

# 第5回講義：プログレッションとエマージェ ンスの統合

---

## 第5回：プログレッションとエマージェンスの統合

### 概要

1. エマージェンスとプログレッションの対比と融合
2. おもちゃ → 遊び場 → ゲーム

おもちゃ (Toy)

遊び場 (Playground)

ゲーム (Game)

3. 構造化された学習曲線

### 原則

Half-Life 2 の例

4. 経済構築ゲーム

Caesarシリーズの分析

5. ミッションとスペースのミスマッチ

ダイアログツリー

Deus Ex の例

6. メカニクスによるプログレッション制御

ロックと鍵のメカニズム

7. ロックと鍵のフィードバックメカニズム

動的フリクション

ダイナミックエンジンパターンの適用

8. プログレスをリソースとして扱う

Warhammer Fantasy Roleplay の進行トラッカー

エスカレーティング・コンプリケーション

エスカレーティング・コンプレキシティ

Seasonsゲームの色魔法経済

### まとめ

# 第5回：プログレッションとエマージェンスの統合

## 概要

ゲームデザインにおいて、**プログレッション**（設計者が用意した固定的な進行）と**エマージェンス**（ルールからの相互作用から生まれる創発的なプレイ）は対立する概念ではなく、統合して用いることで優れたゲーム体験を実現できる。本章では、その統合手法を多角的に分析する。

## 1. エマージェンスとプログレッションの対比と融合

特性	エマージェンス	プログレッション
構造	少数のルールから複雑な挙動が生まれる	設計者が段階的に体験を配置する
プレイヤーの自由度	高い	低い（ガイドされる）
リプレイ性	高い	低い傾向
物語制御	困難	容易

多くの現代ゲームは両者をハイブリッドで採用している。プログレッションで大枠の進行を制御し、各局面でエマージェントなゲームプレイを提供する。

**重要ポイント：**プログレッションとエマージェンスは二項対立ではなく、スペクトラム上に位置する。優れたゲームは両者を目的に応じて使い分け、組み合わせる。

## 2. おもちゃ → 遊び場 → ゲーム

---

ゲームの構造化には段階がある。

おもちゃ→遊び場→ゲームの連続体

おもちゃ→遊び場→ゲームの連続体

### おもちゃ (Toy)

- ・ 定義: 目標のない自由な操作対象
- ・ 例: SimCity (サンドボックスモード)。プレイヤーは街を自由に作るが、勝利条件は存在しない
- ・ ルールは存在するが、ゴールがない

### 遊び場 (Playground)

- ・ 定義: おもちゃに空間的・構造的な文脈を与えたもの
- ・ 複数のおもちゃが組み合わさり、相互作用が生まれる
- ・ 探索や実験が主な楽しみ

### ゲーム (Game)

- ・ 定義: 明確な目標（ゴール）とルールが設定された構造体
- ・ 勝敗や達成条件が定義される
- ・ プレイヤーの行動が評価される

重要ポイント: おもちゃ→遊び場→ゲームは、自由度と構造のトレードオフを示す連続体である。デザイナーはプレイヤーにまず「おもちゃ」として操作を覚えさせ、段階的にゲームへ導くことができる。

---

## 3. 構造化された学習曲線

---

プレイヤーを複雑なメカニクスに段階的に導入する設計手法。

## 原則

1. 一度に一つのメカニクスを導入する
2. 新しいメカニクスを安全な環境で練習させる
3. 既存メカニクスとの組み合わせを徐々に要求する
4. 難易度を段階的に上昇させる

## Half-Life 2 の例

Half-Life 2 は構造化された学習曲線の模範例である。

Half-Life 2の段階的導入

Half-Life 2の段階的導入

- ・ バール → ピストル → SMG → グレネード → 重力銃 と順に導入
- ・ 各武器の導入時には、その武器が最適解となる状況を配置
- ・ 重力銃は最初に物理オブジェクト操作として「おもちゃ」的に紹介され、後にメイン武器として機能する
- ・ 物理エンジンの活用がプログレッション（段階的導入）とエマージェンス（自由な物理操作）を融合

