Analyzing Level Design: A Genre-Specific Approach

Jim Whitehead
With Gillian Smith, Ken Hullett, Michael Mateas, Noah Wardrip-Fruin, Martin Jennings-Teats, Mike Treanor, and Mee Cha

University of California, Santa Cruz
Dept. of Computer Science
ejw@cs.ucsc.edu

DePaul University, November 12, 2010
Univ. of California, Santa Cruz

- One of 10 Univ. of California campuses
Univ. of California, Santa Cruz

- 14,600 undergraduate students
- 1,025 graduate students

Engineering building with Donkey Kong made from lots of Post-It notes

Looking over Santa Cruz from UCSC

Campus mascot: banana slug
Game related research at UC Santa Cruz

- Faculty

- Students
  - Large research group with ~30 graduate students, primarily PhD

- Research
  - Interactive narrative, procedural level design and quest generation, software and media studies, case-based RTS AI, game software engineering, interactive camera control, natural language dialog systems
  - Center for Games and Playable Media

- Web
  - Description of research group and activities
    - games.soe.ucsc.edu
  - Group blog
    - eis-blog.ucsc.edu

Façade
www.interactivestory.net

GAMES & PLAYABLE MEDIA
UC SANTA CRUZ
UCSC Undergraduate Computer Game Design Major

- **Senior Game Design Studio**
  - Work as a member of a team for an entire year to develop a substantial computer game.

- **Game Design Experience**
  - Intro. game design Freshman year
  - Intro. game programming in Sophomore year

- **Solid grounding in Computer Science**
  - 3 course sequence in graphics
    - Includes 3D game engine design
  - 3 course sequence in AI
    - Includes game AI, and narrative AI
  - Built on top of proven CS fundamentals curriculum

- **Digital Media**
  - 2 digital media electives

- **Student enrollments**
  - ~100 freshmen/year
  - ~350 students in major
Foundations of Digital Games Conference (FDG)

- A “big-tent” conference on computer games research
- Seeks to promote the exchange of information concerning
  - Scientific foundations of digital games
  - Technology used to develop digital games
  - Study of digital games and their design, viewed broadly
- Run by Society for Advancement of the Science of Digital Games
  - California-based nonprofit corporation
- FDG 2011
  - June 28-July 1, 2011, Bordeaux, France
  - General Chair: Marc Cavazza, Univ. Teeside, UK
  - Program Co-Chairs: Katherine Isbister, NYU Poly; Charles Rich, Worcester Poly.
  - **Call for papers deadline: 10 February 2010**
  - ACM In-cooperation sponsorship pending
Genre-specific level design
Current State of Level Design

- Designers build levels from experience and intuition
  - A craft tradition
  - Level designers understand their craft, but have generally not written down their design principles or approach. The consequence:
    - No formal understanding of level design
    - Limited language for communication about level design
- No formal understanding of how level design creates gameplay
  - Cause-effect relationships are not widely known

Level from Sonic the Hedgehog
Current approach for teaching level design

- Focus on creating 3D first-person camera games
  - FPS, action-adventure
  - Shallow discussion of other genres
  - Use of a specific level design tool (UnrealEd, Far Cry editor)
- Emphasis on **one-size fits all** approach to teaching level design
  - High level issues or qualities
    - Level narrative, use of puzzles (but not how to design a puzzle), pacing and flow, challenge, goal, rewards
  - Some discussion of different coarse-grain level geometries
    - Hub-and-spoke, linear, branch-and-join
Underlying assumption: principles are sufficient

- Game design is currently taught with an emphasis on teaching broad, universal principles
  - Can see this in *Rules of Play, Fundamentals of Game Design, Chris Crawford on Game Design*
  - Use principles as a lens for thinking about specific game design
  - Useful

- Level design books have a similar assumption
  - For level design, this approach breaks down.
  - Game genres are just too different!
  - Principles-based approach isn’t sufficiently detailed to provide useful guidance for level design.
Genre-Specific Level Design

- Compelling, detailed explanations of how to perform level design require analytical approaches tailored to specific game genres.
Three Examples of Genre-Specific Level Analysis

- 2D platform games
- 2D space shooters
- 3D first person shooters
- An examination of these three will show how each benefits from a genre-specific analytic approach
2D Platformers: Rhythm
2D Platformers

- Simple rules
  - Run, jump
  - Collect items
  - Get to the goal

- Complexity
  - Dexterity challenge
  - Find secret areas
  - Finish levels quickly

- Key Games
  - Super Mario World
  - Donkey Kong Country 2
  - Sonic the Hedgehog
## Level Structure

- **Rhythm Groups**
  - Short, non-overlapping sections of the level
  - Encapsulate challenge

- **Cells**
  - Linear sections of gameplay
  - Contain rhythm groups

- **Portals**
  - Connect cells
  - Provide multiple paths through a level
Level Components

- **Platforms**
  - Player runs along them.

