

神奈川大学オープンキャンパス2022春

体験授業

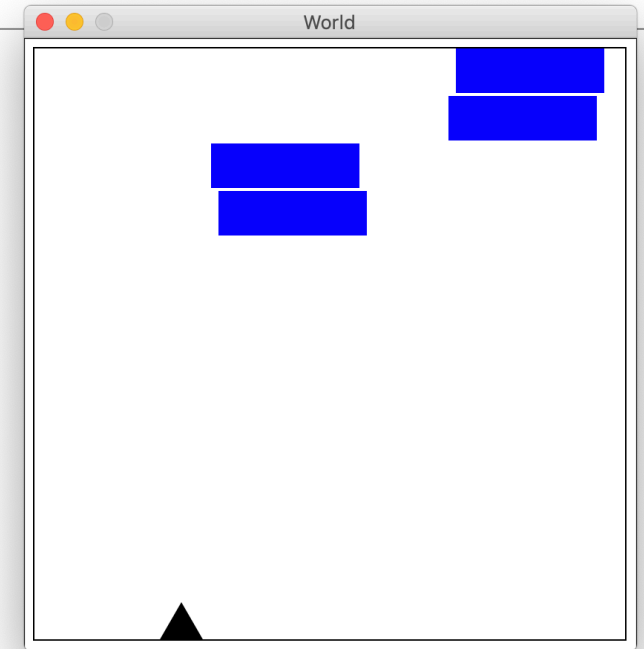
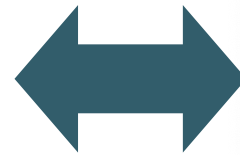
「ゲームをつかって学ぶ関数型プログラミング」

