

vArashi - Artificial Video Game Player

八重樫 剛史

<http://varashi.jp/>

Agenda

- ▷ **vArashi** プロジェクトの紹介
- ▷ **vArashi** システムのハードウェア
- ▷ **vArashi** システムのソフトウェア
- ▷ **事例紹介 (syspuyo2)**

vArashi プロジェクトとは?

▷ 現代の高度なテクノロジーを駆使して、ビデオゲームを人間と同様のインタフェースでプレイしてくれる機械

(vArashi システム)を作る

- 構想 - 2001 年初夏
- スタート - 2001 年秋(LC2001で発表した)
- varashi.jp 取得 - 2001 年 11 月
- プロジェクト構成員 - 1 名

背景

- ▷ ありあまる計算機資源
- ▷ 手軽に入手できる高機能デバイス
 - ワンチップマイコン
 - **CPLD/FPGA**
 - ビデオキャプチャデバイス

技術的なトピック

- ▷ **ビデオゲーム用画像認識の研究**
 - ビデオキャプチャプログラミング
 - マルチメディア命令セットの利用研究
- ▷ **ビデオゲーム機コントローラインタフェース**
 - ハードウェア設計・製作
 - 組み込みソフトウェア開発
 - デバイスドライバ開発
- ▷ **ビデオゲーム思考ルーチンの研究**
- ▷ **リアルタイムプログラミング**

意義

- ▷ コンピュータビジョン研究プラットフォーム
 - リアルワールドに比べてずっと単純
- ▷ 公正なコンピュータプレイヤーの実現
 - イカサマが起きる余地がない
- ▷ ビデオゲームにまつわる必殺技の検証
 - 炎のコマ 使用時は何が起きているのか?

