Real-time Experience-driven Procedural Content Generation

Cameron Alston

University of California, Santa Cruz

calston@ucsc.edu

January 26th, 2015

Table of Contents

Motivation

- Games have become a huge part of today's culture
- Lots of different kinds of players
- Natural to assume that different players will play/experience/enjoy games differently

Motivation

- Games have become a huge part of today's culture
- Lots of different kinds of players
- Natural to assume that different players will play/experience/enjoy games differently

Problem

Today's games don't provide a different experience for individual players!

Motivation - Research Community

- New research area in experience-driven procedural content generation (ED-PCG) addresses this problem.
  - Combines player modeling, machine learning, and procedural content generation methods.
  - Until now, design changes happen offline.
  - Would like to see ED-PCG methods working online.

Motivation - Research Community

- New research area in experience-driven procedural content generation (ED-PCG) addresses this problem.
  - Combines player modeling, machine learning, and procedural content generation methods.
  - Until now, design changes happen offline.
  - Would like to see ED-PCG methods working online.

The primary aim of this research is to study whether experience-driven procedural content generation is useful for improving player experience in games in real-time.

Research Questions

Research Question 1

Does player experience modeling accurately reflect the experience the player while they are playing a game?

Research Question 2

Does player experience modeling give enough information about game content to inform the game's design in real time?

Research Question 3

Does procedurally generating game content in real-time via player experience modeling have the intended effect?

Table of Contents

Related Work

Several existing research areas that combine to form the work being proposed here:

- Player Modeling
- Procedural Content Generation (PCG)
  - Search-based PCG
  - Experience-driven PCG

Player Experience Modeling

Player experience modeling: creating computational models of players in games.

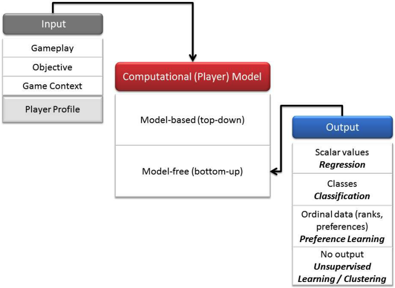


Figure: High level view of player modeling approaches [Yannakakis, 2013]

Player Experience Modeling

- **Subjective based PEM**: refers to the self reporting of player experience through either free-response or forced data collection retrieved through questionnaires.
- **Objective based PEM**: incorporates access to multiple modalities of player input for the purpose of modeling the affective state of the player during play.
- **Gameplay based PEM**: operates under the assumption that player actions and real-time preferences are related to player experience since games may affect the player's cognitive processing patterns and cognitive focus.

PCG Summary

Procedural content generation: methods concerned with the creation of content automatically, through algorithmic means.

PCG Summary

PCG methods can be classified in to several categories [Togelius, 2011]:

- Online vs. offline
- Necessary vs. optional
- Random seed vs. parameter vector
- Stochastic vs. deterministic
- Constructive vs. generate and test

PCG Summary

PCG methods can be classified in to several categories [Togelius, 2011]:

- Online vs. offline
- Necessary vs. optional
- Random seed vs. parameter vector
- Stochastic vs. deterministic
- Constructive vs. generate and test

This Approach

Online, necessary, parameter vector, stochastic, generate and test.

PCG Related to this Work

- PCG genres most applicable to the work being proposed in this study:
  - Search-based PCG
  - Experience-driven PCG

Search-based PCG

Search-based procedural content generation (SB-PCG) is a special case of the generate-and-test approach to PCG, with the following qualifications [Togelius, 2011]:

- The test function grades content using one or a vector of real numbers (fitness function).
- Fitness is based on previously generated content instances.
- Aim is to produce new content with higher fitness.

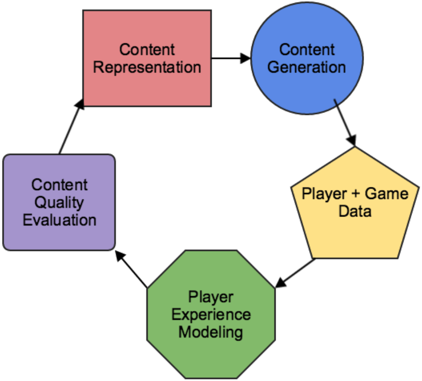
Experience-driven PCG

Experience-driven PCG (ED-PCG) is a field of game research dedicated to understanding what players want to see in a game at a particular time, and presenting that information/content to them [Yannakakis, 2011].

