© Nintendo

CS 672: Spring 2010

Game Programming and Design

Introduction

What is this seminar about?

- Game design
  - Real world abstractions
    - Visuals
    - Interaction
- Design iterations
- Gameplay mechanics
- Rapid prototyping
- Many examples
- **FUN !**

Life meter (Castlevania)

Power ups (Super Mario Bros.)

Bosses (R-Type)

Playing with power
www.1up.com/do/feature?cId=3151392

About myself

- Andrew Nealen
  nealen@cs.rutgers.edu
  http://www.cs.rutgers.edu/~nealen

- Assistant professor of computer science
- Core team member at Hemisphere Games
- Research and teaching
  - Computer graphics + interactive shape modeling
  - Video game design and programming

Agenda

- Definition of a game and game design
- Game design in small teams
- Quick and dirty: rapid prototyping
- Abstraction of the “real world”
- Game “feel”: principles of virtual sensation
- Games as systems/simulations in software
- Prototype design (+ programming) strategies
- Indie Games: what is this all about?

Course structure

- Teams of 3-4 people (start forming this week)
- Weekly (mandatory) meetings
  Thursday noon-3pm, CBIM seminar room
- You will need **at least** two full days outside of class to work on games (likely more)
- First 4-5 weeks: one new game prototype per person/week
- Rest: each team fully develops their favorite and most successful prototype

Requirements

- Most importantly: **time**
- Almost as important: **dedication**
- It would be beneficial to have experience in one or more of the following areas
  - Graphics programming (OpenGL, etc.)
  - General programming and systems design
  - Classical art and painting/sketching/animation
  - Sound and music design
  - Organizational and team leading capabilities

Tools?

- I'm leaving this up to you
- Possibilities
  - Unity 3D
  - Flixel (Flash AS3 libraries)
  - XNA (http://creators.xna.com)
  - Straight up OpenGL and C/C++ or Java
  - SDL (http://www.libsdl.org/)
  - 2D boy framework (http://2dboy.com/2009/05/27/rapid-prototyping-framework/)
  - Whatever works…

Platforms?

- Again, whatever you like
  - PC / Mac
  - Xbox 360 (via XNA)
  - iPhone

- You will need to take care of your own hardware, and be able to present prototypes in class on a weekly basis

What is a game?

And what is game design?

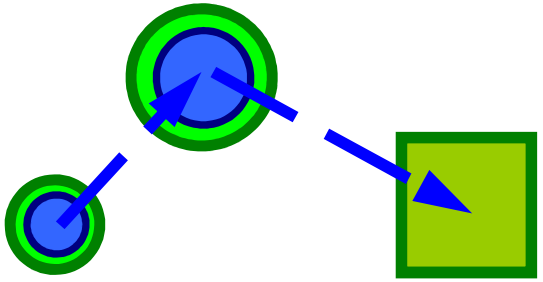
What is a game?

- Many definitions exist
- Common in many definitions:
  - Participants (players)
  - Decisions (interactions)
  - Conflicts (opposition)
  - Resource management (game tokens)
  - Pursuit of a goal (or goals)

  - (and all of this in a closed, formal system)

Aspects of video game design

- What is video game design?
  - Player mechanics and controls?
  - Game rules, dynamics and goals?
  - World and level design?
  - Choice of colors, icons and setting?
  - Interactive sound and music?
  - Intuitive tools and code for designers and coders?
  - Enemy/ally intelligence and behavior?



Aspects of video game design

- Answer:

All of the above (and more)

- Video game design is inherently interdisciplinary
- Try to learn as much as you can about all aspects of a video game, and you will be a better video game designer

Game design

- A form of modern day **Alchemy**
  - If we knew the formula for a great game we would always use it
  - Alchemy has advanced to chemistry, so we can advance to a science too, right ?
- Define rules / guidelines of the game
- Design interesting **Interaction**
  - Risk and reward (e.g. interesting decisions for player)
  - Advance the „plot“ (score, level, skill, narrative, etc.)
  - Keep the player challenged, not frustrated

Fundamental meaningful play

- Descriptive
  - Relationship between player action and system outcome
- Evaluative
  - **Discernable**
    - Perceive the immediate outcome of player action
    - Explosion, sound effect, game state change
  - **Integrated**
    - Outcome of an action is woven into the game system
    - Actions on earlier „levels“ influence gameplay later on

