Media AI
Adaptive. Intelligent.

Project Plan

Nick Burwell
CS 130 – Software Development
Thursday, December 16, 2004
# Table of Contents

1. Introduction..............................................................................................................1
2. Software Development Process Model....................................................................1
3. Schedule...................................................................................................................1
   3.1 Overview ............................................................................................................1
   3.2 Milestones ..........................................................................................................1
   3.3 Time & persons for each module .......................................................................2
   3.4 Week by week detail ...........................................................................................2
4. Personnel..................................................................................................................4
1. Introduction

This document will contain the project plan for the development of Media AI. It contains a detailed schedule and milestones of the development of the project. It also describes the software development process model.

2. Software Development Process Model

The software development process model that will be used is a modified waterfall model. It is the traditional waterfall process with an additional feedback feature. This means that while each phase flows downward to the next phase, at any point a previous phase can be revisited, revised and changes made to everything after that point if necessarily. This method seems the most reasonable since the requirements and architectural design phases have already been completed. The remaining phases will be software design, implementation & unit testing, integration & testing, then formal testing and maintenance. These will be done as a team, and of course the requirements and/or architectural design may change as the project develops.

3. Schedule

3.1 Overview

- Week 1 – Training
- Week 2 – Design and interfaces
- Week 3 – Implementation
- Week 4 – Implementation
- Week 5 – Implementation
- Week 6 – Implementation
- Week 7 – Implementation
- Week 8 – Unit testing, then integration of all the modules
- Week 9 – Testing of the system, finding and fixing bugs
- Week 10 – Final testing, creating user manual, readmes; making it production ready

3.2 Milestones

- User module completed by end of week 3
- Music player interface integrated with engine completed by end of week 4
- The core module completed by end of week 5
- Music server set up, installed and working by end of week 5
- Playlist generator completed by end of week 6
- Week 7 will be used for any incomplete modules
- Integration complete by end of week 8
- Initial testing, and debugging finished by end of week 9
3.3 Time & persons for each module

Manager will be overseeing each module’s design and implementation. The numbers below are not including the manager as a programmer, but he will assist where needed in the actual implementation of the system.

- Training & research – 4 programmers and manager (1 week)
- Design & interfaces – 4 programmers (1 week)
- User module – 2 programmers (1 week)
- Music player – 2 programmers (2 weeks)
- The core – 2 programmers (2 weeks)
- Music server – 2 programmers (1 week)
- Playlist generator – 2 programmers (1 week) + 4 programmers (1 week)
- Unit testing & Integration – 4 programmers (1 week)
- Testing of integrated parts, bug fixes – 4 programmers (1 week)
- Final testing & documentation – 4 programmers (1 week)

3.4 Week by week detail

- Week 1 – Training
  o See training plan document for more details
- Week 2 – Design and create class interfaces
  o Entire team will work together on designing in detail each module
  o The manager of the project will oversee the design
  o Modules will be broken up into classes, the interfaces created
  o Each module will be done individually, one per day
  o A rapid prototype will be attempted, each day another part of it will be created and integrated into the previous day’s work
- Week 3 – Implementation
  o User module will be worked on by team A
    ▪ Database tables for storing user info created
    ▪ Ability to read and write to user info
    ▪ Secure login, session tracking, timeouts will be implemented
  o Music player will be started by team B
    ▪ Existing engine should be functioning, with primitive user interface by end of week 3
- Week 4 – Implementation
  o Music player will be continued and finished by team B
    ▪ User interface completed
    ▪ Installer implemented and tested
    ▪ Be able to get info from a server, while installed on a client machine
  o Core module begun by team A
    ▪ Create database tables for keeping track of what users are listening to what songs
    ▪ Take user info, be able to remotely log out a user
• Week 5 – Implementation
  o Core module continued by team A
    ▪ Work on connecting to music player that is now implemented
    ▪ Work on decision strategy for resolving conflicts between users’ playlists
  o Music server started by team B
    ▪ Database holding all the songs and song info created
    ▪ Few song files inserted into database
    ▪ Get music to stream to a client computer
  o Begin playlist generator by team A
    ▪ Sketch out more designs for ability to be adaptable
    ▪ Create database tables for keeping track of what users rate what sounds, and what the rating is
    ▪ Make table indices, and get some good search and other algorithms, to be used when creating adaptable playlists

• Week 6 – Implementation
  o Team B joins team A on playlist generator
    ▪ Team A brings team B up to speed on progress
    ▪ Before moving on, discuss current design and setup
    ▪ Be able to rate the songs and also receive implicit feedback through whether a song was finished or skipped
    ▪ Implement the adaptive decision processes

• Week 7 - Implementation
  o Any modules not completely finished, finish during this week
  o Review design and requirements documents, verify that everything is taken care of
    ▪ Update documents and/or implementation as needed

• Week 8 – Unit testing & integration
  o Test each unit on its own (although this should also be happening as development is in progress each week); verify it is fully working
  o Integrate all modules; team A and team B working together
  o Begin testing on the integrated system

• Week 9 – Testing of the system & bug fixes
  o Test the integrated system
  o Look for bugs, report them to the entire team
  o Fix bugs, continue testing

• Week 10 – Final testing & documentation
  o Do final testing; ideally will be bug free at this point
  o Install several client music players, on several computers; log into the music server and perform testing with multiple client music players at once
  o Write final documentation, including an installation manual, user guide, and readme files both for users and developers
4. Personnel

For the initial two weeks, all four employees and the manager will be together and training then working on the design as a whole. From week 3 through week 7, the group will be broken into two teams of two programmers each, both which will be overseen by the manager. If there are two programmers with more experience in artificial intelligence, data structures, and algorithms, they will be assigned to team A. Team B will require more knowledge in databases, servers, interaction between clients and servers, as well as user interface design. There will be flexibility as the project progresses as to who is assigned to what teams, if personalities clash or certain programmers have varying skill levels. In weeks 8 through 10, the teams will combine again and work together to integrate and test the entire system. Since there are only 4 programmers and the manager, the communication and efficiency problems will be minimal. The manager will spend the middle weeks of implementation supervising the progress, making sure things are on schedule, helping code when needed, and making sure everything is being documented and backed up. Whenever there is a conflict or question among the teams, the manager will make sure things are resolved and both team A and team B are on the same page.