



"FINAL FANTASY VII REMAKE INTERGRADE" における自動リプレイ、自動探索、自動バグ 分類の理論と実践

太田 健一郎

AIエンジニア SQUARE-ENIX

© 1997, 2021 SQUARE ENIX CO., LTD. All Rights Reserved.

CHARACTER DESIGN: TETSUYA NOMURA/ROBERTO FERRARI

© 2021 SQUARE ENIX CO., LTD. All Rights Reserved.

SQUARE ENIX

アジェンダ

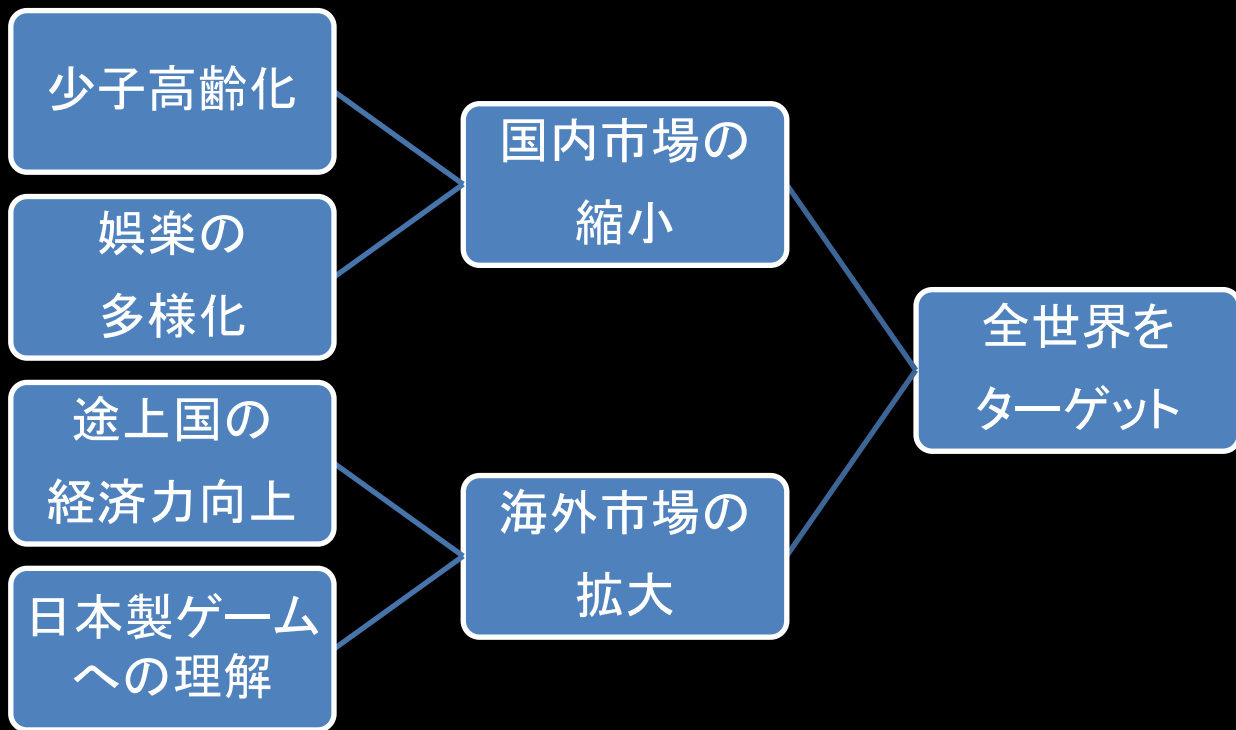
- 担当と経歴
- 開発の背景
- 開発目的と概要
- 各種機能の解説
- 今後の開発

担当と経歴

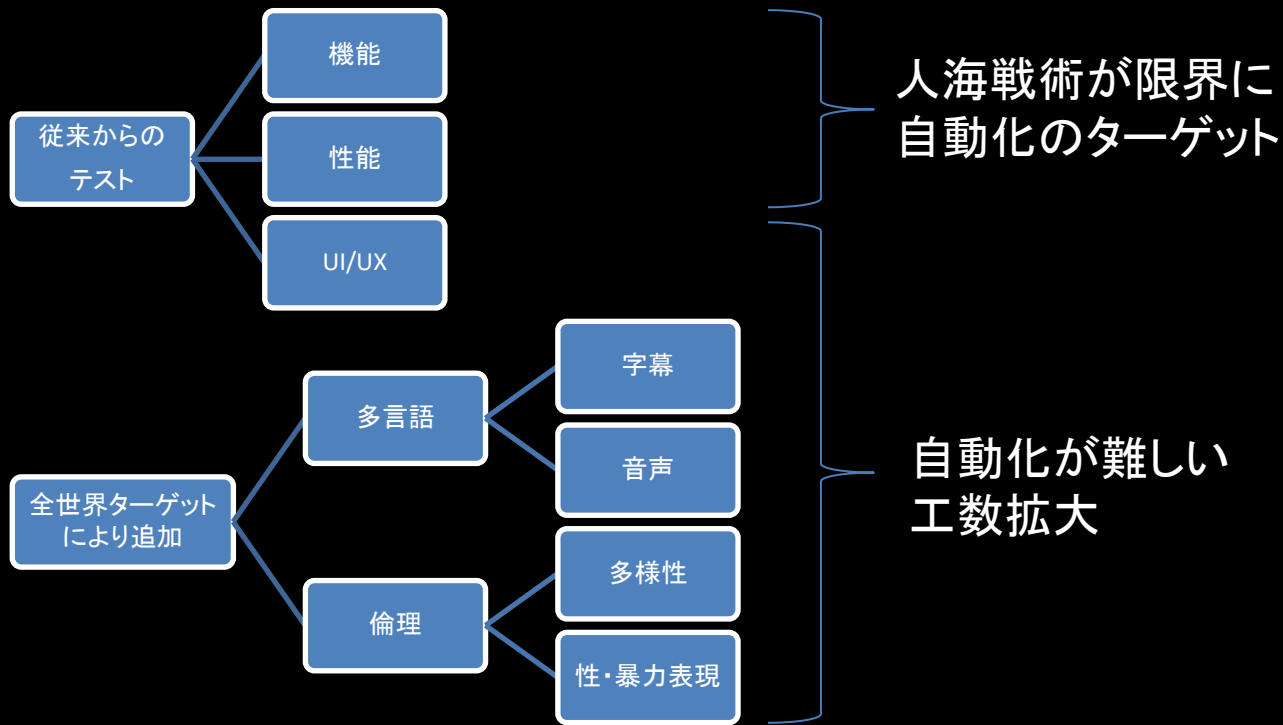
- 担当
 - 自動テストツール バックエンド開発運用
- 経歴
 - ソフトウェアテストのデザインパターンの研究
 - 開発支援ツール開発
 - エンタープライズシステム開発
 - スマートフォンゲームテスト支援
 - システムテスト自動化
 - ゲームテスト自動化 ← 現在