Game Design Teams

For small/medium sized game projects

Small design teams

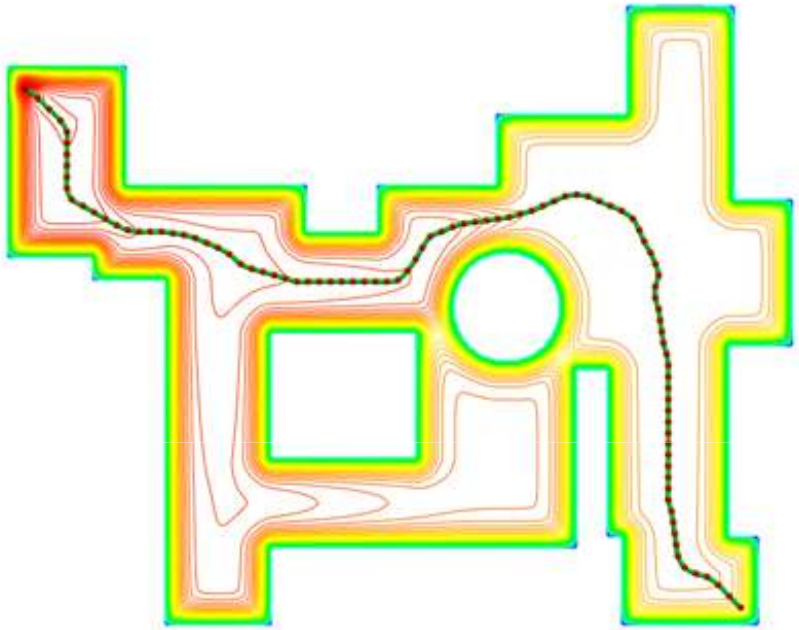
- **Team lead / design**
  - Organize team
  - Schedule milestones
  - Iterate game design and game rules
  - Study game theory etc. See „Rules of Play“, „Theory of fun“ **www.theoryoffun.com**, and „The Art of Game Design“

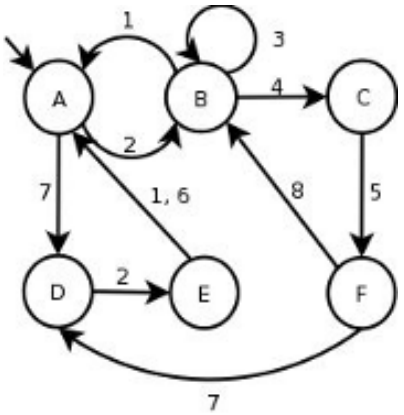
www.experimentalgameplay.com

www.2dboy.com

Small design teams

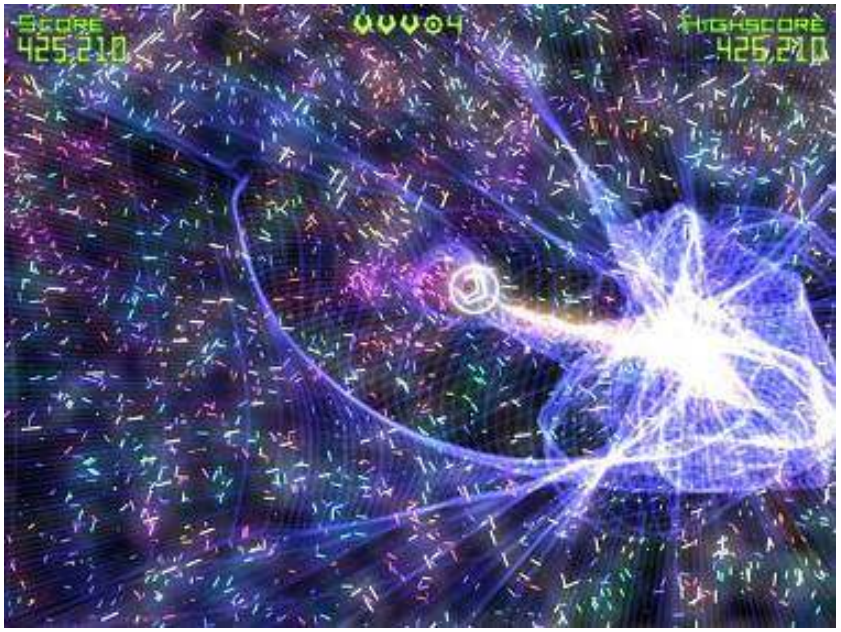
- **Gameplay programmer**
  - The „AI“ of the game: generally refers to rule-based aspects
  - The software engineer
  - Encode „behaviors“, path planning, state machines, etc.
  - Works closely with lead game designer





Small design teams

- **Graphics programmer**
  - Rendering technology
  - Work with content creator on „procedural content“
  - Interface with gameplay programmer (collision detection, events, etc.)
  - Polish the visuals with particle effects, etc.



