Game Postmortems

John Laird and Sugih Jamin

Based on Gamasutra and Game Developer Magazine articles
Post Mortem
Dungeon Siege

Based on Gamasutra article by Bartosz Kijanka 12/18/2002
Datasheet

- **Publisher:** Microsoft
- **Developer:** Gas Powered Games
- **Number of Full-time developers:** 27 at ship date
- **Number of Contractors:** 5
- **Length of Development:** 3 years, 8 months
- **Release Date:** April 5, 2002
- **Platform:** PC
- **Development software used:** MS Dev C++, 3DS Max with Character Studio, Visual SourceSafe, CodeWright, ICQ, RAID (bug tracking), Photoshop, Excel
- **Development hardware used:** Ranged over course of development from 400-1000MHz CPUs with 128-512MB RAM
- **Notable Technologies:** Bink, Miles, SmartHeap
- **Project Size:** Approximately 800,000 lines of source code for game, editor, and associated tools; 60,000 lines of scripts; 21 million total lines of .GAS configuration files; 8,500 textures, 2,000 animations, 2,600 object and actor meshes, 3,700 terrain meshes
Overview

- Gas Powered Games formed in 1998
  - By Chris Taylor (Total Annihilation)
- Dungeon Siege was their first game
  - Forming a company while doing first game
What Went Right

- Exceptional Art
- Great tools for art development
- Special effects, …
What Went Right

• Extreme Flexibility: small company
  • Make decisions very fast
  • Set up temporary test team over a weekend
• Game engine was well designed
• Data-driven design
  • Configuration files
  • Text Files
  • Scripts
• Scripting systems (Skrit)
• Editor built on top of game “world” layer
  • Careless change in world layer could break editor
What Went Right

• Instant Messaging
  • Originally small enough to communicate verbally – all in same room
  • As got bigger used ICQ instant messaging
What Went Wrong

• Extreme Ambition
  • Ambitious people
  • Feature creep, over-optimism
  • Nobody worked on RPG before

• Example features
  • Lip synching
  • Cooperative networked level editing
  • Dual monitor support
  • Wavelet terrain compression
  • Deformable terrain …

• Originally against formal organization
  • Learned need some as company grew
What Went Wrong

• Aborted Efforts
  • Implemented unnecessary features
  • Animation editor discarded - developer left
    • 1 year effort
    • Replaced with 3DSMax – use 3\textsuperscript{rd} party tools!
  • Switched from OpenGL to Direct3D

• Complex Engine
  • Lots of features but
  • Large
    • Local documentation but no global
    • Changes to single-player break multi-player
  • Hard to maintain

• Slipped Schedule
  • Originally aimed to ship in 2 years
  • Building company, not just game
  • Three of original six left company after 1 year
What Went Wrong

Epic Crunch

• Stayed in crunch mode full time
• Didn’t crunch to make up for lost time but “crunched out of uncertainty”
• “At some point we crunched because we could no longer remember doing anything else”
  • Your passion can become your prison
Post Mortem Tron 2.0
Based on Gamasutra article by Frank Rooke 9/10/2003
Why Tron?

• 20 year-old movie
  • Action based – many game elements

• But inspiring to many
  • “It’s why I’m into computer.”
  • “It’s why I’m into 3D graphics.”
  • “It’s why I’m into gaming.”
Datasheet

- Publisher: Buena Vista Interactive
- Developer: Monolith Productions
  - *No One Lives Forever, Alien vs. Predator, The Matrix Online*
  - *Founded 1995 in Kirkland Washington*
- Number of full-time developers: 21
- Number of part-time: 4-5
- Contractors:
  - Cinematic music scoring, motion capture actors, voice actors
- Length of development: 2 years
- Release date: August 26, 2003
- Target platform: PC – 1-2 GHz machines
- Development Software: Lithtech DEdit/ModelEdit, Microsoft Studio (C++), Photoshop, Maya, Editplus 2
- Notable Technologies: Lithtech Jupiter Development System
- Project size: 2,400 files, 853,300 lines of code
What Went Right

• Publisher Compatibility
  • No micro-management on license
  • Strong International standing
  • Access to original talent
    • Syd Mead
      – Original Tron art
      – New super light cycle
    • Richard Taylor and Steven Lisberger (Tron creator)
      – Reviewed game
    • Bruce Boxleitner (Alan Bradley) and Cindy Morgan
      – Original voices from movie
What Went Right

