



WPI

---

# Lindeman's Lectures: Game Development

Robert W. Lindeman

Associate Professor

Interactive Media & Game Development

Human Interaction in Virtual Environments (HIVE) Lab

Department of Computer Science

Worcester Polytechnic Institute

[gogo@wpi.edu](mailto:gogo@wpi.edu)

---

## Five-Lecture Structure

---

- July 10
  - Introduction
  
- July 17
  - Game Development
  
- July 18
  - Game Design
  
- July 24
  - Level Design
  
- July 31
  - Game Presentations (by you!)

## Summary of Syllabus

---

- Lectures and in-class exercises
  - Exercises designed to drive home concepts, or to get you thinking about projects
- Programming projects
  - Take you through the game development process
  - Build something interesting
- Talking in class
  - Please ask questions!!
  - I will too!!
- Computers in class
  - Please focus on class material
  - Don't read email, etc.

# Expected Outcomes of this Lecture Series

---

- Understanding of Game Development Process
  - Why is it hard?
  - What types of people are involved?
  - Why is it important?
- Understanding application areas of games and interactive techniques
- Motivation for future work by you!

## Today's Outline

---

- What is a Game? (next)
- The Game-Development Process
- What Makes a Good Game?

# What is a Computer Game?

## User Perspective

---

- A goal (or set of goals)
  - Save the Princess (solve these puzzles first)
  - Score points (get power ups)
  - Finish first (unlock features)
- A set of rules governing game play
  - Turn taking, like RPGs
  - Reaction to events, like Tetris' falling blocks
  - Legal actions
- Visual (audio, *etc.*) content
- Control techniques
  - Button mappings

# What is a Computer Game?

## System Perspective

---

- A set of resources that are managed to support an entertainment (usually) application
- Graphical (audio, *etc.*) rendering
- A user interface
- Script handling
- Event processing
  - Time, collisions, *etc.*
- File I/O
- Asset-creation tools
  - Models, graphics, sound, *etc.*
- Optional
  - Networking
  - AI

## Elements of a 3D Game

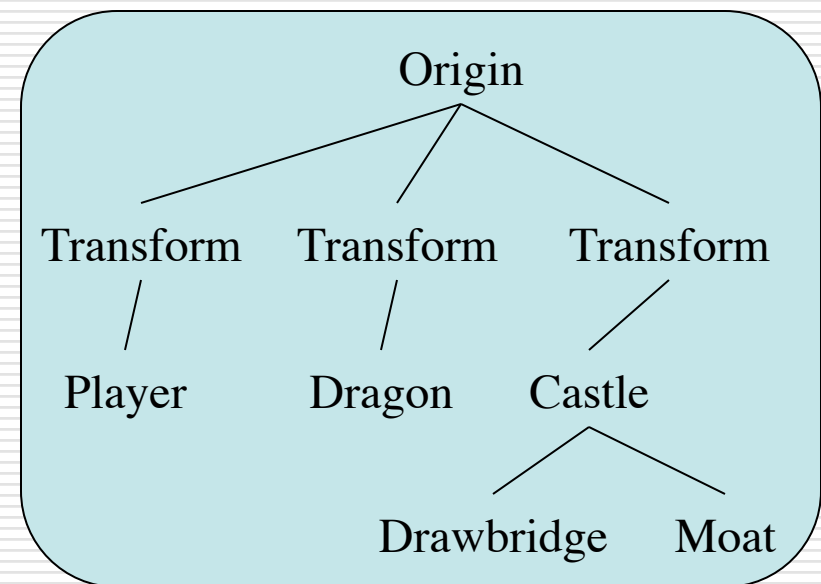
---

- Game engine
- Scripting
- Graphical user interface
- Models
- Animations
- Textures
- Sound
- Music
- Support infrastructure
  - Web site
  - Support forums
  - Admin tools
  - Database



# Game Engine

- Scene graph
  - Representation of the world
  - Includes characters
- Timing is very important
  - Events
    - Time-based
    - Multi-player
  - Synchronization
- Database of objects
- Networking
  - Between server and clients
  - Between servers
  - Between clients



