IMGD 1001 - The Game Development Process: Game Development Timeline

by
Robert W. Lindeman (gogo@wpi.edu)
Kent Quirk (kent_quirk@cognitoy.com)
(with lots of input from Mark Claypool!)

Outline

- □ Game Timeline (next)
- □ Team Sizes

Game Development Timeline (1 of 5)

- □ Inspiration
  - getting the global idea of the game
  - duration: 1 month (for a professional game)
  - people: lead designer, team discussion
  - 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)

Based on notes from Mark Overmars

Concept

- Define game concept
- Define core game features
- Find/Assign developer
- Estimate budget & Due date

Based on notes from Neal Robison, ATI

Concept: Van Helsing (1 of 4)

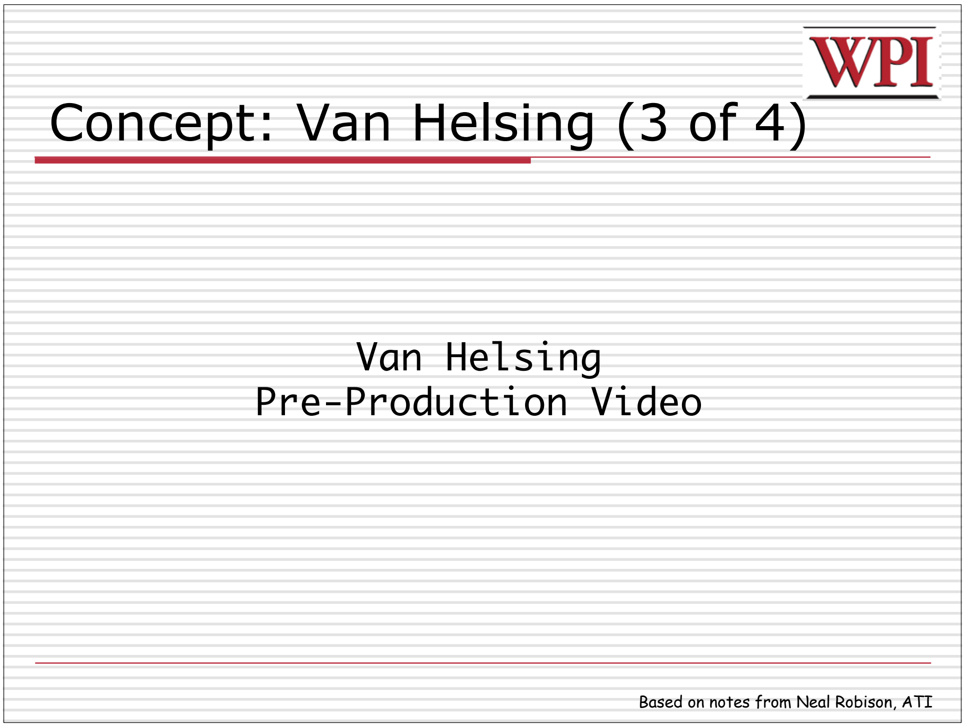
Gameplay:
Still firing after being hit

Based on notes from Neal Robison, ATI

Concept: Van Helsing (2 of 4)

Van Helsing Concepts 10/16/02
confidential
Gameplay: Finishing Move

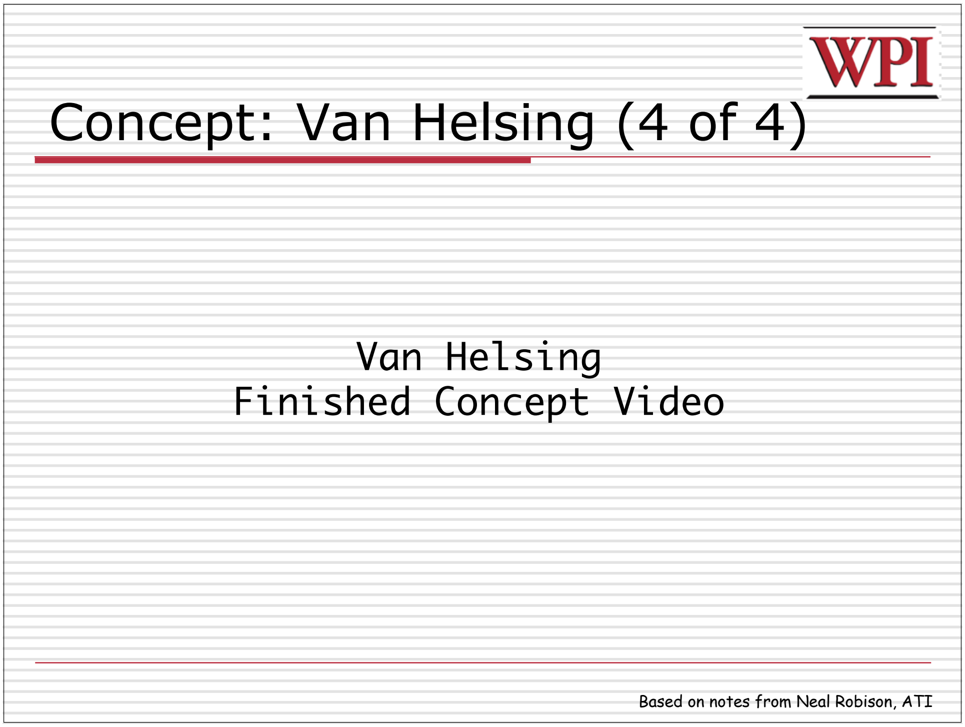
Based on notes from Neal Robison, ATI



Concept: Van Helsing (3 of 4)

