SCALABLE LEVEL GENERATION FOR 2D PLATFORMING GAMES

NEALL DEWSBURY, JAMES TATUM, AIMIE NUNN
MATTHEW SYRETT & TOMMY THOMPSON
MANAGING PCG 'SCALE'

• PCG: A POWERFUL TOOL CAPABLE OF A VARIETY OF INTERESTING CONTENT.

• THE ASSUMPTION OF 'WE CAN HAZ ALL THE CONTENT' CAN BE RATHER NAÏVE.

• KEY ELEMENTS TO APPROACH WITH EACH SYSTEM:
  • VALIDITY OF CONTENT.
  • QUALITY OF CONTENT.
  • THE KNOCK-ON EFFECT OF CONTENT INTEGRATION.
  • WHETHER CONTENT IS 'SCALED' TO THE PLAYERS EXPERIENCE.
VALIDITY AND QUALITY
VALIDITY AND QUALITY
• **IN COMPETITIVE AND SKILL-BASED GAMES, OUR GENERATED CONTENT SHOULD SCALE AS PLAYERS PROGRESS.**

• **AS CONTENT SCALES UPWARDS, IT SHOULD REFLECT THE CHALLENGES THE PLAYER FACES AND THE OVERALL DIFFICULTY OF THE CURRENT EXPERIENCE.**

• **INVESTIGATE A METHOD FOR MANAGING SCALE THROUGH BUDGET-CONSTRAINED GENERATIVE SYSTEMS.**
  • I.E. **WE IMPOSE LIMITS ON EXPRESSIVITY IN THE GENERATIVE SYSTEM PRIOR TO PLAYER METRICS.**

• **APPLY THIS PROBLEM WITHIN AN INFINITE RUNNER.**

**SCALE -> CHALLENGE -> PROGRESSION**
INFINITE RUNNERS

- Platforming game that continually expands the longer you survive.
  - Randomly generated
SURE FOOTING
SURE FOOTING
RELATED WORK
OUR INSPIRATIONS
• **RHYTHM-DRIVEN GENERATION**
  • LAUNCHPAD (SMITH 2009)
  • CONCEPTUALISED THE ACTION SPACE PRIOR TO GENERATION.
  • GRAMMAR-DRIVEN METHODS.

• **DECOUPLING FUNCTION FROM CONSTRUCTION.**
  • DORMAN (2010) IN LEGEND OF ZELDA.
  • LAVENDER & THOMPSON (2015, 2016) IMPLEMENTED THIS IN A PLAYABLE VERSION.
DESIGN PATTERNS

• **DAHLSKOG ET AL. MARIO PATTERN GENERATORS**

• **ENCAPSULATES ASPECTS OF GAME DESIGN AS 'PATTERNS'**.

• **RANGING FROM MACRO TO MICRO LEVELS**.

Steve Dahlskog, Julian Togelius and Mark J. Nelson (2014)
Linear levels through n-grams.

Steve Dahlskog and Julian Togelius (2014)
A multi-level level generator

Steve Dahlskog and Julian Togelius (2014)
Procedural content generation using patterns as objectives.
BUDGET-CONSTRAINED LEVEL GENERATION

A QUICK OVERVIEW OF THE GENERATIVE SYSTEM
SPRINT GENERATION FRAMEWORK

GAME

PLAYER PROGRESS

LEVEL GENERATOR

Action Generator

Budget & Constraints

Action Sequence

Geometry Generator

Generated Platforms

Game Assets

Grammar

Actions

Playable Sequence
ACTION GENERATORS

• **FIXED GRAMMARS THAT GENERATIVE GRAMMAR APPROACH:**
  • TERMINAL SYMBOLS INDICATIVE OF:
    • ACTIONS FOR PLAYER.
    • DESIGN PATTERNS FOR LEVEL CONSTRUCTION.

• **EACH GRAMMAR DEFINES ITS OWN:**
  • SET OF PRODUCTION RULES.
  • NON-TERMINAL SYMBOLS.