開発の背景

現在の家庭用ゲーム市場



家庭用ゲームのテスト



開発プロセスと人員構成

デザイナー



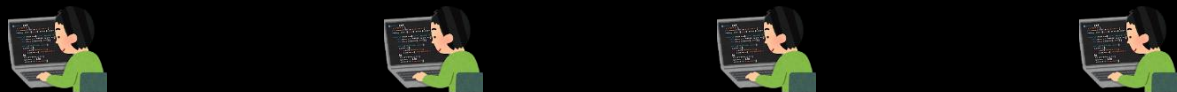
プリプロ

α

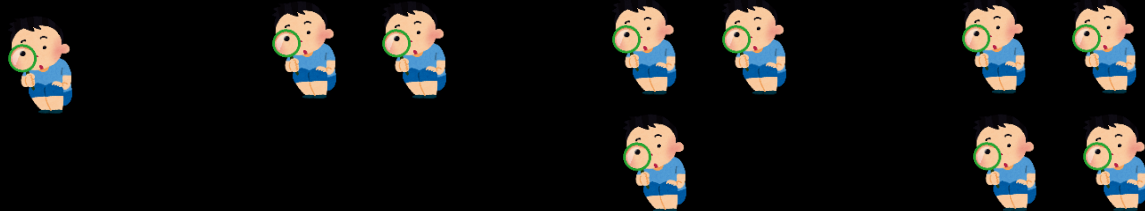
β

マスター

プログラマー



テストエンジニア



ソフトウェアとしての特殊性

- プログラム : コンテンツ比 (量、サイズ)
 - 3 : 97
- 量産するもの
 - コンテンツ (音声、地形、キャラクター)
- プログラミングの位置づけ
 - ゲームエンジンのタイトル向けカスタマイズ
 - コンテンツを作成するための部品の開発

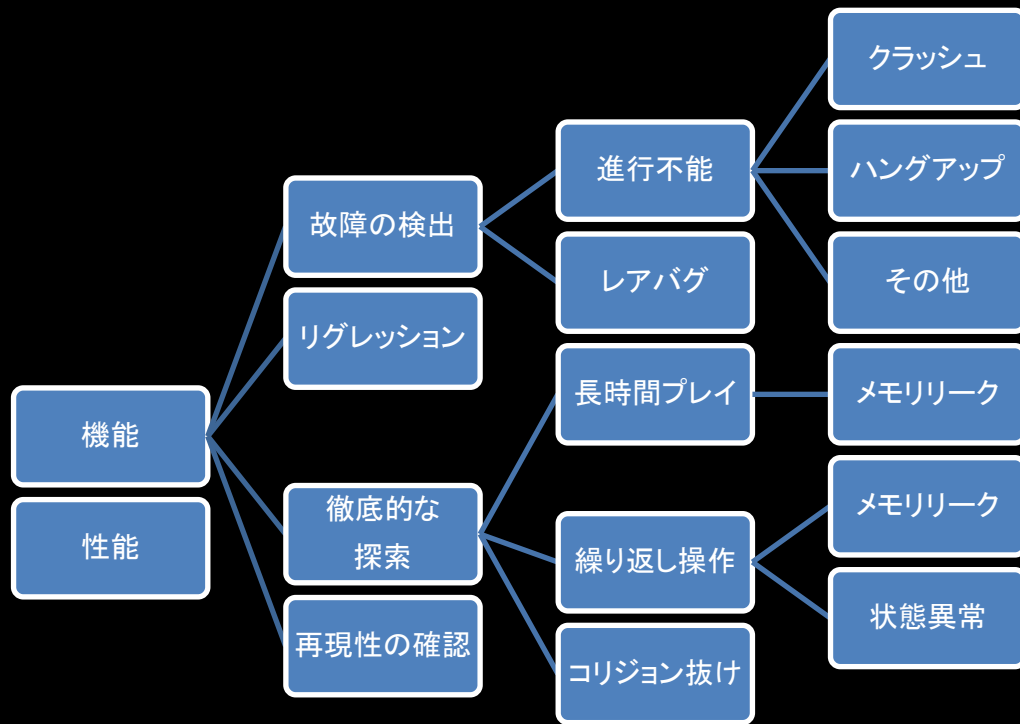
バグの特性

- 発生しやすいバグ
 - コンテンツ間の結合
 - プログラムとコンテンツの結合
- 発見が可能なテストフェーズ
 - システムテスト
- バグ
 - スレッドバグ
 - デッドロック
 - メモリ破壊
 - 検出方法
 - 形式手法
 - 繰り返しテスト
 - 負荷テスト



システムテストで全結合しないと発見できないバグが多い

自動テストの対象

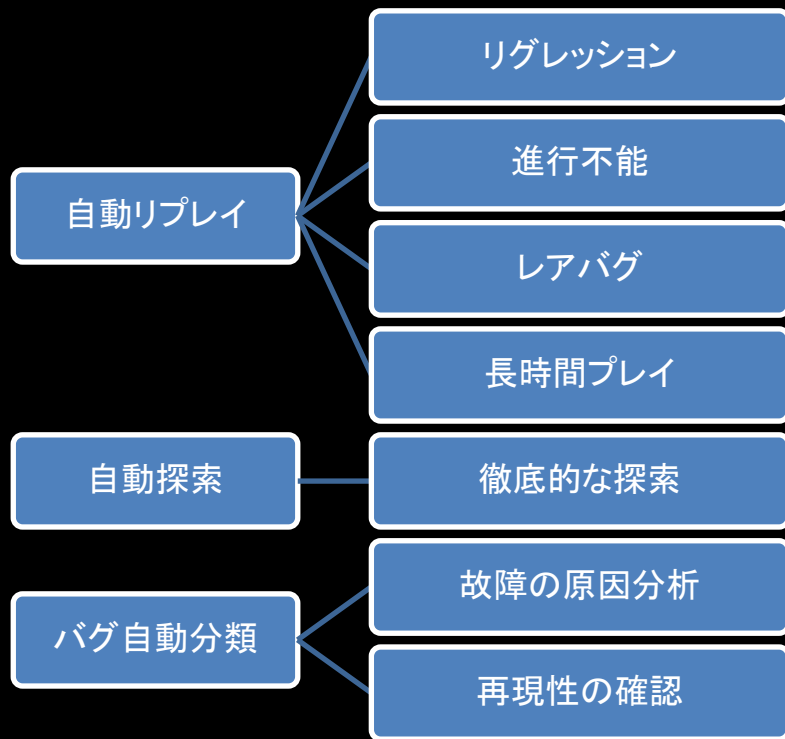


開発目的と概要

開発目的

- 機械が得意なテスト ≡ 手動では苦痛が伴うテスト
 - 同じ操作の繰り返し
 - リグレッションテスト
 - レアバグの発見
 - 長時間プレイ
 - 徹底的な探索
- →手動テストの効果の最大化
 - 本来フォーカスすべきテストに集中
 - ツールの補助により負荷を軽減