- **Obstacles**
  - Cause damage to player.

- **Collectible Items**
  - Provide reward to player.

- **Triggers**
  - Cause change in level.

- **Movement Aids**
  - Help player through the level.

Examples of level components in various video games:
- Super Mario World, Nintendo
- Sonic the Hedgehog 2, SEGA
- Yoshi's Island DS, Nintendo
- New Super Mario Bros., Nintendo
- Yoshi's Island DS, Nintendo
Rhythm Groups: Why Rhythm?

- Foundation of challenge in dexterity games
  - Long sequences without pause
  - Long and complex patterns
  - Reduced time to complete a challenge

Source: Victor Nicollet, “Difficulty in Dexterity-Based Platform Games”, GameDev.net
Rhythm Groups

- Rhythm is that of player performing actions
- Identify challenging areas of a level
- Transitions are place where player can rest

Super Mario World, Nintendo
Cells and Portals

- **Cells**
  - Path within the game
  - Made up of rhythm groups

- **Portals**
  - Connect cells
  - Player makes a choice

![Super Mario World, Nintendo](image)
Case Study

- Cells contain rhythm groups
- Portals connect cells
- Cells and portals let player choose the best path
2D Space Shooters (shmups):
Safe and unsafe space
Leading the player
Level design principles for Shmups

- Safe and unsafe spaces
  - Via walls and the location/trajectory of enemies and bullets
  - Create zones of (relative) safety and danger

- Leading the player
  - Via the placement of enemies and rewards (powerups)
  - Cause the player to move to specific places on screen
  - Can lure the player into spaces that seem safe (but won’t be by the time they arrive)
Leading: Player is attracted to powerup, wants to move there.
Gradius III

Leading: Player has gone for powerup.
Safe and unsafe: safe zone is contracting rapidly, and enemy trajectory will go through powerup
Gradius III

**Leading:** Player narrowly avoids powerup trap, moves to engage line of enemies coming in from right.

**Safe and unsafe:** escaping into bottom of screen safe zone
Safe & unsafe: A line of bullets slices through area in front of powerup, in case player was tempted by a quick snatch. Fast incoming enemies keep player pinned down at bottom in narrow safe zone.
Safe and unsafe: By clearing the line of enemies, the player buys a little time and safety.
Averaging Gradius Video

- 15 gameplay sessions
- Superimposed on one another, with common time base
- Each player has a different color
- Gives sense of “average” player behavior
- Credit: R. LeFeuvre
  http://thenewgamer.com/content/archives/averaging_gradius
First-Person Shooters: Composition of Level Design Patterns
Design Patterns

Architecture, Urban Planning
Alexander 1977

Software Engineering
Gamma et al. 1995

Game Design
Björk & Holopainen 2005
Level Design Patterns

- Recurring arrangements of elements
  - Geometry
  - NPCs
  - Items

- Can be identified in many different games

- Similar effects on player behavior
  - Changing design patterns can change gameplay

- Provide language for describing levels

Halo 3 - Bungie
Pattern Collection

- Patterns for Positional Advantage
  - Sniper location, Gallery, Choke point
- Patterns for Large-scale Combat
  - Arena, Stronghold
- Patterns for Alternate Gameplay
  - Turret, Vehicle section
- Patterns for Alternate Routes
  - Split level, Hidden area, Flanking route
Sniper Location : Halo 3
Sniper Location: Half-Life 2
Pattern Example – Sniper Location

- Description: A position that overlooks some other section of the level
- Affordances:
  - Player v. Enemy
  - Height
  - Size
  - Area overlooked
  - Cover available
  - Access
- Consequences:
  - Enemy: Slow movement, use cover
  - Player: Slow movement, use long range weapon
Procedural Generation of 2D Platform Levels: Making Theories of Level Design Operational
Why Procedural Level Design?

- Theoretical
  - Procedural level designer tools take theories of level design and make them operational – theory testing

- Design
  - Increase replayability of games by creating unique content for each play session
  - Can have very large game worlds via automatic generation of forests, towns, dungeons, etc.

*Level from Super Mario World*
Launchpad: Use model

- Level designer provides as input:
  - Control line
    - Indicates rough preference for slope of level
  - Component frequencies
    - How frequently there are items like springs, moving platforms, enemies, thwompers, etc.

