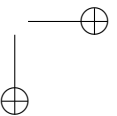
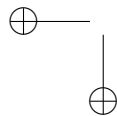
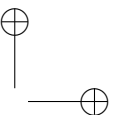
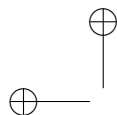


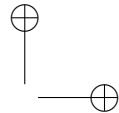
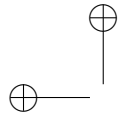
First Person Shooter level generation using Generative Adversarial Networks

Edoardo Giacomello

January 8, 2018

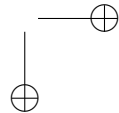
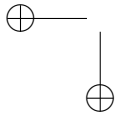


Abstract

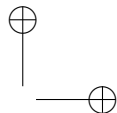
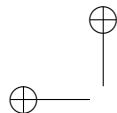


—

—

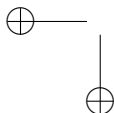


Estratto in lingua Italiana

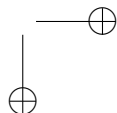


—

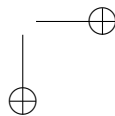
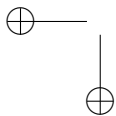
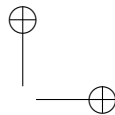
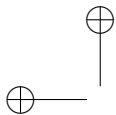
—



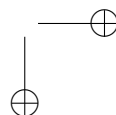
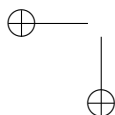
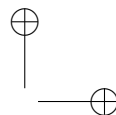
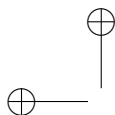
|



To someone...



Acknowledgments

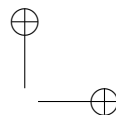
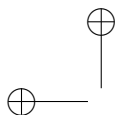


Contents

Abstract	3
Estratto in lingua Italiana	5
Acknowledgments	9
1 Introduction	17
1.1 Background: Level Design	17
1.2 State of the Art	17
1.2.1 Procedurally Generated Content	17
1.2.2 Procedural Content Generation via Machine Learning (PCGML)	17
1.3 Scope	17
1.4 Thesis Structure	17
1.5 Summary	17
2 Toward learn-based level generation	19
2.1 Generative Adversarial Networks	19
2.1.1 Overview	19
2.1.2 Deep Convolutional GAN	19
2.1.3 Wesserstein GAN	19
2.1.4 Wesserstein GAN with Gradient Penalty	19
2.1.5 Recent results	19
2.2 Game of choice: DOOM	19
2.2.1 Description	19
2.2.2 Motivation	19
2.2.3 Level Data Format	19
2.3 Summary	19
3 Dataset and Data Representation	21
3.1 Data Sources	21
3.2 Native Data Format: WAD Files	21
3.3 Feature Extraction	21
3.4 Data Encoding	21
3.5 Input and Output Pipelines	21
3.6 Summary	21

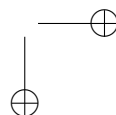
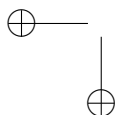
4	System Design and Overview	23
4.1	System Overview	23
5	System Architecture	25
5.1	25
6	Experiment Design and Results	27
6.1	Parameter Tweaking and Training Phase	27
6.1.1	Techniques and ”GAN Tricks” used	27
6.1.2	Resulting Model	27
6.2	Sampling the network	27
6.3	Generated Samples	27
6.4	In-Game Demonstration	27
6.5	Summary	27
7	Results Evaluation and Conclusions	29
7.1	Results Evaluation	29
7.1.1	Evaluation metric	29
7.1.2	Samples Evaluation	29
7.1.3	Loss of accuracy	29
7.2	Summary	29
8	Future Work	31
8.1	Open Problems	31
8.2	Possible Applications and future develops	31

List of Figures

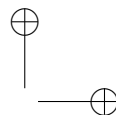
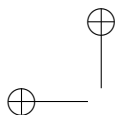