用語の説明

▷ **vArashi (ぶいあらし)**

- 「 virtual ゲームセンターあらし 」
- むかしそういうママガがあった

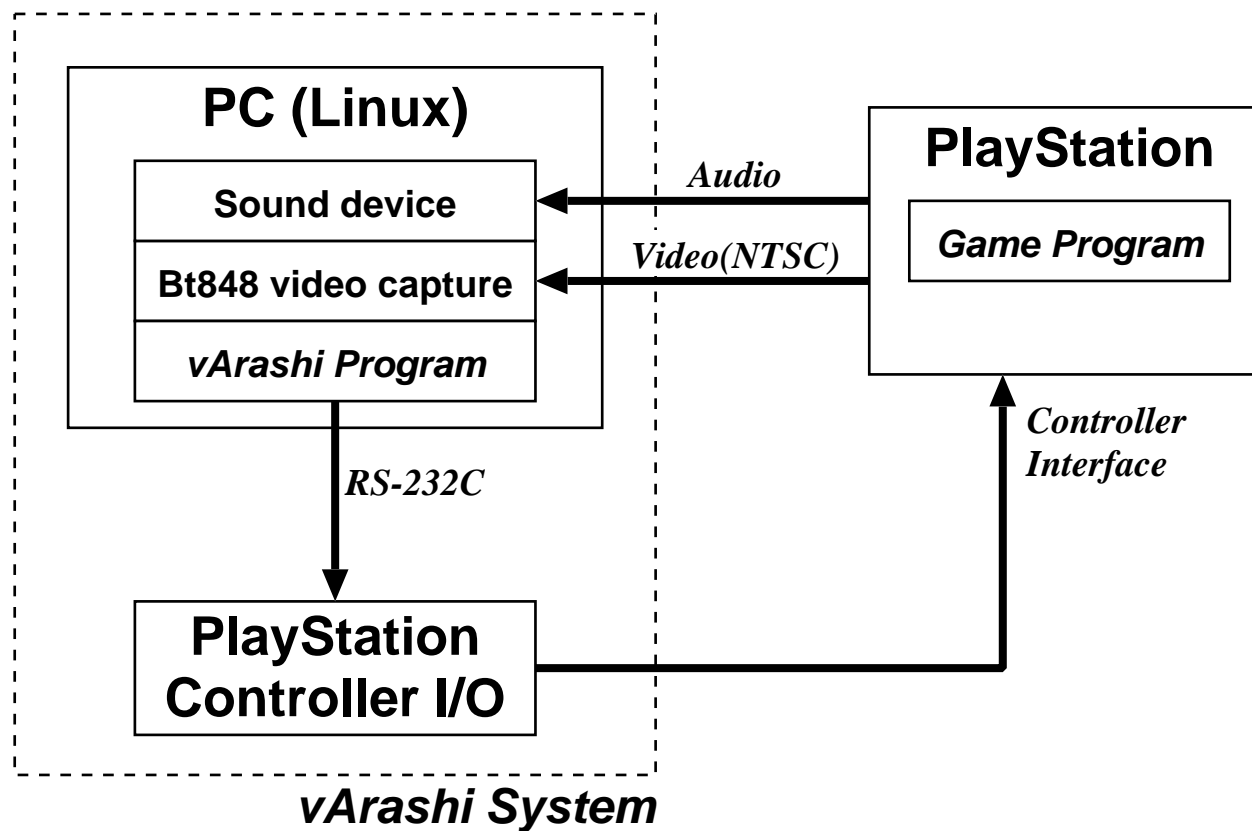
▷ **vArashi システム**

- ビデオゲームを遊ぶ機械 ・ システム

▷ **vArashi プログラム**

- ビデオゲームを遊ぶために作られたソフトウェア

典型的 vArashi システムの構成



典型的 vArashi システムの構成

▷ PC (PC-AT 互換機)

- CPU: Pentium III-S 1.26GHz, メモリ: 256MB

▷ Bt848 系ビデオキャプチャカード

- PC用ビデオキャプチャデバイスの代表

▷ PlayStation

- 最も一般的なビデオゲームコンソール

▷ PlayStation コントローラ入力装置

- vArashi システムの肝となるデバイス

Brooktree Bt848 family devices

▷ PCI Video Decoder

- 現在は **Conexant Systems Fusion 878A** という製品

▷ 安価 ・ 高性能

- 5,000 円** ~ の PCI ビデオキャプチャカード

- ほぼあらゆるフォーマットでキャプチャが実行できる

- 妙なこと (圧縮とか) をしないので、画像処理アプリケーションには最適

- かさばる PCI カードでしか入手できないのが欠点

bttv ドライバの改良と API の拡張

▷ Linux のビデオキャプチャ API

- Video4Linux ... Alan Cox 作
- Video4Linux2 ... 動画ハードウェアなどもサポート

▷ bttv ドライバ

- Brooktree Bt848 ファミリ用 ドライバ
- V4L, V4L2 両方で利用可能

bttv ドライバの改良

▷ Version 0.8.45

- 1 フレーム(2 フィールド)単位のキャプチャのみ可能
 - VSYNC 60Hz のゲームでも 30Hz、2 枚単位でしかキャプチャできない
 - 表示が見苦しくなる (櫛ノイズ)
 - タイミング重視のゲームでは致命的
- 1 フィールドごとに割り込みを発生するように改良

