



RebRank

[<http://www.rebrank.org/>]



Game / Software Creating Circle

RebRank

シューティングの おもてなしシステム

2012/06/15(Fri) 改訂版

Yoko

Twitter @blankvision



Project Blank(P.N.B.)

サークル紹介な

- RebRank (リブランク) …メンバー7人
- 当たり判定のあるゲーム

C68～(2005/08) RebRankとしてコミケに参加

C61～(2001/12) 大学サークルでコミケ参加(Amusement Makers)



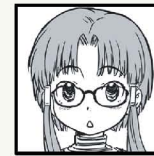
Yoko

- 敵/弾制御スクリプト
- 企画進行・渉外
- 各種グラフィック
- モデリング
- サウンド



Sheile

- メインプログラム
- スクリプトエンジン



らいね

- グラフィック全般
- 背景
- DTP全般



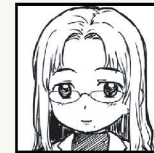
hajime

- サブプログラム
- 操作系・ライブラリ
- アーカイバ
- スクリプトエンジン



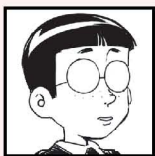
竜ライター

- 音楽
- サウンド



myz

- 世界観設定
- シナリオ



muraro

- キャラクター
- イラストレーション

※似顔絵は、すごく個人差があります



制作物

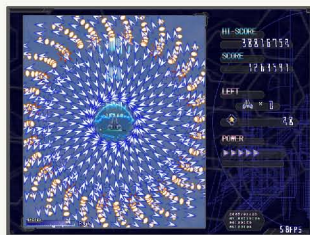
- **RefRain ~prism memories~** 戦略的循環攻撃型弹幕撃込シューティング
→関連作◇RefRain Fragments.c81 ◇RefRain Chronicle[仮称]