理学部 情報科学科 馬谷 誠二



<https://umatani.github.io/ku-oc/>

有名ゲームと今日つくるゲームの違い

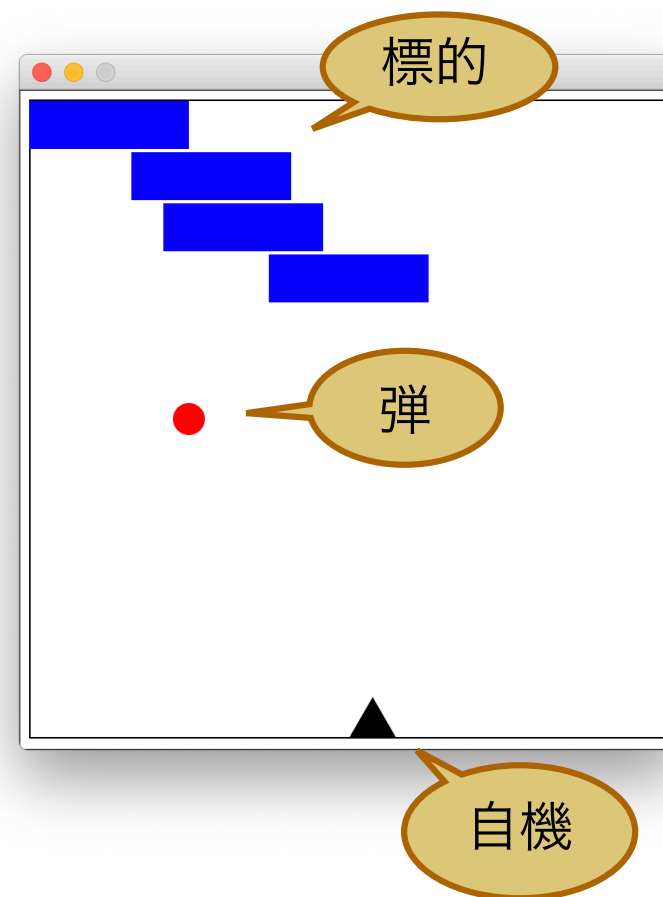


全然違う！

- それでも要素技術は共通（のはず）
 - 図形や画像の描画
 - 時間経過による世界の変化
 - アニメーション
 - キーボード・マウス入力の処理

作成手順

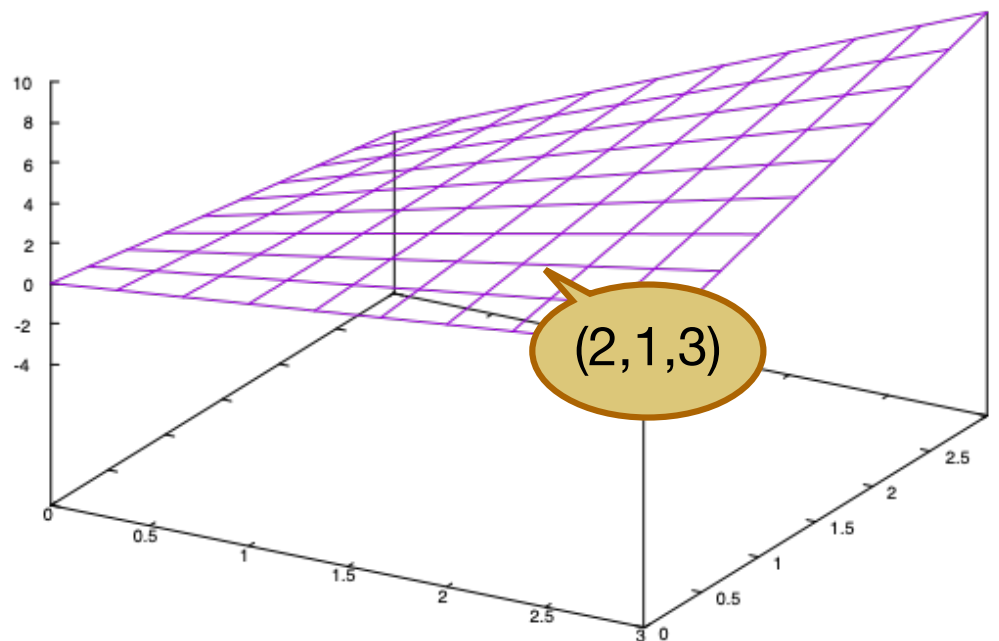
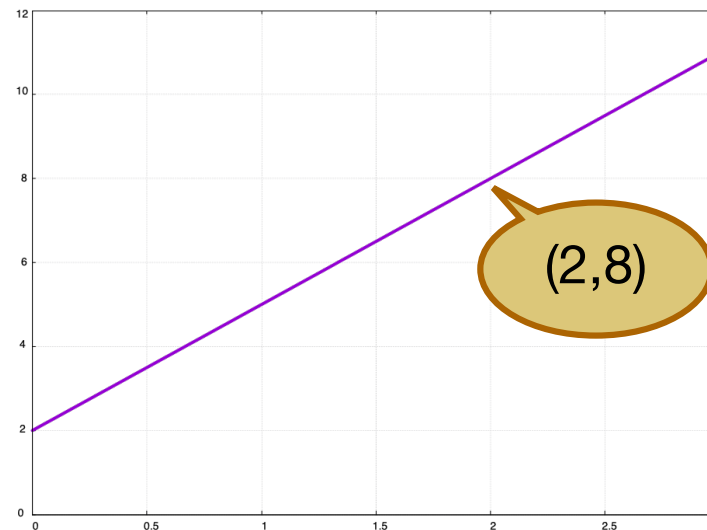
1. 自機の表示と操作
2. 弾の追加
3. 標的の追加
4. 標的の動作の改良
5. 弾の連射機能



1. 自機の表示と操作 ～プログラムの基本要素～

デカルト座標上の関数

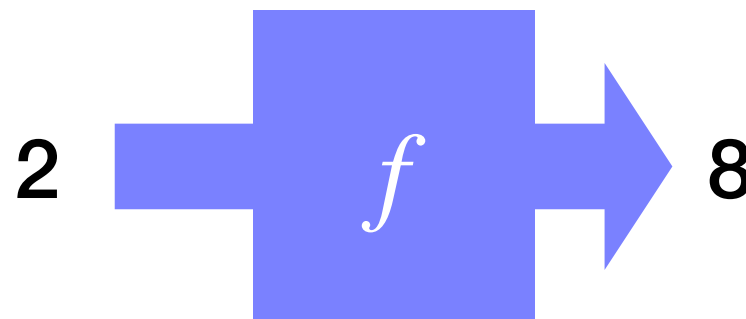
- 1引数関数 $f(x)$
 - $y = f(x) = 3x + 2$
- 2引数関数 $g(x, y)$
 - $z = g(x, y)$
 $= xy + x - y$



計算手続きとしての関数

- 入力を受け取って出力を計算する処理（手続き）

$$f(x) = 3x + 2$$



$$g(x, y) = xy + x - y$$



Racket言語

- 関数型プログラミング言語
 - プログラム = データ処理 = 関数による計算
 - 開発環境（インストール&デモ: `basic.rkt`）
 - インタラクション領域
 - 定義領域
 - 扱うデータの種類
 - 数, テキスト, 画像, etc.