- Combines elements of player experience modeling and search-based procedural content generation research.
- Content seen as indirect building blocks of player experience.
- Content evaluated on the quality of the experience it generates.

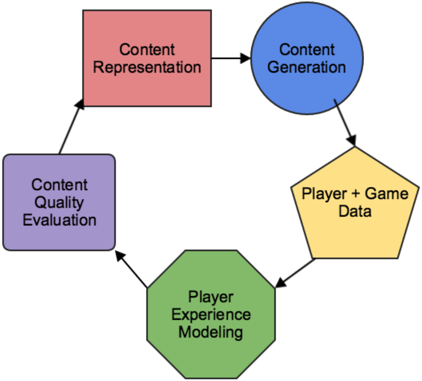
ED-PCG Components

- **Player + game data**: gameplay, objective, and/or subjective data is acquired from players.



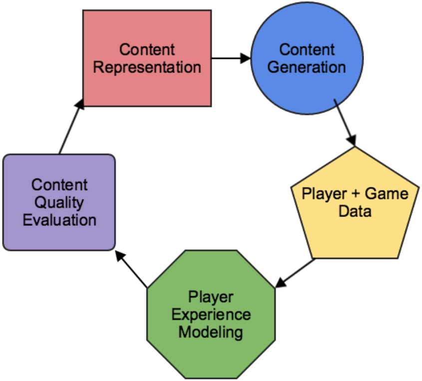
ED-PCG Components

- **Player experience modeling**: player experience is modeled as a function of game content and player behavior.



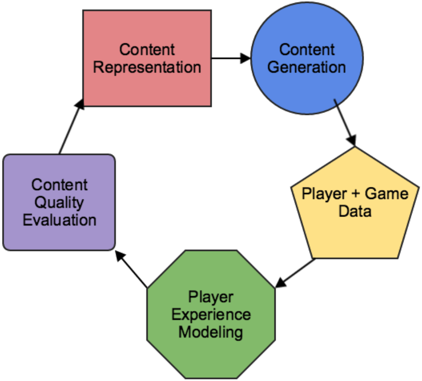
ED-PCG Components

- **Content quality**: content quality is assessed and linked to the modeled experience of the player.



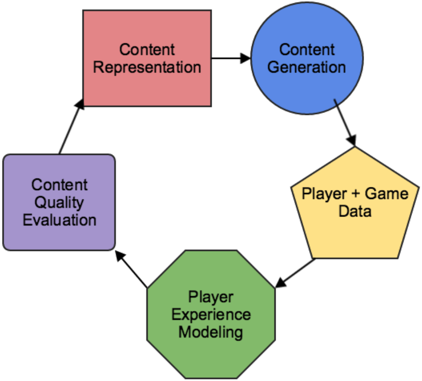
ED-PCG Components

- **Content representation**: content is represented accordingly to maximize efficacy, performance and robustness of the generator.



ED-PCG Components

- **Content generator**: the generator searches through the content that optimizes the experience for the player according to the model.

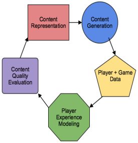


ED-PCG Race Tracks

Togelius, et al., introduced one of the first adaptive game design systems for making tracks in a racing game [Togelius, 2006].

- Evolved personalized race tracks via evolutionary algorithms
- Used AI agents modeled after players
- Skill defined by tendencies to go faster, avoid collisions, and make tighter turns

ED-PCG Race Tracks



- **Player + game data**: human data used to train AI drivers
- **Player modeling**: neural networks trained to drive the car in similarly to humans
- **Content quality**: static simulation-based evaluation function, assessed driving performance of the AI controller
- **Content representation**: directly as fixed-length parameter vectors, interpreted as b-splines (sequences of Bezier curves)
- **Content generation**: cascading elitism evolutionary algorithm

ED-PCG Race Tracks

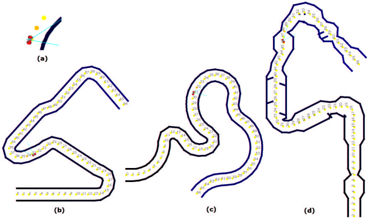


Figure: Hand crafted tracks

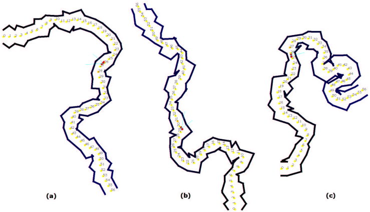


Figure: Evolved tracks

ED-PCG Race Tracks

Findings:

- Promising towards making adaptive systems based on player models.