Van Helsing
Pre-Production Video

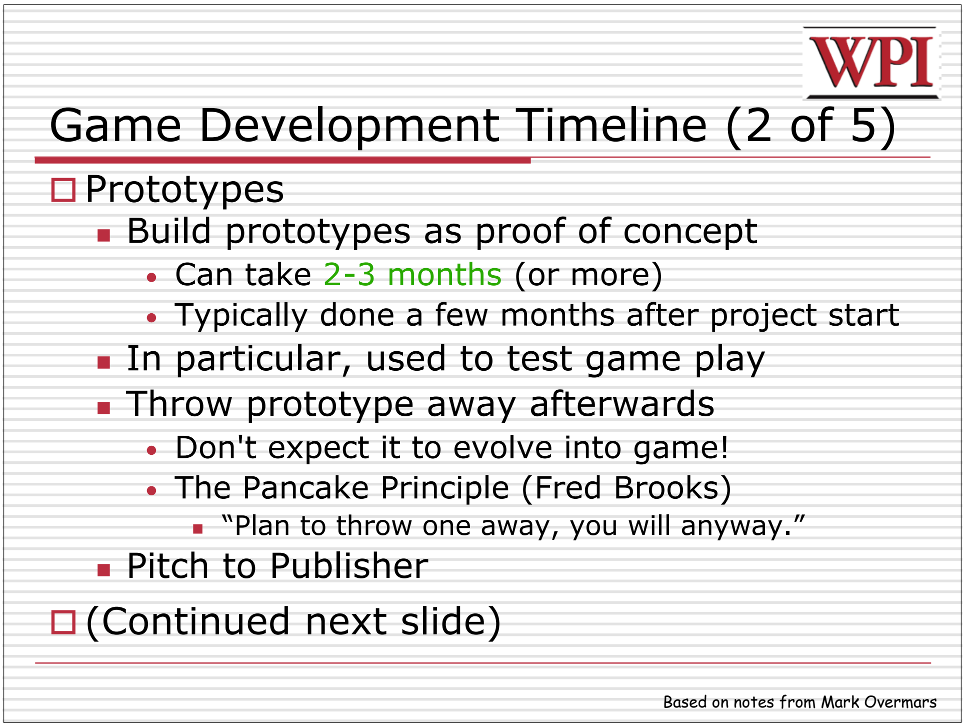
Based on notes from Neal Robison, ATI



Concept: Van Helsing (4 of 4)

Van Helsing
Finished Concept Video

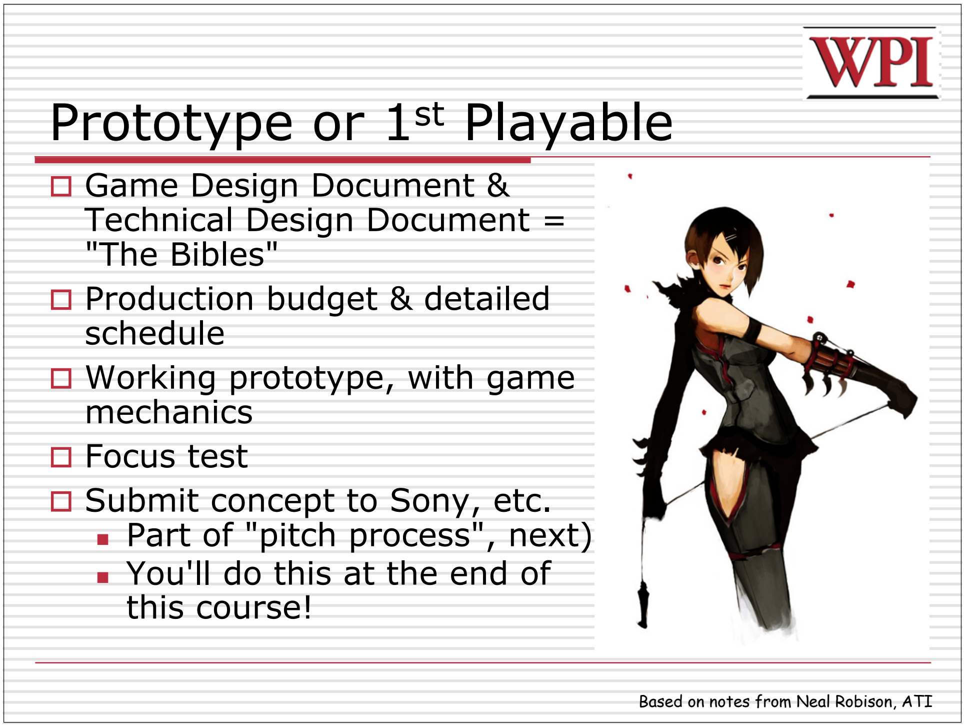
Based on notes from Neal Robison, ATI



Game Development Timeline (2 of 5)

- □ Prototypes
  - Build prototypes as proof of concept
    - Can take 2-3 months (or more)
    - Typically done a few months after project start
  - In particular, used to test game play
  - Throw prototype away afterwards
    - Don't expect it to evolve into game!
    - The Pancake Principle (Fred Brooks)
      - "Plan to throw one away, you will anyway."
  - Pitch to Publisher
- □ (Continued next slide)

Based on notes from Mark Overmars



Prototype or 1st Playable

- □ Game Design Document & Technical Design Document = "The Bibles"
- □ Production budget & detailed schedule
- □ Working prototype, with game mechanics
- □ Focus test
- □ Submit concept to Sony, etc.
  - Part of "pitch process", next)
  - You'll do this at the end of this course!

