
TSE

Oyunlarda procedural generation oyuncu memnuniyetini nasıl etkiler

Author: Samet Zengin [(1)] Mail: smt.zngn02@gmail.com

Abstract

Procedural generation (PCG), oyun dünyalarının, seviyelerinin ve içeriklerinin algoritmik olarak oluşturulmasını sağlayarak oyuncu deneyimini kişiselleştiren ve tekrar oynanabilirliği artıran güçlü bir tekniktir. Bu çalışma, PCG'nin oyuncu memnuniyeti üzerindeki etkilerini araştırmayı amaçlamaktadır. PCG teknikleri kullanılarak oluşturulmuş haritalar ile manuel tasarlanmış haritaların oyuncular üzerindeki etkileri karşılaştırılmıştır. Gönüllü 50 oyuncudan toplanan veriler ele alınarak, PCG teknikleri kullanılarak oluşturulan haritaların oyuncu memnuniyeti ve oyun süresindeki olumlu etkileri istatiksel olarak analiz edilmiştir. Çalışmanın sonuçları, PCG tekniklerinin oyuncuların ilgisini artırdığı, tekrar oynanabilirliği desteklediği ve oyun tasarımı süreçlerinde geliştiricilere önemli avantajlar sağladığını göstermiştir.

Keywords: Procedural Generation (PCG), oyun geliştirme, oyuncu memnuniyeti, hücresel otomata, oyun tasarımı, tekrar oynanabilirlik

1. Introduction

Oyun endüstrisi, oyuncuların ilgisini çekmek ve değişen beklentilerine yanıt verebilmek için sürekli bir ara yıl içerisindedir. Dijital oyunlar, eğlencenin ötesine geçerek birçok oyuncu için sosyal bir platform, meydan okuma, keşif duygusunu ve yaratıcılığı destekleyen bir araç haline gelmiştir. Oyun endüstrisinin son gelişim dönemlerinde, yıllık milyarlarca dolarlık bir gelirle eğlence endüstrisinin en büyük kollarından biri haline gelmiştir. Bu büyüme, sadece teknolojik gelişmelere değil, aynı zamanda oyuncuların deneyimlerinin daha derin ve tatmin edici hale getirilmesine yönelik yenilikçi yaklaşımlara da dayanmaktadır.

Oyunların başarısı, büyük ölçüde oyunculara sundukları deneyimlerin kalitesine ve sundukları görselliğin kalitesine bağlıdır. Oyuncular, tekrar oynama isteği uyandıran, zengin ve çeşitli oyun içerikleri ile oyunlara karşı ilgilerini uzun süre sürdürebilmektedir. Bu açıdan PCG, oyun dünyalarının, seviyelerin ve diğer oyundaki bulunan içeriklerin algoritmalar kullanılarak otomatik olarak oluşturulmasını sağlayan bir teknoloji olarak dikkat çekmektedir. PCG, sabit ve durağan içeriklerin yanı sıra her oyun aşamasında benzersiz bir deneyim sunar. Bu durum, özellikle oyuncuların sürekli olarak yeni oyun dünyaları içerisinde keşif duygusunu arttırarak oyunda kalmasını sağlar. PCG, bu etki sayesinde tekrar oynanabilirliği artırmak için güçlü bir araç haline gelmiştir.

PCG'nin temel mantığı, oyun içeriklerini rastgele ve algoritmik olarak oluşturma yeteneğidir. Örneğin, bir açık dünya oyununda oyuncunun karşılacağı manzaralar, zorluklar veya ödüller farklı olabilir. Bu durum, hem oyunculara her yeni oturumda farklı bir deneyim sunar hem de oyunların uzun ömürlü olmasını sağlar [1]. Ayrıca, geliştiriciler için büyük ölçekli oyun dünyalarını manuel olarak tasarlama zorluğunu ortadan kaldırarak üretim maliyetlerini düşürür ve yaratıcılık için sınırsız olanaklar sunar [2].

Son yıllarda, PCG teknikleri yalnızca oyun dünyalarını oluşturmanın ötesine geçerek, oyuncu tercihlerine uyarlanabilir içerikler oluşturmak için de kullanılmaktadır. Örneğin, roguelike türünde bir oyun olan Rogue, her oynanışta rastgele oluşturulan haritalarıyla oyuncuların uygulayabilecekleri strateji yelpazesini geliştirmekte ve ilgiyi canlı tutmaktadır [3].