Video4Linux2 の問題

▷ フレームバッファ確保 API

□ 一般的な手順

1. VIDIOC_REQBUFS ioctl でフレームバッファを確保

2. mmap(2) でユーザメモリ空間にマップ

□ すでに確保されたユーザメモリ空間 (共有メモリなど) をフレームバッファとする方法がない ... バッファ間コピーの発生、パフォーマンスの低下

□ 新 API の導入 ... VIDIOC_SETBUF ioctl

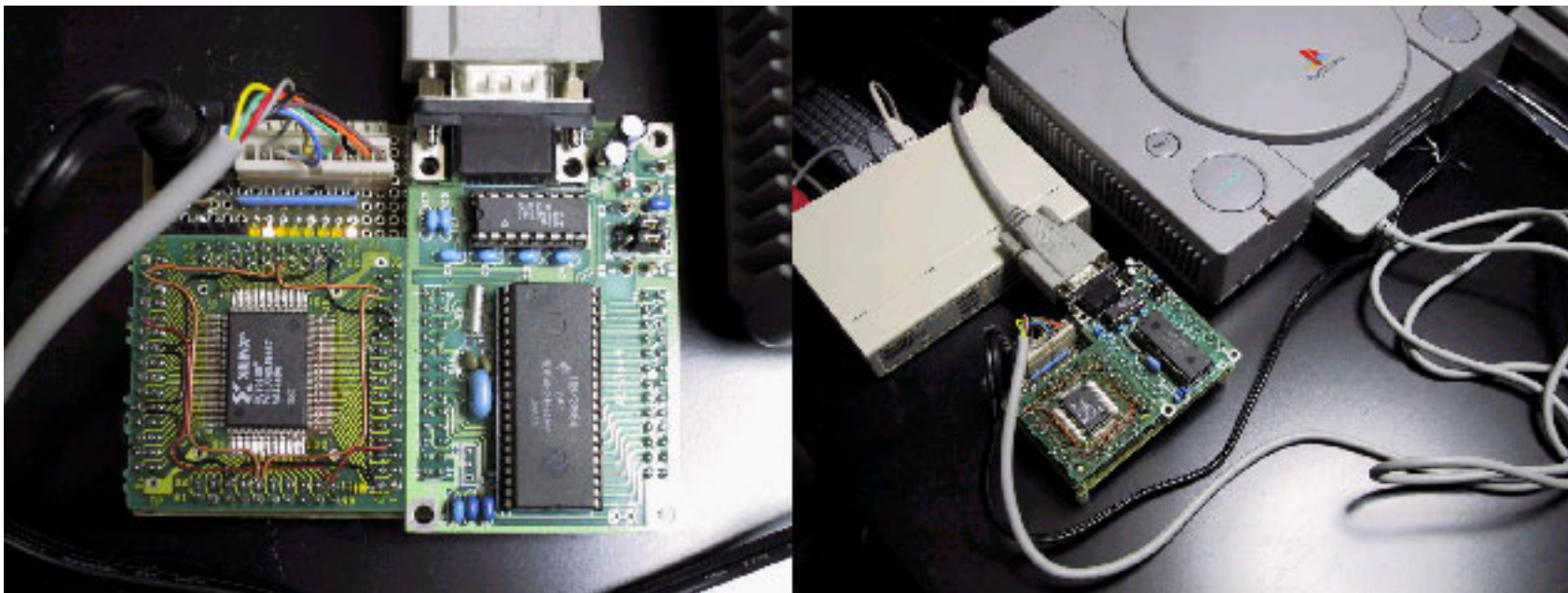
改良版 ドライバ

▷ **sourceforge.jp** の **CVS** から参照可能

- <http://sourceforge.jp/projects/varsahi/>
- まだバグだらけ (引数チェックなどが甘い)
- そのうち upstream にパッチを送りたい