## Game Engine (cont.)

---

- Core utilities
- Rendering system
- Physics
- Artificial intelligence
- Input management

## Core Utilities

---

- ❑ Data structures
- ❑ Game-state management
- ❑ Timers
- ❑ Memory management
- ❑ Journaling services
- ❑ File logging
- ❑ Performance profiling tools
- ❑ Encryption/decryption

## Scripting

---

- Scripting languages provide easier path to building a game
  - Provides access to game-world objects (GWOs)
  - Allows most aspects of the game to be defined
  - Tie all parts of the game together
  - Leverage investment in engine development
  - Trade control for automation
  
- Sample scripting languages for games
  - Lua ([www.lua.org](http://www.lua.org))
  - Torque Script ([www.garagegames.com](http://www.garagegames.com))

## Graphical User Interface

---

- Provides access to
  - Game menus (*e.g.*, save, load, boss)
  - Player status (*e.g.*, health, current speed)
  - Maps
  - Non-Player Character (NPC) dialog
  - Player-to-player chat

## Models

---

- Objects are made from
  - Geometry (a.k.a., polygons)
  - Lighting
  - Textures
  
- Vertices and connectivity
  - Triangles
  - Triangle-strips
  - Meshes
  - Patches/surfaces



## Animations

---

- Making things move believably
- Consider character states
  - Idle
  - Running
  - Jumping
  - Shooting
  - Dying
- More animations means more variability

## Texturing

---

- Created/manipulated using image processing software
  - Photoshop
  - Paint Shop Pro
- Mapped to geometry (models)
- Very powerful image enhancing techniques
  - Can be used for fake shadows, fake reflections, much more



## Sound and Music

---

- One of the most-important elements of any experience is sound
- Sound effects
  - Make things more (hyper-) realistic
- Musical score
  - Sets the mood
  - Builds emotion
- Speech output
- Very important skill

## Support Infrastructure

---

- Front-end for running games
  - Steam
- Web site
  - Promotion, log-in, etc.
- Support forums
  - Cheats, hints, discussion of new ideas
- Admin tools
  - User maintenance
  - Anti-cheating measures
- Database
  - Game-state maintenance

## Game-Development Outline

---

- What is a Game?
- Genres
- The Game-Development Process
- What Makes a Good Game?

## What is a Game? (1 of 3)

---

### □ Movie?

- No *interaction*, outcome fixed

### □ Toy?

- No *goal*, but still fun!
- Players can develop own goals

### □ Puzzle?

- strategy and outcome is the *same* each time

"A computer game is a software program in which one or more players make decisions through the control of game objects and resources, in pursuit of a goal."

Based on notes from Mark Overmars

# What is a Game (2 of 3)

---

- A Computer Game is a *Software Program*
  - Not a board game or sports
  - Consider: chess vs. soccer vs. Warcraft
    - Ask: What do you lose? What do you gain?
  - Lose: 1) *physical pieces*, 2) *social interaction*
  - Gain: 1) *real-time*, 2) *more immersive*, 3) *more complexity*
- A Computer Game involves *Players*
  - *Think* about your audience; the game is not for *you* but for *them*.
  - Don't just think about your story or the graphics or the interface, but consider the *players*.

Based on notes from Mark Overmars

# What is a Game (3 of 3)

---

- Playing a Game is About Making Decisions
  - Ex: what weapon to use, what resource to build
  - Can be frustrating if decision does not matter
  - Want good *gameplay* (major topic later)
- Playing a Game is About Control
  - Player wants to impact outcome
  - Uncontrolled sequences can still happen, but should be sparing and make logical
- A Game Needs a Goal
  - Ex: Defeat Ganandorf in Zelda
  - Long games may have sub-goals
  - Ex: recover Triforce first, then Sword of Power
  - Without game goals, a player develops his/her own (a toy)



Based on notes from Mark Overmars

# What a Game is *Not* (1 of 2)

---

- *A bunch of cool features*
  - Necessary, but not sufficient
  - May even detract, if not careful, by concentrating on features, not game
  