Limitations:

- Only 5 human subjects, AI agents limited in human player representation.
- Theoretical player models were difficult to tune, sometimes would do things erratically (drive into walls, etc..).

Table of Contents

The proposed work to be done will be outlined in the following sections:

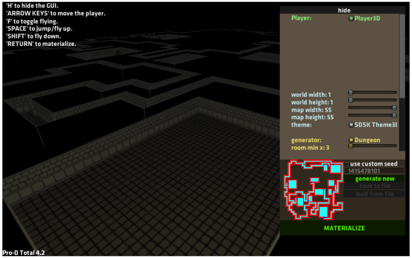
- Test Environment Overview
- Modeling Methodology
- Evaluating Game Content
- Generating Game Content

Pro-D Procedural Dungeon Generator

- Unity plugin allowing for algorithmically generated dungeon maps.
- Many parameters for customizing dungeon maps.

Option	Descriptions
Room min/max x/y	minimum and maximum width and height of any room in the generated map
Room frequency	amount of attempts to place a room on the map
Room retry	amount of retry attempts to place a room upon failing to place a room
Doors per room	amount of doors every room will have
U path reduction	amount of times corridors are simplified during map generation
Doors per room	amount of doors every room will have

Pro-D Interface



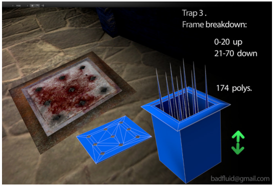
Characters


ENEMIES
Fantasy Horde
PolygonMaker
www.polygonmaker.com


UNITY ASSET
GOBLIN SHAMAN
ANIMATED CHARACTER

Items




Trap 3.
Frame breakdown:
0-20 up
21-70 down
174 polys.
badfluid@gmail.com

Basic Enemy Stats

Enemy	Attack Type	Damage	Health	Speed	Difficulty
Goblin	Melee	Low	Low	Fast	Easy
Orc	Melee	High	High	Slow	Medium
Skeleton	Ranged	Low	Low	Fast	Medium
Zombie	Melee	High	High	Slow	Hard
Vampire	Ranged	High	High	High	Hard

Table: Basic enemy descriptions.

Boss Enemy Stats

Enemy	Attack Type	Damage	Health	Speed
Manticore	Flying Ranged	Highest	Highest	Fast
Golem	Melee	Highest	Highest	Slow
Goblin Shaman	Melee	Lowest	Lowest	Slow

Table: Boss enemy descriptions.

Level Flow

1. Intro
   1. Player appears in room
   2. Goblin Shaman appears, explains player's fate.
2. First section
   1. Player battles through 2 rooms with basic enemies and items while initial data is acquired.
   2. Goblin Shaman appears, cutscene plays describing morphing dungeon.
3. Second section
   1. Player battles through more rooms until mid boss fight.
   2. After fight, Goblin Shaman appears, changes dungeon again.
4. Final section
   1. Player battles through more rooms leading up to boss fight.

PEM Approach in This Work

- Hybrid approach utilizing:
  - Gameplay data (model free)
  - Subjective data (forced response)
- Telemetry: provides gameplay data from players
- User studies: affective pairwise preference data for levels

PEM Approach in This Work

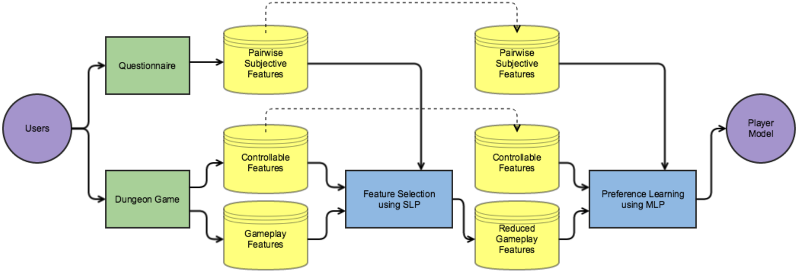


Figure: Flowchart of the PEM methodology used in this study.

Controllable Features

Feature	Description
Level length	number of rooms in the level.
Level topology	difference in room sizes.
Enemy variety	how many different enemies.
Enemy frequency	how many enemies per room.
Trap frequency	number of traps in the level.
Chest frequency	number of treasure chests in the level.
Health frequency	number of health packets to be found in the level.