Based on notes from Neal Robison, ATI

The Pitch Process: Presentation

- Key pitch presentation content:
  - Concept overview & genre profile
  - Unique selling points
    - What makes it stand out from its competitors
  - Proposed technology & target platform/s
  - Team biographies & heritage
  - Outline marketing information, including potential licensing opportunities

The Pitch Process: Prototype

- Key game prototype features:
  - Core gameplay mechanic
  - Game engine / technological proficiency
  - Artistic / styling guide
  - Demonstration of control / camera system
  - Example gameplay goals

The Pitch Process: Project Schedule & Budget

- □ Schedule & budget must:
  - Be detailed and transparent
  - Allow for contingency scenarios
  - Have several sets of outcomes for different size publishers
  - Be realistic

The Deal: Choosing a Publisher Research

- □ Publishers screen Developers
- □ But Developers should also research prospective Publishers:
  - Are they financially stable?
  - Do they have appropriate reach for target?
  - Do they market / PR their games well?
  - Is there a history of non-payment of milestones or royalties?
  - Have they produced many titles?
- □ Sometimes you take what you can get!

The Deal:
IP Rights

- Intellectual Property Rights include:
  - Game name
  - Logos
  - Unique game mechanics & storyline
  - Unique characters, objects & settings
  - Game Source Code including artwork & associated assets
  - Unique sounds and music
- You may not have much power
  - And it probably doesn’t matter

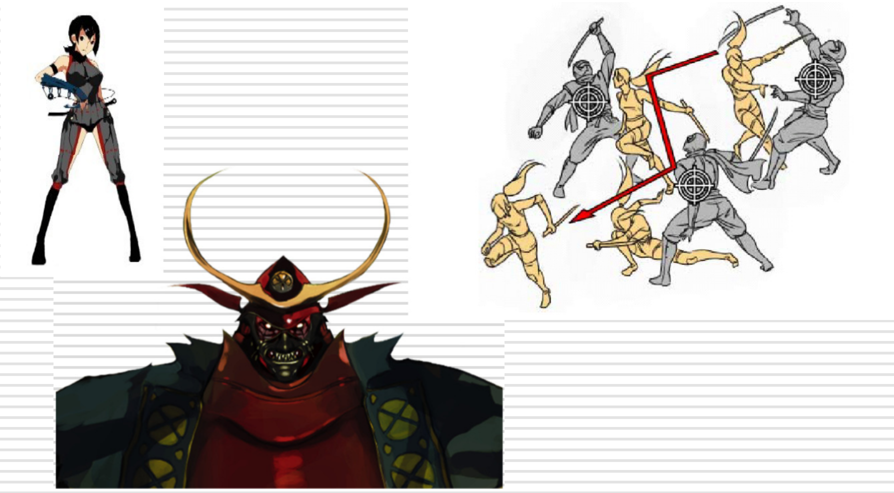
The Deal:
Payment Negotiation (1 of 2)

- Current approximate development costs:
  - $4-5 million for AAA multi-platform
  - $2-3 million for AAA PlayStation 2 only
  - $1 million for A-quality single platform
- Royalties
  - Percentage payments of profits made after recoup of development costs
  - Developer royalties range 0% ("work for hire") to 40%
- Other considerations:
  - Rising-rate royalty: more units sold = higher percentage
  - Clear royalty definition of 'wholesale price' (i.e., including cost of goods etc.)
  - Right to audit publishers books
  - Currency/exchange rate/VAT figures

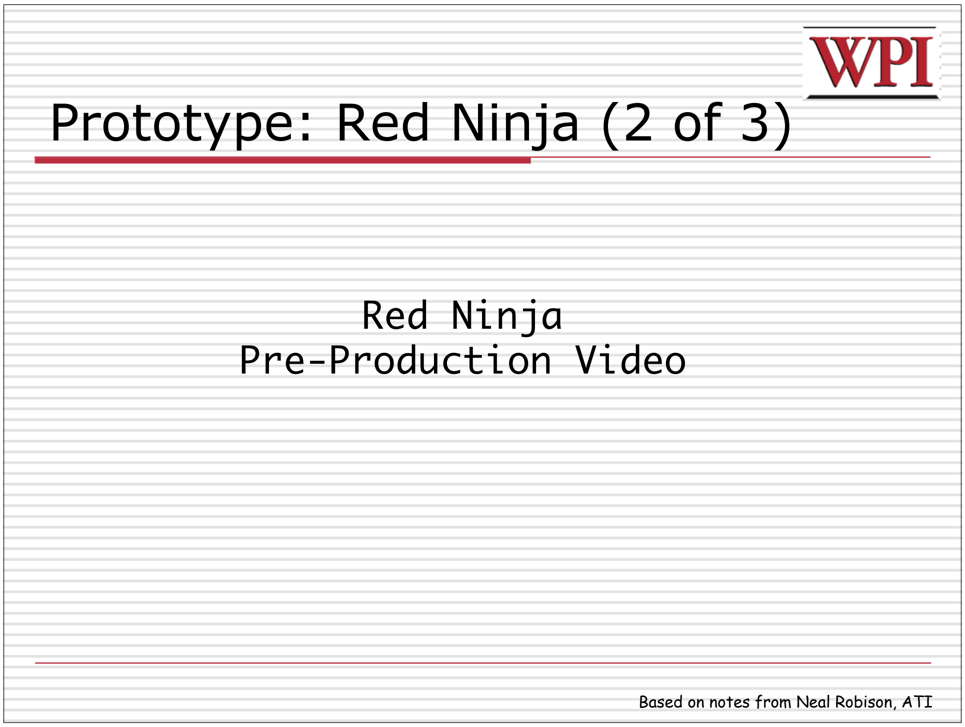
Moving Projects Forward

- □ Most Publishers have a "Green-Light Process"
  - Used to determine which projects go forward
- □ Developers submit to committee at five, mostly independent stages:
  - Concept
  - Assessment
  - Prototype
  - First Playable
  - Alpha
- At each stage, committee:
  - Decides whether or not to continue funding
    - Developers then get next "lump" of money
  - Evaluates market potential
  - Adjusts unit forecasts accordingly

