東方地霊殿AIシステムの改良

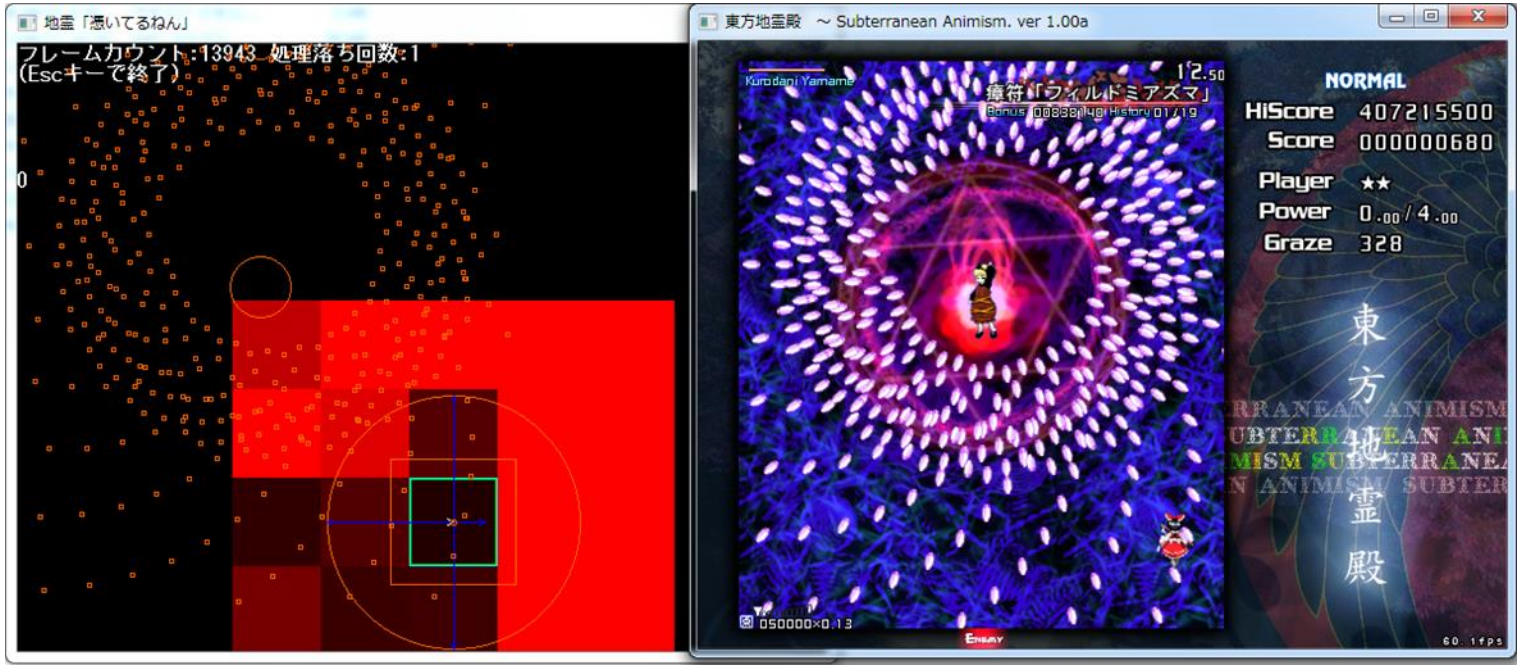
@ide_an

故きを温ねてなんとかかんとか

過去の話

東方地霊殿AIシステム

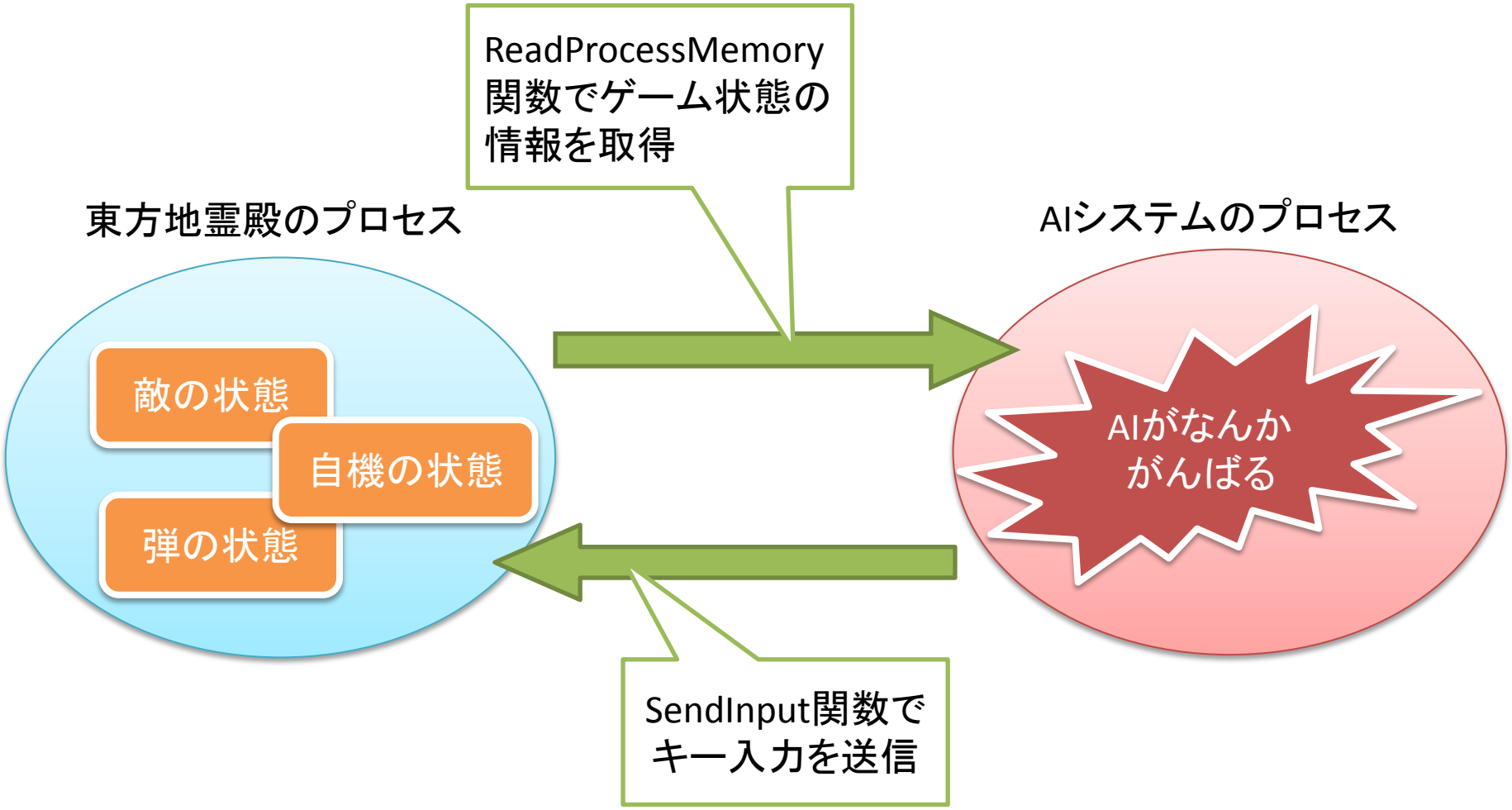
- 東方地霊殿の自動プレイシステムを開発
  - 2012年度新歓展示にて展示
  - 同年度前期研究報告会にて発表



AIシステムの概要

- C++製
  - AIもC++でごりごり書いた（つらい）
- 動作の仕組み
  - 東方地霊殿のプロセスのメモリから
    ゲーム状態の情報を読み取る
  - 個の情報を元に次に行う操作をAIが決める
  - 行う操作をキー入力として送信

動作の仕組み



※ プロセス = 動いているプログラム1個分

または妄想の話

未来の話

突然ですが

花映塚AI作りたい

花映塚とは?

- 東方花映塚
- “対戦型”シューティングゲーム
  - 人間 vs 人間
  - 人間 vs AI
  - AI vs AI

1人間　　1AI

弾を避けつつ雑魚敵を倒したりして
がんばると敵陣に弾とか送れて
たいへんになるゲーム

なぜ花映塚AI?

- AIとの付き合い方が多様になる
  - 作って眺めるだけの関係じゃない
    - 自作AI vs 自分
    - 自作AI vs 他の人
    - 自作AI vs 他の人の自作AI
- 他の人と競う楽しみがある
- 攻撃にも戦略性が生まれる
  - 通常のSTGでは避けに偏りがち