Table: Controllable features of the dungeon game levels.

Gameplay Features

Category	Feature
Combat	% enemies killed
	% enemies killed (each type)
	% attacks landed
	% attacks dodged
	total damage taken
	total damage to enemy
Item	% treasure opened
	% health collected
	% vases smashed

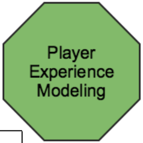
Gameplay Features

Category	Feature
Movement	% areas visited
	# jumps
	% time running
	% time ducking
	% traps avoided
Time	time to complete level
	average time per room
	average time per hall
	% time fighting
	% time exploring

Gameplay Features

Category	Feature
Death	total deaths
	deaths from enemy (each type)
	deaths from boss
	deaths from trap

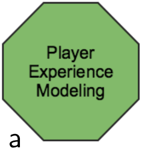
Subjective Features



Affective State	Description
Fun	enjoyment from playing the level.
Challenge	difficulty experienced in completing the level.
Flow	continuity from the sequence of events in the level.
Frustration	irritation or annoyance that arose form playing the level.
Excitement	thrill or elevated exhilaration from playing the level.
Accomplishment	feeling of pride from playing the level.

Table: Subjective preference categories.

Subjective Features



The pairwise preferences will be obtained through the use of a 4-AFC (alternative forced choice) protocol. Under this method, the user expresses preferences in the form:

- A [B] felt more E than B [A] (cf. 2-alternative forced choice)
- both felt equally E or
- neither of the two felt E

A and B represent the different versions of the game level and E is one of the emotional states defined above.

Example

User #1035 reported level 10 felt more frustrating than level 16.

Modeling Methodology

2-phase approach

- Phase1: training single-layer perceptron artificial neural network (ANN)
  - Much easier to perform feature selection on SLP
  - Much easier to attribute single features to affective response
- Phase 2: training multi-layer perceptron ANN
  - More accurate
  - Reduced training time with SLP feature selection

Feature Selection Methods

- nBest: ranks the features used individually in order of model performance
- SFS: bottom-up search procedure adding one feature at a time to the set
- SFFS: adds features but also checks whether removing one will improve
- PFS: search variant of neural pruning
- GFS: genetic algorithm search for accurate feature subset

Preference Learning

An instance preference learning (IPL) problem consists of a set of $\mathcal{X}$ instances which are associated with a total or partial order relation and a set of $\mathcal{D}$ pairwise relations on them.

- Instances: attribute-value features that consist of gameplay and controllable features that are generated during a gameplay session.
- Pairwise relations: subjective data collected from user responses across affective dimensions specified earlier.

Preference Learning

The learning algorithm proceeds as follows:

1. A population of ANNs (1000) is initialized. SLP, fully connected, connection weights random [-5,5].
2. Each ANN in the population gets input two feature vectors, $d_{j,A}$ and $d_{j,B}$ and returns two values, $e_{j,A}$ and $e_{j,B}$, representing the corresponding level of computed emotion in each variant.
3. The two computed outputs for each are fed into a logistic sigmoidal function defined as:

$$g(e, \epsilon) = \frac{1}{1 + e^{-\epsilon e_j}} \qquad (1)$$

where $e_j = e_{j,A} - e_{j,B}$ is the difference of the ANN output values for the pair of games, A and B for pair j, and $\epsilon = 30$ if $A \succ B$ (agreement) and $\epsilon = 5$ if $A \prec B$ (disagreement).

Preference Learning

3. Each member i of the population is evaluated with the fitness function defined as:

$$f_i = \sum_{j=1}^{N_{pairs}} g(d_j, \epsilon) \tag{2}$$

4. Roulette wheel selection is used to determine which chromosomes will be selected for crossover. The roulette wheel selection formula is:

$$p_i = \frac{f_i}{\sum_{l=1}^{N} f_i} \tag{3}$$

Preference Learning

5. Selected parents clone an equal number of offspring so that the total population reaches N members or reproduce offspring by crossover.
6. Gaussian mutation occurs in each gene (connection weight) of each offspring's genome with a small probability $p_m = \frac{1}{n}$, where n is the number of genes.

The algorithm terminates after either a good enough solution is found or after a large number of iterations has completed (e.g. 10000).

Topology Optimization

Player Experience Modeling

After feature selection and SLP training is finished, an MLP is trained using the same approach as above.