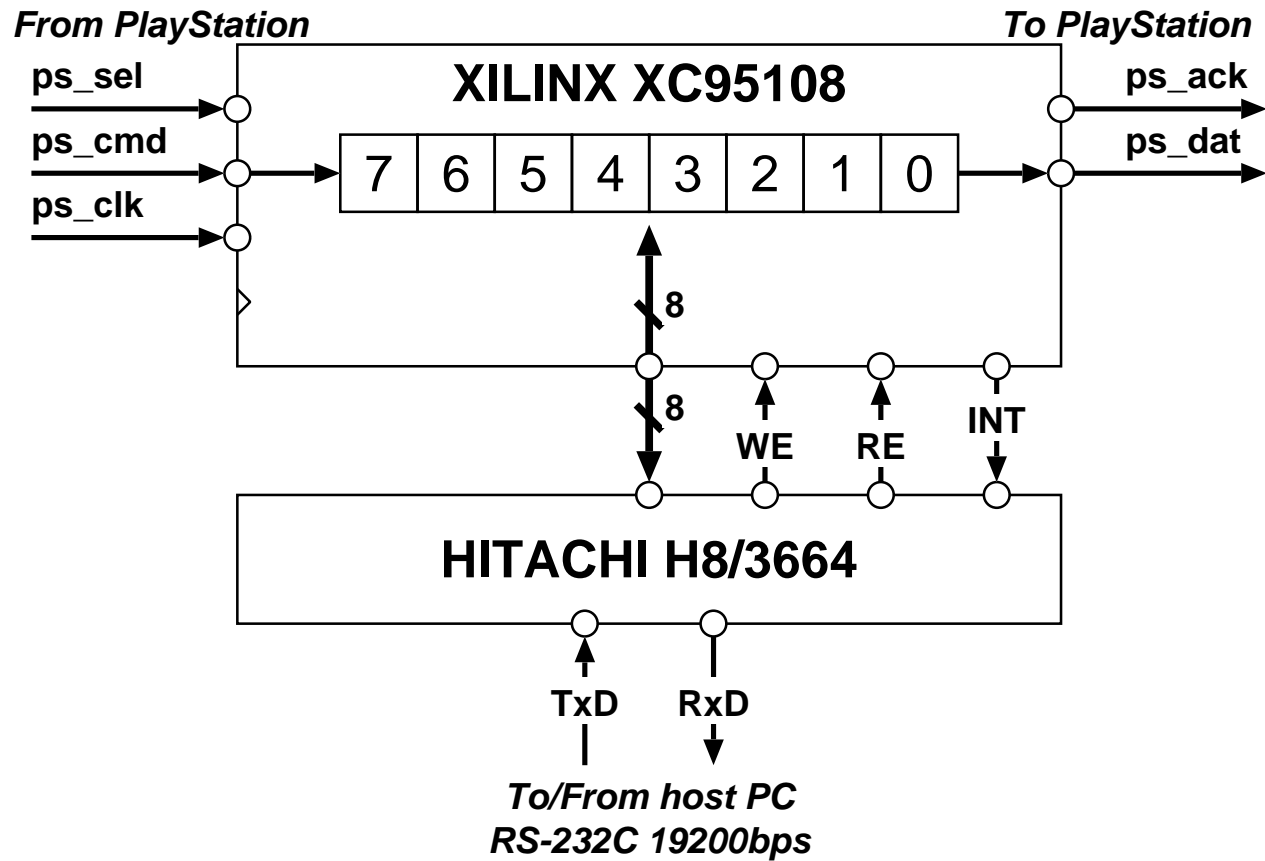
PlayStation コントローラ入力装置

- ▷ PlayStation コントローラを操るためのデバイス
- ▷ ホストとは RS-232C で接続



システム構成

▷ マイコン(H8/3664) + CPLD(XC95108)



開発方法 & 使用方法

▷ 開発方法

- プロトコル解析資料はインターネットに出回っている
- マイコン: **GNU Toolchain** でクロス開発
- **CPLD: VHDL**で記述 ... **Windows** 環境が必要

▷ 使用方法

- ホストから **PlayStation** に読み取らせたいバイト列を送るだけ

PlayStation コントローラ入力装置

- ▷ **製作費 10,000 ~ 20,000 円程度?**
 - 秋月電子通商 H8/3664 マイコンキット
 - 余りものの XILINX XC95108
- ▷ **そのうち回路図やソースコード公開したい**
 - ハードウェアの資料はまとめるのに時間がかかる
 - すでに入手困難なデバイスを使っているのが問題
- ▷ **手に入りやすい部品を使って設計しなおしたい**
 - キット頒布とかできたらなおよい