Bu çalışmada, PCG'nin oyunların tekrar oynanabilirliğine olan etkisi ele alınacak ve bu teknolojinin oyuncu deneyimindeki önemi analiz edilecektir. Literatürdeki çalışmalar ve yapılan analizler sayesinde, PCG'nin oyuncu beklentilerini nasıl karşıladığı ve oyun tasarımına sağladığı katkı değerlendirilecektir.

2. Related Work

PCG, oyun geliştirmede dinamik ve kişiselleştirilmiş içerikler oluşturma konusunda dikkat çekmekte ve oyuncu memnuniyetini artırmada önemli bir araç olarak görülmektedir. Bu bölümde, PCG'nin oyuncu davranışlarına etkisi, uyarlanabilir içerik oluşturma ve kişiselleştirme konularındaki çalışmalara yer verilmektedir.

PCG teknolojisi, oyuncuya farklı içerikler sunarak oyuncunun sürekli aynı bölümleri görmesini engellemektedir. **"Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design"** adlı çalışma, insan yaratıcılığını algoritmik süreçlerle birleştirerek ritim ve hız tabanlı yapılarla oyun seviyelerini optimize etmektedir. Bu sistem, oyuncular için daha ilgi çekici ve dengeli içerikler sunmayı hedeflemektedir [4]. Benzer şekkilde, "Procedural Level Generation in Educational Games from Natural Language Instruction" adlı çalışma, doğal dil işleme (NLP) teknikleri kullanarak, tasarımcıların sezgisel bir şekilde oyun seviyelerini oluşturmalarına olanak sağlamakta ve oyuncu odaklı içerik üretimini genişletmektedir [5].

Oyuncu davranışlarına dayalı içerik üretimi, PCG'nin önemli bir boyutunu oluşturmaktadır. **"Automatic Content Generation in the Galactic Arms Race Video Game"** çalışmasında kullanılan **cgNEAT** algoritması, oyuncuların tercihlerini analiz ederek gerçek zamanlı içerik üretmektedir. Bu sistem, hem oyuncu memnuniyetini artırmak hem de tekrar oynanabilirliği desteklemek için dinamik içerik geliştirme süreçlerini optimize etmektedir [6]. Ayrıca, **"A Generic Approach to Challenge Modeling for the Procedural Creation of Video Game Levels"** adlı çalışma, dinamik zorluk eğrileri kullanarak oyun seviyelerinin eğlenceliliğini artırmayı ve oyuncuların memnuniyetini optimize etmeyi amaçlamaktadır [7]. Benzer şekilde, **"An Approach to Level Design Using Procedural Content Generation and Difficulty Curves"** adlı çalışma, genetik algoritmalar aracılığıyla oyuncu geri bildirimlerine dayalı olarak içeriklerin kişiselleştirilmesini ve zorluk dengelerinin sağlanmasını hedeflemektedir [8].

PCG'nin bilgi transferi ve yeniden kullanım yoluyla içerik oluşturma süreçlerindeki etkisi de dikkate değerdir. **"Procedural Content Generation via Knowledge Transformation (PCG-KT)"** çalışması, oyunlar arasında bilgi transferini ele almakta ve mevcut içeriklerin yaratıcı bir şekilde yeniden kullanımıyla yenilikçi içerikler oluşturulmasını sağlamaktadır [9]. Bu yöntem, hem içerik üretimini hızlandırmakta hem de oyuncuların beklentilerine uygun içerikler sunmaktadır.

Oyuncu davranışlarını anlamak ve kişiselleştirilmiş içerikler oluşturmak, PCG'nin oyuncu memnuniyetine olan etkisini artırmaktadır. **"Modeling Player Personality Factors from In-Game Behavior and Affective Expression"** adlı çalışma, oyuncu davranışları ve kişilik özellikleri arasındaki ilişkiyi inceleyerek, kişilik temelli içerik üretiminin oyuncu memnuniyetini nasıl artırabileceğini göstermektedir [10]. Benzer şekilde, **"Personality and Behavior in Role-Based Online Games"** çalışması, oyuncu davranışlarının demografik değişkenler ve oyun tarzları ile ilişkisini analiz ederek, daha kişiselleştirilmiş oyun deneyimlerinin nasıl yaratılabileceğini ele almaktadır [11].