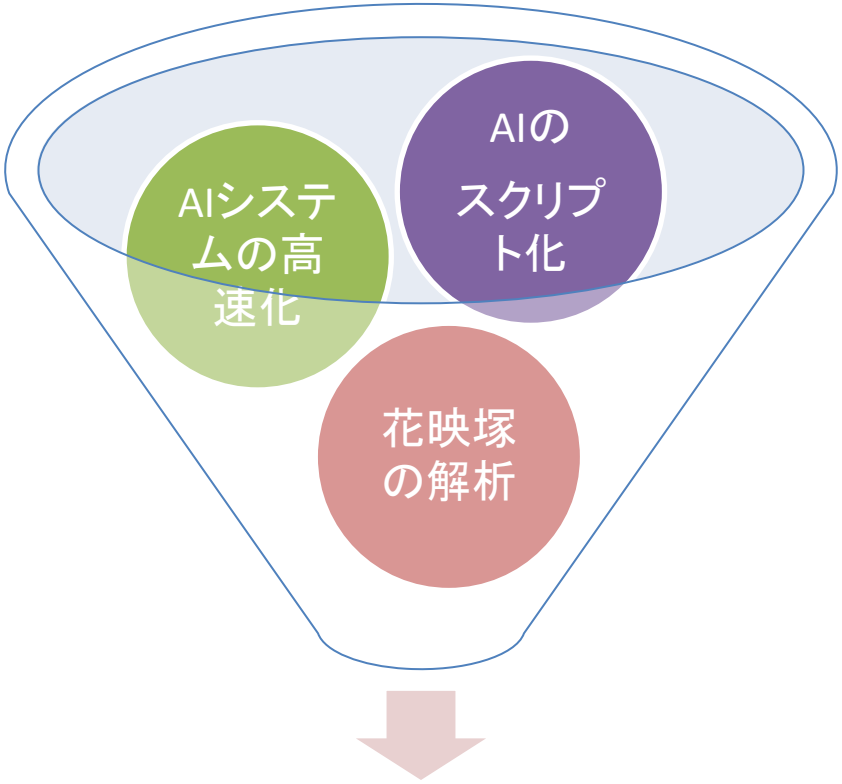
このへんとか

あなたのAI vs 僕のAI

・・・を実現するには

- AIを動かすシステムとAIそのものとを切り離す必要がある
  - AIのスクリプト化
- AIをスクリプト化したら実行速度が落ちるのでは?
  - AIシステム自体を高速化すればいい

できるか？！花映塚AI



花映塚AIシステム!!!

“YOU CAN (NOT) PROGRESS” - @osa_k

現在の話

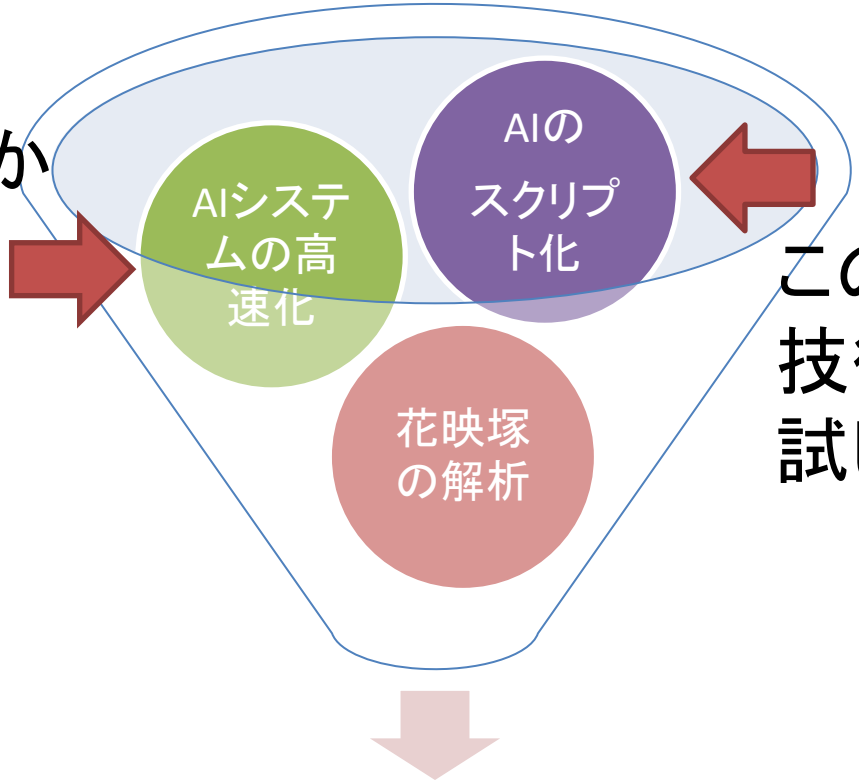
で、

進捗どうですか?