- Previous work has shown that MLP ANNs can produce more accurate models.
- SLP ANN allows for easier attribution of single feature affective response.
- Reduced feature sets allow for faster convergence of model.

Several different topologies (number of layers/number of hidden nodes per layer) will be explored. Previous values of up to 30 and 10 hidden neurons proved appropriate in [Pederson, 2010].

PEM Approach in This Work

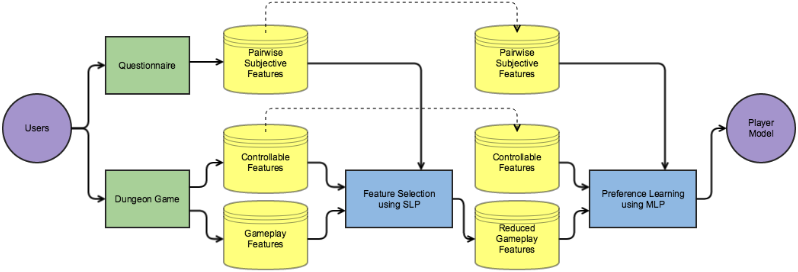


Figure: Flowchart of the PEM methodology used in this study.

Evaluating Game Content

- Content is evaluated on it's usefulness/fitness in producing the desired affective response in the player.
- Different types of evaluation functions for different purposes.
  - Direct
  - Simulation-based
  - Interactive

Dungeon Game Content Evaluation Functions

- Purely data-based direct evaluation function.
- Essentially the weighted sum of the output of the MLP ANN across all affective dimensions.

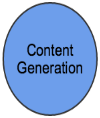
Content Space Search

- Player model $\rightarrow$ evaluation function $\rightarrow$ content generation.
- Content space search.
- Space determined by the dimensionality of the controllable features.
- Search will be performed using an evolutionary (genetic) algorithm.

Content Space Search

- Controllable feature space currently represents 6,000,000 levels.
- Likely that exhaustive search will prove insufficient.
- Parameter values can be tuned if search takes too long.

Searching and Generating Content



ED-PCG has never been used to create content during the execution of a game level as is proposed in this work.

- Previously used to create levels offline.
- Different types of evaluation functions/search methods for different purposes.

This change will result in changes to the geometry and content layout algorithms.

- Gameplay data aggregated over time using current controllable features.
- Incrementally improve the quality of levels by optimizing controllable features.

Searching and Generating Content

Content Generation

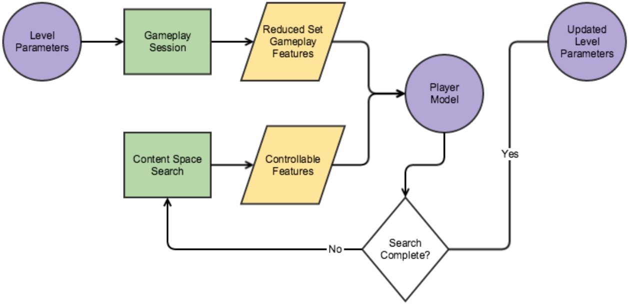


Figure: Content search approach depicted in the flow of a level.

Table of Contents

Player Modeling Accuracy

The neural network will be assessed on:

- **Accuracy**: the theoretical accuracy of the model, after the initial user study
- **Feature subset**: the results of the feature selection approaches, a subset of the entire feature set that results in the highest prediction accuracy.
- **Individual feature contribution**: also a result of the feature selection techniques, the individual contribution of gameplay and controllable features toward subjective preference accuracy will be analyzed.
- **Training time**: the time it takes to train the ANN to it's highest accuracy will be reported for both the SLP and MLP topologies.

Content Generation and Search Evaluation

The content generation search will be evaluated assessed on:

- **Accuracy**: how close to optimal for the affective dimensions that the content search techniques can achieve.
- **Time to converge**: the amount of time it takes the genetic and exhaustive algorithms to converge to a solution.

Overall Effectiveness

The effectiveness of the ED-PCG approach will be evaluated on several criteria:

- The difference between the optimized values of each new set of level parameters and the actual game parameters will be computed at each content generation point.
- Effectiveness can partially be measured by whether the final optimization requires more or less adaptation than the previous content optimizations.
- A further set of user tests will be performed.
    - Users will be asked to compare their feelings between the first, second, and third parts of the game level.
    - 4-AFC protocol but in this case A and B represent the different parts of the game level and E is one of the affective states.