Prototype: Red Ninja (1 of 3)



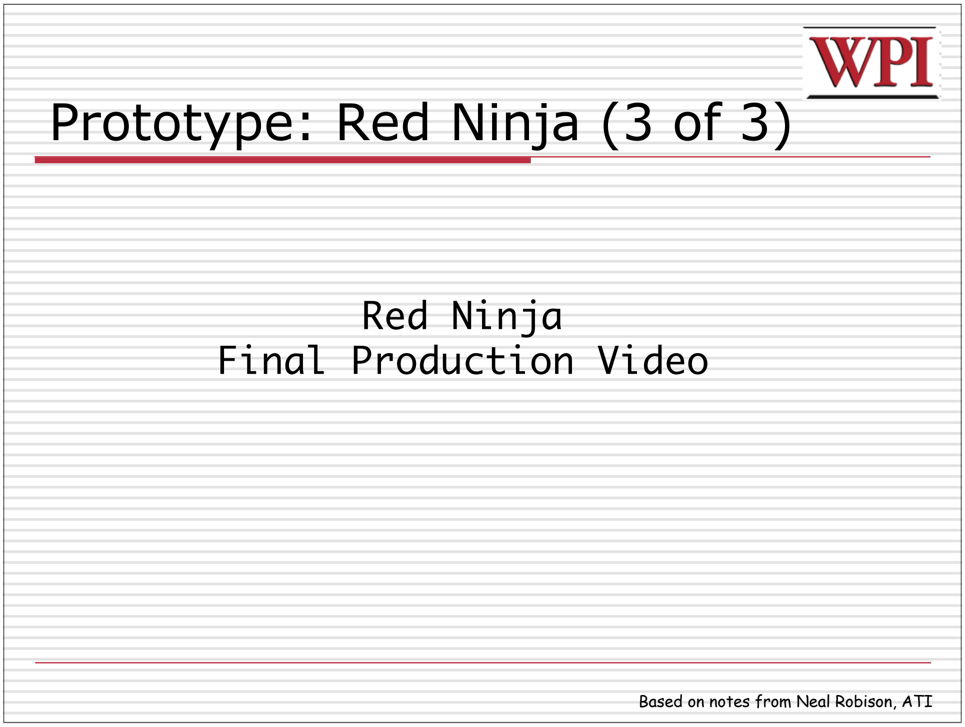
Based on notes from Neal Robison, ATI



Prototype: Red Ninja (2 of 3)

Red Ninja
Pre-Production Video

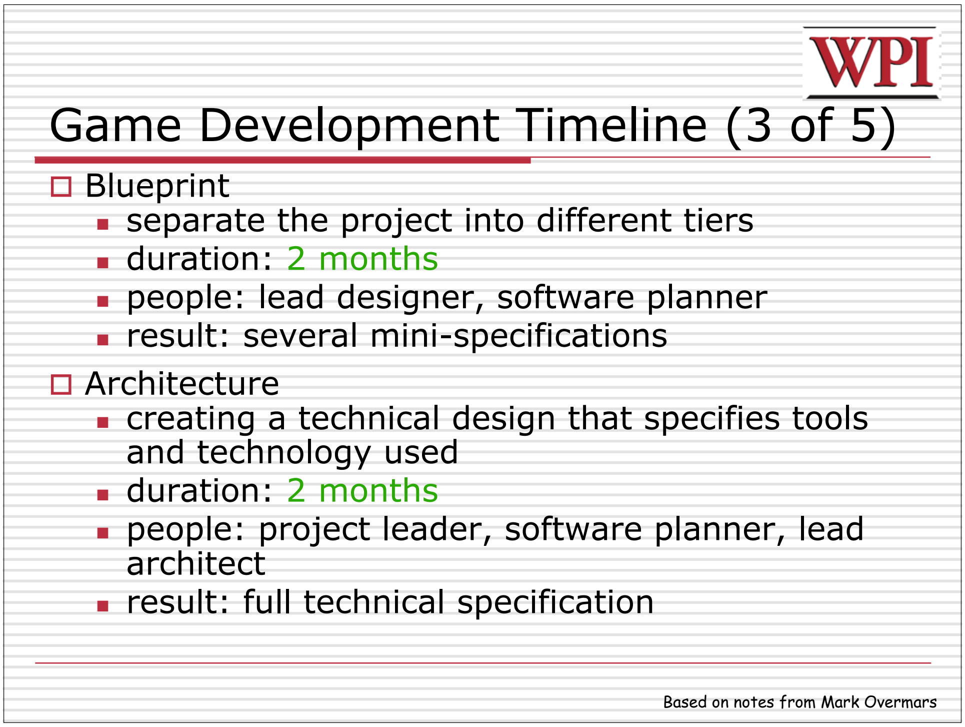
Based on notes from Neal Robison, ATI



Prototype: Red Ninja (3 of 3)