修論があるんや・・・

要素技術を押さえる

この辺とか



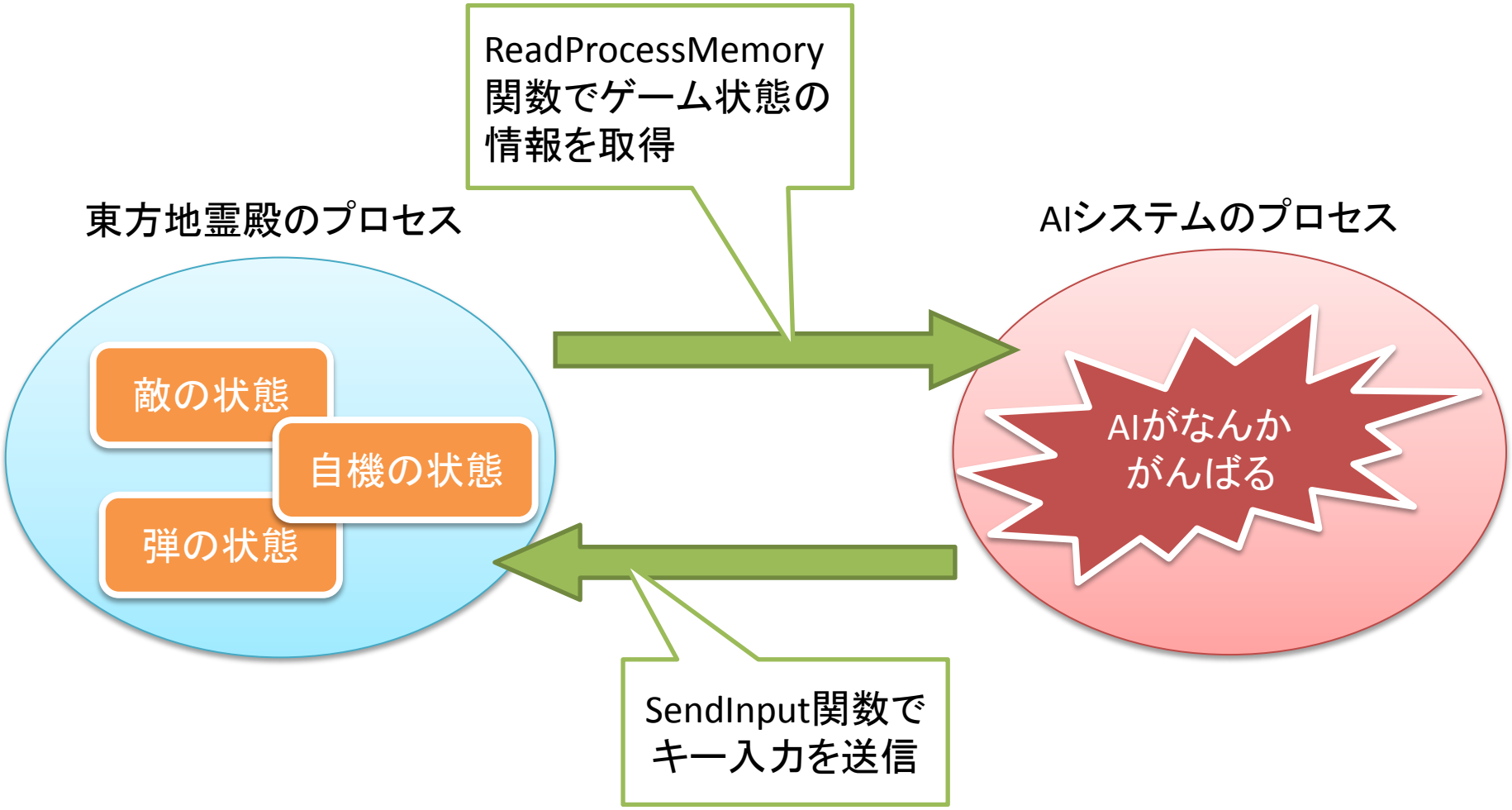
この辺の技術を試してみる

花映塚AIシステム???