Table of Contents

User Testing

- **Pilot study**: to assess the feasibility of training, familiarize work out bugs, initially in-person
  - ~50 users over 2 weeks
- **Training phase**: conducted online and in person, will host an instance of the game on a server
  - ~500 users over 3 months
- **Verification phase**: conducted online and in person
  - ~200 users

User Testers

- Initial user testers needed for acquiring gameplay and subjective data.
- User testers for the final system.
- Recruiting through social networking, GDC connections.
- Potentially use Amazon Mechanical Turk for recruiting.

Undergraduate Programmers

- Undergraduate research assistants earning academic credits.
- This could significantly alleviate the amount of time and work necessary for getting the test environment off the ground.

Winter 2015

- Experimental environment development
  1. Acquire assets from Unity Asset Store and set up project repository (1 day).
  2. Integrate acquired assets to create a single project structure (2 weeks).
  3. Adding objective and gameplay telemetry into game environment (3 weeks).
  4. Create hand crafted levels that target specific affective responses from literature (4 weeks).
- User study
  1. Design user study questionnaire (1 day).
  2. Advertise user study (1 day).
- Paper submission
  - CHI 2015: Doctoral consortium paper
  - FDG 2015: Doctoral consortium paper

Spring 2015

- User study
  1. Implement online testing functionality (1 week).
  2. Open up online environment (1 week).
  3. Acquire data online ($\sim$8 weeks).
  4. Acquire data in person with physiological data ($\sim$4 weeks).
- Player modeling
  1. Implement and test the machine learning algorithms (3 weeks).
- Paper submission
  - AIIDE 2015: Player modeling integrating different data streams for ED-PCG.
  - CIG 2015: Player modeling integrating different data streams for ED-PCG.

Summer 2015

- User study cont.
  1. Continued data acquisition online ($\sim$6 weeks).
- Player modeling cont.
  1. Continued data mining experiments for construction of content evaluation models ($\sim$ weeks).
  2. Run feature selection (2 weeks).

Fall 2015

- Content generation
  1. Integration of content generation functionality to dungeon game (∼8 weeks).
  2. Setting up final user study (∼1 week).
- Paper submission
  - AIIDE 2016: Real-time ED-PCG.

Winter 2016

- User study
  1. Final user study
- Paper submission
  - ACII 2016: Real-time ED-PCG.

Spring 2016

- Thesis writing
- Paper submission
    - FDG 2016: Real-time ED-PCG.

Summer 2016

- Thesis writing/completion

Table of Contents

References

Entertainment Software Association (2014)
Essential Facts About the Computer and Video Game Industry

IndieDB (2014)
Minecraft;
http://www.indiedb.com/games/minecraft/images/tnt-minecart

Gematsu (2014)
No Man's Sky debut screenshots, previews; http://gematsu.com/2013/12/no-mans-sky-debut-screenshots-previews

Artificial and Computational Intelligence in Games (2013)
Player modeling

Proceedings of the 6th International Conference on Foundations of Digital Games (2011)
An Inclusive View of Player Modeling

References

- IEEE Transactions on Computational Intelligence and AI in Games
  Search-based Procedural Content Generation
- IEEE Transactions on Affective Computing
  Experience-driven Procedural Content Generation
- Proceedings of the SAB'06 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games
  Making Racing Fun Through Player Modeling and Track Evolution
- IEEE Transactions on Computational Intelligence and AI in Games
  Adapting Models of Visual Aesthetics for Personalized Content Creation

References

- Unity Asset Store
  Adventurer character;
  http://u3d.as/content/dogzer/adventurer-character/2td
- TutsPlus Game Development
  Create a Procedurally Generated Dungeon Cave System;
  http://gamedevelopment.tutsplus.com/tutorials/
  create-a-procedurally-generated-dungeon-cave-system--gamedev-10099
- PolygonMaker - Mobile Game Art
  Fantasy Horde - Enemies;
  http://www.polygonmaker.com/Products/Products_enemies.html
- Unity Asset Store
  5 Animated Medieval Traps;
  https://www.assetstore.unity3d.com/en/#!/content/6724

References

BitGem 3D
Low Poly Dungeon Entourage Set;
http://shop.bitgem3d.com/collections/low-poly-3d-props/products/low-poly-dungeon-entourage-set

IEEE Transactions onComputational Intelligence and AI in Games
Modeling player experience for content creation

The End