本自動テストツールの機能と対象



各機能の解説

自動リプレイ



自動リプレイのタイプ



リプレイの構成要素	ゲームコードの変更が不要	ゲームコードの変更が必要
入力	時間、スクリーンショット	ゲームの状態 (要コード内へのアクセス)
出力	ゲームパッドの操作	ゲームパッドの エミュレーション
ビルドイメージ	マスター (出荷相当)	テスト (内部検証相当)
非決定要素のサポート	なし	あり
出現要素の変更のサポート	なし	一部
使用開始できるフェーズ	マスター	α以降



自動リプレイコア機能

ゲームの制御

汎用の
状態情報

起動・終了の制御

CLI
ランチャー (UI)
フロントエンド

プロセスの起動

自動テスト
サーバー

操作

死活情報

ゲームの
起動・終了

Watchdog

ゲームを起動

ゲーム

自動テスト
SDK

ゲーム固有
自動テストコード

汎用操作の
ハンドリング

ゲーム固有ボット

プラットフォーム非依存のコードを実行
ゲームの開始と監視



自動サーバー

- ゲーム非依存の情報

- どこで {
 - 位置、レベルID*
 - 速度 / 方向
- いつ {
 - 時間 : ゲーム時間、UTC, 実時間、フレーム数
- 何を {
 - アクション/イベント : キーの入力、パッドの入力
- 今は {
 - ゲームの状態を表したスタック

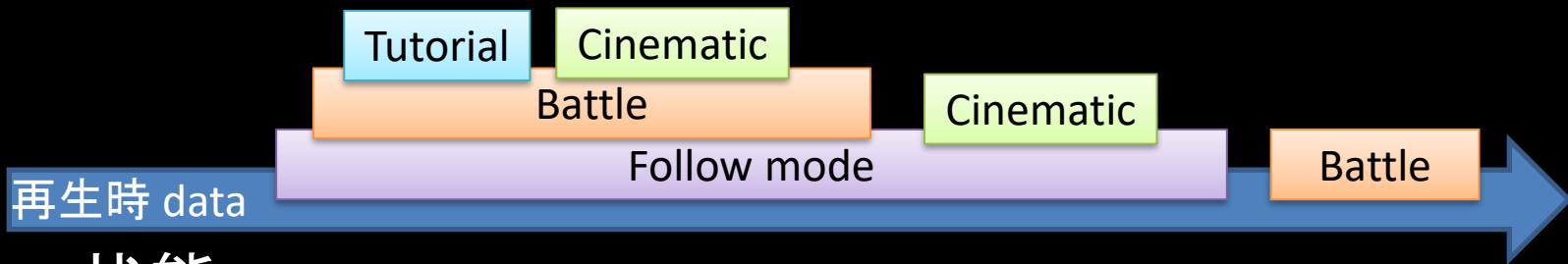


* レベル : ゲーム業界用語、今いる場所、ステージや章のようなもの



自動テストサーバー

- ゲームの状態を表したスタック

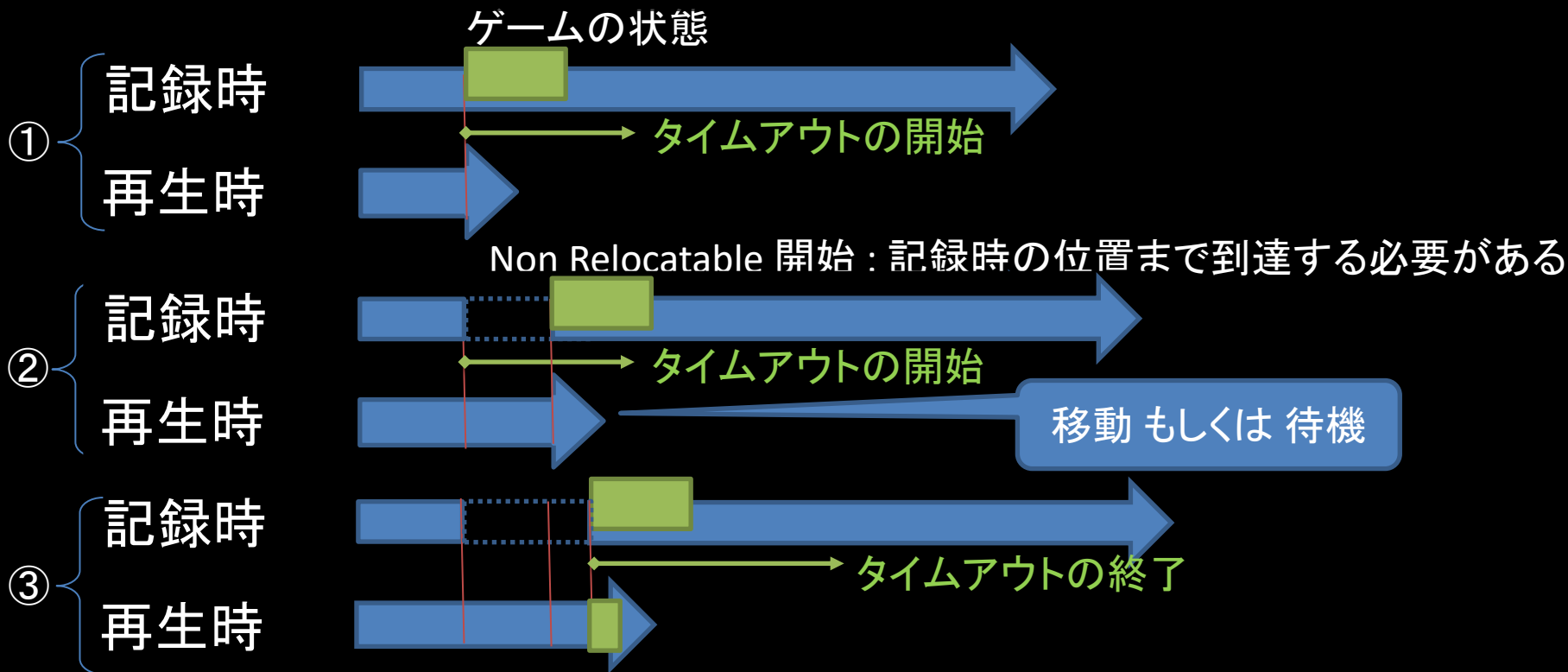


ー 状態

- Id (タイプID/ゲームID)
- ユーザーデータ
- 同期フラグ : unique, client controlled, ...
- タイムアウトのオーバーライド