Red Ninja
Final Production Video

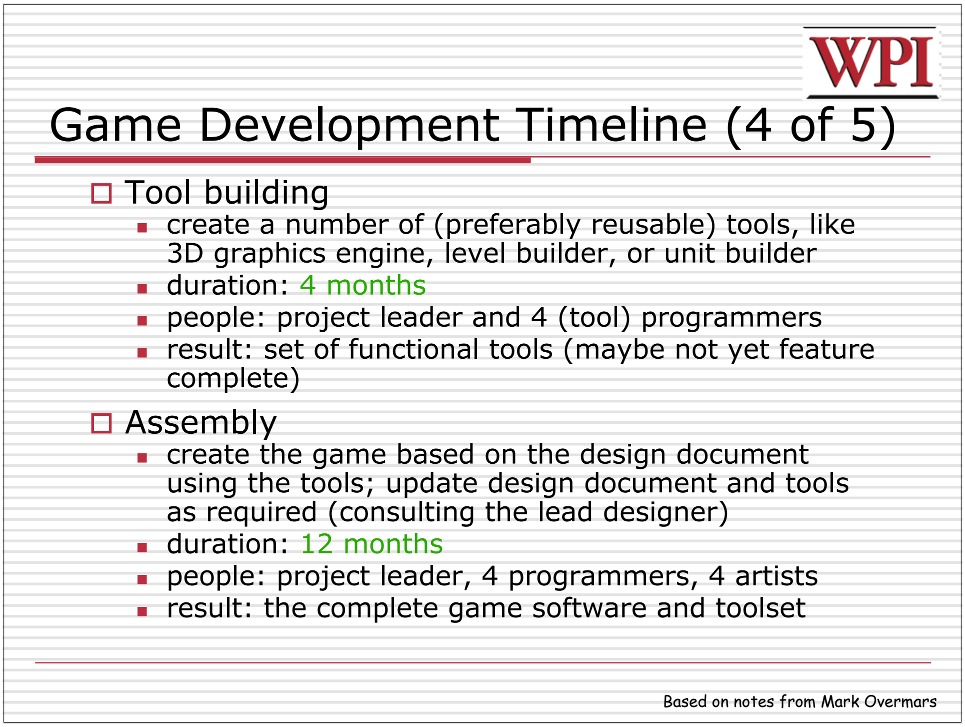
Based on notes from Neal Robison, ATI



Game Development Timeline (3 of 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

Based on notes from Mark Overmars



Game Development Timeline (4 of 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 functional 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

Based on notes from Mark Overmars

Other Milestones: Alpha Definition

- At Alpha stage, a game should:
  - Have all of the required features of the design implemented, but not necessarily working correctly
  - Be tested thoroughly by QA to eliminate any critical gameplay flaws
  - Still likely contains a certain amount of placeholder assets
  - (Continued next slide)

Alpha Definition

- Feature complete
- "Localization" begins
- Focus test
- Play testing
- Marketing continues

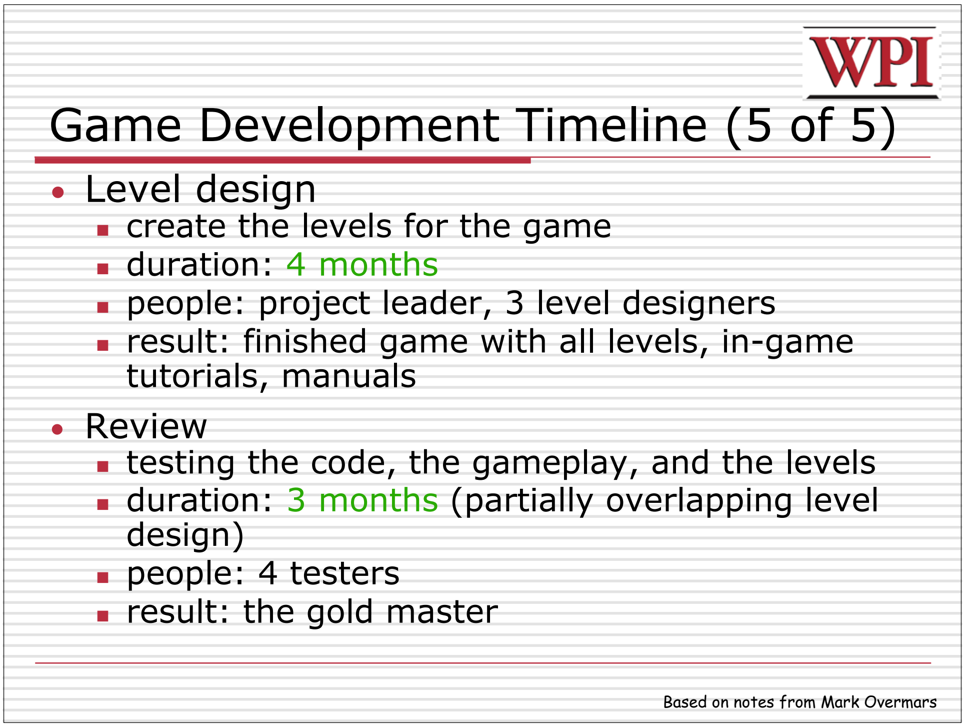
Based on notes from Neal Robison, ATI

Alpha: Crash Bandicoot (1 of 2)

Based on notes from Neal Robison, ATI

Alpha: Crash Bandicoot (2 of 2)

