

# Game Development Process

John Laird

University of Michigan

# Game Development Lifecycle

- Preproduction
  - Should be able to show what is going to make the game good
  - A bunch of different attempts to discover fun
  - <http://www.gamasutra.com/features/>
- Production
  - Iterate on the core gameplay created during preproduction
  - Add features
  - Create level
- Alpha
  - Feature complete but in a rough state
  - Fix bugs
  - Add polish – effects, new help messages, tuning tweaking
- Beta
  - Bug fixes only
- Gold!

# Game Production Timeline

- Inspiration & Preproduction (1 month)
  - Results in game treatment/concept paper
- Conceptualization (3-5 months)
  - Results in design summary/design document
  - Story boards and prototype
- Technical Architecture (2 months)
  - Determine the technical details
  - Results in master technical specification/production document
- Tool Building (4 months)
- Assembly (12 months)
- Levels (4 months)
- Review, Testing, ... (3 months)
- Total (about 2 years)

# Why Need Documentation: Team Dynamics

- Games are not developed by one person anymore
- Collaboration between people of vary different backgrounds – technical vs. artistic
- Designers and programmers request assets from artists and require changes to environments for technical/gameplay reasons
- Gameplay programmers have to work closely with animation
  - Timings have a profound impact on gameplay
  - New moves often have to be created
- Collision geometry needs to meet gameplay requirements

# Three Stages of Design Documentation

- Concept Paper / Design Treatment
  - Used to determine if this idea worth development.
- Design Document / Design Summary
  - Blueprint for the game
  - Game script/storyboard or prototype
  - Possible markets
  - Basis for in depth evaluation
- Technical Design Specification / Production Document
  - Used to write the code and manage product

# Design Treatment/Concept Paper (Project Proposal)

- Purpose:
  - Basis to decide if worth investing any serious effort
  - Sales tool
- Many sections that summarize game:
  - The “high concept” of game
  - Basic design choices: game genre, ...
  - Features that make game exceptional
  - Game story: plot, characters: no more than 1 page
  - Game play: what does the player do?
  - Technical development: platform, development environment, ...
  - Expected audience
  - Estimated cost [development time]
  - Risk analysis
- Keep it short and simple: 1-2 pages
  - Does not have all plot twists -- just main ideas
- May include a mini-prototype of key ideas

# Design Treatment:

## *High Concepts*

- Should be able to describe in 1-2 sentences:
  - it better be *cool* and *interesting*!
- Ping-Pong on the computer
- An ordinary technician battles trans-dimensional monsters after an accident at a secret research facility
- Armies based on ancient civilizations battle each other
- A fighting game based on knocking opponents out of the stage
- An escaped prisoner attempts to prevent the opening of the gates to a hellish realm
- Simulate playing rock music on a plastic guitar-shaped peripheral.
- A cybernetically-enhanced human super-soldier and his AI companion battle an alliance of alien races
- Control the evolution of a species from a unicellular organism, through development as an intelligent and social creature, to interstellar exploration

# Design Treatment:

## *Game Story*

It's 2025, 5 years after a major nuclear war has destroyed many things. Australia has been divided into three segments, a penal colony, the badlands – a swath of land made up of villages of policed mutants, and other crazed beings, and the free zone where a group of outcasts are reestablishing democracy. It is up to you, the player, to escape and make it to the free zone. You have only martial arts expertise to help you. You must run a foot race to the free zone to lead a revolution.

# Design Treatment:

## *Game Play and Look*

The game features 3D graphics which scroll toward the player (the movement somewhat like a raycaster like Doom except the movement is 180 degrees not 360. Imagine “Auto Racing Doom”). The player is represented by a third person polygon figure (like Virtua Fighter) controlled with a joystick. The faster you run and avoid obstacles, the better chance you have to survive. Too slow a pace allows your chasers to catch up to you.

# Design Treatment:

## *Game Play and Look 2*

The player runs through a continuous stream of towns, forest, brush and other settings attempting to maintain speed, jumping onto, over and around, while mutants and security guards and more enemies attack them. At your disposal are weaponry you pick up, water and food for health and a series of “Virtua Fighter” like martial arts moves including flips and summersaults.

The trick is to use the flips and rolls to jump over things, on to building rooftops (in towns) and to use kicks, punches and weaponry to take out oncoming enemies. ...

# Design Treatment: Technical

This game will feature our new 3D engine, coupled with animated 3D backgrounds. There are roughly 50 scenes we are constructing, each with several views; these scenes will be rendered with 3D Studio Max. We will use a PC development platform and later port to the Xbox 360. The development language will be C++, using several APIs developed in and out of house. Overall, we expect a development time of roughly 10-16 months.