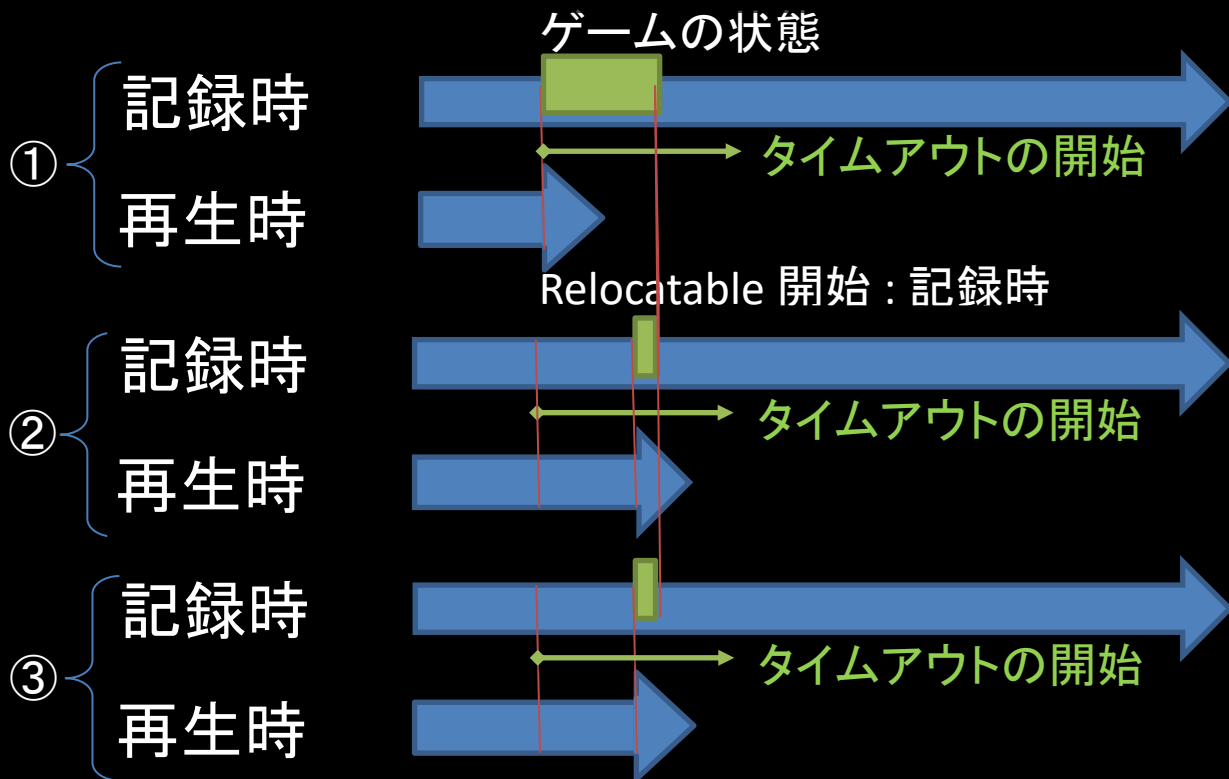


同期メカニズム Non Relocatable



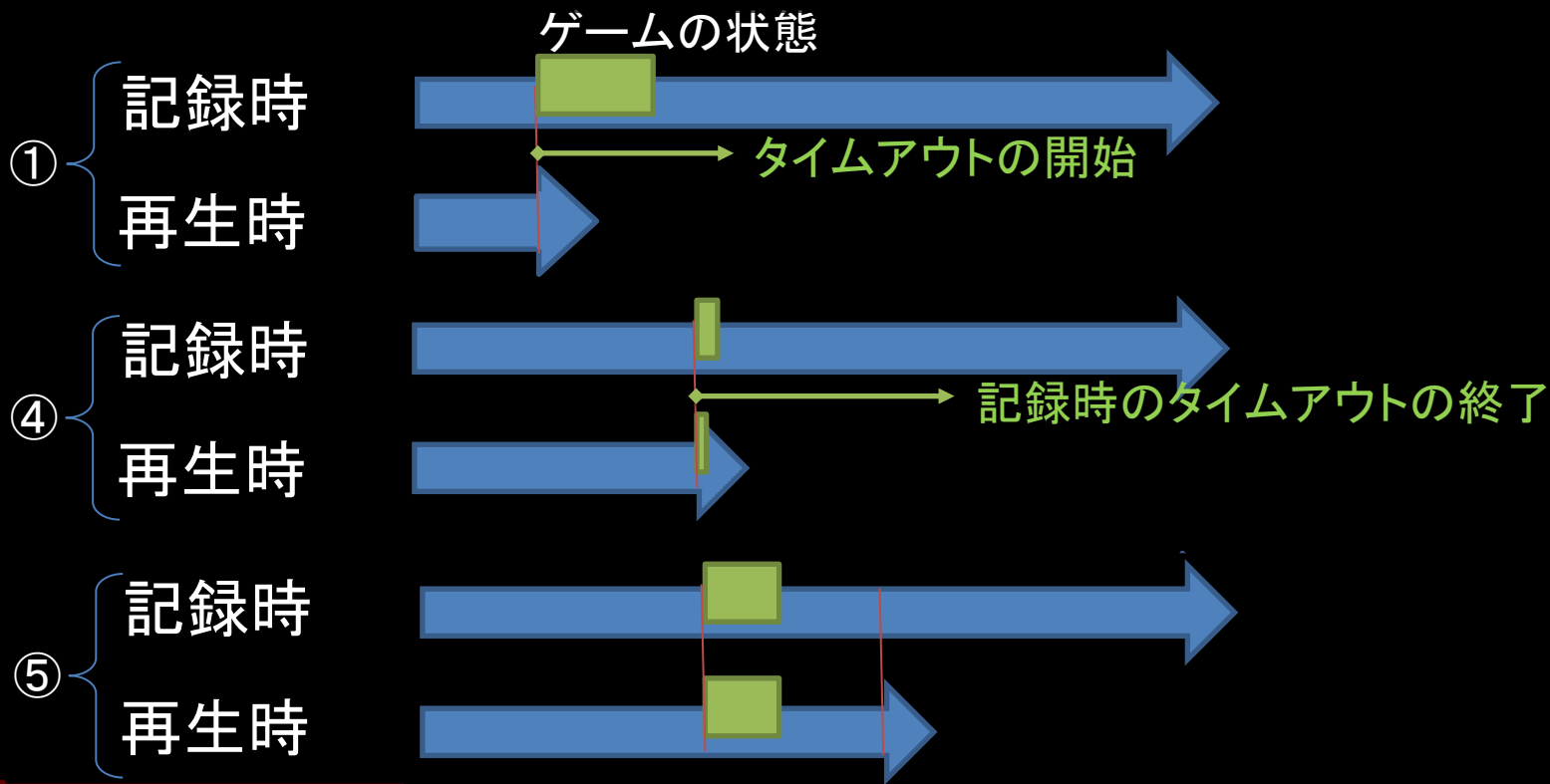


同期メカニズム Relocatable





同期メカニズム Relocatable

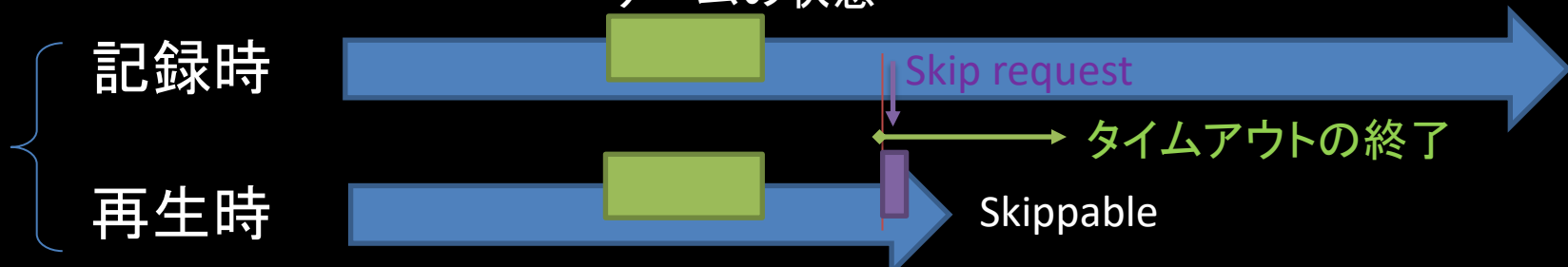




同期メカニズム

Skippable / Startable

ゲームの状態



記録時