- *A lot of fancy graphics*
  - Games need graphics just as hit movie needs special effects, but neither will save weak idea
  - Game must work without fancy graphics
  - Suggestion: Should be fun with simple objects

"When a designer is asked how his game is going to make a difference, I hope he ... talks about gameplay, fun and creativity – as opposed to an answer that simply focuses on how good it looks." – Sid Meier (*Civilizations, Railroad Tycoon, Pirates*)

Based on notes from Mark Overmars

# What a Game is *Not* (2 of 2)

---

- *A series of puzzles*
  - Most games have them, but they are not the game
- *An intriguing story*
  - Good story encourages immersion, but will mean little without good gameplay
  - Example: *Baldur's Gate* is a linear story.
  - Going wrong way gets you killed.
  - Not interactive: interaction in world all leads to same end.

---

Based on notes from Mark Overmars



# Games are Not Everything

---

- Most important
  - Is it *fun, compelling, engaging*?
- Computers are good at interactivity
  - Allow for interactive fun
- Examples:
  - *SimCity*
    - Very compelling, but mostly no goals.
    - More of toy than a game, but still fun.
  - *Grim Fandango*
    - Good visuals, story, etc., but need to do puzzles to proceed
    - Could have skipped to just watch story
    - Would still have been *fun* without the gameplay.

Based on notes from Mark Overmars

## Definition Revisited

---

"A computer game is a software program in which one or more players make decisions through the control of game objects and resources, in pursuit of a goal."

□ What's missing from this definition?

# Welcome to the Entertainment Business!

---

WPI

“To be boring is the worst sin of all.”

Stanley Kubrick

## Game-Development Outline

---

- What is a Game?
- The Game-Development Process (next)
- What Makes a Good Game?

# 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

*Based on notes from Mark Overmars*

## Concept

---

- Define game concept
- Define core game features



Based on notes from Neal Robison, ATI

## Concept: Van Helsing (1 of 4)

---



Based on notes from Neal Robison, ATI

## Concept: Van Helsing (2 of 4)

---



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

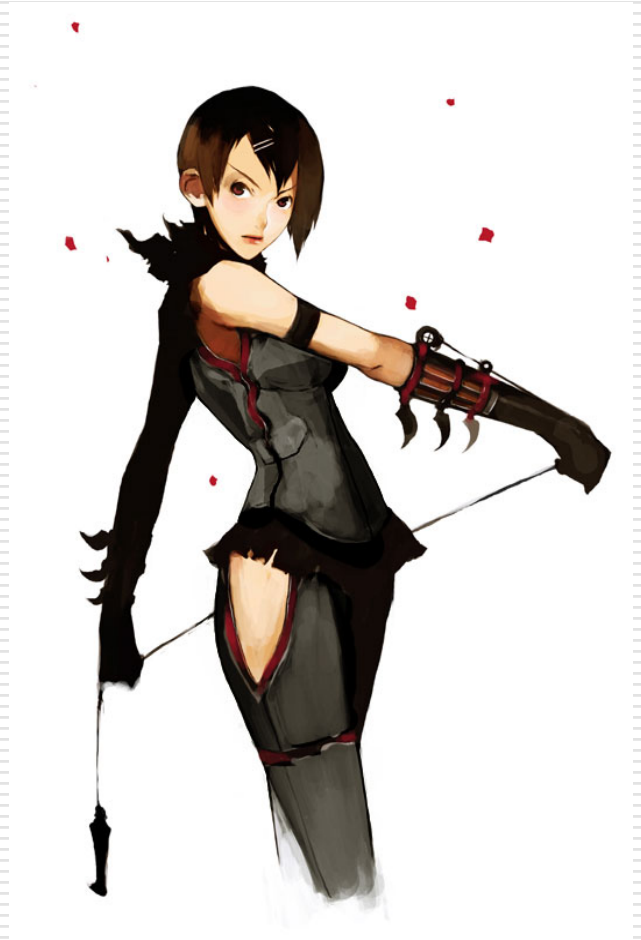
- Build prototypes as proof of concept
  - Can take **2-3 months** (or more)
  - Typically done a few months after project start
- In particular, used to test game play
- Throw prototype away afterwards
  - Don't expect it to evolve into game!
  - The Pancake Principle (Fred Brooks)
    - “Plan to throw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

