7. Interactive Media and Game Development process
Game Types

- Arcade Games
- Puzzle Games
- Role Playing Games
- Strategy Games
- Adventure Games
- First-Person Shooters
- Third-Person Action
- Sports Games
- Racing Games
- Simulators
- Party Games
- Educational Games
Game Studios – Vertical Structure

- Developers
- Publishers
- (Distributors)
- Retailers

- Much like a mini-Hollywood
Developers

- **Design and implement games**
  - Including: *programming*, *art*, *sound effects*, and *music*
  - Historically, small groups
  - Analogous to book authors

- **Structure varies**
  - May exist as part of a Publisher
  - May be “full-service” developers or may outsource some
    - Motion Capture (to replicate realistic movement)
    - Art and Animation (can be done by art house/studio)

- Many started on PC games (console development harder to break into)

- Typically work for royalties & funded by advances
  - Do not have the capital, distribution channels, or marketing resources to publish their games
  - Often seen that developers don’t get equitable share of profits
  - Can be unstable
Publishers

• *Fund development of games*
  – Including: manufacturing, marketing/PR, distribution, and customer support

• Publishers assume most of the risk, but they also take most of the profits

• Relationship to developers
  – Star Developers can often bully Publishers, because publishers are desperate for content
  – Most Developers are at the mercy of the almighty Publisher Originally grew out of developers

• Massive consolidation in recent years

• Most also develop games in-house
Retailers

- Sell software
- Started with mail-order and computer specialty stores
- Shift in 80’s to game specialty stores, especially chains (Today 25%)
  - EB Games, GameStop
- Shift in 90’s to mass market retailers (Today 70%) (ask)
  - Target, WalMart, Best Buy
- Retailers generally earn 30% margin on a $50 game
- Electronic download of games via Internet still in infancy
  - Big but not huge (Today 5%)
Game Development Process (1/5)

• **Inspiration**
  – getting the global idea of the game
  – duration: 1 month (for a professional game)
  – people: lead designer
  – result: treatment document, decision to continue

• **Conceptualization**
  – preparing the "complete" design of the game
  – duration: 3 months
  – people: designer + prototype programmers/artists
  – result: complete design document
  – (continued next slide)
Game Development Process (2/5)

• Prototypes
  – Build prototypes as proof of concept
    • Can take 2-3 months (or more)
    • Typically done a few months in
    – In particular, use to test game play
    – Throw prototype away afterwards
      • Don’t expect it to evolve into game!
    – Pitch to Publisher
Game Development Process (3/5)

• **Blueprint**
  – separate the project into different tiers
  – duration: 2 months
  – people: lead designer, software planner
  – result: several mini-specifications

• **Architecture**
  – creating a technical design that specifies tools and technology used
  – duration: 2 months
  – people: project leader, software planner, lead architect
  – result: full technical specification
Game Development Process (4/5)

• **Tool building**
  – create a number of (preferably reusable) tools, like 3D graphics engine, level builder, or unit builder
  – duration: 4 months
  – people: project leader and 4 (tool) programmers
  – result: set of functionally tools (maybe not yet feature complete)

• **Assembly**
  – create the game based on the design document using the tools; update design document and tools as required (consulting the lead designer)
  – duration: 12 months
  – people: project leader, 4 programmers, 4 artists
  – result: the complete game software and toolset
Game Development Process (5/5)

• **Level design**  
  – create the levels for the game  
  – duration: 4 months  
  – people: project leader, 3 level designers  
  – result: finished game with all levels, in-game tutorials, manuals

• **Review**  
  – testing the code, the gameplay, and the levels  
  – duration: 3 months (partially overlapping level design)  
  – people: 4 testers  
  – result: the gold master
Managing IM&G Development with Alienbrain
Alienbrain

http://www.softimage.com/products/alienbrain/
What is Alienbrain

• Industry standard for file management
  – in professional media and entertainment projects.

• Systems for creative teams
  + DAM: Digital Asset Management
  + SCM: Software Configuration Management

• Tools for any kind of file for import, version, manage and share
Alienbrain Features

• Secure File Management and Version Control
  – Comprehensive Version History, Rollback, Powerful Search Tools

• Visual Workflows
  – Intuitive User Interface, Previews and Thumbnails, Local File State Icons, Integrations for Leading Art Tools

• Collaborative Environment
  – Image Annotations, Integrated Messaging, Reporting

• Software Configuration Management
  – Labels, Change Sets, Parallel Development and Branching

• Architecture and Administration

• Customization and APIs
  – Custom Metadata, Triggers & Events, Command Line Tool
Client / Server architecture

Alienbrain Studio integrates all team members
Alienbrain Server

• maintain the asset files and up-to-date information
  – file sizes/version/attributes
  – raw file data
  – optimized object-oriented database.

• controls and co-ordinates access
  – Security/access collisions
  – download any version
  – modification and upload new versions
Alienbrain Client

• Asset management command center
  – browse the project databases
  – import new files or view, lock and edit.
  – display thumbnail images and preview
  – workflow functions.