# Three Stages of Design Documentation

- Design Treatment / Concept Paper
  - Used to determine if this idea worth development.
- **Design Summary / Design Document**
  - **Blueprint for the game**
  - **Game script/storyboard or prototype**
  - **Possible markets**
  - **Basis for in depth evaluation**
- Technical Design Specification / Production Document
  - Used to write the code and manage product

# Design Document

- Purpose
  - Exhaustively details *everything* that will happen in the game
    - Cheaper to evaluate and change than a computer implementation
  - Basis for estimating cost and tracking progress
  - Allows others to start designing implementation
  - Is the game “bible” - over 100 pages
- Created by the Game Designer and Technical Leads
  - Usually not same as the game developer
- If it isn't in this, it might not be in the game
  - Depends on development philosophy

# Design Document

- More formal and complete than concept treatment
  - What does the player do?
  - What is the interface?
  - What is the plot?
- Details! Leave nothing to the imagination
  - What are all the levels?
  - What are all the characters?
  - How can they interact?
  - What can you do to them?
  - Art bible – what will be the look of the game
- Development plan/schedule
  - What is going to be done
  - When is it going to be done
  - Who is going to do it

# Design Document Sections

- Product Specification
- Game Overview
- Gameplay and Mechanics
- Story, Setting, Character
- Levels
- Interface
- AI
- Art

# Design Document: Product Specification

- Production team:
  - Members and experience
- Time: game play, shelf life
- Target platform
- Production tools:
  - Microsoft project, excel, PhotoShop, ...
- Schedule with milestones and deliverables
- Etc.
  - Localization: Languages, art work, ...
  - Packaging and Documentation

# Design Document: Game Overview

- Game Concept
- Feature Set
- Genre
- Target Audience
- Game Flow Summary
  - How does the player move through the game. Both through framing interface and the game itself.
- Look and Feel
  - What is the basic look and feel of the game? What is the visual style?
- Project Scope – A summary of the scope of the game.
  - Number of locations
  - Number of levels
  - Number of NPC's
  - Number of weapons
- Etc.

# Game Play

- Game Progression
- Mission/challenge Structure
- Puzzle Structure
- Objectives – What are the objectives of the game?

# Mechanics

- Rules to the game, both implicit and explicit. This actually can be a very large section.
- Physics – How does the physical universe work?
- Movement
  - General Movement
  - Other Movement
- Objects
  - Picking Up Objects
  - Moving Objects
- Actions
  - Switches and Buttons
  - Picking Up, Carrying and Dropping
  - Talking
  - Reading
- Combat – If there is combat or even conflict, how is this specifically modeled?
- Artificial Intelligence
- Economy – What is the economy of the game? How does it work?

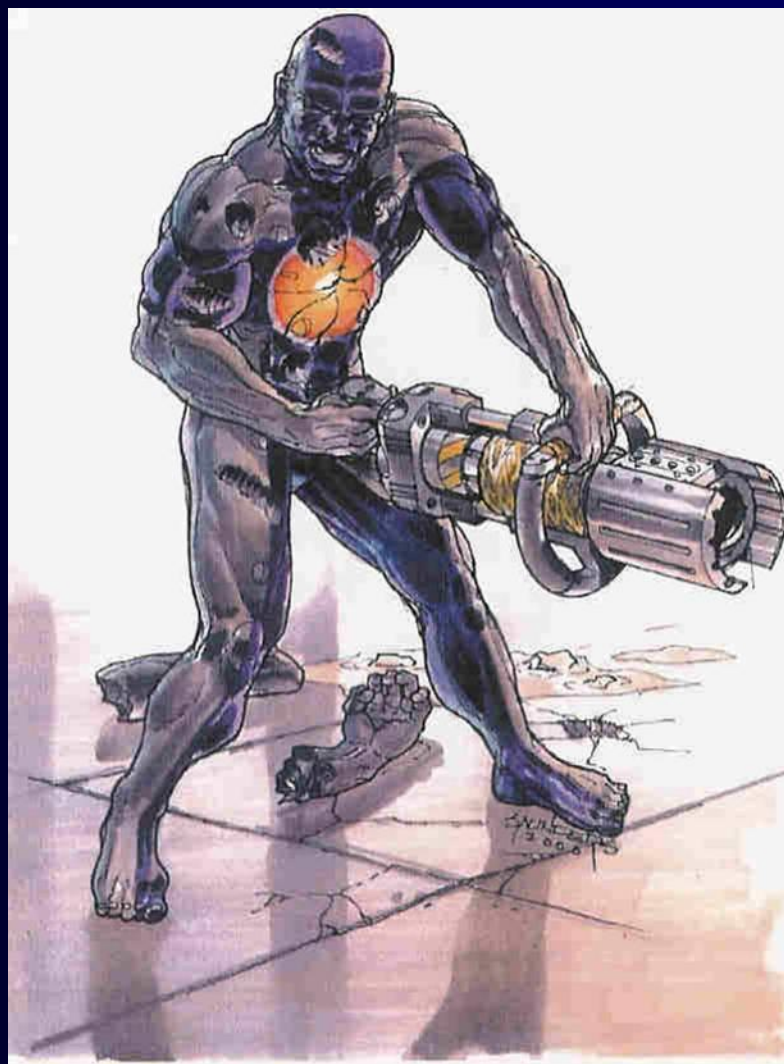
# Story, Setting and Character

- Story and Narrative –
  - Specific details like scripts and cut scenes may not be in this document but be in the Story Bible.
- Game World
  - General description of world structure, look
- Characters
  - Back story, personality, back story, profile.

