

大規模RPGのための シナリオ記述言語と 統合開発環境の開発

東京大学大学院情報理工学系研究科

清木 昌

プロジェクト管理組織

株式会社メディアフロント

概要

● ゲームシナリオ記述に特化した言語と環境

対象: ストーリー展開を本質とするゲーム

- ロールプレイングゲーム(RPG)
- アドベンチャーゲーム(ADV)

1. 様々なゲームで汎用的に利用可能なエンジン
2. データの管理がしやすい開発環境の整備

ゲーム開発における生産性の向上

- 特に自動検証によるデバッグコストの削減

背景

● ゲームシナリオの持つ特徴

- 膨大なデータ量・複雑な条件分岐



● 非力なテキストベースのスクリプト言語

- if - goto による条件分岐が基本
- BASIC や C のサブセット or それ以下

制作現場での問題

- 膨大で複雑なシナリオデータの入力・管理コスト

目的

- 生産性の向上への2つのアプローチ
 - シナリオ記述言語のモデルを作り替える
 - if - goto のみの世界からの脱却
 - 状態遷移を基本とした実行モデルへ
 - 開発環境を整備する
 - テキストエディタだけでは実現できないサポート
 - 編集対象を関連シーンで絞り込む機能等
 - シナリオの整合性の自動検証
 - デバッグコストの多大な低減
- 「汎用で使用可能なベースシステム」の提案

機能

● GUI によるシナリオ編集機能

- ブロック(シーン)毎にツリー状に管理
- 関連するブロックの検索・絞り込みも可能

● 自動検証

- 発生しないイベントの検出
 - 初期状態から到達できないイベントを発見
- ハマリ状態の検出
 - 遷移の向きを逆にし、エンディングから到達できない状態が存在すれば、そこでハマリが存在

今後の展開

● 実用レベルへの改善

- UIの向上(テキストエディタよりも使いやすく！)
- 自動検証機能を強力に(CTL程度の記述力は)
- 既存エンジン用スクリプトへの変換出力を可能に
- データの整形出力機能(確認用・アフレコ台本)

● ネットワークゲーム対応へ

- 人力の整合性検査が困難 自動検証の重要性
- 安心して動的にシナリオを変更可能に

Appendix