• Different client applications types
  – Essentials for Artists for creative users.
    • 3-D authoring tool integrations
  – Essentials for Programmers for programmers.
    • source code file merging.
  – Advanced the complete package.
    • Programmers and Artists, workflow management functionality.
  – Alienbrain Reader
    • read only access to project data.
Functions

- file management
- version control
- change management
- configuration management
- workflow
- access control
- archiving
- visual working
File Management & Sharing

- import
- browse and view
- move, rename and delete
- automated operations by scripts
- edit files
- Check out / check in / multiple check-out
Version Control

• version history/ get version
• Rollback
• show differences between versions
• Text comparison and merge tool
  – Araxis Merge Professional
Change Management

- change sets as shielded containers
- default change set/create change sets
- (active change set) delete, rename, check out, modify and check in, until submit
Configuration Management

- maintain multiple configurations without duplicating its content
- root branch
- branching manager
- branch selection drop-down list
- integrate changes wizard
Workflow

• An asset-based workflow based on a range of configurable workflow states (work in progress, awaiting modification, awaiting approval, approved, approved-and-locked)
  – assign assets to a user,
  – change the workflow status of an asset
  – set a due date
  – Track and review/approve the changes
Access Control

• Access rights.
• simplified set of role definitions (default permissions)
  – authors,
  – Contributors
  – Editors
  – reviewers
Archiving

• archiving system
  – Offline/online
  – multiple language or platform variants
  – images and 3-D models.
Alienbrain evaluation

• download it from

• a fully functional version of Alienbrain 8.1
  – for an unlimited time
  – with a maximum of two simultaneous client connections and five projects.
Best practices

• Alienbrain administration
• Customizations
• process management
• workflow optimization
Version control
Version control for programmer

- CVS
- Subversion (SVN)

- Git
  - was initially created by Linus Torvalds for Linux kernel development
Subversion (SVN)

- Since 2000

- a free version control system which operated much like CVS

- used by SourceForge
Subversion filesystem can be described as a three dimensional filesystem.
Subversion properties

- name=value pairs of text
- used in two different places in the Subversion filesystem
  - filesystem entries, i.e., files and directories
  - revisions themselves
Subversion properties

- filesystem entries
  - svn:executable
  - svn:mime-type
  - svn:ignore
  - svn:keywords
  - svn:eol-style
  - svn:externals
  - svn:needs-lock
  - svn:special

- revisions themselves
  - svn:date
  - svn:author
  - svn:log
Branching and tagging
Software that uses Subversion

- TortoiseSVN, a Windows shell (i.e. Explorer) extension
- Xcode is Apple's Mac OS X IDE
- Microsoft Visual Studio
  - AnkhSVN is a Visual Studio .NET addin
  - VisualSVN is simple and reliable Subversion integration for Visual Studio 2003 and 2005
• **windows**平台上的SVN客户端软件
• 易于控制

• 教程
  • [https://www.se.auckland.ac.nz/courses/SOFTENG254/resources/TortoiseSVN.pdf](https://www.se.auckland.ac.nz/courses/SOFTENG254/resources/TortoiseSVN.pdf)
创建版本库
(The Repository)

- 使用命令行工具创建版本库
  - 创建一个名为SVN(例如D:\SVN\)的空文件夹，作为你的所有版本库的根。
  - 在D:\SVN\里创建另一个目录MyNewRepository。
  - 打开命令行窗口(或DOS窗口)，进入D:\SVN\目录，输入

    `svnadmin create --fs-type bdb MyNewRepository`
创建版本库
(The Repository)

- 使用TortoiseSVN创建版本库
- 打开资源管理器
- 创建一个新的文件夹，命名为SVNRepository
- 右键点击新创建的目录，
  - TortoiseSVN → Create repository here ...
访问版本库

- 本地:
  - file:///C:/SVNRepository/

- 网络:
  - file://ServerName/path/to/repos/
Icon Overlays

Resources
ColumnProvider.cpp
ContextMenu.cpp
deregister.registry
Guids.h
IconOverlay.cpp
ItemIDList2.cpp
ItemIDList.cpp
PIDL.h
PreserveChdir.cpp
register.registry
RemoteCacheLink.cpp
Context Menus

- SVN Update
- SVN Commit
- SVN Check for modifications
- TortoiseSVN

- Show log
- Repo-browser
- Revision graph
- Resolved...
- Update to revision...
- Rename...
- Delete
- Revert...
- Clean up
- Get lock...
- Release lock
- Branch/tag...
- Switch...
- Merge...
- Export...
- Relocate...
- Add...
- Create patch...
- Apply patch...
- Properties
- Help
- Settings
- About
Authentication

<http://svn.collab.net:80> TortoiseSVN repository

requests a username and a password

Username:

Password:

☐ save authentication

OK  Cancel
Import Data

Select the command TortoiseSVN → Import...
Check Out
Update data

- TortoiseSVN → update
Add data

• TortoiseSVN → add ..
Commit data

- Conflicts?
See difference

- Text?
- How about image?
Merge

• version ...

• ?