

Building an educational adventure game: Theory, design, and lessons

1. はじめに

遊び (play) は学習を促進 (Quinn, 1994; Rieber, 1996)

遊び = 本質的に動機付けられた自発的活動

ゲーム が教育に有効 (Quinn, 1994, 1997; Amory, et al, 1999B)

- 挑戦とファンタジー (Malone 1980, 1981 a,b)
- 好奇心を刺激 (Thomas & Macredie, 1994)
- 新奇性と複雑性 (Carroll, 1982; Malone, 1984; Rivers, 1990)
- ゴールの構築と生徒間の競争 (Neal, 1990)

シミュレーションゲーム = 教育現場で多用

- 単一のゴール
- 自分のペースで進行し，生徒間の競合が少ない

あくまで現実世界のシミュレーションだから，シミュレーションだけでは駄目で，現実の活動が必要
ゲームそのものが日常生活の一部であること，楽しむためにゲームをやることを認めない。

マイクロワールド

- 特定の領域を学習するというよりも生徒はシナリオの一部になる
- 自己管理が重要
- ゲーム中の学習者はメタ認知的/行動的にアクティブになる
- 自己評価が促進
- ドラマを提供し，チューターの役割をする

アドベンチャーゲーム (Adventure game) がマイクロワールドを学習者に提示する有効な方法

本稿の目的：教育ゲームを概念化したモデルの構築，モデルにしたがった教育ゲームの開発例，レッスン（著者自身の経験談）

2. ゲームデザイン

2.1. Quinn (1994) のゲーム開発に関するモデル

システム vs ユーザー中心，楽しさの要素，教示方法，ゲーム開発の段階が議論
教育理論とゲームデザインの関係は不明

2.2. Game Object Model (GOM) Figure 1

教育哲学とゲーム要素の弁証法的関係を構築することを狙い

コンポーネント (囲まれた四角) とそれに属するインターフェース (丸) が存在

インターフェースは重要度の順にリスト形式で配置

- インターフェースの種類
教育目的 (抽象) に関わるインターフェース (○)
主に外側のコンポーネントに含まれる

ゲーム要素 = 現実化 (具体) に関わるインターフェース ()

主に内側のコンポーネントに含まれる

• コンポーネントの種類

相互に独立しているコンポーネント

他のコンポーネントの一部となるコンポーネント

親コンポーネントのインターフェースを継承

2.3. ゲーム : Zadarh

教育目的 : 2・3年生が保持する生物学に関する誤概念を解消

トピック : メンデルの法則 (分子生物), 光合成 (呼吸作用), 人類の進化

各トピックでの誤概念を調査

調査された誤概念のそれぞれに対応する部屋 (パズル) からゲームは構成

ゲームの概要

1. 新たなウィルスを開発していた科学者が自身の生成物によって命を落とす
2. 研究所は外から隔離
3. プレーヤーはバイオハザードスーツに身を包み研究所に進入
4. 科学者が生前, 研究所に隠した抗体を発見することが目的
5. 迷宮 ($5 \times 2 \times 5$) をパズルを解きながら探索 (ナビゲーション)
6. リフトにたどり着くと次のレベルに移行
 - 人類進化レベル : RNA を回収する
 - 光合成レベル : 酸素を回収し, 二酸化炭素で火を消し, DNA を回収
 - メンデルレベル : 抗ウィルス性に分子変換された DNA を発見しワクチンを構成

ナビゲーションはマウスの使用により進行

各シーンでプレーヤーは前進, 後退, 左向き, 右向きになれる

迷宮中にはオブジェクトが落ちており, それを使ってパズルを解く

2.4. GOM の Zadarh への適用

教育ゲームの開発に関する 3 つの視点 (Figure 2)

1. 対象領域の同定 (リサーチベース)
2. ゲームリソースの生成
3. ソフトの開発

背景 : 探索と問題解決により同定された誤概念が解消されるはず

1. 誤概念と学習すべき問題を同定 (Content Area Identification)
2. 学習目標を規定 (Misconception Identification)
3. 学習目標からパズルの内容を決定 (Puzzle conception)
4. 誤概念を各部屋に割り振り, Goal formation (Discovery) インターフェースを定義
5. Goal formation インターフェースに関連した問題インターフェース (manipulation, memory, logic, mathematics, reflexes) を各部屋に組みこむ
6. パズルの解決, 新規オブジェクト・情報の発見により Goal completion インターフェースが実現できるように設計
7. 部屋の見た目/雰囲気 (graphics, sound, technology インターフェース) を定義 = 各部屋で学習すべき概念と関連したもの
8. パズルを統合しストーリーラインを完成 . 抽象と具体のマッチングがとれ, 概念レベルは完成 .
- 9.
10. リソース開発, ソフトウェア開発 (ゲームレベル, ユーザーインターフェース) = アクションリサーチを繰り返して作りこむ

3. ゲームリソースの開発

パズル、ストーリーの作成と同時進行でゲームページエディタ/プレイバックエンジンを開発 (Pascal プログラミングツール, Delphi, Borland, ver3)