- **RedRive** スクウェアアクション
→関連作◇ SimpleTwitterClient

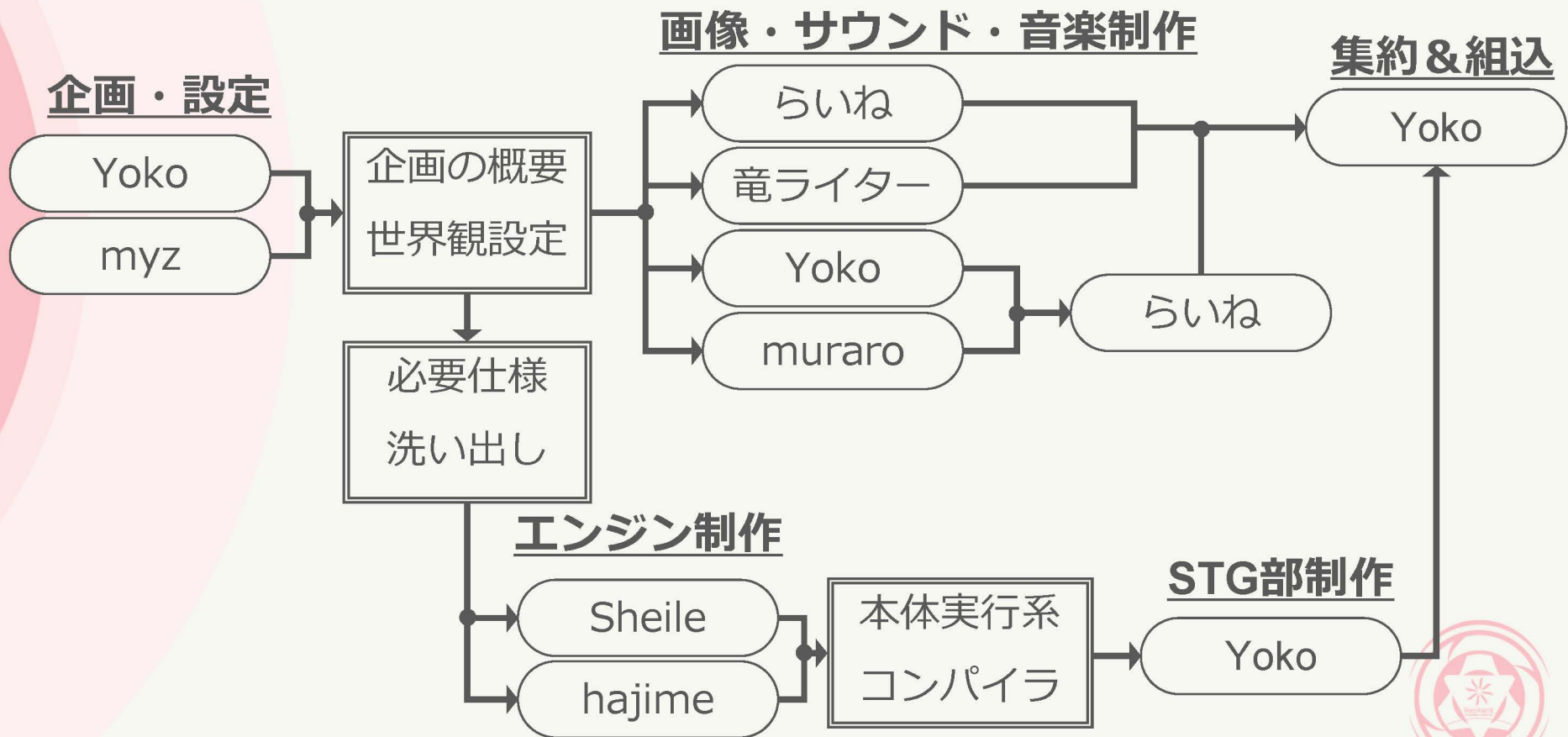


- **五月雨 ~samidare~** 戦略的近接撃込型弹幕シールドシューティング
→関連作◇SAMIDARE ARCHIVE - M.R.S REPORT -



おおよその作業の流れ

- 大体こんな流れで作ることが多い



作業環境

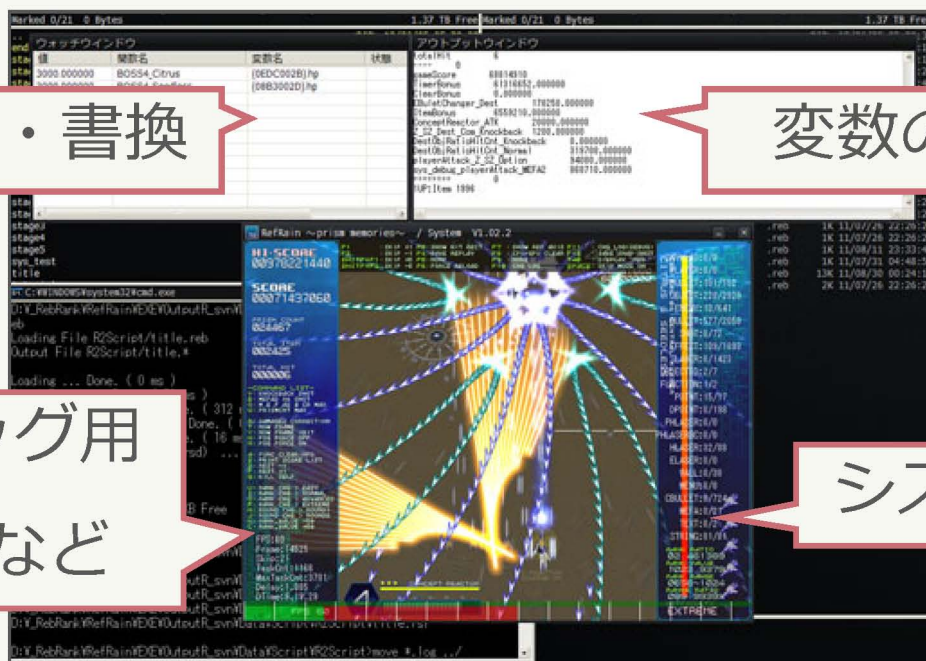
- 独自スクリプト言語 (R2Script / SMDScript)
 - C言語風のコンパイラ型言語
 - ゲーム内容は全てスクリプトで記述
- リアルタイムで変数出力・監視・書換が可能