vArashi フレームワーク

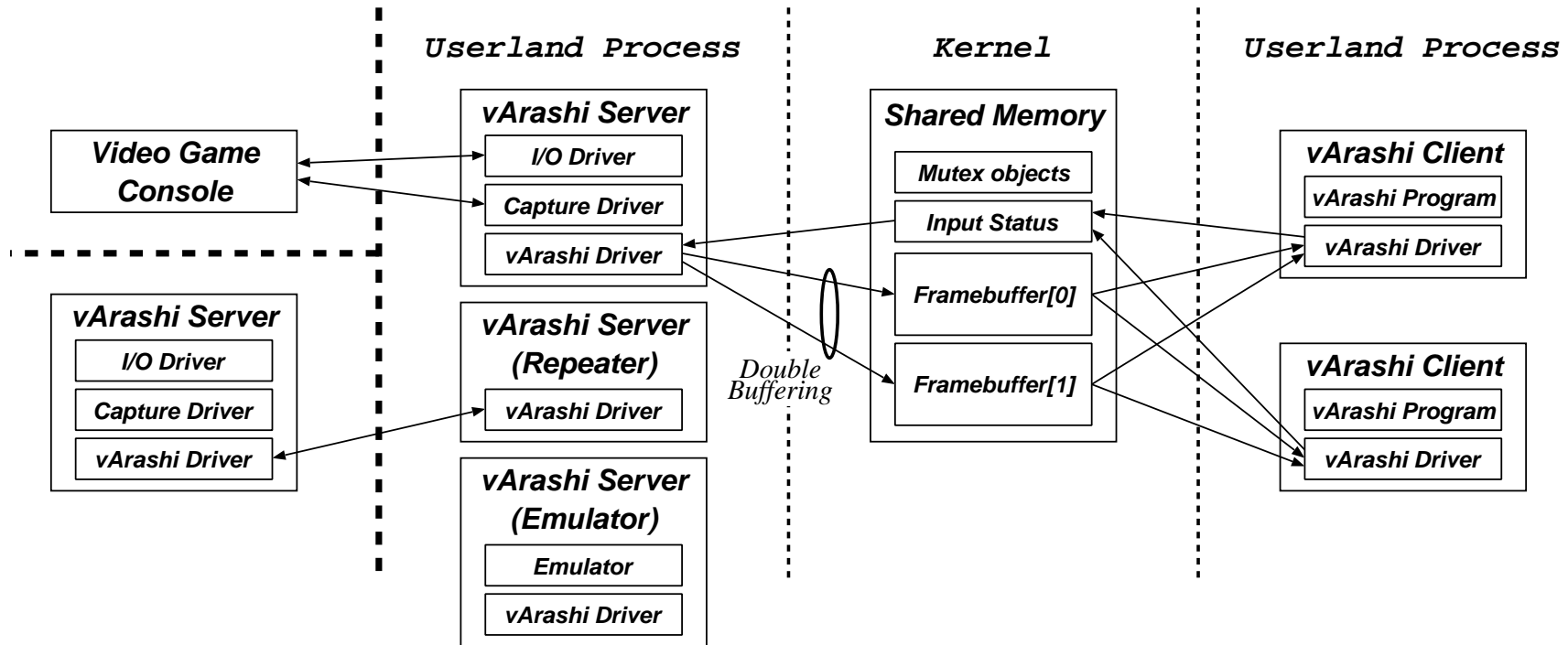
▷ これまでの vArashi システム

- ビデオキャプチャ、画像認識、ゲームアルゴリズムなどがひとつの vArashi プログラムに渾然一体

▷ 新しい vArashi システム

- 役割にあわせて複数のプロセスに分割
- 共有メモリやセマフォ、TCP/IP などを用いて通信
- 柔軟に vArashi システムを構成することが可能

vArashi フレームワーク



xmame.vFubuki

▷ xmame.vFubuki とは

- アーケードビデオゲームエミュレータ XMAME に、vArashi サーバ機能を組み込んだもの
- 特殊なハードウェアが必要なく、手軽に始められる
- ノイズが入りこむ余地もなくなるため、画像認識がやさしくなる

syspuyo2

▷ シスぷよ(syspuyo)