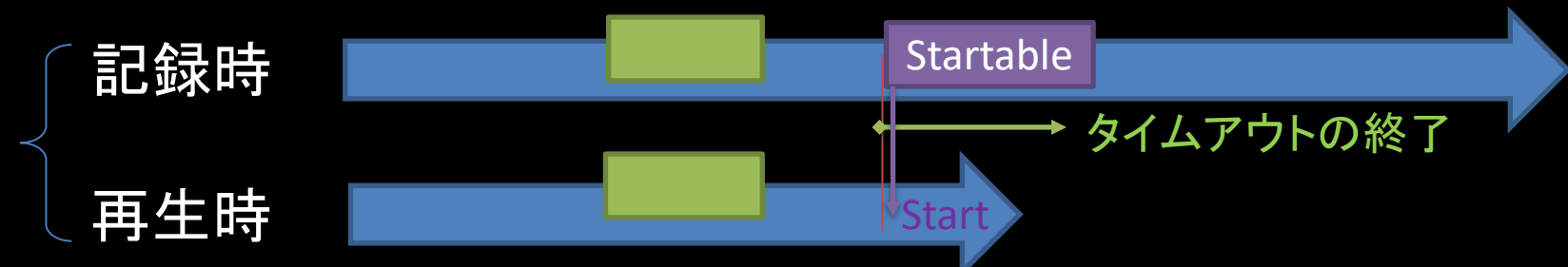
Skip request

タイムアウトの終了

再生時

Skippable

クライアント側でのアップデートを許可する、例：メニューを閉じる



記録時

Startable

タイムアウトの終了

再生時

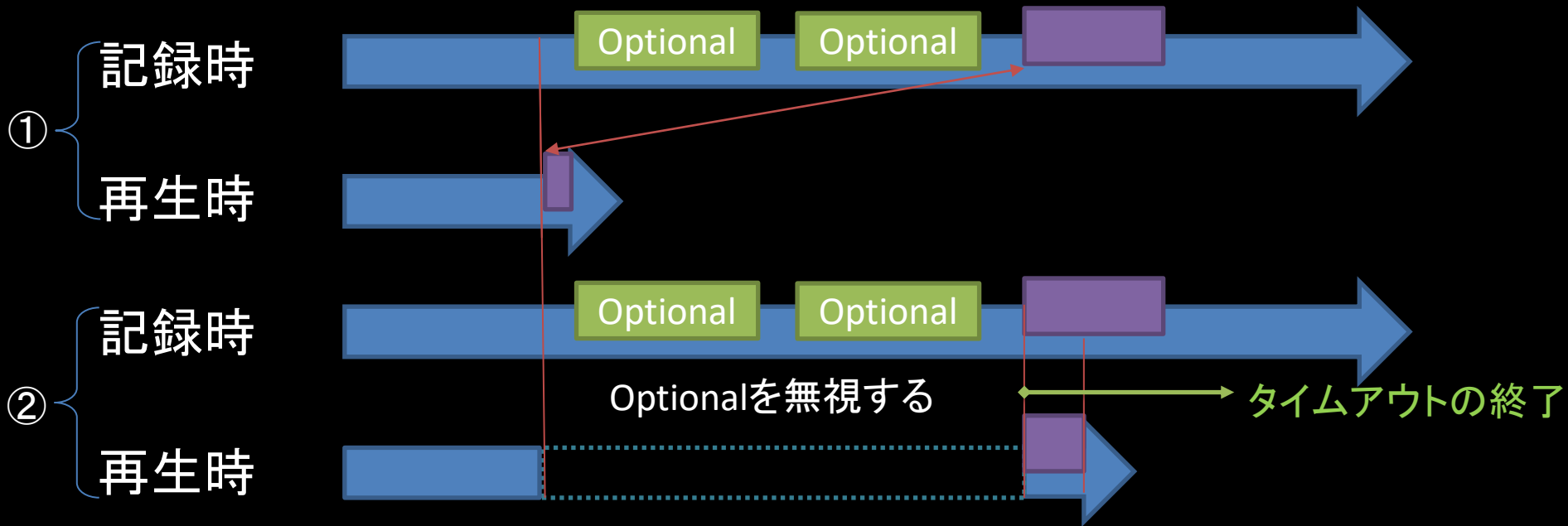
Start

クライアント側でのアップデートを許可する、例：インタラクションの開始



同期メカニズム

Optional





ゲーム固有コード (BOT)