Based on notes from Mark Overmars

## Prototype or 1<sup>st</sup> Playable

---

- Game Design Documents
  - Vision Document
  - Technical Design Document
  - Artistic Design Document
  - Together == "The Bibles"
- Production budget & detailed schedule
- Working prototype, with game mechanics
- Focus test
- Submit concept to Sony, etc.
  - Part of "pitch process"



Based on notes from Neal Robison, ATI

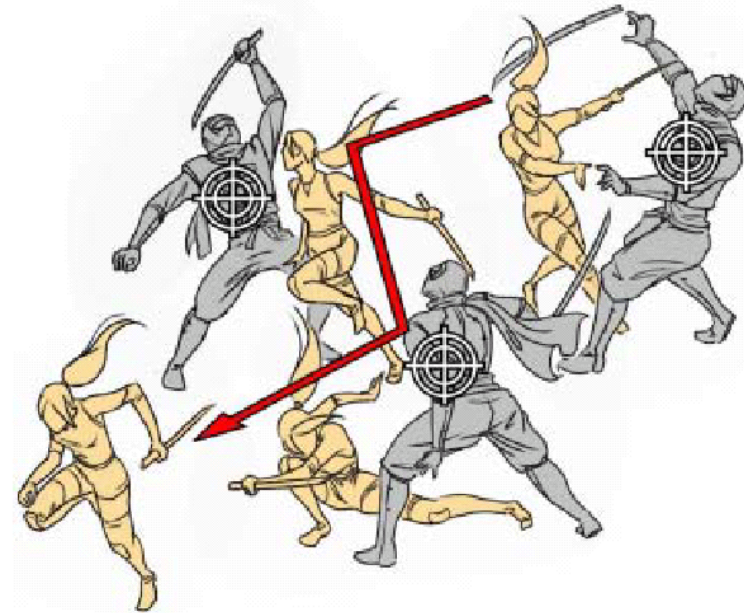
## Prototype

---

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

## 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

## Alpha Definition

---

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



Based on notes from Neal Robison, ATI

## 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

## Game-Development Outline

---

- What is a Game?
- Genres
- The Game-Development Process
- What Makes a Good Game? (next)

# What Makes a Good Game?

---

- "A great game is a *series* of *interesting* and *meaningful choices made by the player* in pursuit of a *clear* and *compelling goal*."  
- Sid Meier
  
- "Natural Funativity"
  - Survival-skill training
  - Need to have player develop a set of skills with increasing levels of difficulty
  - Putting them to the test = mission, quest, level, *etc.*
  - Prize at the end (or in the middle)