—

—

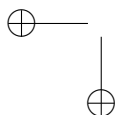


List of Tables

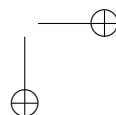


—

—



|



Chapter 1

Introduction

1.1 Background: Level Design

1.2 State of the Art

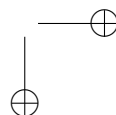
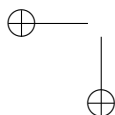
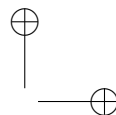
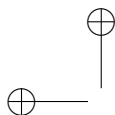
1.2.1 Procedurally Generated Content

1.2.2 Procedural Content Generation via Machine Learning (PCGML)

1.3 Scope

1.4 Thesis Structure

1.5 Summary



Chapter 2

Toward learn-based level generation

2.1 Generative Adversarial Networks

2.1.1 Overview

2.1.2 Deep Convolutional GAN

2.1.3 Wesserstein GAN

2.1.4 Wesserstein GAN with Gradient Penalty

2.1.5 Recent results

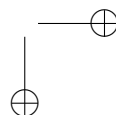
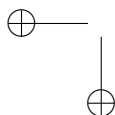
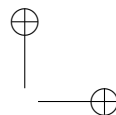
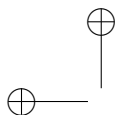
2.2 Game of choice: DOOM

2.2.1 Description

2.2.2 Motivation

2.2.3 Level Data Format

2.3 Summary



Chapter 3

Dataset and Data Representation

3.1 Data Sources

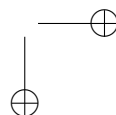
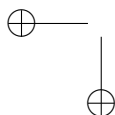
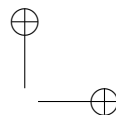
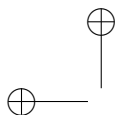
3.2 Native Data Format: WAD Files

3.3 Feature Extraction

3.4 Data Encoding

3.5 Input and Output Pipelines

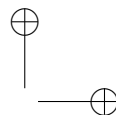
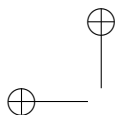
3.6 Summary



Chapter 4

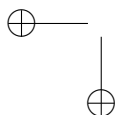
System Design and Overview

4.1 System Overview

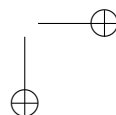


—

—



|

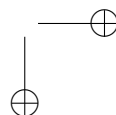
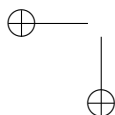
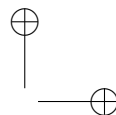
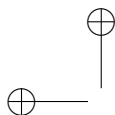


Chapter 5

System Architecture

5.1 Component View

5.2 Neural Network Architecture



Chapter 6

Experiment Design and Results

6.1 Parameter Tweaking and Training Phase

6.1.1 Techniques and ”GAN Tricks” used

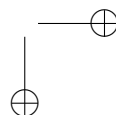
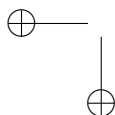
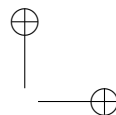
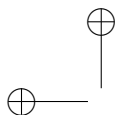
6.1.2 Resulting Model

6.2 Sampling the network

6.3 Generated Samples

6.4 In-Game Demonstration

6.5 Summary



Chapter 7

Results Evaluation and Conclusions

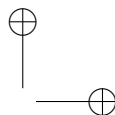
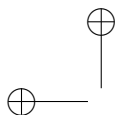
7.1 Results Evaluation

7.1.1 Evaluation metric

7.1.2 Samples Evaluation

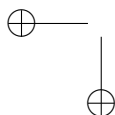
7.1.3 Loss of accuracy

7.2 Summary

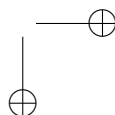


—

—



|

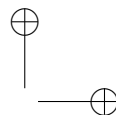
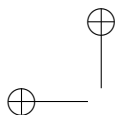


Chapter 8

Future Work

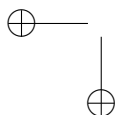
8.1 Open Problems

8.2 Possible Applications and future develops



—

—



|