AIシステムの高速化

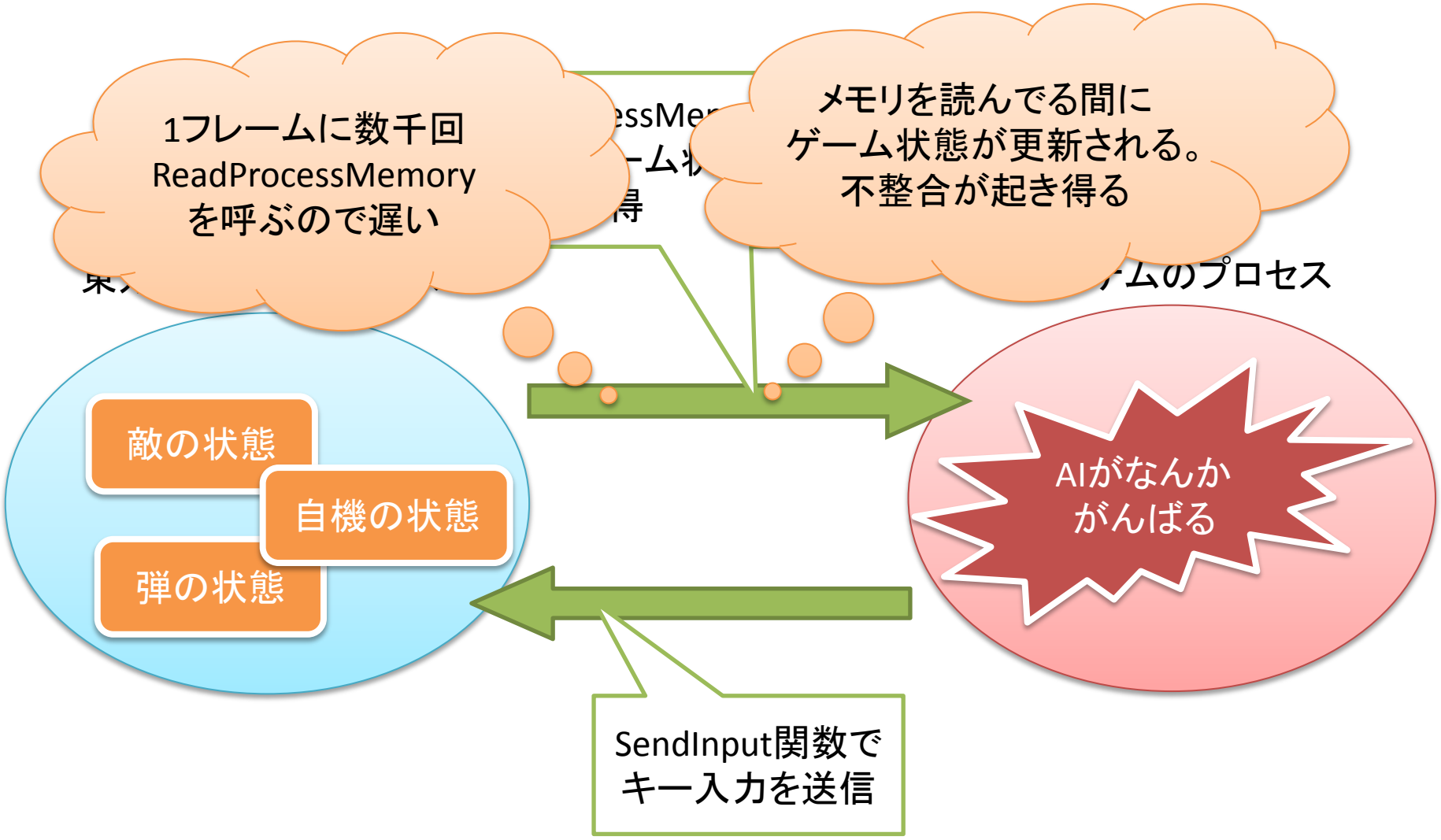
- どうする?
  - 地霊殿プロセス内でAIを動かす
- なぜ?
  - 別プロセスのメモリを読むのはオーバーヘッドが大きい。これを回避したい
  - データの不整合を解消できるという御利益もある