## Structure of Games

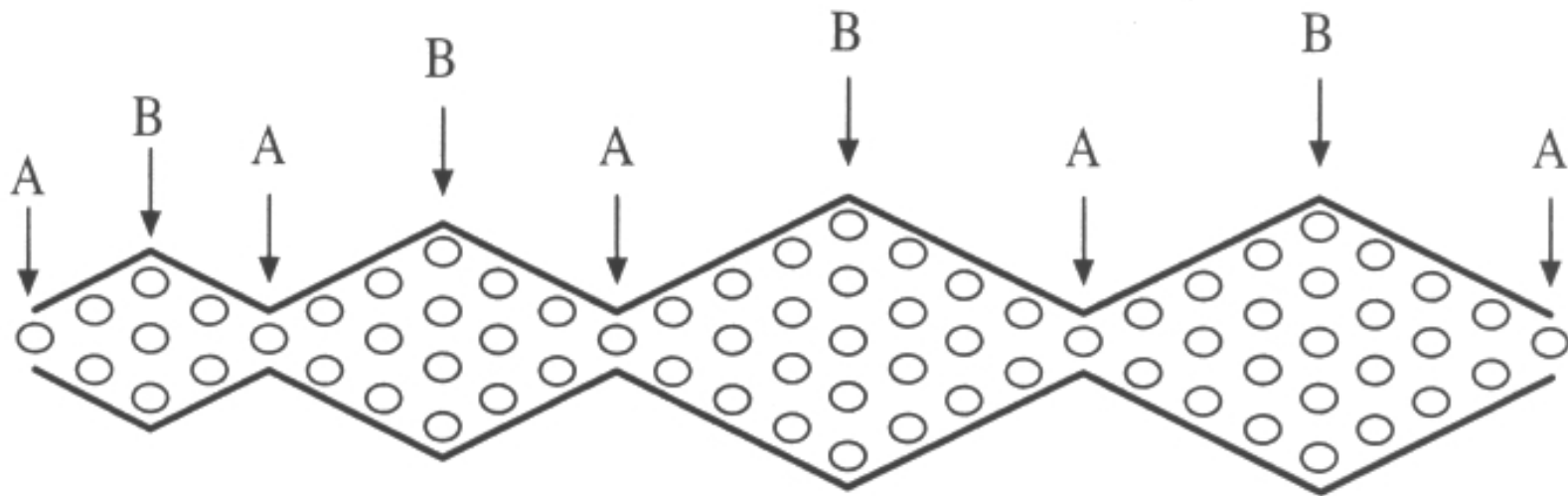
---

- Movies have linear structure
  - No choice by viewer
- Games must provide "interesting and meaningful choices"
  - Otherwise, user is not in control
- Random death is frustrating!



