What Does It Take to Make a Successful Persistent Online World

Raph Koster
Rich Vogel

GAMEDevelopers CONFERENCE 2001

Introductions & Overview

Who are these people, & what games have they worked on?

Raph Koster

Creative Director, Verant Austin

Lead Designer: Ultima Online & Ultima Online: Second Age

Creative Director: Star Wars Galaxies

www.legendmud.org/raph/gaming/

Rich Vogel

Executive Producer: Verant Austin

Producer: Meridian 59 and Ultima Online

Executive Producer: Star Wars Galaxies

Introductions & Overview

Why is the development process for large scale persistent online worlds different from single player/multiplayer games?

- The content in these games have to last from 6 to 12 months for the average player
- The average life-span of these games is around 5 years
- It is not fire and forget software. The software must be scalable and maintainable for years.
- Client/Server Architecture
- Supporting thousands of players instead of a few
- Environments are huge.
- It is a service not a product

Introductions & Overview (RK)

What special challenges do designers of large-scale gaming worlds face?

- Creating systems and programs that scale gracefully (e.g. player profiles, achievement ladders, small group infrastructure)
- Accommodating player diversity (playstyles, expectations, interests)
- Managing expectations when bringing an existing brand into Cyberspace
- Creating an online society – whether you intend to or not
- Managing player confusion and intra-player conflicts
- Building a game that offers both deep and shallow gameplay

Introductions & Overview

What special challenges do programmers of large-scale gaming worlds face?

- Creating systems and programs that scale gracefully (e.g. ease of adding new content, and infrastructure)
- Handling flash crowds (large amount of players in an area)
- Syncing clients and server over the internet (dead reckoning, smoothing)
- Security
- Load balancing (Server)
- Backend support (Tools…Tools…Tools)
- Database management (managing character and world data)

Introductions & Overview(RK)

What special challenges do artists of large-scale gaming worlds face?

- Creating assets that can easily be modifiable by the player
- Developing character customization schemes
- Developing assets that have small footprints
- Developing huge environments that don't slow the client down
- Developing scalable art assets (LOD)
- Developing dynamic and static art
- Developing scalable and customizable UI
- Database management (managing huge amounts of art)

Introductions & Overview

What is the Development Process?

- Prototyping - Reaching a production pipeline
- Production Pipeline - Asset creation
- Live Development
- Test and Launch

Prototyping- Reaching a Production Pipeline

Overview

- Develop Organizational Chart (hire leads)
- Develop Vision Document
- Develop Schedule Process (Milestone Layout)
- Develop Detail Design Document
- Prototype concept
- Develop Production Tools
- Develop High Level Technical Overview Document

Prototyping- Reaching a Production Pipeline

Develop Organization Chart

- Hire core team
  - –Executive Producer
  - –Producer
  - –Associate Producer
  - –Art Director
    - - World Lead
    - - Character Lead
  - –Lead Programmer
    - - Server Lead
    - - Client Lead
  - –Lead Designer
    - - Systems Lead
    - - World Lead
- Scale your hiring

Prototyping- Reaching a Production Pipeline(RK)

Develop Vision Doc

- Provides mission statement
- Defines game look and feel
- Defines minimum feature set
- Defines leadership roles
- States the goals of the game
- Defines system requirements

Prototyping- Reaching a Production Pipeline

Develop Schedule Process

- Develop deliverables doc
- Develop milestone task list
- Define milestone 1 (DDR)
- Develop milestone schedule after DDR
- Develop project WAG
- Define milestone length (6 weeks)
- Define detail milestones (2 ahead)
- Revaluate schedule after production has been reached
- Always front-load your highest risks

Prototyping- Reaching a Production Pipeline(RK)

Develop Detail Design Doc

- Outline of systems in the game
- High level spec for each System
- Include goals and priorities for each System
- Include examples of how each system works
- Provide player experiences (1st hour, 1st session, 1st week)
- Provide a vision for the entire product (including after Launch)
- Provide Milestone Schedule and WAG