従来のシステム



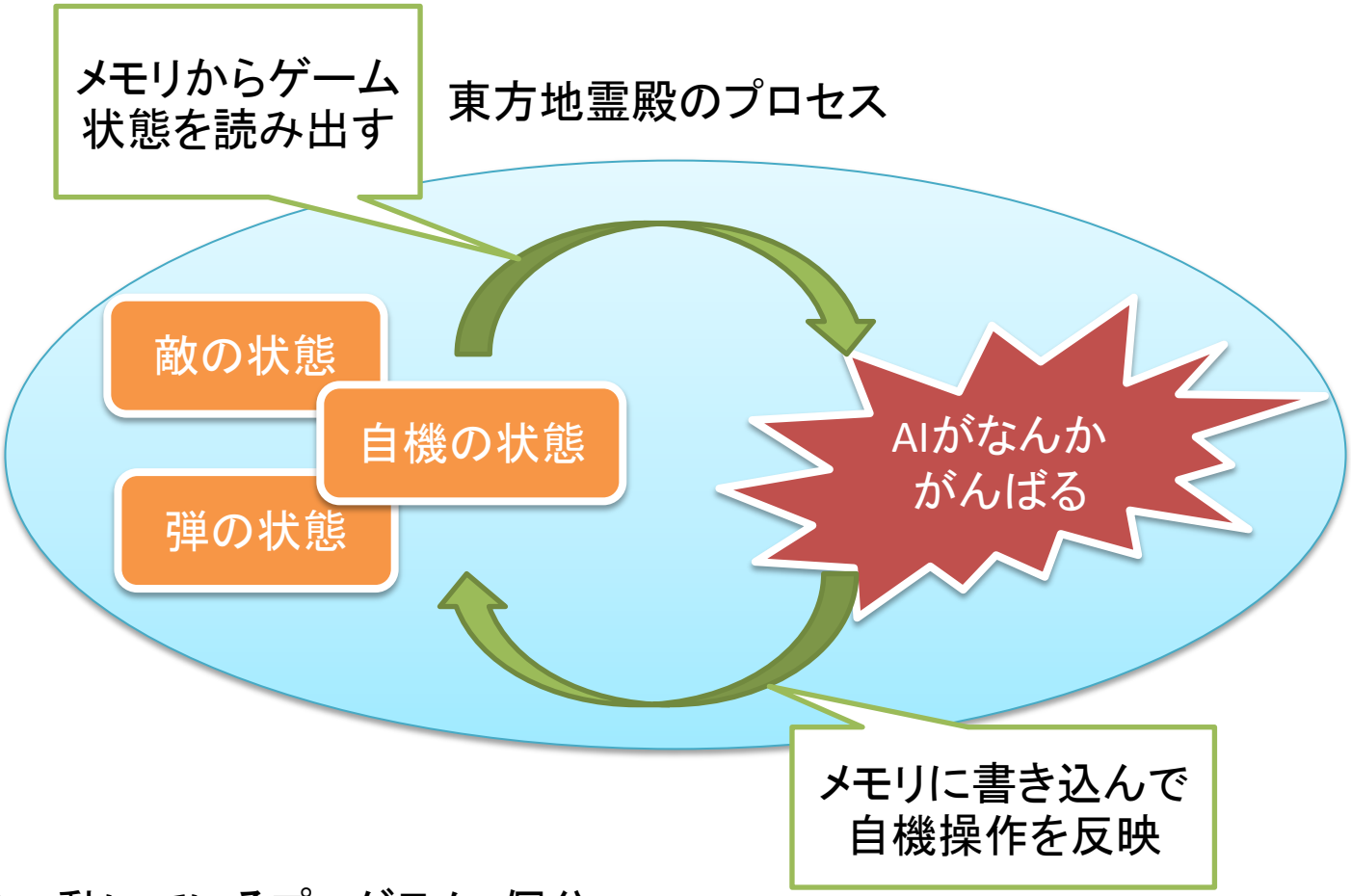
※ プロセス = 動いているプログラム1個分

従来のシステム



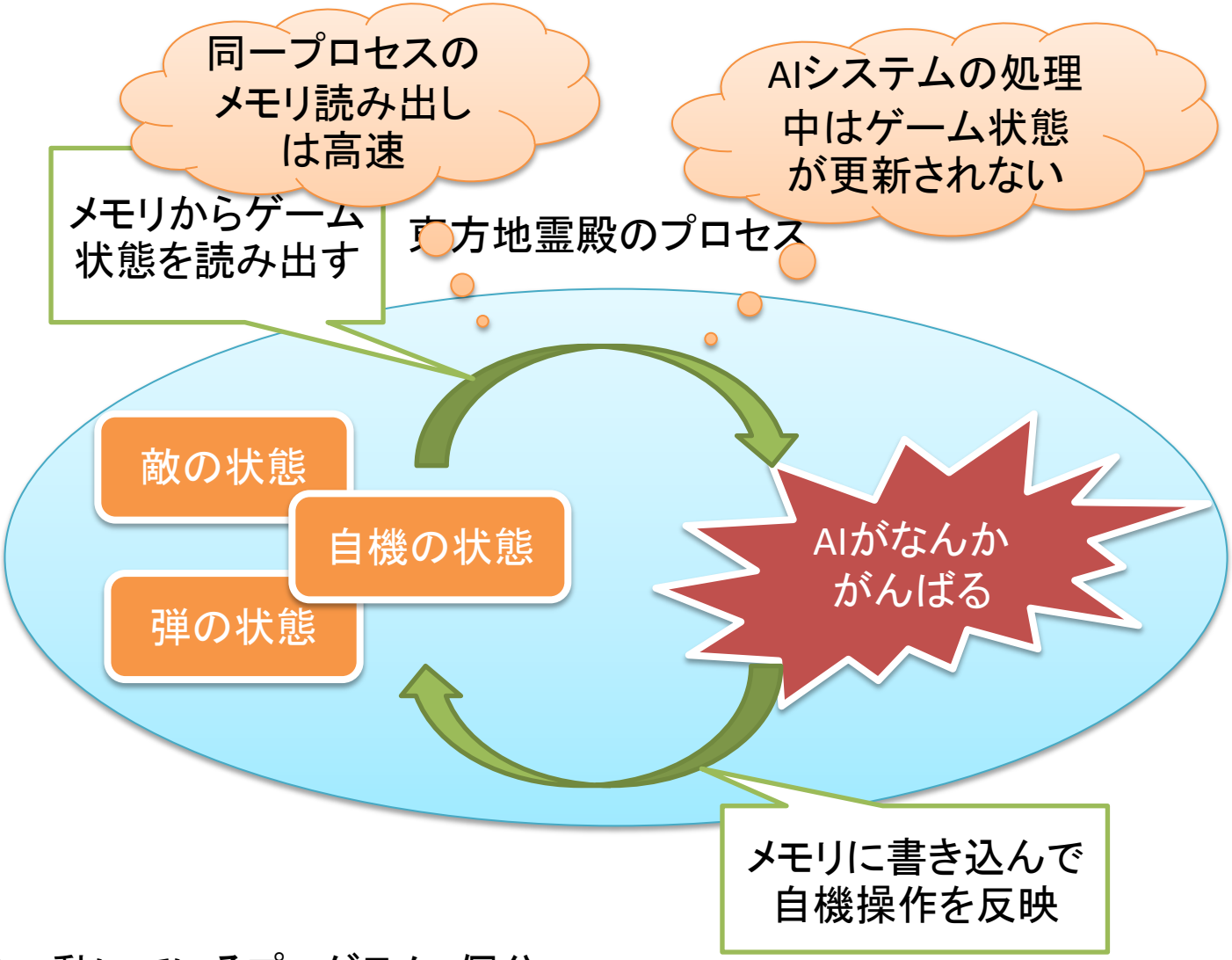
※ プロセス = 動いているプログラム1個分

同一プロセスで動かす



※ プロセス = 動いているプログラム1個分

同一プロセスで動かす



※ プロセス = 動いているプログラム1個分

同一プロセスで動かすには?

- DLLインジェクションを用いる
  - AIシステムをDLLとしてロードさせる
  - ロードさせたらメモリ上の地霊殿プログラムを書換えて毎フレームAIシステムを呼ばせる
- 詳細は研究報告書を参照

