

# ネイティブゲーム開発 を支える Jenkins

KLab株式会社 李 承益

使い方  
を間違えた  
Jenkins


# 自己紹介

- 李 承益 (りちえんい)
- 台湾人
- 2008日本上陸
- 大学院でOSの研究
- 2012新卒

**Victor Lee**

konyavic

---

 Tokyo, Japan

 Joined on Jan 31, 2011

---

# Jenkinsの間違った使い方

- どんなプロジェクト？
- どうやって開発してる？
- Jenkinsをどんな風に酷使してる？
- それでみんなが幸せになった？

どんなプロジェクト？

# スマートフォンゲーム

- リッチな画面と演出！
  - アセットファイル
  - 画像
  - 音声
- マルチプラットフォーム！
  - iOS、Android
- クライアント・サーバ

どうやって開発してる？

Unity?

Cocos2d?

\_\_人人人人人人人人\_\_  
> 突然の自社エンジン <  
\_\_Y^Y^Y^Y^Y^Y^Y^Y^Y^Y^\_\_



# 自社エンジン

- Lua
  - ゲームロジック
- マルチプラットフォーム対応
  - iOS、Android
  - Windows
- アセット周り→開発ツール

# 開発ツール

- アセットの設計
  - UI、演出
  - テキスチャー配置
- 各PFの形式にPublish
  - 画像、音声データの圧縮

# コードを書くことから、リリースまで

- コードを書く
- データを突っ込む
- 画像を突っ込む
- Windowsで動作確認する
- Publishする
- おもむろにMacにコピーする
- Xcodeを開いてコンパイルする

普通

# マスターデータのコピー地獄

- マスターデータとは
  - ゲーム中のあらゆるパラメータ
  - 100を超えるExcelシート！
  - 複数人・同時・分散作業！
- シート間の整合性を守ることが絶望的！

# マスターデータをGoogle Docsに置く

- Google Docs
  - 複数人で！同時に！同じファイルを編集できる！
  - そこにあるものを開けば常に最新！常に正である！
- Google Docsからの  
マスターデータDLツールを作った！

# 問題

- クライアントとサーバ
  - レポジトリが分かれている
  - 両方への更新が必要
  - 両方が同じものを見ていることを保証したい
- git submodule導入

# やりたいことはとても簡単

- マスターデータをダウンロード
- マスターデータのレポジトリを更新する
- クライアントとサーバのレポジトリを  
更新する
  
- ただ面倒くさいだけ



# git submoduleでのワークフロー

```
git fetch
```

```
git reset --hard
```

```
git clean -df
```

```
git checkout $BRANCH
```

```
git rebase origin/$BRANCH
```

```
git submodule foreach reset --hard
```

```
git submodule foreach clean -df
```

```
git submodule update
```

```
...
```

# git submoduleでのワークフロー

- gitより複雑
- クライアントとサーバ、  
片方漏れると問題がでる
- なんとなく怖い

# Jenkinsのジョブを作った



54 分前 (#403)

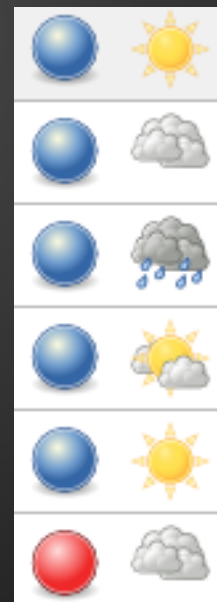
2 時間 0 分前 (#400) 25 分



```
git fetch
git reset --hard
git clean -df
git checkout $BRANCH
git rebase origin/$BRANCH
git submodule foreach reset --hard
git submodule foreach clean -df
git submodule update
...
```

# 結果

- ワンクリック
- 誰でも実行できる
  - 開発
  - 企画
  - J( 'ー` )し
- 実行結果がわかりやすい



その日から、  
我々のJenkins依存が始まった...

# 自動ビルドもやってみよう！

- Windowsでpublish
- Macでビルド
- サーバー間のデータ転送どうしよう？
  - backlogも取っておきたい
  - publishデータのバージョンを選択可能にしたい
- とりあえず全部gitに突っ込む

