem, with any new functions if available. Test the game to ensure it works as before the update. If this reveals no problems the new version of the game can be released.
2. Carry out more testing to narrow down the part of the game software in which the problem is caused. Once the general area of the problem is known the Game Maker community can be asked if they have had similar problems. This can be done by posting the section of code on the Game Maker forum [2]. Copyright over the section of code posted would need to be stated to protect it from being illegally copied and used. If this did not solve the problem, the creator of Game Maker can be e-mailed as there may be a bug in Game Maker.
3. Many standard functions in Game Maker have equivalents that perform the same task in different ways. Carry out more testing on the game to narrow down the part of the game software in which the problem is caused. Once the area of the problem is known these equivalent functions can be used in place of functions that could be causing the problem.

Commentary

The source code has been commented to explain any difficult sections. Any code written after the game is released should also be commented in this way. The date the code was changed, a brief summary of the change or changes made and the problem those changes solve should be placed at the top of the section of code as a comment.

If the task or tasks performed by a script is changed, its new task must be summarised and placed at the top of the script as a comment. Any new scripts written must also be commented in this way.

References

- [1] Title: Game Maker by Mark Overmars
Website URL: <http://www.gamemaker.nl/>
- [2] Title: Game Maker Community (Powered by Invision Power Board)
Website URL: <http://forums.gamemaker.nl/>
- [3] Title: Wikipedia - Patch (computing)
Website URL: [http://en.wikipedia.org/wiki/Patch_\(computing\)](http://en.wikipedia.org/wiki/Patch_(computing))