- 追記: 参考書籍
  - たのしいバイナリの歩き方
    - http://www.amazon.co.jp/dp/4774159182
    - 生協に置いてある

高速化の結果

環境	実装	1フレーム当たりのゲーム状態取得にかかる時間(平均)
いい環境 (CPU: 3GHz/RAM: 4GB)	従来の実装	3.358 msec
	高速化した実装	0.098 msec
しょぼい環境 (CPU: 1.2GHz/RAM: 4GB)	従来の実装	6.613 msec
	高速化した実装	0.229 msec

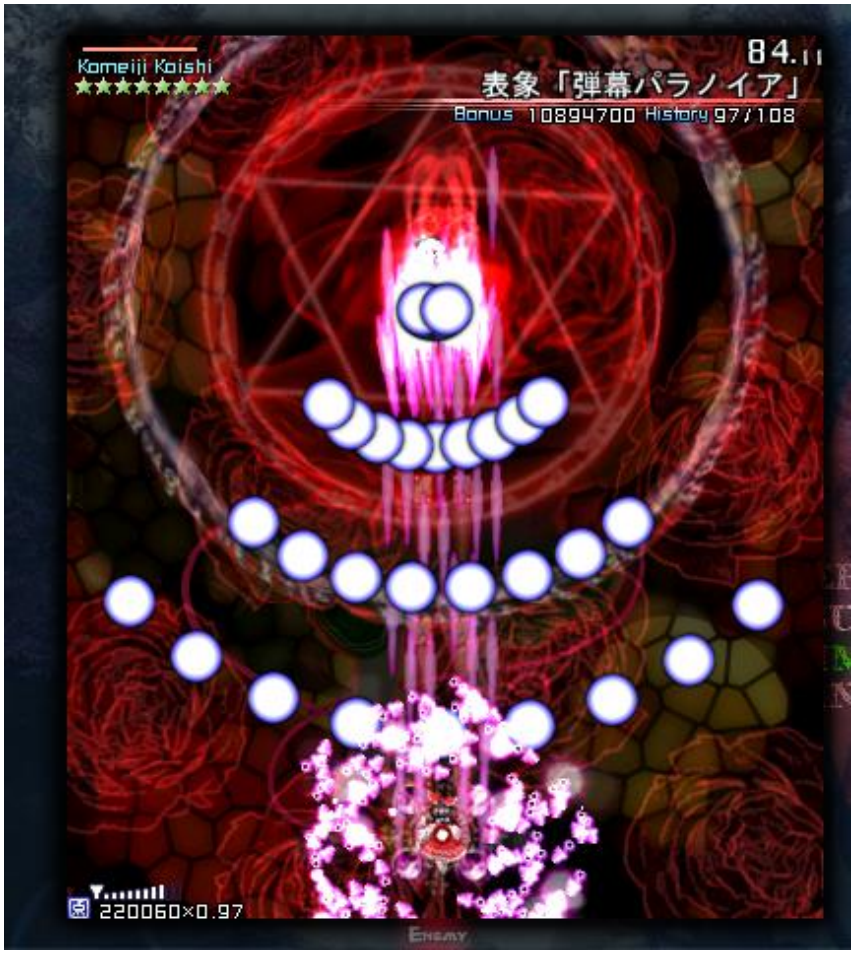
- 30倍の高速化を実現
  - 最良ケースで100倍、最悪ケースでも10倍速い

