TO: Jeff Salvage, Toban Emanuel

FROM: Kevin Galloway, Nicholas Goede, James Hess, Brian Lawrence, William Mongan

Date: January 7, 2005

RE: **NLIVE:** A proposal to design and implement a non-linear video editor

The following is a proposal to design and develop a non-linear video editor. We outline the need, the functional description, the environment and implementation details, and the corresponding business model identifying our schedule and resources required to successfully create this application.

Non-linear video editing is one of the most common and flexible forms of video editing available today. This process can easily edit and re-order videos in whole or part without the generational quality loss inherent in traditional analog transfers done by video editors. The process was once confined to video editing shops, but is now available for the average home user without the need to acquire prohibitively expensive video editing equipment. This is a result of an explosive increase in processing power of the home computer.

Few open source editors

Commercial programs for non-linear video editing currently range from professional, high end solutions to those for the amateur enthusiast. However, due to their relatively high costs across the board, there exists no viable option for the casual video maker. Furthermore, there are few open source alternatives; of these, none is available for the Windows environment.

Functionality

NLIVE executes under the Windows XP operating system with DirectX 9.0b, utilizing a graphical user interface. The user manipulates one or more video files for which a video codec is installed for input. The user may cut and splice together parts of these input files as well as to add transitions and other effects to the video. The user will be able to add "title cards" and other still images to the video file, and to remove or replace the audio track of the videos being edited. Once video editing is complete, the user can generate a new video file using one of several possible compression routines and file formats.

Environment and Implementation

The primary programming environment is the Microsoft .NET platform, specifically Visual C#, utilizing the Direct X 9.0 SDK throughout the development process.

Subversion, an open source version control tool provided by Tigris.org, handles version control. Subversion contains many features similar to CVS along with additional features not provided by CVS.

NUnit, a .NET unit-testing framework, handles unit testing. NUnit was ported from JUnit, a popular regression-testing framework for the Java programming language. NUnit utilizes Clover.NET to help analyze the unit testing and assure complete testing coverage. Clover.NET is available under an open source license with the understanding that all of our code remains open source. We may also consider Nester, a test tester for C# NUnit tests, during the unit testing process.

During the design stage of this project, each requirement is verified through prototyping. This assures the feasibility of every requirement outlined in the requirements specification document before the finalization of this document.

Before the development stage begins, prototypes act as a proof of concept model to assure that each requirement can be implemented within the constraints of the time schedule and environment.

Business Need / Business Model

We plan to utilize the waterfall software process model for our development. Concurrent to the later stages of design, implementation, and testing, there is a prototype phase. This is necessary both to act as a test bed for ideas and to flush out implementation kinks in an unfamiliar technology.

Due to the licenses we have agreed to, we are indelible for financial gain. Many of the development support suites we are using demand that our code remain open source. Our team's driving motivation is to create usable software that benefits the whole group, and to receive college credit for our efforts. After completion of the integration and testing phases, our intentions are to release our executable and source code to the public to allow for further development and general use. Little maintenance is offered on this software because we will have graduated and will most likely not be available to provide end user support. We hope to find a software group after our graduation to shepherd the project at that time.

We do not anticipate any budgetary demand or need. With the exception of Microsoft Visual Studio .NET, which is provided to us via a site license through Drexel, all of our supporting software suites are free to use. We do not foresee any need for software or hardware purchases.

Our deliverable schedule coincides with the required date of completion of each document and code component. We have a technically sound group and estimate a high probability of success in the required course objectives as well in our own group goals.