Prototyping- Reaching a Production Pipeline

Prototype Concepts (RK)

- Build a small prototype of the environment
- Play with UI concepts
- Get basic walk and talk working
- Play with visibility, camera, and movement
- Use this push an iterative development process
- Always prototype systems that are not as defined before implementation

Prototyping- Reaching a Production Pipeline

Develop Production Tools

- Take time to build good tools (Team Effort)
- Develop art production tools
  - Viewers (character/creature, textures…)
  - Conversion tools (plug-ins)
- Develop world building tools
- Make them scalable
- Develop data manipulation Tools

Prototyping- Reaching a Production Pipeline

Develop High Level Technical Overview Doc

- Provide a high level (30,000 feet) overview of all core systems
- How core systems interact together
- Coding standards
- Scalability plan
- Integration plan (login servers, patching, etc)

Production Pipeline - Asset Creation

Overview

- Asset Creation/Revision Loop
- Game System Creation
- Software Development Environment (Client/Server/Engine)

Production Pipeline - Asset Creation(RK)

Asset Creation/Revision Loop (Art)

- Develop detailed Assets list (database)
- Alien Brain or equivalent source control system
- Art Rules Doc
- Production Process
  - define roles
  - define methods
  - Prototype
- Communication between groups and within groups

Production Pipeline - Asset Creation(RK)

Game System Creation

- Develop detail system spec
- Develop scripting guideline
- Develop scripting environment
  - Source control environment
  - Editing environment
  - Debugging environment
- Develop God Client

Production Pipeline - Asset Creation

Software Development Environment

- Modular design (Separate Engine from Game Code)
- Develop server model
  - Development Servers
  - Test Servers
  - Branching of code
- Develop client model
- Develop network layer
- Source control system
- Editing, debugging, profiling (Tools)
- Iterative Development as soon as possible (publishing/patching system)
  - Beg, borrow, buy technology
  - Don’t get into the “not invented here syndrome”

Live Development

Overview

- Forming a community around your game
- Forming a live team
- Deployment of the live team
- Integration - publishing process (code and assets)

Live Development(RK)

Forming A Community Around Your Game

- Hire a community manager early
- Develop an website for the game
- Control information flow by making your game website “the” place for all information about your game
- Run message boards and hire moderators to police them
- Provide a link (Community to game developer)

Live Development

Forming A Live Team

- Hire a Live Team before testing begins
- Why another team
- Team makeup
- Integration into an existing team

Live Development

Deployment of The Live Team

- When product is launched
- Working with existing development team (Transition Period)
- Responsible for service