# Convexity of Game Play

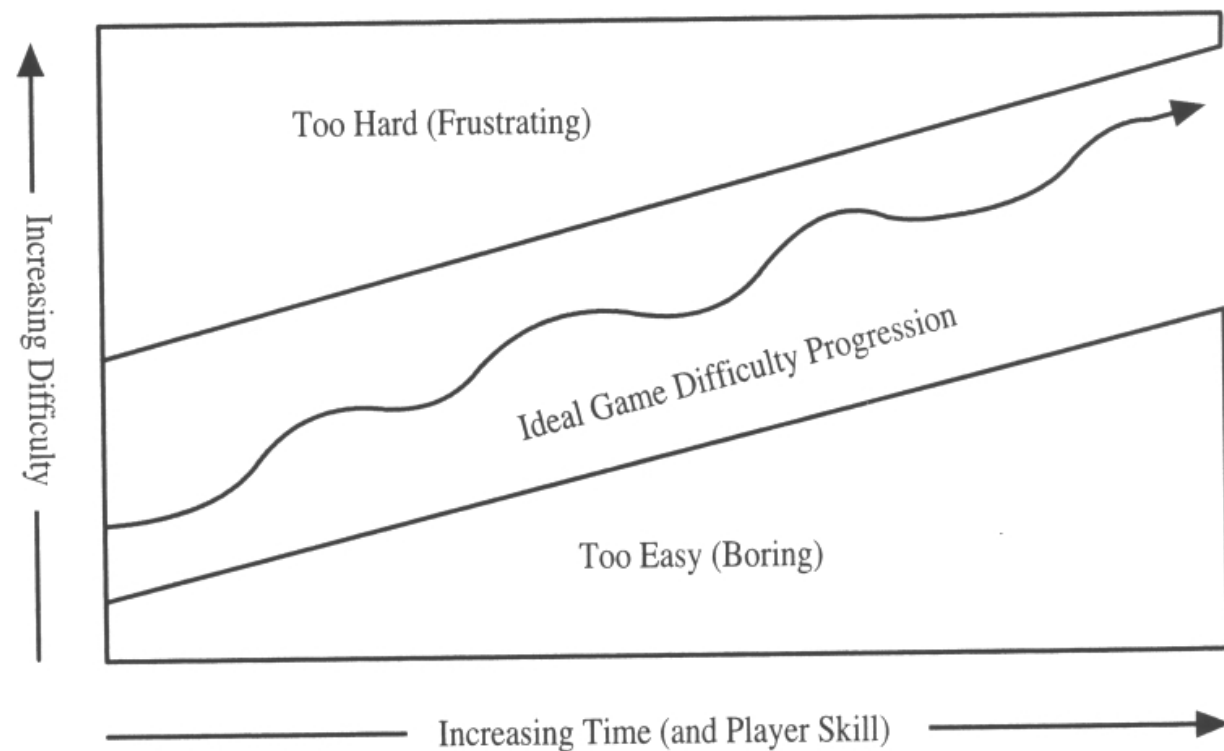
- Need to provide choices



**FIGURE 2.1.6** *A series of convexities.*

## Flow

- Getting the balance right is the key to success



M. Csikszentmihalyi,  
"Flow, The Psychology of  
Optimal Experience"

FIGURE 2.1.8 *A better flow.*

Chapter 2.1, *Introduction to Game Development*

# Convexity + Flow

- Utilizing both can lead to a great game

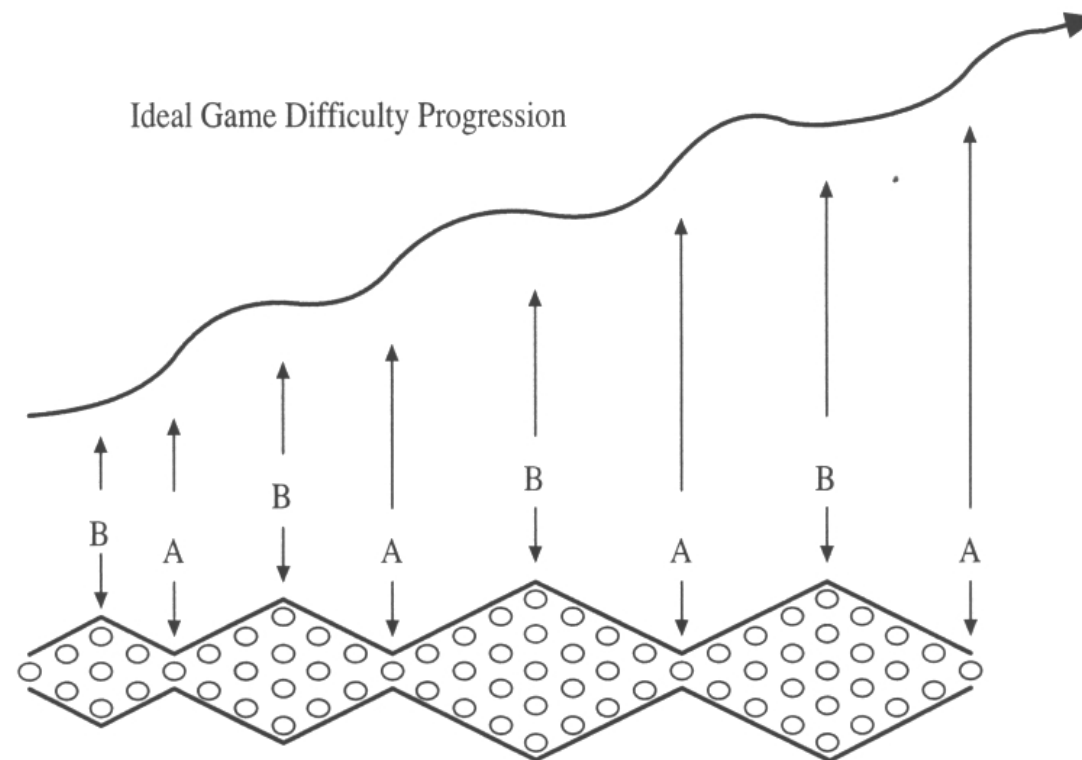


FIGURE 2.1.9 *Better flowing through convexities.*

## Other Thoughts

---

- Theatre:
  - Show, Don't Tell
  
- Games
  - Do, Don't Show
  
- Hal Barwood on Cut Scenes
  - Cut, edit, and cut some more until the writing is just as brief and concise as possible. At that point, the scene is probably about twice as long as it should be.