数を扱う簡単な関数

- 例: 絶対値関数

- 定義: $|x| = f(x) = \begin{cases} x & (x \geq 0 \text{ のとき}) \\ -x & (\text{それ以外}) \end{cases}$

- 呼出し: $|3| \implies 3, f(-8) \implies 8$

- Racketで呼出しはこう書く (定義は後述)

```
↳ (absolute 3)  
3  
↳ (absolute -8)  
8  
↳
```

- 関数には意味のある名前をつける
- 括弧の付け方が全然違う (全体を囲む)

数の四則演算

- 簡単な演算も、実は関数の一種
 - 普段書いている書き方は、関数を呼び出す一般的な書き方の省略形

省略形 (数学)

一般形 (数学)

Racketコード

$1 + 2$

$+(1,2)$

`(+ 1 2)`

$1 - 2$

$-(1,2)$

`(- 1 2)`

1×2

$\times(1,2)$

`(* 1 2)`

$1 \div 2$

$\div(1,2)$

`(/ 1 2)`

定数の定義

- 何度も出てくる値に名前をつける → 読みやすくなる
 - 数学でもよく用いられる手法

球の半径 (3) を r , 円周率 (3.141592) を π とする.

球の体積は $\frac{4}{3}\pi r^3$, 表面積は $4\pi r^2$ によって求められる.

- Racketコード

```
(define r 3)
(define pi 3.141592)
(define volume (/ (* 4
                    (* pi
                      (* r r r)))
                  3))
(define surface (* 4 (* pi (* r r))))
```

関数の定義

- 特定の球ではなく，任意サイズの球の体積を計算したい！
 - 数学でもよく用いる

円周率 (3.141592) を π とする.

球の体積は 関数 $f(r) = \frac{4}{3}\pi r^3$ によって求められる.

例えば，半径3の球の体積は $f(3) = \frac{4}{3}\pi \cdot 3^3 \doteq 113.1$ である.

- Racketコード

```
(define pi 3.141592)

(define (volume r)
  (/ (* 4
        (* pi
            (* r r r)))
     3))
```

$$\pi = 3.141592$$

$$\text{volume}(r) = \frac{4}{3}\pi r^3$$

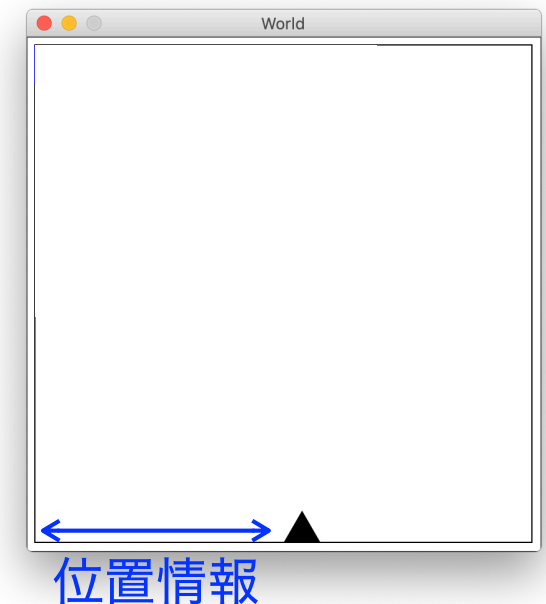
ゲームの世界の数学表現

- 世界 = 時間とともに変化していく状態
 - 状態 = ゲームの世界に存在するモノの性質
 - 画面上に見える全てを記録する必要はない
- 例：自機だけからなるシューティングゲーム