|  |                        |                     |
|--|------------------------|---------------------|
| ● [buildserver] auto publish 2013/03/14_13:28:11 | lee-c <lee-c@klab.com> | 2013-03-14 13:31:08 |
| ● [buildserver] auto publish 2013/03/14_12:32:01 | lee-c <lee-c@klab.com> | 2013-03-14 12:35:03 |
| ● [buildserver] auto publish 2013/03/14_12:13:10 | lee-c <lee-c@klab.com> | 2013-03-14 12:16:12 |
| ● [buildserver] auto publish 2013/03/14_11:08:32 | lee-c <lee-c@klab.com> | 2013-03-14 11:11:40 |
| ● [buildserver] auto publish 2013/03/13_23:45:28 | lee-c <lee-c@klab.com> | 2013-03-13 23:48:35 |
| ● [buildserver] auto publish 2013/03/13_23:09:49 | lee-c <lee-c@klab.com> | 2013-03-13 23:13:03 |
| ● [buildserver] auto publish 2013/03/13_22:58:14 | lee-c <lee-c@klab.com> | 2013-03-13 23:01:20 |
| ● [buildserver] auto publish 2013/03/13_21:43:07 | lee-c <lee-c@klab.com> | 2013-03-13 21:46:04 |
| ● [buildserver] auto publish 2013/03/13_21:16:03 | lee-c <lee-c@klab.com> | 2013-03-13 21:18:52 |
| ● [buildserver] auto publish 2013/03/13_20:56:37 | lee-c <lee-c@klab.com> | 2013-03-13 20:59:43 |
| ● [buildserver] auto publish 2013/03/13_20:38:57 | lee-c <lee-c@klab.com> | 2013-03-13 20:41:57 |
| ● [buildserver] auto publish 2013/03/13_20:28:40 | lee-c <lee-c@klab.com> | 2013-03-13 20:31:46 |

# さらに必要なものと要らないものを 次々と導入...

- マスターデータの静的チェック
- ダウンロードパッケージの処理
  - パッケージの生成
  - パッケージのアップロード
  - Windows -> Mac -> Linux
- コンパイル時のパラメータ調整
- 音を出す



ジョブが長すぎる！



分割して  
モジュール化したい

# Build Pipeline

The screenshot displays the Jenkins web interface for a build pipeline named "My pipeline". The interface includes a navigation bar with "Jenkins" and "Build pipeline" tabs, a search bar, and a "log out" link. Below the navigation bar, the pipeline is visualized as a sequence of steps in two runs.

**Run 8 (Current):**

- Pipeline version 8** (No parameters)
- Test** (Jun 26, 2012 5:30:48 PM, 8 min 10 sec)
- Release** (Jun 26, 2012 5:31:03 PM, 8 min 12 sec)
- Generate docs** (Jun 26, 2012 5:31:20 PM, 9.1 sec)
- Deploy to Test** (Jun 26, 2012 5:31:46 PM, 6 min ...)
- Deploy to Pre-Prod** (Pending)
- Deploy to Prod** (Pending)

**Run 7 (Previous):**

- Pipeline version 7** (No parameters)
- Test** (Jun 26, 2012 5:26:25 PM, 7 min 10 sec)
- Release** (Jun 26, 2012 5:26:40 PM, 7 min 12 sec)
- Generate docs** (Jun 26, 2012 5:26:57 PM, 7 min 9.1 sec)
- Deploy to Test** (Jun 26, 2012 5:27:13 PM, 5 min 13 sec)
- Deploy to Pre-Prod** (Jun 26, 2012 5:27:31 PM, 4 min 10 sec)
- Deploy to Prod** (Jun 26, 2012 5:27:46 PM, 4 min 15 sec)

At the bottom of the page, there is a link to "Help us localize this page" and a footer indicating "Page generated: 26-Jun-2012 17:31:39 Jenkins ver. 1.470".

# Build Pipeline

- Parameterized Build
- Parameterized Trigger
- 本当の意味での「パイプライン」じゃない
  - レポジトリcheckoutでジョブ間の依存性が発生する
  - ジョブの再利用が難しい
  - ジョブコピーしないと並列化できない