Diğer araştırmalar, prosedürel içerik üretimi süreçlerinde pekiştirmeli öğrenmenin (Reinforcement Learning) entegrasyonuna dikkat çekmektedir. **"Deep Reinforcement Learning for Procedural Content Generation of 3D Virtual Environments"** adlı çalışma, DRL ajanları kullanarak oyun seviyelerini dinamik bir şekilde oluşturmayı ve oyuncu beklentileriyle uyumlu hale getirmeyi hedeflemektedir [9]. Son olarak, **"Search-Based Procedural Content Generation: A Taxonomy and Survey"** çalışması, içerik üretiminde optimizasyon algoritmalarını kategorize ederek, oyuncu tercihleriyle uyumlu içeriklerin nasıl geliştirilebileceğini tartışmaktadır [12].

Bu çalışmalar, prosedürel içerik üretiminin oyuncu memnuniyetini artırmadaki potansiyelini vurgulamaktadır. Dinamik algoritmalar, oyuncu davranışlarını dikkate alan yaklaşımlar ve yaratıcı framework'ler sayesinde PCG, oyun tasarımında hem eğlenceliliği artırmakta hem de tekrar oynanabilirliği desteklemektedir.

3. Materials And Method

Bu bölümde, PCG tekniklerini uygulayabileceğimiz algoritmalardan biri olan hücresel otomata (Cellular Automata) algoritmasını kullanacağız. Hücresel otomata sayesinde oyun haritalarını oluşturarak kullanıcılara memnuniyet anketi yapacağız.
Hücresel Otomata: 1970'lerde bir matematikçi olan John Conway, *The Game Of Life* adında bir simülasyon tanımlamıştır. Bu simülasyon aslında bir oyun değil, daha çok ızgaradaki hücrelerin (ya yaşayan ya da ölü) basit kurallar çerçevesinde değişimini gösterir.

3.1. Hücresel Otomata ile Prosedürel İçerik Üretimi

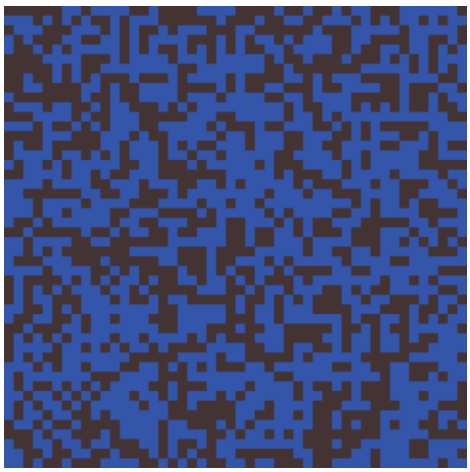
Hücresel otomata, her bir hücrenin komşu hücrelerin durumuna göre değiştiği ve geliştiği bir matematiksel modeldir. Bu çalışmada, oyun dünyalarının oluşturulması için rastgele başlatılmış bir ızgara üzerinde hücresel otomata kuralları uygulanmıştır.

1. **Başlangıç Izgarasının oluşturulması:**
   - Boyutları tanımlanmış bir ızgara (width x height) oluşturulmuştur.
   - Her hücre, belirli bir olasılıkla (chanceToStartAlive) "dolu" veya "boş" olarak rastgele atanmıştır.
   - Başlangıç durumu şu şekilde tanımlanmıştır:
   - "Dolu" hücreler, oyuncu tarafından geçilemez katı yapıları temsil eder.
   - "Boş" hücreler, oyuncunun hareket edebileceği alanları ifade eder.

```
Algorithm InitialiseGrid(width, height, chanceToStartAlive)
    Input: width (int), height (int), chanceToStartAlive (float)
    Output: grid (2D Boolean array)

    // Izgarayı oluştur ve başlat
    grid ← Create 2D array of size [width][height]
    for x from 0 to width-1 do
        for y from 0 to height-1 do
            randomValue ← GenerateRandomFloat(0, 1)
            if randomValue < chanceToStartAlive then
                grid[x][y] ← true  // Hücre dolu
            else
                grid[x][y] ← false // Hücre boş
            end if
        end for
    end for
    return grid
End Algorithm
```

Görsel 1: Başlangıç ızgarasının oluşturulma algoritması



Görsel 2: Simülasyon adımından önceki rastgele mağaramız.