ゲームの制御

汎用の
状態情報

起動・終了の制御

自動テスト
サーバー

ゲーム

CLI
ランチャー (UI)
フロントエンド

プロセスの起動

操作

自動テスト
SDK

ゲーム固有
自動テストコード

死活情報

ゲームの
起動・終了

汎用操作の
ハンドリング

ゲーム固有ボット

Watchdog

ゲームを起動

プラットフォーム非依存のコードを実行
ゲームの開始と監視

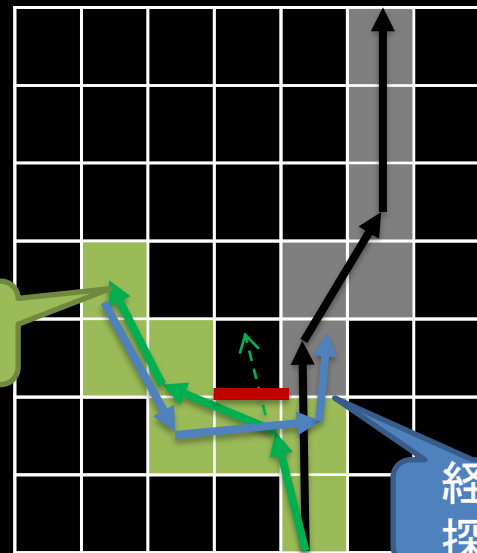


サーバー上でのパス探索

記録時の
マップ

- 3Dの格子型マップを作成
- 記録と再生の両方のマップを使用
- 実際の動作が命令と一致しなかった => 壁と判断
- パスが発見できたらリプレイに復帰
- ブロックされた場合には単純推論を使用する (ゲームのパス検索も利用可能)

再生時の
マップ



経路
探索



ゲーム間で共有するコード

ゲーム間で共有

ゲームの制御

汎用の
状態情報

起動・終了の制御

CLI
ランチャー (UI)
フロントエンド

プロセスの起動

自動テスト
サーバー

操作

死活情報

ゲームの
起動・終了

自動テスト
SDK

汎用操作の
ハンドリング

ゲーム

ゲーム固有
自動テストコード

ゲーム固有ボット

Watchdog

プラットフォーム非依存のコードを実行
ゲームの開始と監視

FF7RIにおける自動リプレイの評価

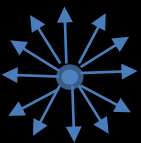
- メリット

- 再現性0.3%レアバグでも発見できた
- 複数レベル(20章)に渡るテストを同時に繰り返し実行出来た
- クラッシュをもたらすようなコードの変更を検出できた
- フレームレートが低下しても実行できた
- 24時間365日のテストができた

- 課題

- テスト対象が記録時の経路に限定される

自動探索



自動探索の評価

- メリット

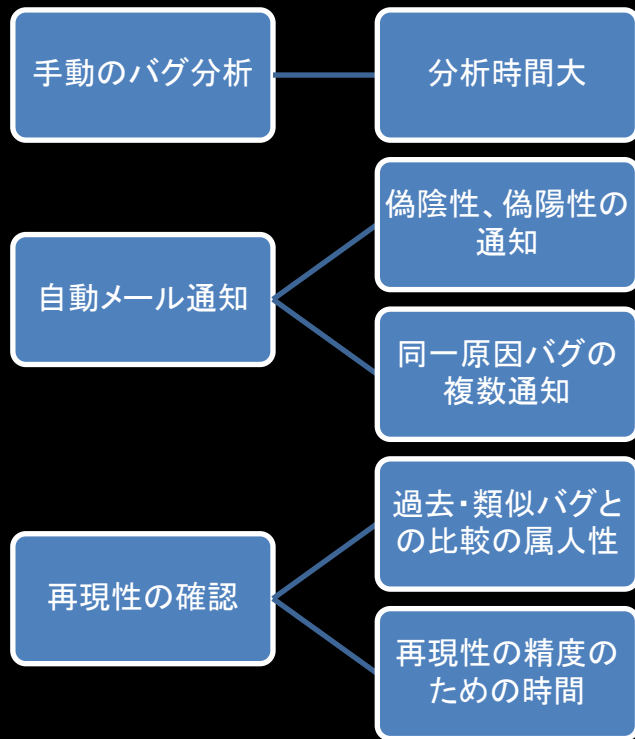
- 記録された経路以外でのバグも発見できる
- パフォーマンスの分析もできる (FPSやメモリ消費など)
- コリジョン抜けの確認に使用できる