- Level generator outputs:
  - Randomly generated level that is the best match
  - Generate 1000 levels, toss all but one (generate and test)

- No possibility of further modification by designer
  - Generate-and-play
Generation Algorithm Overview

Rhythm Group

- Rhythm
- Physics
- Geometry

Candidates...

Critics

- Base Level
- Coin Decoration
- Final Level

Style
Generation Algorithm Overview

- Rhythm
- Physics
- Geometry
- Critics
- Base Level
- Coin Decoration
- Final Level

- Style

- Candidate
- Candidate
- Candidate
- Candidate
Rhythm

- **Type**
  - Regular
  - Random
  - Swing

- **Length**
  - 5 seconds
  - 10 seconds
  - 15 seconds
  - 20 seconds

- **Density**
  - low
  - medium
  - high
# Rhythm

- **Type**
  - Regular
  - Random
  - Swing

- **Length**
  - 5 seconds
  - 10 seconds
  - 15 seconds
  - 20 seconds

- **Density**
  - low
  - medium
  - high

<table>
<thead>
<tr>
<th>Action</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>jump</td>
<td>2</td>
<td>2.25</td>
</tr>
<tr>
<td>move</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>jump</td>
<td>8</td>
<td>8.75</td>
</tr>
</tbody>
</table>
### Rhythm

**Type**
- Regular
- Random
- Swing

**Length**
- 5 seconds
- 10 seconds
- 15 seconds
- 20 seconds

**Density**
- low
- medium
- high

<table>
<thead>
<tr>
<th>Action</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>jump</td>
<td>2</td>
<td>2.25</td>
</tr>
<tr>
<td>move</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>jump</td>
<td>8</td>
<td>8.75</td>
</tr>
</tbody>
</table>

![Timeline diagram showing start and stop times for different actions.](image-url)
Rhythm

- **Type**
  - Regular
  - Random
  - Swing

- **Length**
  - 5 seconds
  - 10 seconds
  - 15 seconds
  - 20 seconds

- **Density**
  - low
  - medium
  - high

<table>
<thead>
<tr>
<th>Action</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>jump</td>
<td>2</td>
<td>2.25</td>
</tr>
<tr>
<td>move</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>jump</td>
<td>8</td>
<td>8.75</td>
</tr>
</tbody>
</table>
Rhythm

- **Type**
  - Regular
  - Random
  - Swing

- **Length**
  - 5 seconds
  - 10 seconds
  - 15 seconds
  - 20 seconds

- **Density**
  - low
  - medium
  - high

<table>
<thead>
<tr>
<th>Action</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>jump</td>
<td>2</td>
<td>2.25</td>
</tr>
<tr>
<td>move</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>jump</td>
<td>8</td>
<td>8.75</td>
</tr>
</tbody>
</table>
Physics: From Rhythm to Geometry

- Physics Model – ensures levels are playable!
  - Avatar physics (size, velocity, jump heights...)
  - Geometry constraints (jump time in air)
- Turns “player actions” into “avatar states”
  - Geometry dependent

<table>
<thead>
<tr>
<th>Time (seconds)</th>
<th>State</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Moving</td>
<td>2</td>
</tr>
<tr>
<td>1.5</td>
<td>Jumping</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>Moving</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>Moving</td>
<td>1.5</td>
</tr>
<tr>
<td>4</td>
<td>Stop</td>
<td></td>
</tr>
</tbody>
</table>
Geometry Types

- **Moving**
  - Flat Platform
  - Sloped Up
  - Sloped Down

- **Waiting + Moving**
  - Stomper

- **Waiting + Moving + Waiting**
  - Horizontal moving platform
  - Vertical moving platform
    - Going Up
    - Going Down

- **Jumping**
  - Flat Gap
  - Up (no gap)
  - Up (gap)
  - Down (no gap)
  - Down (gap)
  - Enemy (kill)
  - Spring
  - Fall
## Rhythm Group Example

<table>
<thead>
<tr>
<th>Rhythm</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0-8</td>
</tr>
<tr>
<td>jump</td>
<td>2-2.25</td>
</tr>
<tr>
<td>jump</td>
<td>4-4.25</td>
</tr>
<tr>
<td>jump</td>
<td>6-6.50</td>
</tr>
<tr>
<td>move</td>
<td>10-12</td>
</tr>
<tr>
<td>move</td>
<td>14-20</td>
</tr>
<tr>
<td>jump</td>
<td>16-16.25</td>
</tr>
<tr>
<td>jump</td>
<td>18-18.25</td>
</tr>
</tbody>
</table>