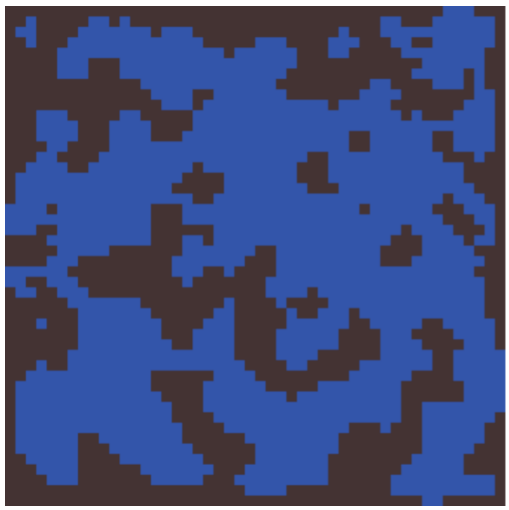
2. Simülasyon Adımları:

- Rastgele oluşturulan ızgaraya belirli kurallar uygulanmıştır:
  - Bir hücre, yeterli komşusu yoksa ölür (boş hale gelir).
  - Boş bir hücre, yeterli komşusu varsa "doğar" (dolu hale gelir).
- Simülasyon, oyuncu deneyimi için daha düzgün ve ilgi çekici haritalar oluşturana kadar belirli sayıda tekrarlanmıştır.

```
Algorithm doSimulationStep(oldGrid, deathLimit, birthLimit)
    Input: oldGrid (2D Boolean array), deathLimit (int), birthLimit (int)
    Output: newGrid (2D Boolean array)

    newGrid ← Create 2D array of size [width][height]
    for x from 0 to oldGrid.width-1 do
        for y from 0 to oldGrid.height-1 do
            aliveNeighbours ← countAliveNeighbours(oldGrid, x, y)
            if oldGrid[x][y] == true then
                if aliveNeighbours < deathLimit then
                    newGrid[x][y] ← false // Hücre ölür
                else
                    newGrid[x][y] ← true  // Hücre hayatta kalır
                end if
            else
                if aliveNeighbours > birthLimit then
                    newGrid[x][y] ← true  // Hücre doğar
                else
                    newGrid[x][y] ← false // Hücre ölü kalır
                end if
            end if
        end for
    end for
    return newGrid
End Algorithm
```

Görsel 3: Simülasyon adımlarının uygulanma algoritması



Görsel 4: Hücresel Otomata iki kez uygulandıktan sonra oluşan mağara

Görsel 2 ve 4 karşılaştırıldığında, başlangıçta hiçbir anlam ifade etmeyen noktalar birleşerek bir harita oluşumunu ortaya çıkartmıştır. Basit bir ön izleme ile kahverengi noktalar sınırları, mavi noktalar ise oyuncuların erişebileceği noktaları betimlemektedir. Kullanılan teknik sayesinde, sadece harita oluşumunu değil içerikleri de yerleştirebiliriz.

3.2. Oyuncu Verilerinin Toplanması ve Analizi

Player ID	**PCG**	**Game Time (minutes)**	**Satisfaction Score (1-10)**
Player_1	Evet	95	9
Player_2	Hayır	45	5
Player_3	Evet	110	10
Player_4	Evet	85	8
Player_5	Hayır	30	4
Player_6	Evet	100	9
Player_7	Hayır	50	6
Player_8	Evet	90	8
Player_9	Hayır	40	5
Player_10	Evet	105	9
Player_11	Evet	115	10
Player_12	Hayır	20	3
Player_13	Evet	80	7
Player_14	Evet	95	9
Player_15	Hayır	35	4
Player_16	Evet	120	10
Player_17	Evet	90	8
Player_18	Hayır	55	5
Player_19	Evet	100	9

Player_20	Hayır	40	4
Player_21	Evet	85	7
Player_22	Evet	110	9
Player_23	Hayır	25	3
Player_24	Evet	120	10
Player_25	Hayır	50	6
Player_26	Evet	100	9
Player_27	Evet	95	8
Player_28	Hayır	35	4
Player_29	Evet	115	9
Player_30	Hayır	30	2
Player_31	Evet	110	10
Player_32	Hayır	45	5
Player_33	Evet	105	9
Player_34	Hayır	40	4
Player_35	Evet	90	8
Player_36	Hayır	55	6
Player_37	Evet	120	10
Player_38	Hayır	20	3
Player_39	Evet	115	9
Player_40	Hayır	25	3
Player_41	Evet	100	8
Player_42	Hayır	50	5
Player_43	Evet	110	9
Player_44	Hayır	35	4
Player_45	Evet	105	10
Player_46	Hayır	30	3
Player_47	Evet	95	8
Player_48	Hayır	55	6
Player_49	Evet	120	10
Player_50	Hayır	40	4

Tablo 1: Kullanıcı memnuniyet tablosu

Tablo 1'e bakıldığında, PCG'nin oyuncu memnuniyeti üzerindeki etkisini incelemek amacıyla 50 gönüllü oyuncudan veri toplanmıştır. Oyuncular, dinamik olarak oluşturulan ve sabit tasarlanmış oyun haritalarını oynayarak değerlendirmelerde bulunmuşlardır. Veriler, oyun süresi ve memnuniyet puanı gibi kritik metriklere dayanmaktadır.