# Build Flow

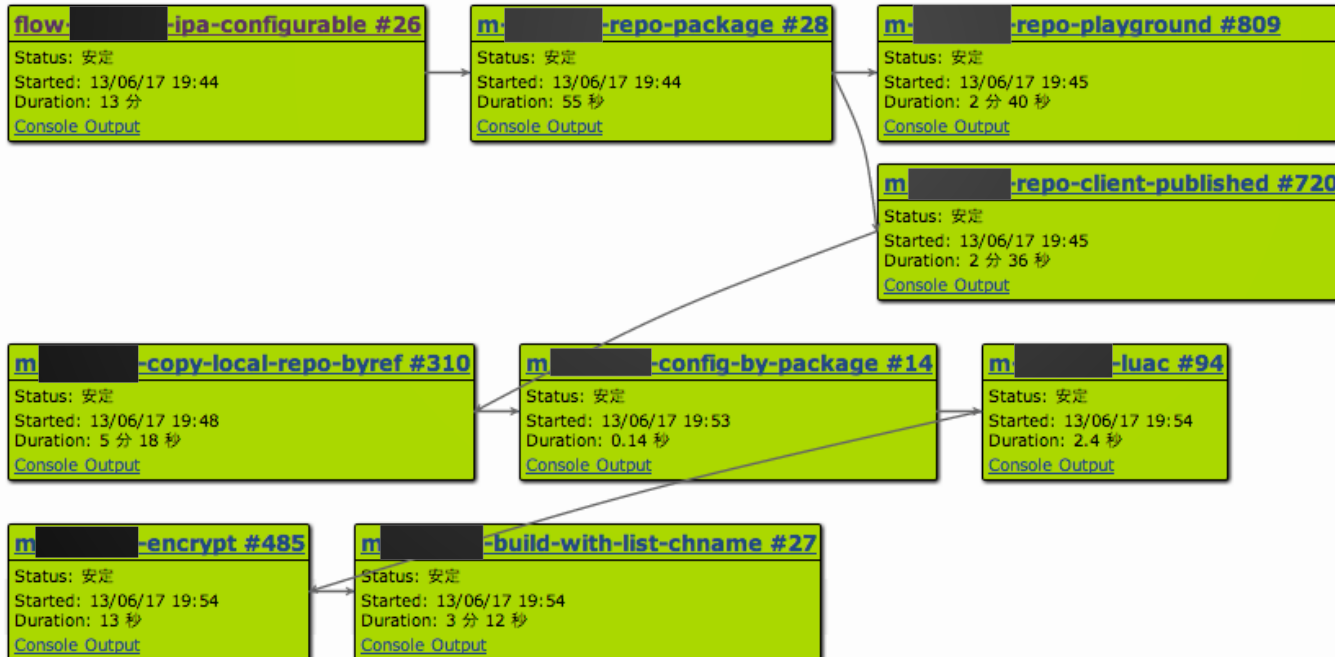
# Build Flow

- DSLでジョブの組み合わせができる

```
guard {  
  build("update",  
    ENGINE_BRANCH: params.ENGINE_BRANCH,  
    CLIENT_BRANCH: params.CLIENT_BRANCH  
  )  
  build("build")  
} rescue {  
  build("reset")  
}
```

# Build Flow

| Parameters     |                                    |
|----------------|------------------------------------|
| SERVER_ENV     | review                             |
| SERVER_VERSION | dev, dev2, dev3, stg, review, prod |
| ENGINE_VERSION | 3.0                                |
| ENGINE_CONFIG  | fb27679e18b091d131880fd87369d7f5d  |
| ENGINE_CONFIG  | Release                            |
| BUILD_TARGET   |                                    |
| FULL           | false                              |



# Build Flow

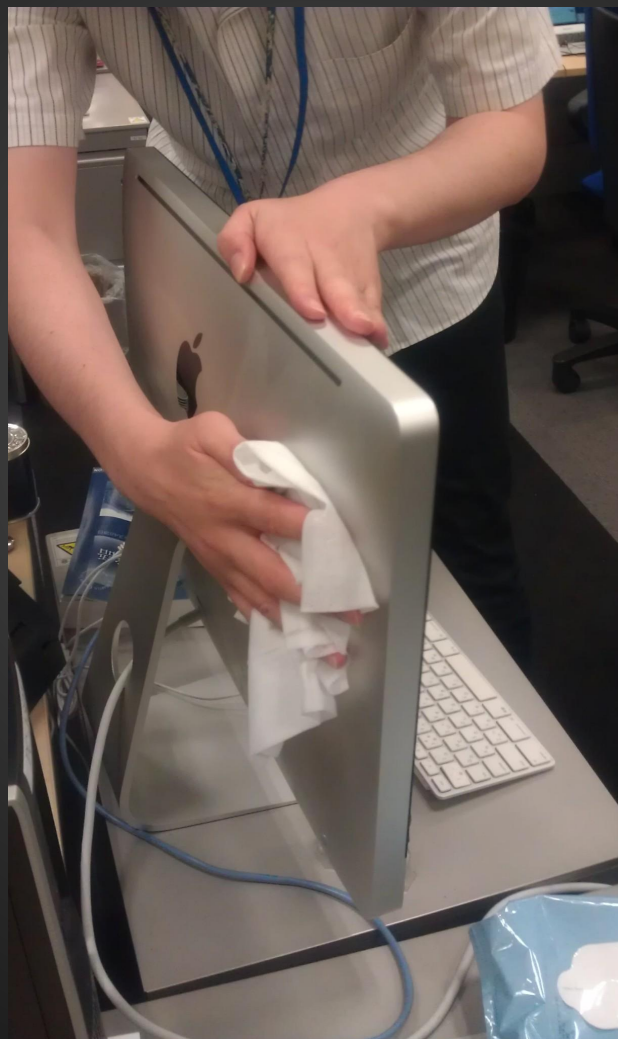
- 分かりやすい結果表示
  - どこで失敗したか一見で分る
  - カッコイイ!(・▽・)
- フロー毎にワークスペースを持っている
  - フロー毎のデータ格納先
  - git shallow cloneでコピー量減らす
  - 複数フローの並列実行





Jenkins「はたらくぞ！」

# Jenkins氏、熱で倒れる・・・



で、  
みんなが幸せになった？

# 良くなったところ

- 実行ステータス可視化
- ステップの長い、  
複雑なタスクをカプセル化
- 実行の敷居が低くなった

# 不幸になったところ

- 中身が分かりにくくなる
- 複雑なことが簡単になったわけではない
  - 言語 (DSL, bash, python, ruby, PowerShell)
  - OS (Windows, Mac, Linux)
- Jenkinsの介護係というポジションの誕生
  - 不幸になった人
- 気づいたら属人化が進んだ

# まとめ

- 複雑なことが、  
簡単に実行できるようになった
- Jenkinsのジョブに  
複雑度を集約することになった