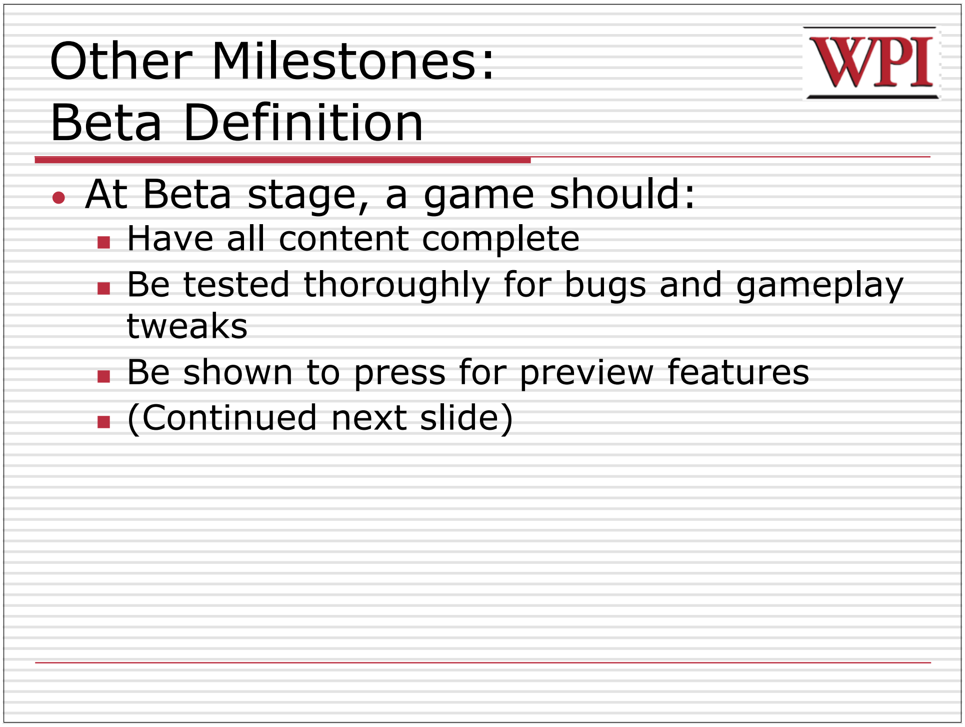
Crash Bandicoot Video



Game Development Timeline (5 of 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

Based on notes from Mark Overmars



Other Milestones: Beta Definition

- At Beta stage, a game should:
  - Have all content complete
  - Be tested thoroughly for bugs and gameplay tweaks
  - Be shown to press for preview features
  - (Continued next slide)

Stages of Development: Beta

- Polish, polish, polish
- Game balancing
- Localization continues
- Demo versions

Based on notes from Neal Robison, ATI

Other Milestones: Gold Master Definition

- At Gold Master stage, a game should:
  - Be sent to the platform holder/s (where applicable) for TRC (Tech. Req. Checklist) testing
  - Be sent to press for review
  - Be sent to duplication for production
  - Be backed up and stored
  - (Continued next slide)

Based on notes from Mark Overmars

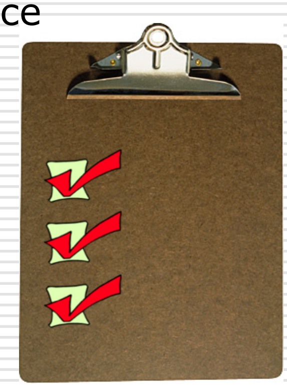
Final/GMC/Gold

- The Game is "Done"
- Testing, testing, testing
- Intense pressure
- Submit to console developers
- Manufacturing timing

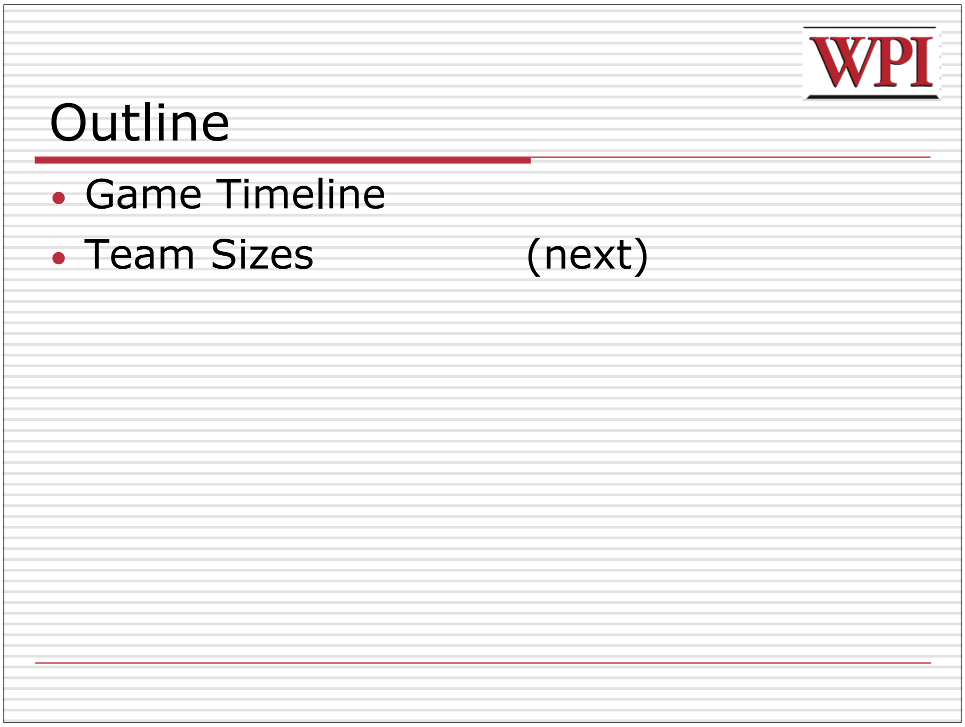
Based on notes from Neal Robison, ATI

Post-Mortem

- Analysis of PR, marketing
- Analysis of production, source Code
- What went right
- What went wrong
- Archive all assets
- Kick-off the Sequel!

