

CONFIGURATION MANAGEMENT REPORT

By

MIR

N. Özlem ÖZCAN - 1250596

Hamza KAYA - 1250448

Mustafa Özturun - 1250638

Ozan Şimşek - 1250760

TABLE OF CONTENTS

1. Introduction	3
1.1 Purpose of CMP	3
1.2 Scope of Document	3
1.3 Definitions, Acronyms and Abbreviations	4
1.4 References	4
2. The Organizations CM Framework	4
2.1 Organization	4
2.2 Responsibilities	5
2.3 Tools & Infrastructure	6
3. The CM Process	6
3.1 Identification	6
3.2 Management and Control	7
3.3 Configuration Status Accounting	8
3.4 Auditing	8
4. Project Schedule and Milestones	9
5. Project Resources	10
6. Plan Optimization	10

1. INTRODUCTION

1.1 PURPOSE of CMP

This configuration management plan defines the system that will be used by MIR company for developing the 3D adventure game named `1956`. The purpose of this document is to establish the process of identifying, documenting, monitoring, controlling, and approving changes in coding during the project. By setting up a formal configuration management plan, preserving consistency and preventing any conflicts between the group members will be easy, change control is effectively established and maintained and the development of documentation is handled.

As software configuration management organizational role, everyone in the team will do the SCM activities. For instance, if a member of the team makes a change then the others will check to see if it is properly done. So this will ensure that parallel changes made by two different people do not overwrite one another. Also the changes will be documented as the change, why it was done, when, by who, on which level. We will be able to refer to them in the future phases of the project.

1.2 SCOPE of DOCUMENT

In this document all configuration management activities of `1956` project are described. It defines the identification, verification, change and version control, audits, responsibilities, schedules and procedures for controlling the system configuration during the development, testing, and deployment of `1956`.

1.3 DEFINITIONS, ACRONYMS and ABBREVIATIONS

CCB	Configuration Control Board
CI	Configuration Item
CM	Configuration Management
CMP	Configuration Management Plan
CSA	Configuration Status Accounting
CVS	Concurrent Versions System
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan

1.4 REFERENCES

MIR Requirement Analysis Report

MIR Initial Design Report

MIR Detailed Design Report

www.cvshome.org

2- THE ORGANIZATIONS CM FRAMEWORK

2.1 ORGANIZATION

Our project group consists of four people. Besides dealing with the design of the phases, all of the team members will work in implementation of the project. We adopt a democratic-decentralized type of management so nobody has special dominance over others, nobody is the leader of the group, so everyone is equal and the team is self-organizing. Decisions about the problems and the solutions to these problems will be made by group members together in the weekly meetings.

The management should be decentralized because the project team is very small and the members are not specialized in any kind of duties. Every member is responsible to finish the assigned part on time and discuss it with others. There are no specific roles like

manager, designer, programmer, tester, documenter etc. Also by using the democratic type, we believe that solving the problems we encounter will be easier.

Another point is that project management will be up to all group members rather than one person. In this way we are planning to make sure that responsibility of the project is shared equally between the members of the project team. However, a leader is useful for holding the group together and organizing the events such as meetings. That's why we chose having a leader, but this leader does not decide for us, he organizes us. We believe that is fair and sensible since benefits and income of a successful project will be equal for all members.

As a result of democratic-decentralized management approach, decisions and approaches will be taken by group consensus. This will have a fair affect on sharing the responsibilities in an equal manner while there is a leader that organizes us.

2.2 RESPONSIBILITIES

Although the group members are all responsible for design-related parts of the project like creating the models and the textures, there will be some specific tasks that are assigned to specific members. Here is a list of them:

Hamza Kaya:

Model loading, physics calculations(ODE), Rasterization

Mustafa Özturun:

Game sound, Window Handling, Game Logic

Ozan Şimşek:

Scripting, Model Loading, Input Handling

Özlem Özcan:

Map generation, Level Design, Character Setup, Game Menu

All the tasks about these modules are assigned in the meetings and a deadline is defined for each task. Every member is responsible to finish his/her task in time. However the

borders of these tasks are not strictly apart from each other. Sometimes the group members work on the same task together.

2.3 TOOLS & INFRASTRUCTURE

We use the MS WINDOWS applications of Visual C++, MS Office and Visio, OpenGL, Glut, the physics engine ODE, Adobe Photoshop, Corel Painter & DRAW and 3D Studio Max to do our project requirements. For software configuration management we will use CVS (Concurrent Version System). CVS is an open source version control system. The main reason for using CVS is that it helps the group members work on the same project at the same time. It keeps a central repository that includes source code, past versions of the system, and logs document changes to the system. Team members may check out a directory containing source code, check in new versions of files, and view logs to learn the modification history of a particular file or directory.

We do two meetings per week and discuss the progress of the team members. After the discussion we integrate the mature to the baseline. Those members who have completed the task are assigned a new task and the mechanism works like this.