- コンパイルのビデオゲーム「ぷよぷよ」をターゲットにした vArashi システム
- なぜこのゲームを選んだのか？
 - 誰でも知ってる ・ ルールが簡単
 - 画像認識がやさしい
 - 連鎖など奥が深くおもしろい
 - リアルタイムな駆け引き

▷ シスぷよ通 (syspuyo2)

- 「ぷよぷよ通」に対する syspuyo 実装

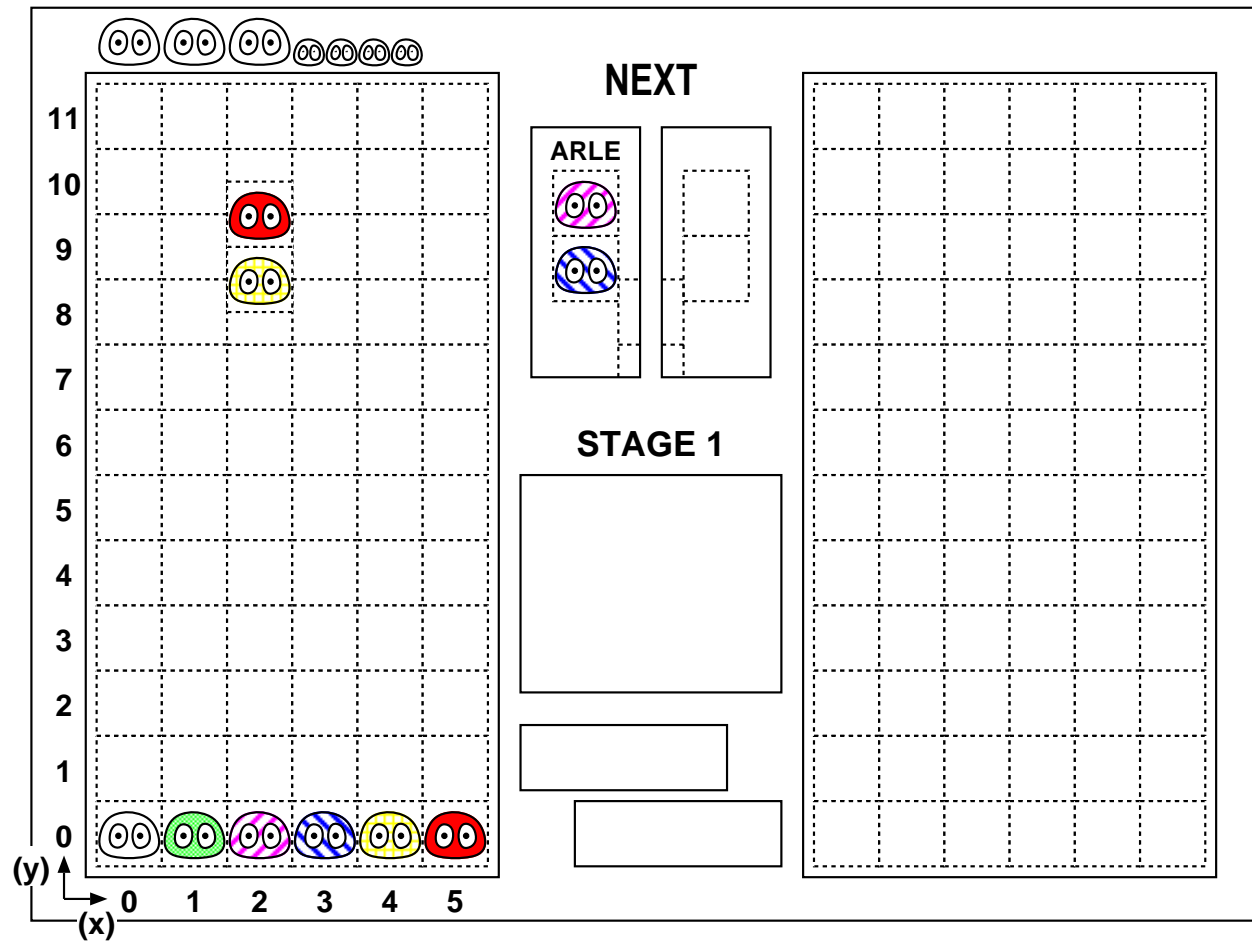
syspuyo2: ぷよぷよ

▷ ルール

- プレイヤーごとに 12 段 × 6 列のフィールドがある
- フィールドにランダム色の「色ぷよ」が 2 匹一組で降りてくる
- 同じ色が 4 匹以上隣接した「色ぷよ」は消える
- ぷよを消すと相手のフィールドに「おじゃまぷよ」が降る
- 「おじゃまぷよ」は隣接した「色ぷよ」が消えるとき一緒に消える
- 「おじゃまぷよ」を沢山降らせて相手フィールドをぷよで埋めると勝ち