![Timeline Diagram]
Rhythm Group Example

### Rhythm

<table>
<thead>
<tr>
<th>Rhythm</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>jump</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
</tr>
</tbody>
</table>

### Timeline

```
start  jump  jump  jump  stop  start  stop  start  jump  jump  stop
```

0  2  4  6  8  10  12  14  16  18  20
### Other Geometry Interpretations

#### Rhythm

<table>
<thead>
<tr>
<th>Action</th>
<th>Time 1</th>
<th>Time 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>jump</td>
<td>2</td>
<td>2.25</td>
</tr>
<tr>
<td>jump</td>
<td>4</td>
<td>4.25</td>
</tr>
<tr>
<td>jump</td>
<td>6</td>
<td>6.50</td>
</tr>
<tr>
<td>move</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>move</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>jump</td>
<td>16</td>
<td>16.25</td>
</tr>
<tr>
<td>jump</td>
<td>18</td>
<td>18.25</td>
</tr>
</tbody>
</table>

#### Timeline

```
0 2 4 6 8 10 12 14 16 18 20
start jump jump jump stop start stop start jump jump stop
```

![Graph showing timeline and movements](image-url)
Other Geometry Interpretations

<table>
<thead>
<tr>
<th>Rhythm</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>jump</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>jump</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>jump</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>move</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>jump</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>jump</td>
<td>16</td>
<td>18</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>jump</td>
<td>18</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>
Other Geometry Interpretations

**Rhythm**

<table>
<thead>
<tr>
<th>Action</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jump</td>
<td>2</td>
<td>2.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jump</td>
<td>4</td>
<td>4.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jump</td>
<td>6</td>
<td>6.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>move</td>
<td>10</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>move</td>
<td>14</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jump</td>
<td>16</td>
<td>16.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jump</td>
<td>18</td>
<td>18.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Timeline**

```
0 2 4 6 8 10 12 14 16 18 20
start jump jump jump stop start stop start jump jump jump stop
```
Other Geometry Interpretations

<table>
<thead>
<tr>
<th>Rhythm</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>move 0</td>
<td>0</td>
</tr>
<tr>
<td>jump 2</td>
<td>2</td>
</tr>
<tr>
<td>jump 4</td>
<td>4</td>
</tr>
<tr>
<td>jump 6</td>
<td>6</td>
</tr>
<tr>
<td>move 10</td>
<td>10</td>
</tr>
<tr>
<td>move 14</td>
<td>14</td>
</tr>
<tr>
<td>jump 16</td>
<td>16</td>
</tr>
<tr>
<td>jump 18</td>
<td>18</td>
</tr>
</tbody>
</table>

0 2 4 6 8 10 12 14 16 18 20

Timeline:

- start
- jump
- jump
- jump
- stop
- start
- stop
- start
- jump
- jump
- stop
Generation Algorithm Overview

- **Rhythm Group**
  - Rhythm
  - Physics
  - Geometry
  - Candidate
  - Candidate
  - ... (multiple candidates)

- **Critics**
  - Base Level
  - Coin Decoration
  - Final Level

- **Style**

Diagram shows the flow of concepts from Style through Rhythm to Physics, then to Geometry, with multiple candidate options along the way, leading to Critics, Base Level, Coin Decoration, and finally, the Final Level.
Style: Human Designer Control

- Series of “style knobs” presented as probabilities that influence the generator
- Control line
- Verified in generate-and-test by critics
- Open Question: What do designers really want control over?
Generation Algorithm Overview

Rhythm Group

- Rhythm
- Physics
- Geometry

Candidates

- Candidate
- Candidate
- Candidate

Critics

- Style

Base Level

- Coin Decoration

Final Level
Generate and Test

- Level: a set of non-overlapping rhythm groups
Critic Measures

- **Line Distance**
  - Best fit to a control path
  - Minimize distance of platforms from line

- **Component Style**
  - Style parameters for component probabilities
Generation Algorithm Overview

Rhythm Group

- Rhythm
- Physics
- Geometry

- Candidate
- Candidate
- Candidate
- Candidate

Critics

Base Level

Coin Decoration

Final Level
Coin decoration

- Coin decoration
  - Over gaps to guide the player
  - Along otherwise ‘boring’ long platforms
  - Performed after creation of geometry
Demonstration of generated levels
Observations

- **Complexity**
  - Generated levels do not have the same level of complexity and interest as existing hand-authored levels
  - Missing:
    - Secret areas
    - Complex interactions between geometry and enemies
    - Puzzles
    - Areas where player reverses direction
    - Elements of surprise, delight
    - These suggest areas for further refinement of level design theory

- **Artwork**
  - Not a research focus, but clearly not at professional levels
Questions?