- 課題

- 複雑なゲームではレベルの最後までたどり着けない

バグ自動分類

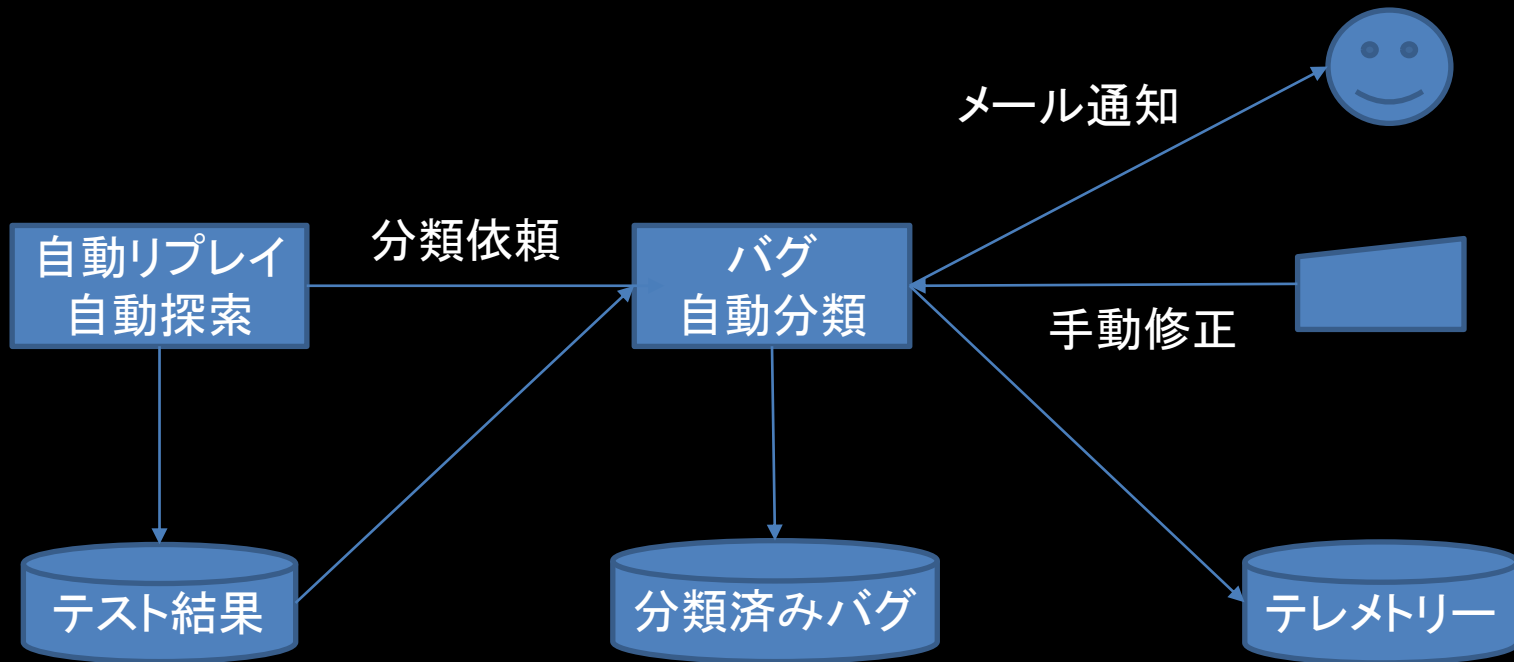
自動リプレイ・探索の課題



バグ自動分類の目標

- 手動のバグ分析の時間を削減
- 偽陽性通知の低減
- 故障の検出から通知までの時間を短縮
- ゲームチームへの精度の高いメール通知
- ゲームチームの原因分析の短縮化

アーキテクチャ

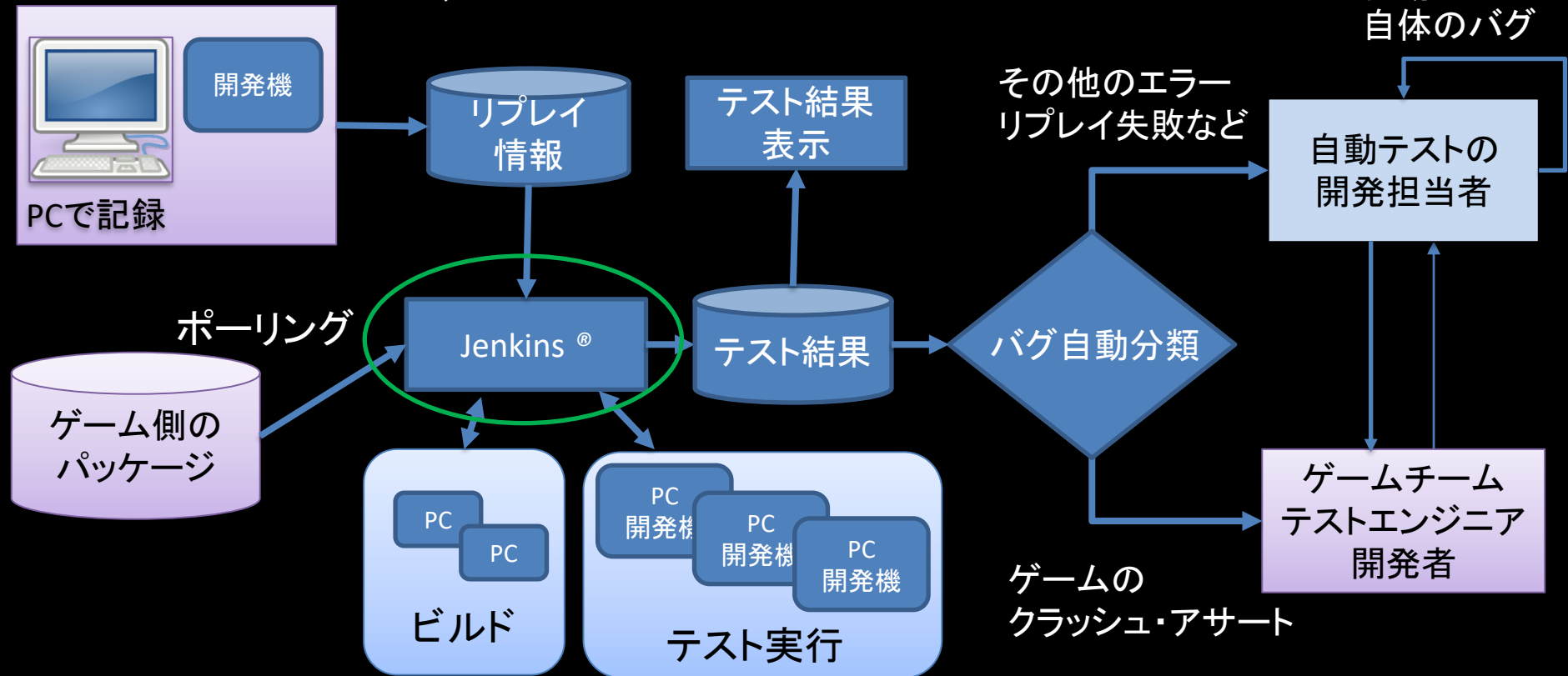


対象バグとマッチング・アルゴリズム

- バグ
 - クラッシュ
 - アサート
 - ハングアップ
 - リプレイ失敗
 - 未定義動作エラー
 - アドレス違反エラー
- アルゴリズム
 - 完全一致
 - 部分一致
 - 重複一致
 - 手動一致