重要ポイント: 優れた学習曲線は、プレイヤーが「教わっている」と感じないように設計される。環境自体が教師となり、メカニクスの理解を自然に促す。

## 4. 経済構築ゲーム

### Caesarシリーズの分析

Caesar（シーザー）シリーズは、経済構築ゲームにおけるプログレッションの好例である。

Caesarの経済メカニクス構造

## Caesarの経済メカニクス構造

- **段階的な経済の複雑化:**
  - 初期ミッション: 住宅と基本的な食料供給のみ
  - 中期ミッション: 水道、宗教施設、娯楽施設が追加
  - 後期ミッション: 貿易、軍事、複数の資源チェーンが必要
- **ミッションが経済メカニクスのアンロックを制御:**
  - 各ミッションで利用可能な建築物が制限される
  - プレイヤーは限定された要素で経済の基礎を学ぶ
  - 後のミッションで要素が増え、既習のメカニクスの上に新しい層が積まれる
- **内部経済の相互依存:**
  - 住民の満足度 → 人口増加 → 税金 → 建設投資 というフィードバックループ
  - 各ループが段階的に開放される

**重要ポイント:** 経済構築ゲームでは、プログレッション（ミッション構造）がエマーゼンス（経済の創発的挙動）の複雑さを制御するゲートとして機能する。

## 5. ミッションとスペースのミスマッチ

ミッション設計とゲーム空間が完全に一致しない場合、プレイヤーに選択の自由を与える手法。

### ダイアログツリー

ダイアログツリーの構造例

ダイアログツリーの構造例

- 会話内で選択肢を提示し、**分岐する物語進行**を実現
- 各選択がゲーム状態（敵対関係、情報取得、アイテム獲得など）に影響
- プログレッションの中にプレイヤーの**エージェンシー（主体性）**を埋め込む

## Deus Ex の例

- 一つのミッションに対して複数の解法が存在：
  - 戦闘: 正面から敵を排除
  - ステルス: 見つからずに目的を達成
  - ハッキング: セキュリティを電子的に突破
  - 社会的: 会話や交渉で問題を解決
- ゲーム空間（スペース）が複数の経路を物理的に提供
- ミッション目標は固定だが、達成手段がエマージェント

重要ポイント: ミッション（プログレッション）とスペースの間にあって「遊び」を持たせることで、線形的な進行の中にもエマージェントな体験を実現できる。

## 6. メカニクスによるプログレッション制御

### ロックと鍵のメカニズム

ゲーム空間の進行をメカニクスで制御する古典的パターン。

ロックと鍵のMachinations表現

ロックと鍵のMachinations表現

### Machinationsによる表現

- 鍵（Key）：プレイヤーが獲得するリソースまたは能力
- ロック（Lock）：鍵がなければ通過できないゲート
- Machinationsダイアグラムでは、リソースの有無を条件としたゲートで表現

### 森の神殿のスペースグラフ拡張

ゼルダの伝説シリーズの「森の神殿」をMachinationsで拡張する例：

森の神殿のスペースグラフ

### 森の神殿のスペースグラフ

- スペースグラフの各部屋をノードで表現
- ロックされた扉をゲートメカニクスで表現
- 鍵の取得がリソースフローとして可視化される

### 複数鍵メカニズム

- 1つのロックに対して**複数の鍵**が必要な場合
- AND条件: すべての鍵が必要
- OR条件: いずれか1つの鍵で開放可能
- これにより非線形な探索パターンが生まれる

### ボムリングメカニズム

- 爆弾とブーメランの連携による複合メカニクス
- ブーメランで遠距離の爆弾を起動
- 単独では機能しない能力の**組み合わせ**がロック解除条件となる
- プレイヤーに複数メカニクスの統合的理解を要求