• Avoiding simple translation of the movie to game
• Identifying iconic elements from film
  • Disc
    • Unique game play and combat
  • Light Cycle
    • Well known
• Glowing backgrounds and artwork
• Techie metaphors
• Bit, Tanks, and Recognizers
What Went Right

• Sharing Code
  • Trailed development of No One Lives Forever 2 by eight months which developed the Jupiter engine

• Evolved Art Direction
  • Started by redoing actual sets of film
  • Art was major asset of game: Glow
    • Colorful Architecture
    • Glowing Streams of Energy
    • Creative Level Design
      – Met or surpassed the movie
      – Alternative to hyper-realistic military games

• Challenging to represent abstract computer concepts
  • Firewall, CPU, RPC, …
What Went Wrong

• Short on initial resources
  • Only 4-5 core developers on preproduction
  • Significant ramp up time on tools and ideas
  • Unusual nature of game environment not appreciated

• Levels unplayable until late in project
  • No working prototype
  • No working cycle racing, disc combat until late in project

• Sharing Code
  • Jupiter was targeted to “realistic” environments
  • Required lots of tweaking of engine
What Went Wrong

• Loose Review Process
  • Reviews were flexible
  • Worry about meeting milestones, but many problems
  • Solved by having regular in depth reviews
    • Weekly or even daily

• Problems with Commercial 3D Software
  • Originally had home-grown game editor
  • Switched to commercial 3D package
    • More flexibility and power
    • Lost ability to move back and forth between building and testing
Datasheet

- **Publisher:** Microsoft
- **Developer:** Ensemble Studios
- **Number of Full-time developers:** 50 employees; 15 programmers
- **Number of Contractors:** 10 quality assurance
- **Length of Development:** 30 months
- **Release Date:** October 31, 2002
- **Platform:** PC
- **Development software:** MS Visual Studio 6. Source Safe, 3DS Max 4.0, Photoshop
- **Notable Technologies:** Bink, Granny
- **Project Size:** 1,500,000 lines of code
- **Gamespot:** 9.2 – superb – editor’s choice
Overview

- Third in “Age of Empires” Series
  - Good understanding of what people like
  - Existing engine
    - Although dated already
  - Already have to do lists from prior projects
  - Fans want something different
    - But others that don’t want any changes…
  - Expectations of continued growth
  - How can they top Age of Kings?

- First “Age” in 3D
What Went Right

• Prototype early

• Lots of iteration – tweak until it is fun

• Example: God Power
  • Originally tied directly to Heroes
    • Tactics devolved into Hero killing
  • Lightning rods?
  • Buy god powers?
  • Final – Heroes separate from God powers
    • God powers global and single use
  • Important to title so spent lots of time trying to get them right
What Went Right

• Prototype often
• Everyone plays at least once/week
• Internal feedback in addition to some external feedback
  • Avoid attempting to please everybody
• Keeps everyone up on design
• Gives everyone pride and ownership
• Find bugs
What Went Right

• Small Meetings
• For first two titles, had entire team involved in design meetings
  • Became unmanageable

• Age of Mythology
  • Lead meetings on management
  • Design meetings restricted to 4-5
  • Sometimes went off site to avoid interruption
  • Announce results to company
What Went Right

• Data-driven tools
  • Took lots of time at start of project
  • Designers could implement content without programmers

• Focus on scenarios (campaigns)
  • Prior “Age” titles did not emphasize scenarios
  • Made a big feature in Mythology
    • Custom animations and cinematics
What Went Wrong

• Design drove too much
  • Sometimes programmer could have done things faster than tool + designer
  • Specs were too detailed
    • “When you click this button, it should appear depressed until the user releases the mouse button, at which time it should revert to looking un-depressed; clicking the button in this manner should cause a sound to occur, the sound should be kind of like a twig snapping …,”
  • New employees didn’t feel empowered

• Scenario scriptwriting problems
  • Big script required for campaign
  • Requiring lots of dialog, etc.
  • No experience
  • Lots of revision
What Went Wrong

• Consensus is hard in big groups
  • Consensus worked well with small company
  • Stalemated when company got bigger
  • Empowered Design team to make tough decisions
    • Still would lead to lots of email discussions

• On sequels and expansions: How different is “different”??
  • How much change to the game play?
  • Arguments between changes and staying with the tried and true throughout development
  • In future plan to have crisper definition of difference from beginning
What Went Wrong

• Unfinished Tools
• Developed lots of tools
• Many unfinished
• Developers left waiting for features
  • Hacked in content while waiting
  • Had to unhack it when tool became available