• **BUDGET FACTORS:**
  • COST OF USING EACH RULE.
## Action Generators

### The ‘Safe-Random’ Grammar:

<table>
<thead>
<tr>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow rS$</td>
</tr>
<tr>
<td>$S \rightarrow tpA$</td>
</tr>
<tr>
<td>$S \rightarrow ruA$</td>
</tr>
<tr>
<td>$S \rightarrow spA$</td>
</tr>
<tr>
<td>$S \rightarrow hpS$</td>
</tr>
<tr>
<td>$S \rightarrow sdS$</td>
</tr>
<tr>
<td>$A \rightarrow rA$</td>
</tr>
<tr>
<td>$A \rightarrow sA$</td>
</tr>
<tr>
<td>$A \rightarrow \epsilon$</td>
</tr>
</tbody>
</table>

### The ‘Height Intensity’ Grammar:

<table>
<thead>
<tr>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow GF$</td>
</tr>
<tr>
<td>$G \rightarrow suG$</td>
</tr>
<tr>
<td>$F \rightarrow sdF$</td>
</tr>
<tr>
<td>$G \rightarrow \epsilon$</td>
</tr>
<tr>
<td>$G \rightarrow hpG$</td>
</tr>
<tr>
<td>$F \rightarrow rdF$</td>
</tr>
<tr>
<td>$F \rightarrow \epsilon$</td>
</tr>
<tr>
<td>$F \rightarrow rF$</td>
</tr>
<tr>
<td>$G \rightarrow spG$</td>
</tr>
<tr>
<td>$G \rightarrow ruG$</td>
</tr>
</tbody>
</table>
GEOMETRY GENERATORS

- Translates the action sequence into a playable gameplay *sprint*.

- Interprets action string using their own built-in model of level generation.

- Budget interprets the 'cost' of each action in-game.
GEOMETRY GENERATORS

(a) Budget Cost: 1

(b) Budget Cost: 2

(c) Budget Cost: 4

(d) Budget Cost: 5
Analysis

We assessed a specific configuration of the system:

Four action generators:
- Random
- 'Safe-Random'
  - Random selection with some constraints.
- Intensity
  - Actions are modelled based on (Smith & Whitehead, 2010) metric.
- Height-Intensity
  - Action costs modelled w.r.t. vector between start and end-points.

One geometry generator:
- Prefab generator with hand-crafted segments.

“Analyzing the Expressive Range of a Level Generator.”
Gillian Smith and Jim Whitehead
ANALYSIS

• **ASSESSED UNDER THREE CONFIGURATIONS OF THE SYSTEM:**

  • **FIXED ACTION & GEOMETRY BUDGET.**
    • FOCUSED SOLELY ON VARIETY OF ONE BUDGET SETTING.

  • **FIXED ACTION BUDGET, INCREASING GEOMETRY BUDGET.**
    • INDICATING VARIETY IN SPRINT CONSTRUCTION WHEN ACTION BUDGET IS FIXED.

  • **INCREASING ACTION BUDGET, FIXED GEOMETRY BUDGET.**
    • MORE ACTIONS PERMITTED BUT GEOMETRY CAPPED TO BASIC INTERPRETATIONS.
FIXED ACTION & GEOMETRY BUDGETS

'RANDOM'

'SAFE-RANDOM'

'INTENSITY'

'HEIGHT INTENSITY'

LINEARITY & LENIENCY
FIXED ACTION BUDGET, INCREASING GEOMETRY BUDGET

'RANDOM'

'SAFE-RANDOM'

'INTENSITY'

'HEIGHT INTENSITY'

LINEARITY & LENIENCY
INCREASING ACTION BUDGET, FIXED GEOMETRY BUDGET

'RANDOM'

'SAFE-RANDOM'

'INTENSITY'

'HEIGHT INTENSITY'

LINEARITY & LENIENCY
FIXED ACTION & GEOMETRY BUDGETS

'RANDOM'

'SAFE-RANDOM'

'INTENSITY'

'HEIGHT INTENSITY'

LENGTH & VERTICALITY
FIXED ACTION BUDGET, INCREASING GEOMETRY BUDGET

LENGTH & VERTICALITY
INCREASING ACTION BUDGET, FIXED GEOMETRY BUDGET

'REANDOM'