変数のモニタ・書換

変数の出力

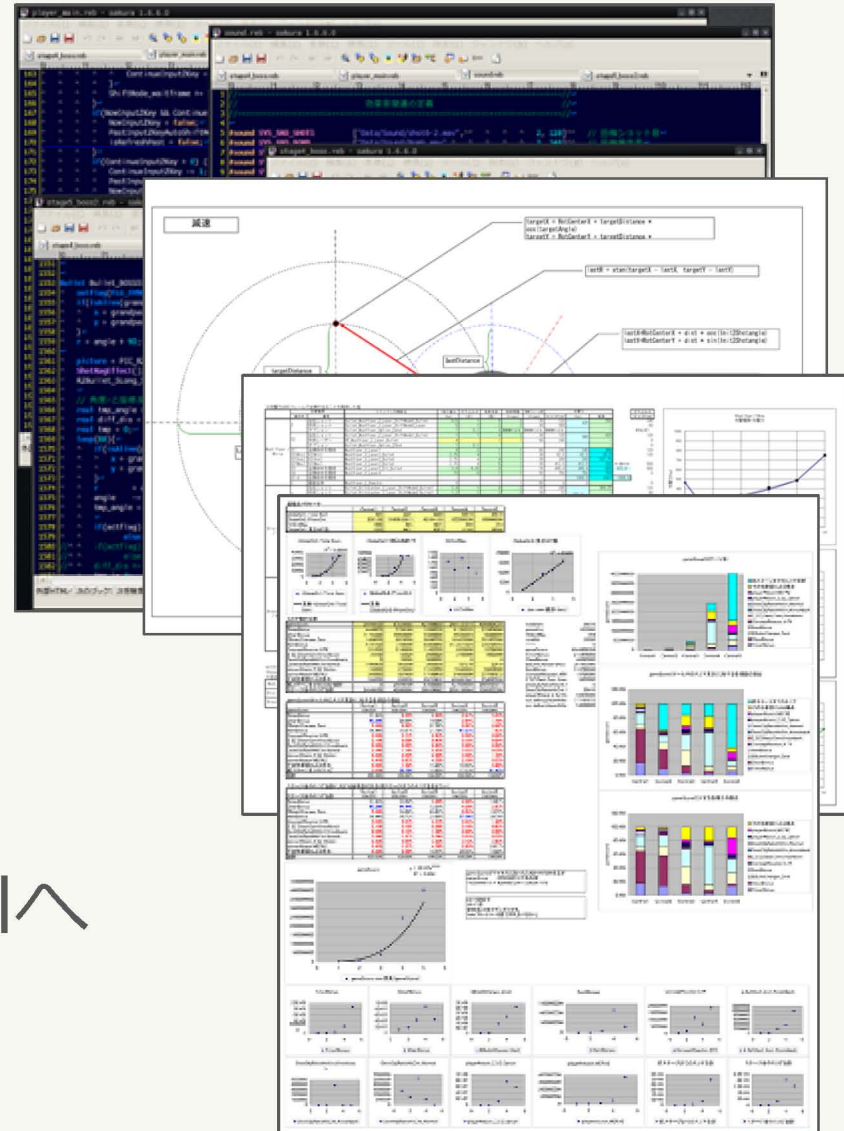
各種デバッグ用
コマンドなど

システム値表示



スクリプト作業時の環境

- 作業方法
 - サクラエディタを利用
 - ひたすら頑張る
 - コンパイラに投げる
 - 実行
- パラメータの管理方法
 - デバッグ出力値をExcelへ
 - 値の整理、予測と検証



本題

シユューティングの おもてなしシステム



おもてなしのシステム

ゲームの内容・手ごたえを変化させて
より心地よく死んでいただくために
初心者にはゆるめ、上級者には手ごたえある

たのしい内容を提供する

弾幕によるおもてなし例



おもてなしシステムの例

①難易度のシステム

1. 静的な設定…選択設定
2. 動的な設定…**ランクシステム**

②得点のシステム…**スコアシステム**

加算方式・減算方式…だいたい加算方式
→値が増えていく方が楽しい



理想のランクシステム

プレイヤーの行動から腕前を評価し
それ合わせた難しさを提供する



適度な緊張感 & 今日の俺TUEEE感を演出

- 1980年代のゲームで既に見かける
- 基本的にランク上昇の方向で調整されている
- 同人でもさりげなく搭載のものが多い？
- ゲームの難易度をそれとなく微調整
より心地よく死んでいただくために