© Bizarre Creations

Small design teams

- **Content creator**
  - 2D Sprites (Photoshop)
  - 3D Models
  - Textures, Normalmaps
  - 2D/3D animations
  - Sound design
- Role of an art director
  - Coherent look-and-feel
  - Adapt style to game mechanics and „story“

God of War™, © SCEA

Small design teams

- None of these roles are carved in stone
  - The previous example is one way of splitting work among a group of four
  - Many successful projects have been completed by groups of two, and also solo developers
    - Tip/trick: realistically adjust your design ambitions to the expected man-hours you will be able to invest

Resources

Stand on the shoulders of giants

Resources

- Game development / careers
  - **gamecareerguide.com,gamasutra.com, gamedev.net,igda.com**
- Game news
  - **kotaku.com,1up.com,gamespot.com**
  - **tigsource.com,indiegames.com/blog/**
- Podcasts
  - **1up.com,gamespot.com, brainygamer.net**

Resources

- Books
  - Fullerton, **Game Design Workshop**
  - Norman, **The Design of Everyday Things**
  - Swink, **Game Feel**
  - Salen/Zimmerman, **Rules of Play**
  - Schell, **The Art of Game Design**
  - Koster, **A Theory of Fun**
- Research
  - Jesper Juul, **`www.jesperjuul.net`**
  - Game studies, **`www.gamestudies.com`**

Homework

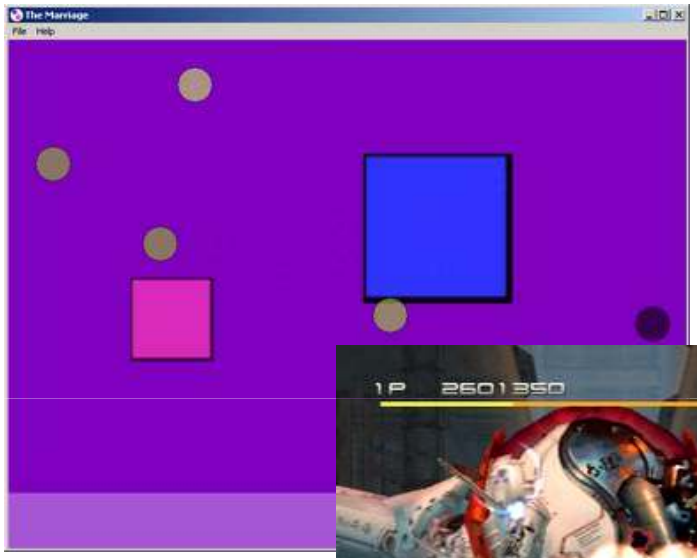
Your very first prototype

For next week…

- Each participant
  - Select a toolset, and make a very small game!
- Constraints
  - No sound or music whatsoever
  - Only art assets (bitmaps) allowed:
    circles and squares
    what you create in code is up to you…
  - Input: at most **four** keys on a keyboard, or **one** thumbstick and **one** button, or simple motion controls **or** taps on a touch-screen (not both)

Some Inspirations

- The Marriage
  - `http://rodvik.com/rodgames/marriage.html`
  - Simple game rules + simple rendering
  - Interesting mechanics
- Ikaruga (shooter)
  - Simple polarity principle
  - Looks beautiful, but would be equally playable if not



Tips

- Keep it VERY simple!
  - You only have one week for everything
- Code quick and dirty
  - You will not be using this code ever again
- Quickly converge on a toolset + platform
  - Based on previous experience, preference, methods of distribution, etc.
- Have friends play test your game
  - This will always be helpful throughout the course

From *The design of everyday things*

- Design must convey the **essence of a device's operation**; the way it works; the possible actions that can be taken; and through **feedback**, just what it is doing at any particular moment

- **When people have trouble with something, it isn't their fault—it's the fault of the design**

From *The design of everyday things*

- The surest way to make something easy to use, with few errors, is to make it impossible to do otherwise—to **constrain the choices**

- A good designer makes sure that **appropriate actions are perceptible** and **inappropriate ones are invisible**