# Character Profile

Color	Cost	Player Controlled Abilities	Combat	Speed	Lifespan	Hitpoints
Green	Low	Scout and errand ditto. Can use few weapons.	Low	Fast	Medium	Low
Gray	Medium	Minor combat skill, use some weapons. Can infiltrate some restricted human	Medium	Medium	High	Medium
Yellow	Medium High	Good combat ability. Can use most weapons Strong Defense.	High	Medium	Medium	High
Purple	Very High	Strong Combat, can use any weapon. Seek and Destroy Abilities.	Very High	Medium to Fast	Low	Very High
Ebony	Medium High	Hacker. Detect and disable traps. Hack and Reprogram Robots.	Very Low to None	Slow	Medium Low	Low

# Example Art



# Level Design

- Name of the Section/Level/Scene
- Physical and audio appearance
- Background - sketch if possible
- Foreground objects and characters
  - What actions can they perform?
  - Which objects are animated?
  - Sound effects
- Objectives
- Transitions

# Level Design

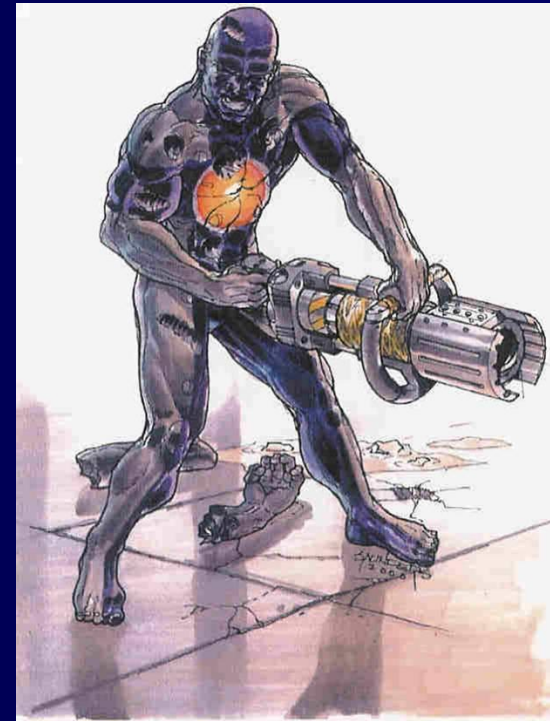
- Consumer testing (not by game designers)
  - Where do they get bored?
    - “Basically, I fell and I died. Instantly. The game consisted only of levels where I wanted to give up before I’d reached the end.”
  - Where did they die?
  - How much time is there between challenges?
  - Should be ~20 minutes to finish a level
    - Sometime move the player along if they’re having trouble
      - Double the number of checkpoints, easier problems
  - Not hard to make level more difficult
    - Hard to make level easier

# Interface

- Visual System
  - HUD - What controls
  - Menus
  - Rendering System
  - Camera
  - Lighting Models
- Control System – How does the game player control the game? What are the specific commands?
- Audio
- Music
- Sound Effects
- Help System

# Design Document: Art Specification

- Define overall style of art work
  - Example backgrounds
- Include example sketches of all main characters



# Key Points to a Good Design Document

- Philosophy needs to be explicit (goals of game)
- Make it readable, well written
  - Use tables, graphics as appropriate
- Give priorities to the *ideas* so everyone knows what is important and what has been rejected
  - Indispensable, Important, If Possible, Rejected
  - Include reasons!
- Give all the details
  - Tables of what happens when X hits Y
- How will you do things: motion capture/animation

# Three Stages of Design Documentation

- Design Treatment / Concept Paper
  - Used to determine if this idea worth development.
- Design Summary / Design Document
  - Blueprint for the game
  - Game script/storyboard or prototype
  - Possible markets
  - Basis for in depth evaluation
- **Technical Design Specification / Production Document**
  - Used to write the code and manage product

# Technical Design Specification

- Purpose: details of implementation
  - Written before you spend lots of \$\$'s
  - Allows many people to work at once
- Further details of art, music, sounds, video, etc.
  - Definition of data structures and interfaces
  - Pseudo-code
    - Can debug ideas before “cut bits”
    - Provides documentation
- Development environment
- Schedule with milestones, releases

- <http://www.youtube.com/watch?v=Jd3-eiid-Uw>
- <http://www.youtube.com/watch?v=QgKCrGvShZs>
- <http://www.youtube.com/watch?v=eLy9AWi67XU>
- <http://www.youtube.com/watch?v=nZ-VjUKAsao&feature=channel>