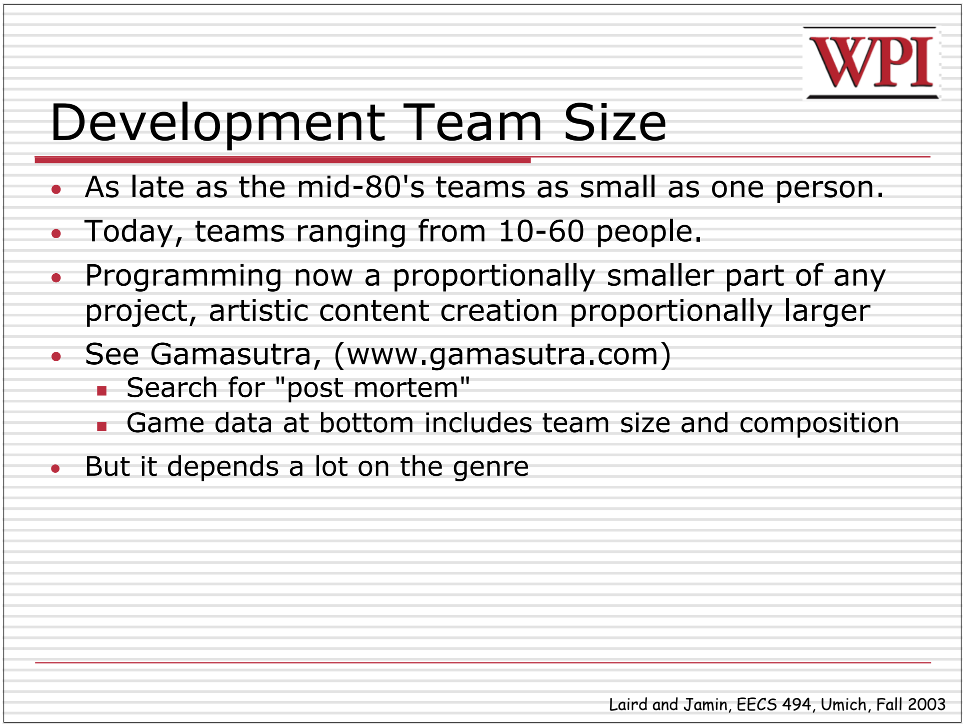


Based on notes from Neal Robison, ATI



Outline

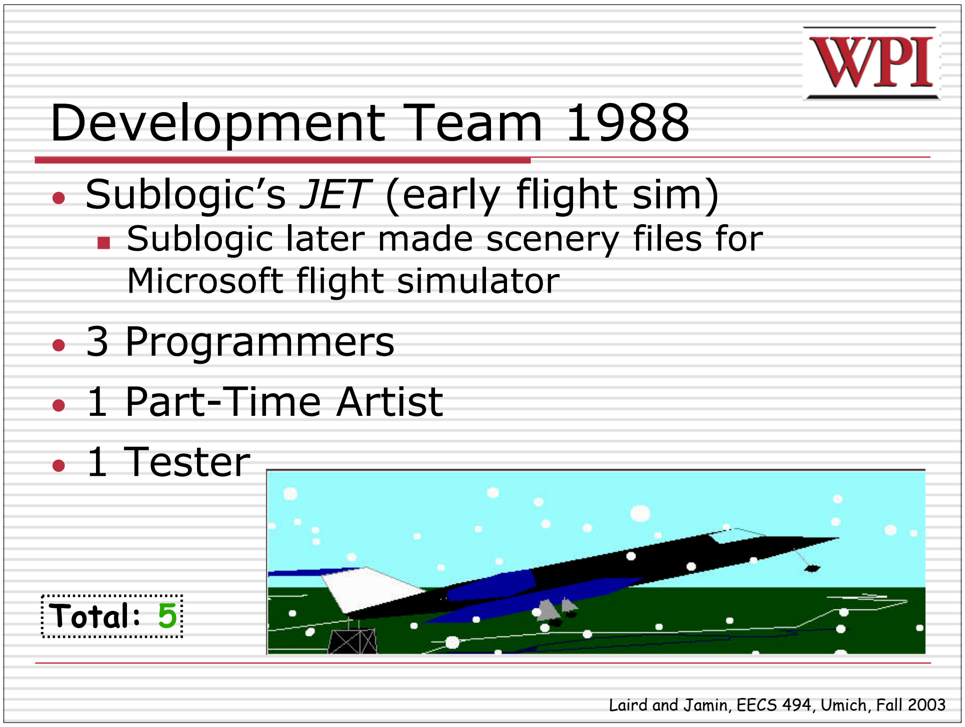
- Game Timeline
- Team Sizes (next)



Development Team Size

- As late as the mid-80's teams as small as one person.
- Today, teams ranging from 10-60 people.
- Programming now a proportionally smaller part of any project, artistic content creation proportionally larger
- See Gamasutra, (www.gamasutra.com)
  - Search for "post mortem"
  - Game data at bottom includes team size and composition
- But it depends a lot on the genre