各コンポーネント (ゲームロジック, イメージ, エフェクト, ナビゲーションエリア, パズル) をゲームページエディタで統合し, ゲームファイルを作成 プレイバックエンジンで再生
3次元画像は 3D StudioMax (Kinetix), その他の画像は PhotShop (Adobe) を使用して生成

3.1. ゲームの構成

開発チームのメンバーは始めに市販のゲームをプレイ (Myst, Rama, Zork Nemesis)

市販ゲームの技を身に付ける

画像を作成する前に全体的なグラフィックのビジョンをもつ

各ナビゲーションポイントでプレイヤーは 4 方向 (東西南北) に移動可能にデザイン オブジェクトの配置, カメラアングル, シーン間の連続性, テクスチャ・マッピング, ナビゲーションの補助を注意深くデザイン

3.2. オブジェクトの配置

現実性のため, オブジェクトを正確にスケーリング, 論理的な位置に配置

- 異なる視点から各オブジェクトが正確に配置されることをチェックするために, 視野を一定に保つ
- 存在したりしなかったりするオブジェクトを複数の視点から見ることでできるシーンを作らない. 壁を配置するなどオブジェクトを複数方向から見るできないように工夫

3.3. カメラ

カメラアングル (高さ, 角度) を一定にたもつ = 1.7M の高さ, 下向き 4° に設定

カメラアングルはプレイヤーの目. こころと変わると混乱してしまう

3.4. スケール

50mm レンズは歪む 35mm レンズを使い人の視野を再現

カメラの視野にしたがって, オブジェクトのスケールが自然に変化するよう工夫

3.5. テクスチャ・マッピング

色, きめ, 輝度, 透過性といった表面の特徴をオブジェクトに割り当て

- 大きいオブジェクトに素材を並べて貼るときには, なるべく自然に見えるように工夫
ビットマップのエッジ (上下左右) の色調を統一し, エッジを目立たなくする
- 現実性を出すためにオブジェクトに汚れをつける
汚れを再現するために, 素材が貼られた後に汚れのマークを微妙にくっつけた
- 本棚のイメージを構成する際にファイルサイズ, テクスチャの貼りつけ時間が膨大に増加
本 1 冊ずつにビットマップを貼りつけていたけど, 本棚全てを覆うテクスチャを貼りつけることにした

3.6. フロアのプランとイメージの生成

- 部屋の中での位置の区切りは 3 文字のエリアコードで表示
- グラフィックファイルは xxxyyd 形式でラベル化
xxx = エリアコード, yy = カメラ位置, d = 方向
- 最初のグラフィックファイルは 100 万色, 540 × 405 ピクセル, サイズは 640kb
作業行程のなかで 256 色, 214kb まで減らす

- 各グラフィックファイルの保管に最低 1.25MB のディスクスペースが必要
ゲームの最終バージョンは 8Gb の SCSI ハードドライブをもつコンピュータに収集

3.7. ゲームページの開発

ゲームページデザイナーによりグラフィックファイル，ホットスポット，インタラクティブロジックを統合

ゲームファイルは単一のフロアエリアに対応

全ての要素を単一のゲームファイルに組みこんで保存

各エリアに関連付けられたイメージ，ゲームロジックファイルを含む

多数の視点を含む部屋の時にファイルサイズが 250Mb を超えたので，大きな部屋は部分（エリアコードに対応）に分けて保存

3.8. プレーヤー

200MHz の CPU，64Mb の RAM，16 ビットカラー，Windows NT 上のマシンで DirectX を使用してプレイ．各端末は LAN に接続

効果（水や火），アニメーションはゲームエンジンに組みこむ．パズルはゲームエンジンの外部（DLL）に構成
ゲームエンジンとパズルを区別することで，エンジン外部に実行コードが貯蔵され，複雑で異なるインタラクシオンタイプが実現

4. ユーザーインターフェース

生徒のプレイ中のインターフェース使用を観察し，改良を加えながら構成

ホットキー：ほとんどの生徒はコンピュータを使用した経験があまりなかった．経験のある生徒は F1 でヘルプが呼び出せることにすぐに気付いたが，ノービスは気付かなかった．そこで，全ての機能をマウスで呼び出せるようにインターフェースを改良（Figure 3）

アイコンの理解：アイコンを理解できなかった．よって，各インターフェースは明確に同定できるように，マウスをアイコンに置いたときにラベルが表示されるように改良．

ナビゲーションの補助：矢印を表示することでプレーヤーの進行方向を示した（前進，後退，左に向く，右に向く）．しかし，初心者はこれがよく分からない．よって，方向を変更するときは二つの画像をブレンドした．これによって，カーソルの動きと視野の動きを繋げることが可能に．

5. 結論

Adventure ゲームの開発に関する理論的な裏付けを示し，開発のプロセスを説明

特に Zadarh を開発する上で生じた問題とその解決を提示

GOM は教育哲学をゲーム開発に繋げる上で有用であった

ゲームリソースの生成は複雑である

本稿のように，一般的な問題の解決策を示すことで，今後のゲーム開発がスムーズに