Live Development

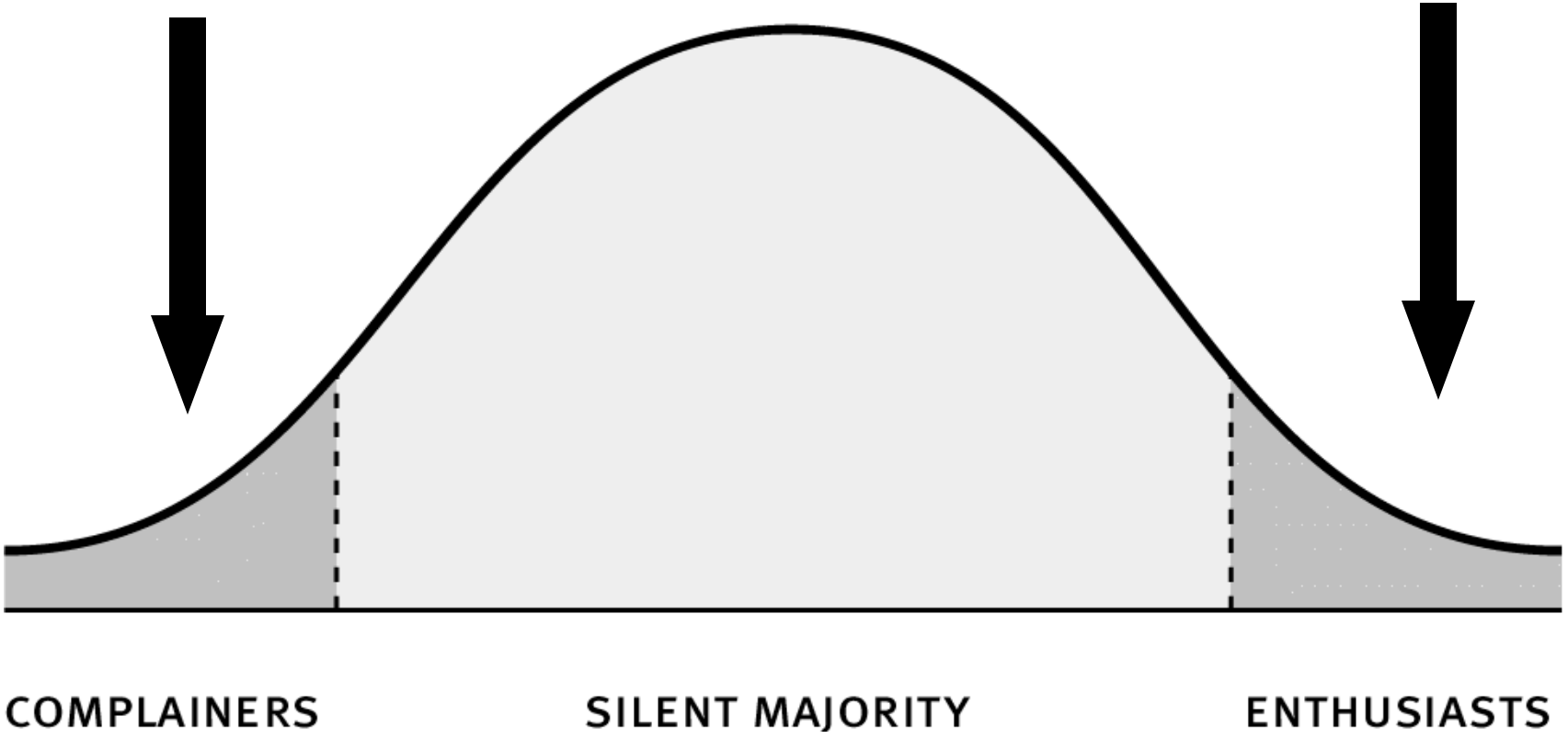
Integration

- Publishing Process
  - Publishing Plan (What is being changed)
  - Test Plan
  - Publish code to test centers
  - Test Code
  - Patch/Publish Code (Operations Group)
- Feedback Loop (Players/CS to Development Team)

Live Development(RK)

Forming A Community Around Your Game

Managing your community



Testing and Launch

Overview

- Develop high level test plan
- Testing Process
- Integration - publishing process (code and assets)

Testing and Launch

Develop High Level Test Plan

- Define testing cycles
- Define player testing rollout plan
- Integration of bug tracking system
  - Player entered bugs
  - Tester entered bugs

Testing and Launch

Testing Process

- Time needed to test these games (6 months)
- Testing cycles
  - –Closed Testing
    - 1st playable (Testing Department/Team)
    - Alpha (100) – 4 weeks
    - Beta 1 (500 – 1,000 people) - 6 weeks
    - Beta 2 (3,000 – 10,000 people) - 6 weeks
    - Beta 3 (10,000 – 25,000 people) – 6 weeks
  - –Open Testing
    - Beta 4 (25,000 – 50,000 people) – close a week before product launch
- Need bug tracking mechanism for player reported bugs
- Need a community test manager
- Need a CS support staff operational at launch of Beta 1
- You are really running a live service at this point

Testing and Launch

Integration – Publishing Process

- Fully operational login process with account authentication
- Fully operational patching system
- Fully operational CS support system
- Fully operational bug tracking system (player/test)
- Fully operational test centers
- Fully operational publishing process

Audience Q&A