Oyuncuların değerlendirmeleri aşağıdaki üç temel veri kategorisine ayrılarak incelenmiştir:

1. **Harita Türü:** Haritaların PCG teknikleri kullanılarak veya manuel olarak tasarlanmış olduğu belirtilmiştir
2. **Oyun Süresi (dk):** Oyuncuların bir gün içerisinde belirli bir haritada geçirdikleri toplam süre kaydedilmiştir. PCG teknikleri ile oluşturulan haritalar, oyuncuların oyun süresini artırarak daha fazla ilgi çekmiştir.
3. **Memnuniyet Puanı (1-10):** Oyuncular, oynadıkları haritalarla ilgili genel memnuniyetlerini bir ölçek üzerinde değerlendirmiştir. PCG teknikleri ile oluşturulan haritalar, manuel tasarımlarla oluşturulmuş haritalara kıyasla daha yüksek memnuniyet puanlarıyla sonuçlanmıştır.

References

[1] Hendrikx, M., Meijer, S., van der Velden, J., & Iosup, A. (2013). Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications and Applications*, *9*(1). https://doi.org/10.1145/2422956.2422957

[2] Gao, T., Zhang, J., & Mi, Q. (2022). Procedural Generation of Game Levels and Maps: A Review. *4th International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2022 - Proceedings*, 50–55. https://doi.org/10.1109/ICAIIC54071.2022.9722624

[3] Risi, S., & Togelius, J. (2019). Increasing Generality in Machine Learning through Procedural Content Generation. *Nature Machine Intelligence*, *2*(8), 428–436. https://doi.org/10.1038/s42256-020-0208-z

[4] Smith, G., Whitehead, J., & Mateas, M. (2011). Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design. *IEEE Transactions on Computational Intelligence and AI in Games*, *3(3), 201–215.* https://doi.org/10.1109/TCIAIG.2011.2159716

[5] Kumaran, V., Carpenter, D., Rowe, J., Mott, B., & Lester, J. (2024). Procedural Level Generation in Educational Games from Natural Language Instruction. *IEEE Transactions on Games.* https://doi.org/10.1109/TG.2024.3392670

[6] Hastings, E. J., Guha, R. K., & Stanley, K. O. (2009). Automatic Content Generation in the Galactic Arms Race Video Game. *IEEE Transactions on Computational Intelligence and AI in Games*, *1(4), 245–263.* https://doi.org/10.1109/TCIAIG.2009.2038365

[7] Sorenson, N., Pasquier, P., & DiPaola, S. (2011). A Generic Approach to Challenge Modeling for the Procedural Creation of Video Game Levels. *IEEE Transactions on Computational Intelligence and AI in Games, 3(3), 229–244.* https://doi.org/10.1109/TCIAIG.2011.2161310

[8] Adrian, D. F. H., & Ana Luisa, S. G. C. (2013). An Approach to Level Design Using Procedural Content Generation and Difficulty Curves. *IEEE Conference on Computational Intelligence and Games.* https://doi.org/10.1109/CIG.2013.6633640

[9] Sarkar, A., Guzdial, M., Snodgrass, S., Summerville, A., Machado, T., & Smith, G. (2024). Procedural Content Generation via Knowledge Transformation (PCG-KT). *IEEE Transactions on Games, 16(1), 36–50.* https://doi.org/10.1109/TG.2023.3270422

[10] Habibi, R., Pfau, J., & El-Nasr, M. S. (2023). Modeling Player Personality Factors from In-Game Behavior and Affective Expression. https://arxiv.org/abs/2308.14224v1

[11] Wang, Z., Sapienza, A., Culotta, A., & Ferrara, E. (2019). Personality and Behavior in Role-Based Online Games. *IEEE Conference on Computational Intelligence and Games.* https://doi.org/10.1109/CIG.2019.8848027

[9] López, C. E., Cunningham, J., Ashour, O., & Tucker, C. S. (2020). Deep Reinforcement Learning for Procedural Content Generation of 3D Virtual Environments. *Journal of Computing and Information Science in Engineering, 20(5).* https://doi.org/10.1115/1.4046293/1074423

[12] Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2011). Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Transactions on Computational Intelligence and AI in Games, 3(3), 172–186.* https://doi.org/10.1109/TCIAIG.2011.2148116