位置情報

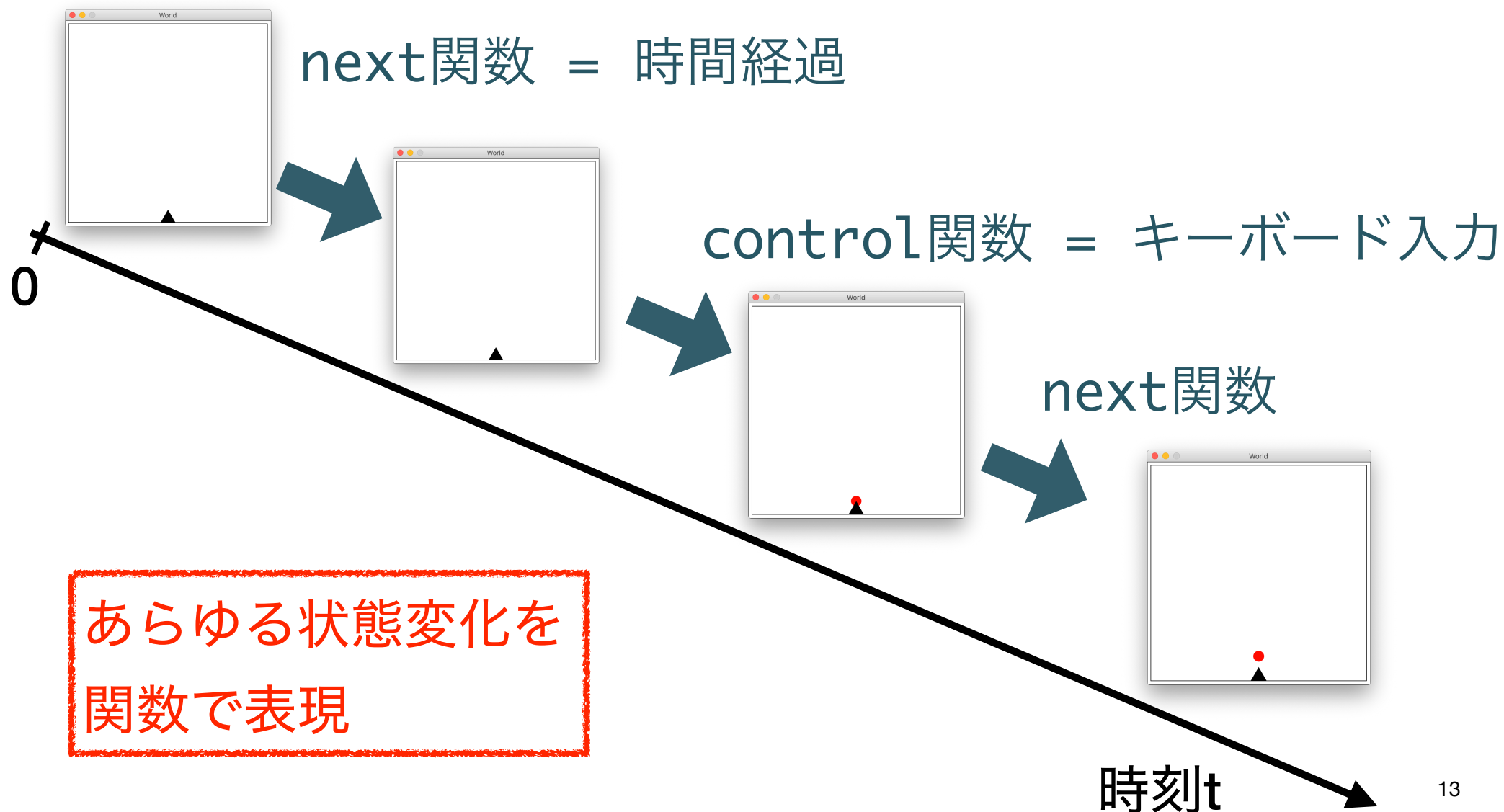
100

- 自機サイズ
 - 三角形であること
- などの「不変データ」は状態の一部ではない



状態変化の関数による

初期状態



参考情報

- Racketホームページ（英語）
 - <https://racket-lang.org>
 - ソフトウェアのダウンロード, マニュアル
- このスライド・ソースコード
 - <https://github.com/umatani/ku-oc.git>
- Schemeの簡単な入門書（日本語訳）
 - <https://www.sampou.org/scheme/t-y-scheme/t-y-scheme.html>
- Schemeを使った情報科学の入門書（世界的に有名）
 - <https://github.com/hiroshi-manabe/sicp-pdf/blob/japanese/jsicp.pdf>
- 私のメールアドレス
 - umatani@kanagawa-u.ac.jp