バックエンド

自動テスト・ワークフロー

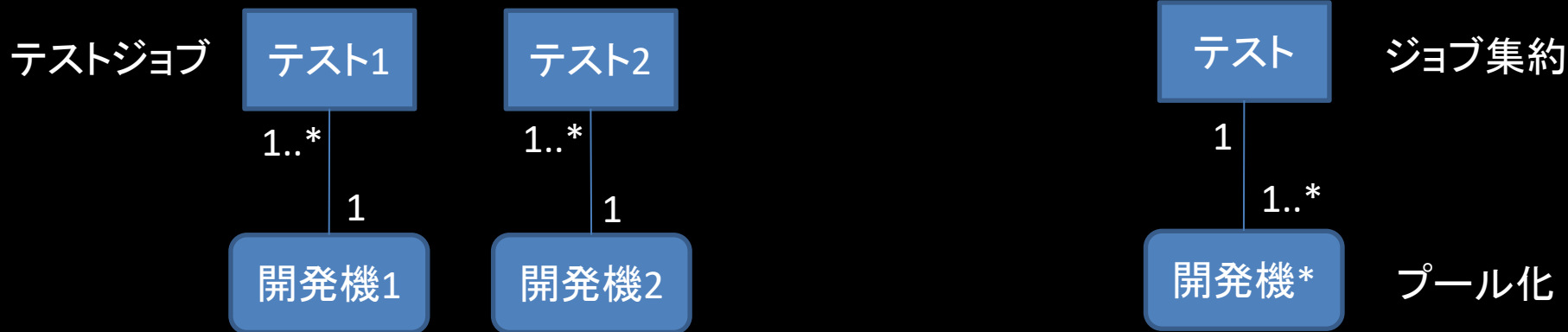


バックエンド改善 V2.0

• V1.0

• V2.0

パラメータ

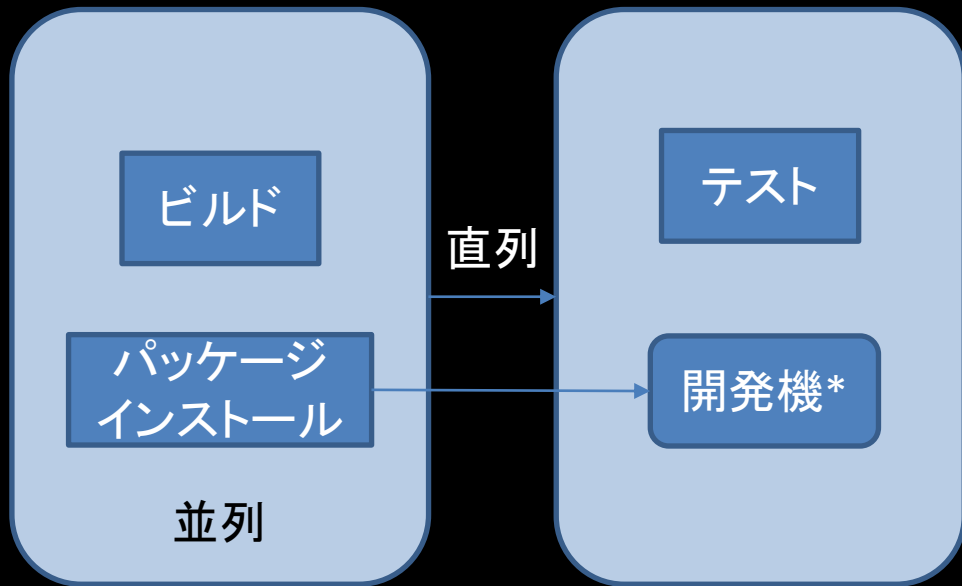


テストと開発機の対応が固定
ジョブ数大、構成変更工数大

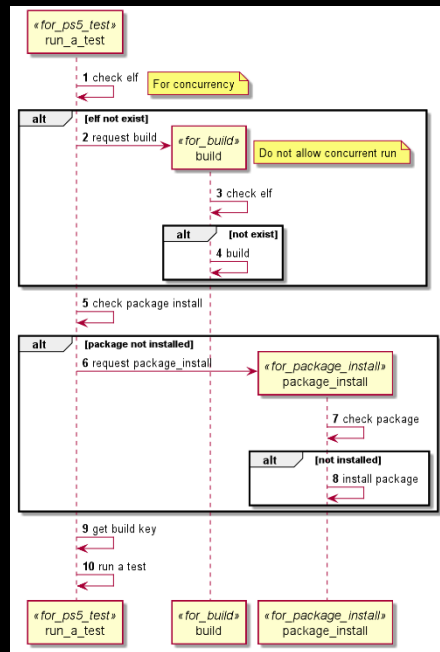
効果：管理工数低下
残課題：ビルド→テストが依然必要

バックエンド改善 V2.5

• V2.0



• V2.5



オンデマンド
ビルド
インストール
共有
ビルド

→

実行時間減
任意のテスト実行

フロントエンド

フロントエンド改善

- 目的

- ゲームチームの方の役立つツールとする

- レアバグの再現
- カットシーンの検証

- 手動のテストエンジニアの作業効率を向上させる

- 長時間テストの実行
- バグの再現性の精度向上

無限リプレイとデバッグ補助

バグの発生直前の停止設定

レアバグの再現率を見る
対象のバグ以外は無視

一時停止予約 ゲーム設定ファイル

00:00:10 探索時間スケール: 0

動画を記録

異常終了時にコアダンプを出力する

無限リプレイ

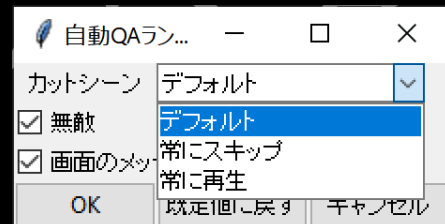
エラー発生時、ゲームを起動したままにする

時間: 00:00:00 から 00:00:10 最後 ▾

クラッシュ、ハングアップ以外の進行不能バグ発生時、デバッグ可能

カットシーンの制御と切り出し

- カットシーン*の検証用
 - リプレイ動画からカットシーンを切り出し
 - ゲームチームのご要望により作成
 - 切り取りタイミング
 - 記録時の時間



* カットシーン：ゲーム上で動いているキャラクターをそのまま使用したアニメーション
固定のムービーよりも自然なアニメーションとなるが、CPU、GPU負荷が高い

自動テストビルド取得

- 機能
 - 自動テスト込みのビルドをUIから取得
 - Jenkinsでビルド済みのパッケージを取得
- 目的
 - 手動のテストエンジニアのツール利用を容易に
- 従来
 - かなり前：ローカルで自動テストを組み込んでビルド
 - 今まで：CLIでローカルにダウンロード
 - →CLIでも敷居が高かった

今後の開発

開発予定機能

- C++自動テストサーバー
 - PythonサーバーのGILによるFPS低下解消
- バグ自動分類
 - アルゴリズム精度向上
 - フロントエンド
 - バグシステム統合
- テレメトリー
 - FPS, メモリ消費可視化
- バックエンド
 - 実行テスト決定エンジン

ご清聴ありがとうございました

商標

- “PlayStation”, “PS4” and “PS5” are registered trademarks or trademarks of Sony Interactive Entertainment Inc.
- *Jenkins® is a registered trademark of Software in the Public Interest, Inc.*