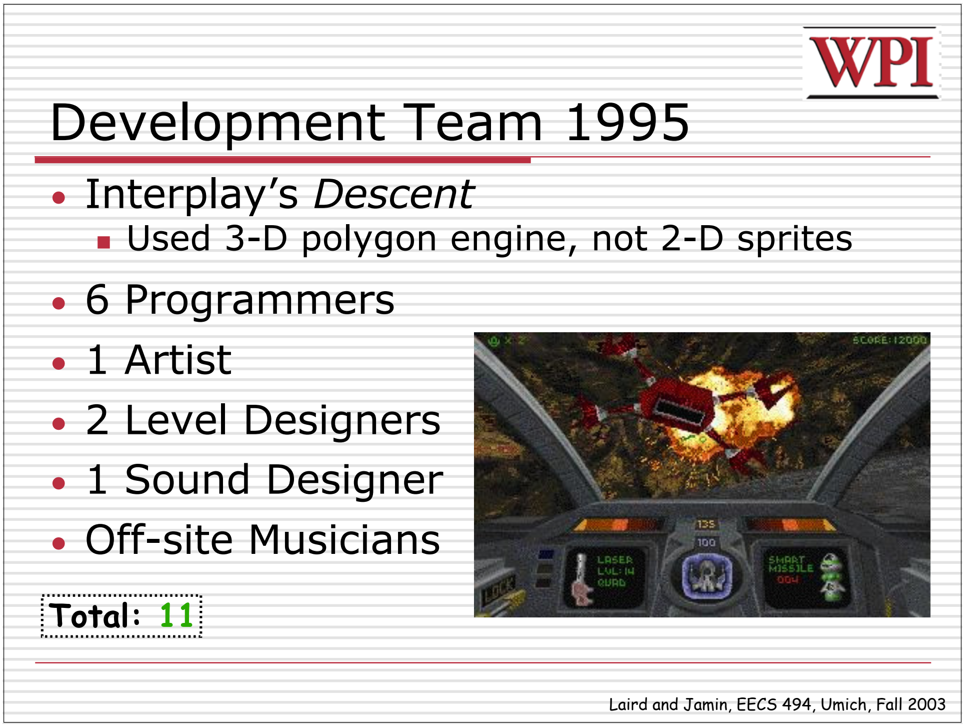
Laird and Jamin, EECS 494, Umich, Fall 2003



Development Team 1988

- Sublogic's *JET* (early flight sim)
  - Sublogic later made scenery files for Microsoft flight simulator
- 3 Programmers
- 1 Part-Time Artist
- 1 Tester

Total: 5



Development Team 1995

- Interplay's *Descent*
  - Used 3-D polygon engine, not 2-D sprites
- 6 Programmers
- 1 Artist
- 2 Level Designers
- 1 Sound Designer
- Off-site Musicians

Total: 11

Development Team 2002

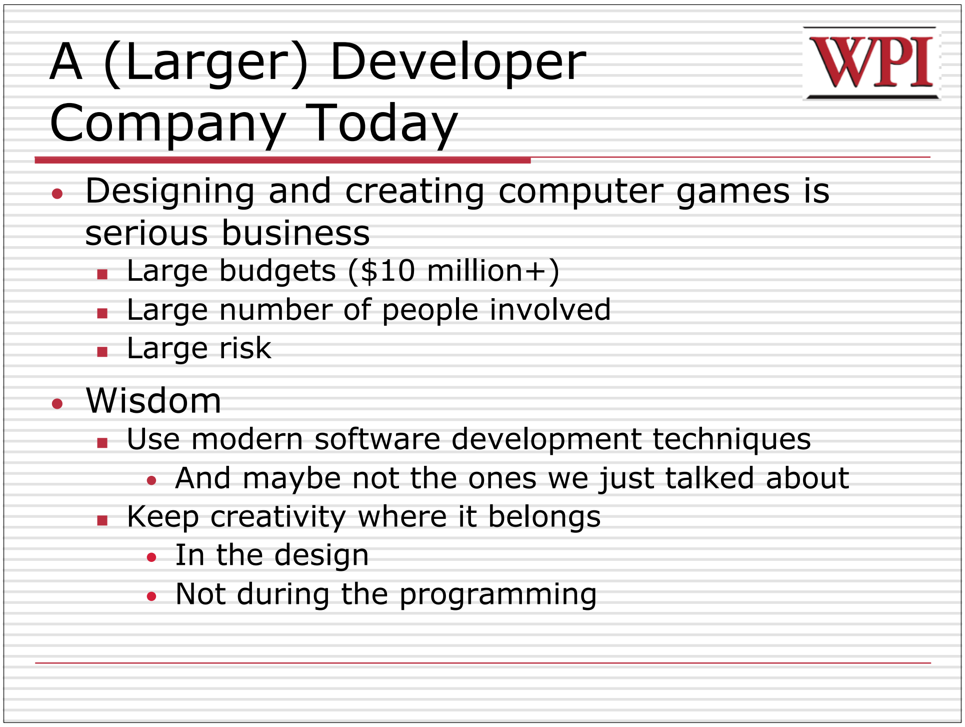
- THQ's *AlterEcho*
- 1 Executive Producer
- 1 Producer
- 4 Programmers
- 2 Game Designers
- 1 Writer
- 3 Level Designers

- 3 Character Modelers and Animators
- 1 2d and Texture Artist
- 1 Audio Designer
- 1 Cinematic Animator
- 1 QA Lead and Testers

Total: 19+

Development Teams for Online Games

- Star Wars online (2003?)
- Development team: 44 people
  - 50% Artists
  - 25% Designers
  - 25% Programmers
- 3 Producers
- "Live" Team (starting at Beta, 6 months before done)
  - 8 Developers
  - 50-60 Customer support (for 200K users)
  - 1000 Volunteer staff (for 200K users)



A (Larger) Developer Company Today

- Designing and creating computer games is serious business
  - Large budgets ($10 million+)
  - Large number of people involved
  - Large risk
- Wisdom
  - Use modern software development techniques
    - And maybe not the ones we just talked about
  - Keep creativity where it belongs
    - In the design
    - Not during the programming