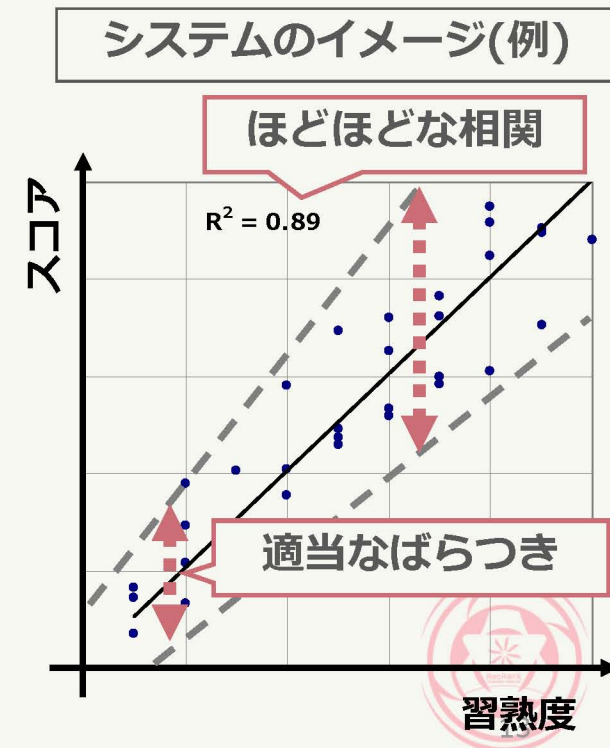
現実のランクシステム

- パラメータが多いと調整がむずかしい
 - 調整次第では良い意味で別ゲームになる
 - 体感難易度が丁度いい感じになるメリットもある
- ゲームが歪む or 理想的に働かない例も…
 - ランクが上がりすぎてクリアできない…
 - パワーアップアイテムを取らずに落とせばよい
 - クリアには適度なボムと自殺が近道
 - 筐体の電源をOFFすればランクが下がる
 - 無駄弾を撃ってはならない
 - ランクを上げて物理で殴れば撃込点が稼げる
 - 赤い鳥に撃ち込んで稼ごう



スコアシステム

- プレイ内容を得点で評価
 - 客観的な「値」で示す（値以外の手段があればいいのですが…）
- 得点を稼ぐ = 特定の行動を推奨
 - ①理想的なプレイ
 - ②危険行為を推奨（≠理想的プレイ）
- システムは外乱に強い必要がある
 - 安定性・ロバスト性
 - ほどほどの相関とばらつき
 - 腕に応じた見返りを返す



例：五月雨の場合

- 難易度のシステム
 - 固定難易度
 - Normal/Extraの2構成
 - 簡単な表、少し難しい裏



Normal



Extra

- スコアシステム
 - プレイスタイルとスコアが直結する傾向
 - おおよそ線形近時
 - 理想的プレイ = 高いスコア
 - 近接破壊でアイテム回収、ハイパーで倒せば高倍率
 - クリアにはパターン化が必要
 - システムを理解しないと進むことが出来ない



例：RedRiveの場合

- 難易度のシステム

- ルート選択式
- プレイヤーの意思で決定
- 下エリアほど高難易度

易しい

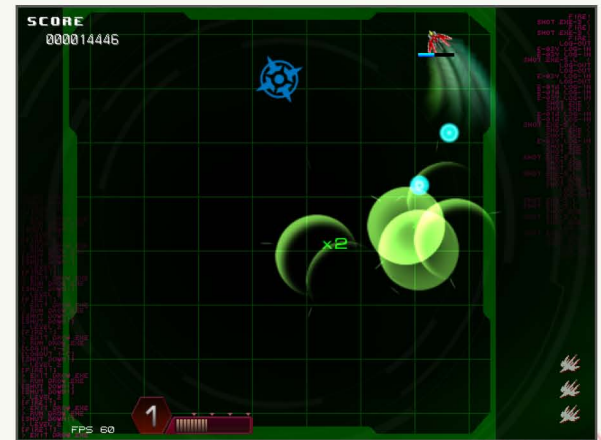


難しい



- スコアシステム

- ややストイックな方向
より近くで敵を囲むと高倍率
素早くクリアすれば高得点



例：RefRainの場合(1)

- 初期難易度の選択

- ① 初期選択：4種類

- ② 周回数：2周程度

計8難易度 3キャラクタ



- ランクシステム

- 難易度各々でランク値の階段を設ける

- 階段数：浮動小数点数型 有効数字7ケタ

- プレイヤーの行動からランク係数を算出

- 係数をもとにして各種パラメータを自動補正

- 意図的なランク調整は意味がない程度に抑える

ランク係数
ランク実行値
変動範囲
変動率
… etc



例：RefRainの場合(2)

- スコアシステム
 - プレイスタイルのスコア影響を見直し
 - 理想的プレイ = 必ずしも高いスコアではない
 - 自機システムを使いきる = 最終的にスコアが高い
高いスコア ≡ 魅せプレイになることも
 - システムの循環 … おおよそ累乗近時
 - 自機行動 & 要素をリストアップして相関付け
 - 特定箇所 & 特定条件で各種分岐
- クリア要素とスコア要素
 - 1UPとスコア周りは分離、死亡時には救済補正



まとめ

- おもてなしシステム
 - やりこみ（深度）を深くする方向に働く
 - ゲームの幅は必ずしも広がらない傾向
 - パラメータ設定と見極めがかなり難しい
- スコア稼ぎ
 - 針の穴に糸を通し続ける作業
 - ①スコアとゲーム性を一致させる場合
 - ②スコアとゲーム性を必ずしも一致させない場合
 - 各々でメリットとデメリットがある