AIのスクリプト化

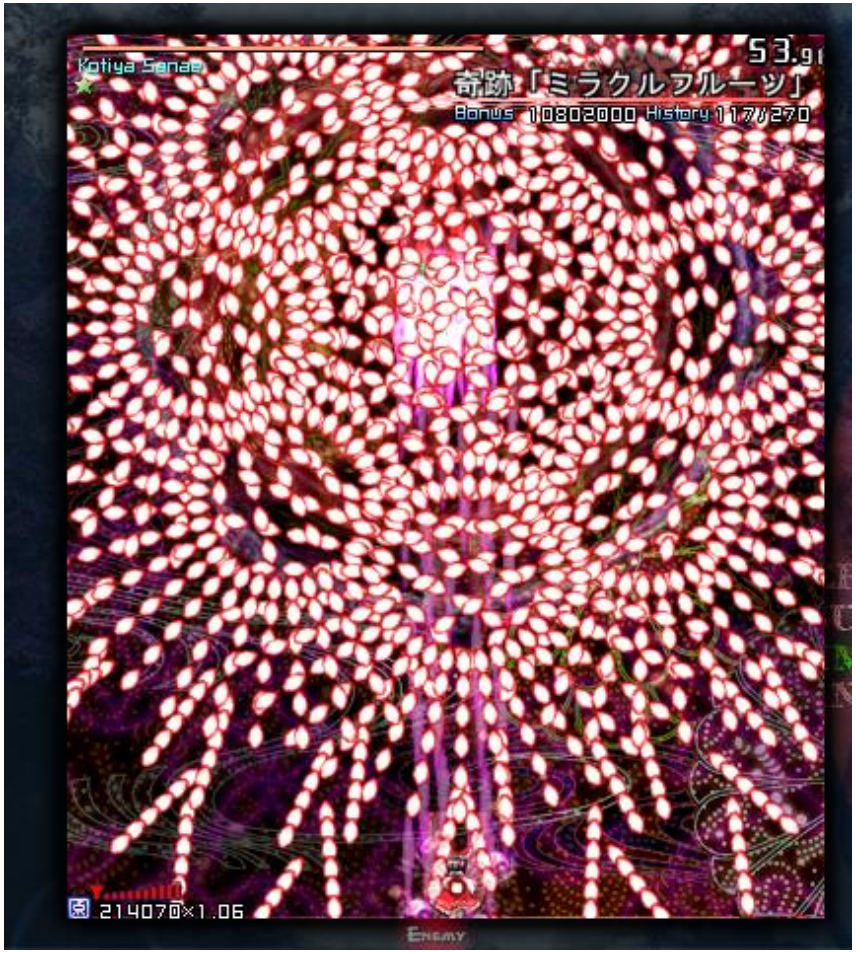
- AIシステムにLuaを組み込む
- 使用した実装
  - 本家 Lua 5.2 スタティックライブラリ
    - DLLを使おうとしたらうまく行かなかった
      (設定ミス? DLLインジェクションとの関連?)
  - LuaJIT 2.0.2の組込みについても確認
    - こっちは普通にDLLが使えた

実行速度について

- いい環境なら無問題
- しょぼい環境だと・・・
  - 当たり判定処理が多いとしんどい
    - 1フレームに4千回以上だと処理落ちする
  - GCの時間がかかるケースもある?
    - 現状のAIシステムはオブジェクトの使い捨てが多い
      - 弾が多い場面だとGCが頻発
    - LuaのGCはインクリメンタルだが、1ステップ当たりの停止時間を調整できない?



自機の周りに弾が多い
→当たり判定処理を減らしづらい
→処理落ちする



弾が多い(2千個近い)
→オブジェクトを大量に生成して大量に捨てる
→GC頻発、しかも長引く
→処理落ちする

スクリプト向けAPIの構成

- 毎フレームmain関数が呼ばれる
  - 戻り値として次回の自機の操作を返す
- ゲーム状態の取得
  - グローバル変数で提供。
    毎フレームAIシステムが更新する
- 当たり判定処理
  - 自機との当たり判定のみを行う関数を提供

スクリプト向けAPIの構成

- 正直今のAPIは微妙
  - 汚い
  - 面倒を避けようとしてかえって面倒臭くなってる
  - スクリプトからできる操作を増やすべき
- セキュリティへの考慮が必要
  - ファイル操作系やos.execute()などは少なくともそのまま提供してはいけない

デモ

- 回転移動縛り #とは

まとめ

- DLLインジェクションによる高速化
  - 実によろしい
- AIのスクリプト化
  - GCを頻発しないAIシステムへ変えるべき
  - API構成やセキュリティを詰めていく必要あり

- 花映塚AIを作りたい人生だった

Appendix. 東方界隈のAI

- th075booster
  - 萃夢想AI。リプレイ等から学習する。公開停止
- AIに『ダブルスポイラー』をプレイさせる本
  - 画像処理でゲーム状態を取得するAIの同人誌
- th105_ai(のちにth123_ai)
  - 緋想天/非想天則AI。AIをLuaで記述。別プロセスから監視
- th135_ai
  - 心綺楼AI。AIをmrubyで記述。DLLインジェクションして監視。SquirrelのVMを奪取する