

2018年度 後期 金曜日3時限

モーションプランニング

前期の WEB デザインから発展させ
JavaScript / Animate CC などを使った
WEB アニメーションを学びます

WEBアニメーションの主な実装方法

CSS / DOMアニメーション

HTMLの要素に対して行うアニメーション。

CSSのみで行う

CSS TransitionやAnimationプロパティを使ってのアニメーション

メリット

- ・比較的簡単に実装できる

デメリット

- ・ツイーンなど複雑なアニメーションは難しい
- ・ユーザーインタラクションを取得できない、しづらい（マウスイベント、タップイベント、スクロールイベント、フォームイベントなど）

JavaScript

JavaScript(以下JS)を使って要素のStyleを変更することで行う。また、ユーザーアクション・環境などを取得してインタラクティブなアニメーションが可能

メリット

- ・自由度が高い
- ・様々なアニメーションに対応したライブラリが豊富にある

デメリット

- ・JSに慣れていないと実装が難しい

〈主なライブラリ〉
TweenMax
anime.js
velocity.js
kute.js
など

Canvas / WebGL

HTML内のcanvas要素に描画する。

Animate CC

AdobeのAnimateCCで制作し、JavaScript/HTMLを書き出す方法

メリット

- ・アプリなので直感的に操作できる

デメリット

- ・パーティクルなど大量のオブジェクトを扱うには向いていない
- ・ユーザーインタラクションを扱うにはJavaScriptを使用するなど工夫が必要

JavaScript

JSで描画・アニメーション・ユーザーイベントなどを扱う。

メリット

- ・自由度が高い
- ・複雑なグラフィック、アニメーション、ゲームなどを制作できる。
- ・3Dや物理エンジンなども実装可

デメリット

- ・JSに慣れていないと実装が難しい
- ・CSS、HTMLとは違うOpenGLやCanvasAPIなど知識が必要

〈主なライブラリ〉
PixiJS (2D WebGL)
three.js (3D WebGL)
create.js (HTML5 canvas)
など

授業の流れ

サンプルなどを使った
基礎演習
1~4回(予定)

上の技術などを使って
自由課題制作

JavaScript 基礎
↓
JS での DOM アニメーション
↓
Animate CC
↓
Canvas / WebGL

課題制作
「アニメーションや
ユーザーインタラクションを用いた
WEB コンテンツ」(仮)
↓
提出・講評