重要ポイント: ロックと鍵のメカニズムは、空間的プロGRESSIONをメカニクスのリソースフローとして形式的にモデル化できる。複合鍵メカニズムは非線形な探索を実現する。

## 7. ロックと鍵のフィードバックメカニズム

### 動的フリクション

進行に伴って難易度を動的に調整するメカニズム。

動的フリクションのパターン

動的フリクションのパターン

- プレイヤーの進行度合いに応じて「摩擦（フリクション）」を増減させる

- ・ 進みすぎたプレイヤーには高い抵抗を、遅れたプレイヤーには低い抵抗を提供
- ・ ゲームバランスを自動的に維持

## ダイナミックエンジンパターンの適用

- ・ 第4章で学んだダイナミックエンジンのパターンをプログレッション制御に応用
- ・ リソース生成率と消費率をプレイヤーの進行状態に連動させる
- ・ フィードバックループにより、プログレッションの速度が自己調整される

重要ポイント: 動的フリクションは、プログレッション（固定的な進行）にエマージェンス（動的な調整）を組み込む具体的手法である。ダイナミックエンジンパターンがその基盤となる。

## 8. プログレスをリソースとして扱う

進行そのものをMachinationsのリソースとして表現し、メカニクスに組み込む手法。

### Warhammer Fantasy Roleplay の進行トラッカー

- ・ キャラクターの成長を進行トラッカーとしてモデル化
- ・ 経験値（XP）をリソースとして蓄積し、閾値を超えると新しい能力が解放
- ・ 進行自体がゲーム内経済の一部として機能

### エスカレーティング・コンプ리케이션

時間経過とともに障害が増大するパターン。

エスカレーティング・コンプ리케이션

エスカレーティング・コンプ리케이션

- ・ **Pac-Man:** ゴーストの速度が段階的に上昇、パワーペレットの効果時間が短縮
- ・ **Space Invaders:** エイリアンの数が減るほど移動速度が上昇
- ・ 同じメカニクスの中でパラメータが変化することで難易度が上昇
- ・ プレイヤーの行動（エマージェンス）は変わらないが、環境が厳しくなる



## エスカレーティング・コンプレキシティ

時間経過とともにシステムの複雑さ自体が増大するパターン。

- **Tetris:** ブロックの落下速度が上昇するだけでなく、積み上がったブロックが状態の複雑さを増す
- プレイヤーが管理すべき**状態空間**が拡大していく
- エスカレーティング・コンプリケーションとの違い: パラメータではなく**構造的な**複雑さが増大

## Seasonsゲームの色魔法経済

Seasonsの色魔法経済モデル

Seasonsの色魔法経済モデル

- 季節ごとに利用可能な**色付きリソース（魔法エネルギー）**が変化
- プレイヤーは季節の変化を先読みしてリソース管理を行う
- 進行（季節の推移）がリソース供給を制御する
- プログレッション（季節サイクル）とエマージェンス（リソース管理戦略）の統合

**重要ポイント:** 進行をリソースとしてモデル化すると、時間経過による難易度上昇を形式的に設計・分析できる。エスカレーティング・コンプリケーション（パラメータ変化）とエスカレーティング・コンプレキシティ（構造的複雑化）は異なるアプローチである。

## まとめ

手法	プロGRESSION側	エマージェンス側
学習曲線	メカニクスの段階的導入	各段階での自由なプレイ
経済構築	ミッションによるアンロック	経済の創発的挙動
ロックと鍵	空間進行の制御	鍵の取得順序の自由度

手法	プログレッション側	エマージェンス側
動的フリクション	進行の速度制御	フィードバックによる自己調整
エスカレーション	時間経過による変化	プレイヤーの適応行動

プログレッションとエマージェンスの統合は、ゲームデザインの中核的な課題であり、両者のバランスがプレイヤー体験の質を大きく左右する。