'REIGHT INTENSITY'

'SAFE-RANDOM'

'HEIGHT INTENSITY'

LENGTH & VERTICALITY
CONCLUSIONS & FUTURE
FUTURE WORK

• RECENT WORK:
  • CONTINUED ANALYSIS OF THIS DATA SET.
    • PAPER (& TALK) ONLY CONSIDERS 4 METRICS.
      • LINEARITY, LENIENCY, VERTICALITY & LENGTH.
  • IN-GAME TOOLS RECORD ADDITIONAL METRICS:
    • ACTION VARIATION, ACTION DENSITY, GEOMETRIC DENSITY, BASE RHYTHM.
  • CURRENTLY LOOKING TO ADD MORE OF THE (CANOSSA & SMITH, 2015) METRICS TO THE GAME WHERE POSSIBLE.

“TOWARDS A PROCEDURAL EVALUATION TECHNIQUE: METRICS FOR LEVEL DESIGN”
ALESSANDRO CANOSSA GILLIAN SMITH
FUTURE WORK

• CURRENTLY IN DEVELOPMENT:
  • GEOMETRY GENERATORS:
    • ADOPTION OF SEARCH-BASED GENERATION FOR PLATFORM PLACEMENT.
    • PREFAB PLATFORM PLACEMENT.
    • CELLULAR-AUTOMATA AND L-SYSTEMS.
  • ADAPTIVE LEVEL GENERATION:
    • 20+ LEVEL GENERATION PARAMETERS.
    • PREFERENCE LEARNING VS. IN-GAME PERFORMANCE.

<DifficultySettings>
  <Difficulty>
    <DifficultyName>STROLLER</DifficultyName>
    <StartSpeedMultiplier>0.8</StartSpeedMultiplier>
    <ActionBudgetStart>5</ActionBudgetStart>
    <GeometryBudgetStart>5</GeometryBudgetStart>
    <ActionBudgetIncrease>1</ActionBudgetIncrease>
    <GeometryBudgetIncrease>1</GeometryBudgetIncrease>
    <DeletionWaveStart>0.2</DeletionWaveStart>
    <DeletionWaveIncreaseMin>0.02</DeletionWaveIncreaseMin>
    <DeletionWaveIncreaseMax>0.005</DeletionWaveIncreaseMax>
    <DeletionWaveDelay>2.0</DeletionWaveDelay>
    <ActionGenerators>
      <Generator>RANDOM</Generator>
      <Generator>SAFE_RANDOM</Generator>
    </ActionGenerators>
    <RelaxConstraints>false</RelaxConstraints>
    <Constraints>
      <Constraint>TwoPath</Constraint>
      <Constraint>StairUp</Constraint>
      <Constraint>Hopscotch</Constraint>
      <Constraint>RampUp</Constraint>
      <Constraint>RampDown</Constraint>
    </Constraints>
    <ObstaclesActive>false</ObstaclesActive>
    <ObstacleTypes/>
  </Difficulty>
</DifficultySettings>
FUTURE WORK

• CURRENTLY IN DEVELOPMENT:

• VALIDATION PLAYTHROUGHS:
  • DEVELOPING AGENTS TO PLAY THROUGH AND TEST GENERATED LEVELS.
  • USEFUL FOR TESTING GENERATION AND POTENTIAL TOOL FOR DESIGNERS AND PLAYERS.
FUTURE WORK

• LONG-TERM WORK:

  • PLAYER LOADOUT RECOMMENDATION:
    • 4 CHARACTERS, 9 POWER-UPS, 10 BUFFS, 10 COSTUMES EACH.
    • PREDICT 'GOOD' COMBINATIONS FOR IMPROVED PLAYER PERFORMANCE.

  • IDENTIFY 'BAD' LEVEL DESIGN:
    • COMPILe DATA ON PLAYER DEATHS.
    • EXTRAPOLATE WHETHER REGULAR GENERATED PATTERNS PROVE MORE FATAL THAN OTHERS.
THANK YOU!

TOMMY THOMPSON
@TABLEFLIPGAMES
@SUREFOOTINGGAME
@GET_TUDA_CHOPPA