3- THE CM PROCESS

3.1 IDENTIFICATION

Configuration identification makes sure that all project management disciplines have a common reference point through standard numbering. Configuration identification describes the selection of technical documents that describe the functional characteristics of a CI. It refers to the document or set of documents that define the approved configuration, or partial configuration of a CI, and the establishment of configuration baselines for the CIs.

Plans, Specifications, and Procedures approved by the team members will be under configuration control and marked or numbered using a project approved standardized numbering scheme.

A baseline is designated when the CI's configuration identification (document or set of documents) is deemed to be complete and correct. Against these baselines, changes, and deviations may be proposed, evaluated and dispositioned. Baselines and approved changes from those baselines comprise the current configuration identification.

Until now, fundamental structure of the project has been finished. We have started to develop code recently and members are focusing on important functional specifications and level design. Having a modular design gives important flexibility to work independently for the members. Furthermore, they can handle small problems independently instead of communicating with the others. This will speed up the project. However, any important changes in modules require discussion between members in audits.

3.2 MANAGEMENT AND CONTROL

All of our configuration items will be kept in repository and we will have local copies that will be backed up frequently. During the initial stage of our development, we may not effectively control or record the changes to the parts of the product. After the product reaches the minimum quality level we need to keep track of changes to these parts. The automatic version control of CVS will be used after this point.

When product reaches a desired quality, which may become a baseline, change control mechanism will start to apply for that part. In the initial stage of the development only the libraries we use (lib3ds, ODE) will be baselined.

CVS automatically does a version control on files. When somebody “checks out”, modifies a file and then “checks in”, the version of the file increases automatically. This prevents from checking in old version of files.

When a part of code is considered to reach the desired quality by all of the team members, it will be established as the baseline. To make a change to a baseline, it is

required to be analyzed by at least 2 members for its impact on the baseline and other parts of the product. They will decide on whether to make that change or not. In case of a modification on the baselined product, the new version will be established as the new baseline.

3.3 CONFIGURATION STATUS ACCOUNTING

The communication between the team members is generally through e-mails and group meetings. A yahoo-group account exists for communication. The announcement of minor changes and the requests for minor changes that are out of responsibility will be through e-mails that are sent to each member of the team. It is the team member's responsibility to announce the changes that he/she does.

Other than the e-mails, all group members write comments inside the source code of the files whenever they make changes.

3.4 AUDITING

Configuration auditing is the way to ensure that both the technical and administrative integrity of the product are being met throughout the project development life cycle.

Adopting a democratic-decentralized team structure there won't be a CM manager. Also, there will be no unique procedure in audits. Meetings will be held twice a week. In each meeting, members will explain each of important changes, problems in their work.

During audits, team members will explain important differences between the baseline configuration and current configuration. These audits provide verification that each item in the detailed design report confirms with corresponding item in the current configuration. During audits, team will try to answer questions of members and decisions will be taken at the end of each audit.

The configuration audit will be held after members finish an important version of the product. The purpose of this audit is to verify that system performs all the functions described in the system documentation (Detailed Design Report and Software

Requirement Specification Report). Since the modules are defined, every team member will test his/her modules independently under development and after finishing modules.

Also, there will be test plans, descriptions, test reports, and test results will be compared with the expected ones. The results are checked for their completeness and accuracy. Deficiencies will be noted in the audit report, and coder(s) of the module are expected to fix the deficiency.

4. PROJECT SCHEDULE and MILESTONES

- **1 Mar - 7 Mar** Implementation of classes related to windows class
- **8 Mar - 15 Mar** Implementation of classes related to graphics class
- **15 Mar - 22 Mar** Implementation of classes related to environment class
- **22 Mar - 30 Mar** Implementation of classes related to physics class
- **31 Mar - 8 Apr** Implementation of classes related to utility class
- **9 Apr - 16 Apr** Implementation of classes related to sound class
- **17 Apr - 24 Apr** Implementation of classes related to form class
- **25 Apr - 1 May** Implementation of classes related to menu class
- **2 May - 16 May** Black-box and white-box testing
- **17 - 31 May** Debugging applications
- **1 June - 7 June** Documentation
- **8 June - 11 June** Final Release

Above is the latest schedule of `1956` project by MIR Company. Estimated completion dates of each module and milestones are given. All the CM activities will be performed after implementing each module and at milestones. Furthermore, additional CM activities will be performed throughout the testing phase.

5. PROJECT RESOURCES

MIR Company has four members working on `1956` project. In order to perform CM activities for this project, the team members need to use a tool to take care of changes and development of the code and documentation. The main tool for CM activities is CVS

provided by METU Computer Engineering Department. It is the mostly used open-source version control system. It is useful for large and distributed teams.

Newsgroup and e-mail are used for communication in between group members. In addition, all documents, resources and living schedule are hold in our web page of the project.

6. PLAN OPTIMIZATION

As being a small team each member should take care of maintaining configuration management plan. Moreover, the updates will be applied each week since we have weekly progresses. The changes are committed to the CVS Server so that all personnel will be aware of the change and proper versioning, check-in and check-out activities are done safely. If there is a need, plan will be updated accordingly.