

# ga-1. ゲームエンジン

2次元と3次元, ゲームエンジンの構成要素,  
ゲームの構成物, ゲームエンジンの例

URL: <https://www.kkaneko.jp/db/game/index.html>

金子邦彦



# 2次元のオブジェクト



## 種類

- ポイント (点)
- ライン (線分, 折れ線)
- エリア (多角形など)
- 文字列 (テキスト)

## 属性

- 色 ※ R, G, B 成分. 各成分は 0 から 255 の数値とすることが多い
- 位置  $x, y$

ここに書いた種類, 属性以外にもいろいろ

# 3次元のオブジェクト



## 種類

- メッシュ（多角形の集まり）
- カメラ（視点と視野）
- ランプ（光源）

## 属性

- 色 ※ R, G, B 成分. 各成分は 0 から 255 の数値とすることが多い
- 位置  $x, y, z$

ここに書いた種類, 属性以外にもいろいろ

# ゲームエンジンの構成要素



## • フレームワーク

コンピュータゲームの主要な機能をまとめたもの  
(ソフトウェアの部品)

## • 開発環境, 動作環境

- 三次元世界の表現,
- 物理シミュレーション (衝突, 摩擦など) ,
- ゲーム制作やテスト用のツール

ゲーム開発の効率化. ゲーム以外 (コンピュータグラフィックス, インタラクティブなアプリケーション) にも

# ゲームエンジンの例

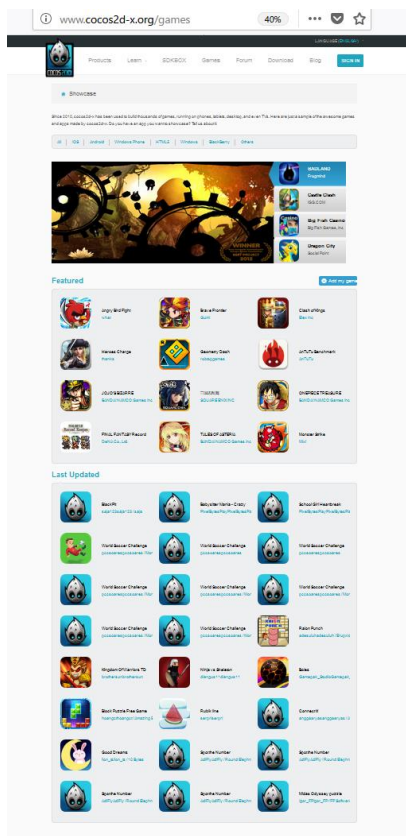


- Unity 3次元. Mono など
- Unreal Engine 3次元. ブループリント, C++ など
- pygame 2次元. Python
- cocos2d 2次元. Python
- cocos2d-x 2次元. C++ など  
(iOS, Andoroidでも動く)

# cocos2d-x のショーケース



- cocos2d-x を用いて制作されたゲームの紹介
- <http://www.cocos2d-x.org/games>



# ゲームの構成要素と相互作業



## ゲーム画面

- ゲームのビジュアル・インターフェース
- プレイヤーに情報を表示
- ゲームの世界を視覚的に提示

## オブジェクト

- ゲーム内の要素（キャラクター、アイテム、障害物など）
- それぞれが独自の属性（位置、サイズ、色など）を持つ

## イベント

- ゲーム内で発生するアクションや状況の変化

例：キャラクターの移動、得点の獲得

## イベントハンドラ

- イベントに対するゲームの反応を定義するプログラム

例：プレイヤーが特定のボタンを押すとキャラクターがジャンプ

# ゲームのオブジェクトとイベント



## • オブジェクト

- 文字列（テキスト）
- 画像（スプライト）
- 2次元のグラフィックス
- 3次元のグラフィックス

## • イベント

- マウス
- キーボード
- タイマー
- 特定の条件



# ゲームエンジン cocos2d でのイベント



- `on_key_press` キーが押された
- `on_key_release` キーが離された
- `on_mouse_motion` マウスが動いた
- `on_mouse_drag` マウスのドラッグ
- `on_mouse_press` マウスのボタンが押された

## 衝突検出

- 衝突が発生した際に、オブジェクトの動きや状態を適切に変更

## 力と運動

- オブジェクトに作用する力（重力、推進力など）を計算し、それに応じてオブジェクトの速度や方向を変更

## 摩擦と反発

- オブジェクトが他の表面と接触する際の摩擦力、衝突時のオブジェクトの反発（跳ね返り）

## 剛体と柔軟体のダイナミクス

- 剛体：変形しない固体オブジェクト
- 柔軟体：布や液体など、変形するオブジェクト

## ジョイント

- オブジェクト間の物理的な関係や制約（例：ヒンジ、ばね）を定義し、それに従ってオブジェクトを動かす。