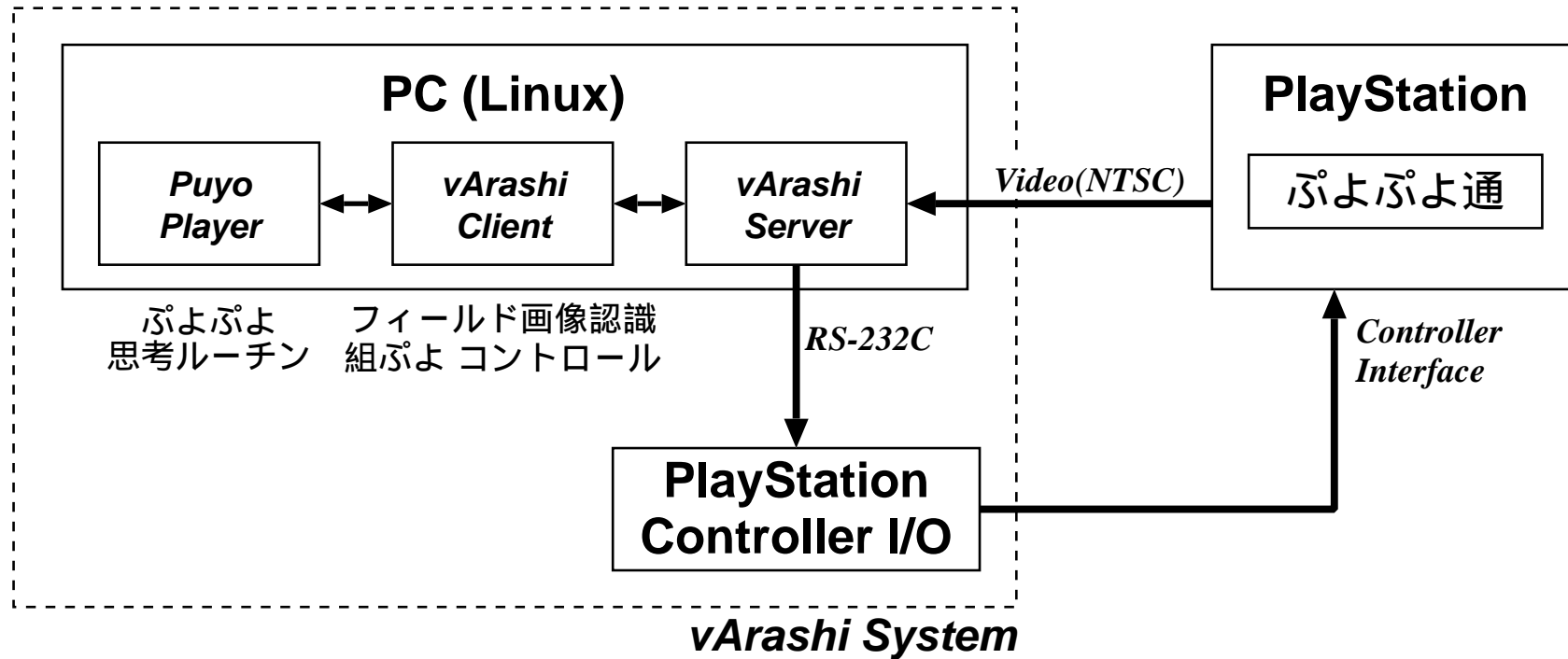
syspuyo2: ぷいよぷいよ

▶ ぷいよぷいよ通の画面レイアウト



syspuyo2: システム構成

▷ vArashi システムの構成



syspuyo2: 画像認識

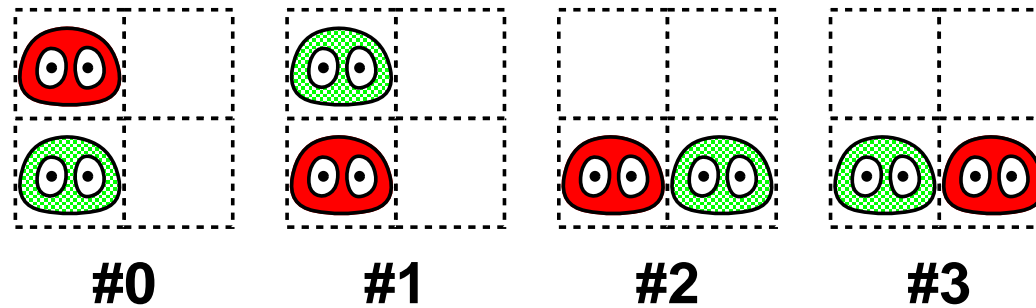
▷ 画像認識の手法

- 各セル内ピクセルの YUV の合計値から判断
- 単純だが外乱に弱い (連鎖表示 ・ 汗 ・ 涙 ・ 「全消し」)
- **NEXT** ぷよ表示部を観察することにより、「配ぷよ」のタイミングを知る

syspuyo2: プレイヤー

▷ プレイヤーのアルゴリズム

- 「組ぶよ」が取りうる姿勢はたかだか 4 通り



- 選択できる「手」の種類は $5 \times 4 + 2 = 22$
- NEXT, NEXT2 ぶよの表示により最大 3 手先までの結果が読める フ
フィールド評価関数を定めて、最善手を求めて探索

syspuyo2: プレイヤー

▷ プレイヤープログラムとのインタフェース

□ プレイヤープログラムは UNIX 的なフィルタとして記述

(input)

Field:OPYGP

Next: PGYB

(output)

Move: 10

□ Ruby で記述した思考ルーチン

- 評価関数方式で 1 手先(22通り)を全探索
- おそい: 反応速度 100ms ~ 300ms
- よわい

syspuyo2: Demonstration

syspuyo2: 今後の改良方針

- ▷ 画像認識アルゴリズムの改良
 - 外乱に強いアルゴリズム
 - マルチメディア命令の使用
- ▷ プレイヤーアルゴリズムの改良
 - 評価関数の改良: 連鎖の定石の認識
 - ニューラルネットワークによる学習 (妄想)
- ▷ 対戦機能の強化
 - 相手プレイヤーフィールドの監視
 - リアルタイムな戦略の変更

まとめ

- ▷ **PlayStation** コントローラ入力装置
- ▷ **Video4Linux2 API** と **bttv** ドライバの改良
- ▷ **vArashi** フレームワークの考案と実装
- ▷ シスぷよ通 (**syspuyo2**) 事例紹介

質疑応答