

Software Project Plan

Introduction

Project Scope

GameForge is a graphical tool used to aid in the design and creation of video games. A user with limited Microsoft DirectX and/or Visual C++ programming knowledge will be able to construct a basic 2D-arcade game. The idea is to limit the amount of actual code written by the user. It will also assist experienced programmers in generating the Microsoft DirectX and Microsoft Windows9x overhead necessary for basic game construction, allowing them to concentrate on more detailed game design issues and implementation.

Critique: Bounding is a critical element of the project scope and the project plan. It would be a good idea to try to "bound" all the general statement of scope noted here. For example, "a basic 2D arcade game" is open to very broad interpretation. What is basic to one reader might be unacceptable to another.

The software will consist of a number of inputs, graphically assisting the user in creating on-screen objects including the following:

- User Created Objects (player character, creatures, static objects)
 - Bitmaps (with animation)
 - Collision Detection Areas
 - Movement Routines
 - Additional Object Attributes
- Backgrounds
- Input Device Setup
- Sound Events

The software will also consist of a number of graphical processing functionalities including the following:

- Defining/Editing Objects (including characteristics)
- Object Positioning
- Opening/Closing/Saving Game Project Files
- Exporting Game Projects to compilable C++ Files

Outputs include:

- User Created Sprite Objects
- Bitmaps
- Microsoft VC++ (with DirectX code) Files
- Game Project Files
- Text Files (containing sprite attributes)

Timeline Chart

K = Ken, J = Jon, M = Matt, B = Bill
 < = See next chart

	2-Jan	9-Jan	16-Jan	22-Jan	30-Jan	6-Feb	13-Feb
	s m t w r f s	s m t w r f s	s m t w r f s	s m t w r f s	s m t w r f s	s m t w r f s	s m t w r f s
Requirements Spec. & Design							
Requirements specification	K, M, J, B						
Engine architecture design		K					
Interface layout and design			M				
Interface Task Breakdown							
Level Editor			M				>
New Project wizard				M			>
New Sprite wizard				M			>
Database (DB) construction			M				>
DB communication w/ interface			M				>
Exporting game files ability							
Exporting .cpp files ability							
Engine Task Breakdown							
Object Handler			K				>
Sprite Handler			K				>
Image Handler (DDraw)			K				>
Sound Handler (DSound)				J			>
Input Handler (DInput)				K			>
Text Handler				K			>
Logic Handler				K			>
Attribute Handling					K		>
Unit Pathing					K		>
File I/O Parser				K			>
Help Task Breakdown							
Interface Help							
Engine Help							
FAQ							
Game building tutorials							
Manual							
Testing Task Breakdown							
Unit testing				B			>
Integration testing							
Validation testing							
Performance testing							
In-house Alpha testing							
Outside beta testing							
Documentation							
System Requirements Specification		B					
Software Requirements Specification		K, M					
Software Quality Assurance Plan						K	>
Risk Management Plan						B, M	>
Software Configuration Mgmt.						J, K	>
Project Plan						J	>
Design Document							
Test Plan							

More Timeline Chart

K = Ken, J = Jon, M = Matt, B = Bill
 < = See previous chart

