

テーマ1

enchant.js で ゲーム作り

(1)

2015/04/13

苫小牧高専 ソフトウェアテクノロジー部 新人研修

前回のまとめ

ソフテクについて

- ・ ソフトウェアテクノロジー部。
プロコン班とETロボコン班に分かれる。
- ・ 1年生は、プロコン・ETロボコン班関係なく、
「プログラミングでものを作る」 感覚を掴んでもらうことに
する。
- ・ 1、2ヶ月くらいの短いスパンで、
いろいろなテーマに触れる。
- ・ 毎週火曜日 15時40分から情報棟3階 システム実習室に集合！

enchant.jsについて

- ・ 2Dゲームを作るためのフレームワーク（≒便利ツールキット）。
- ・ Javascriptというプログラミング言語で書く。
- ・ 9leapというプログラミングコンテストが開催されている。

開発に必要なツールについて

- ・ enchant.js本体
あとで配る。
- ・ エディタ
TeraPadを使います。
(好みのものがあればそれでOK)
- ・ ブラウザ
Internet Explorer 11 (学校のPCに入っている)

今日のテーマ

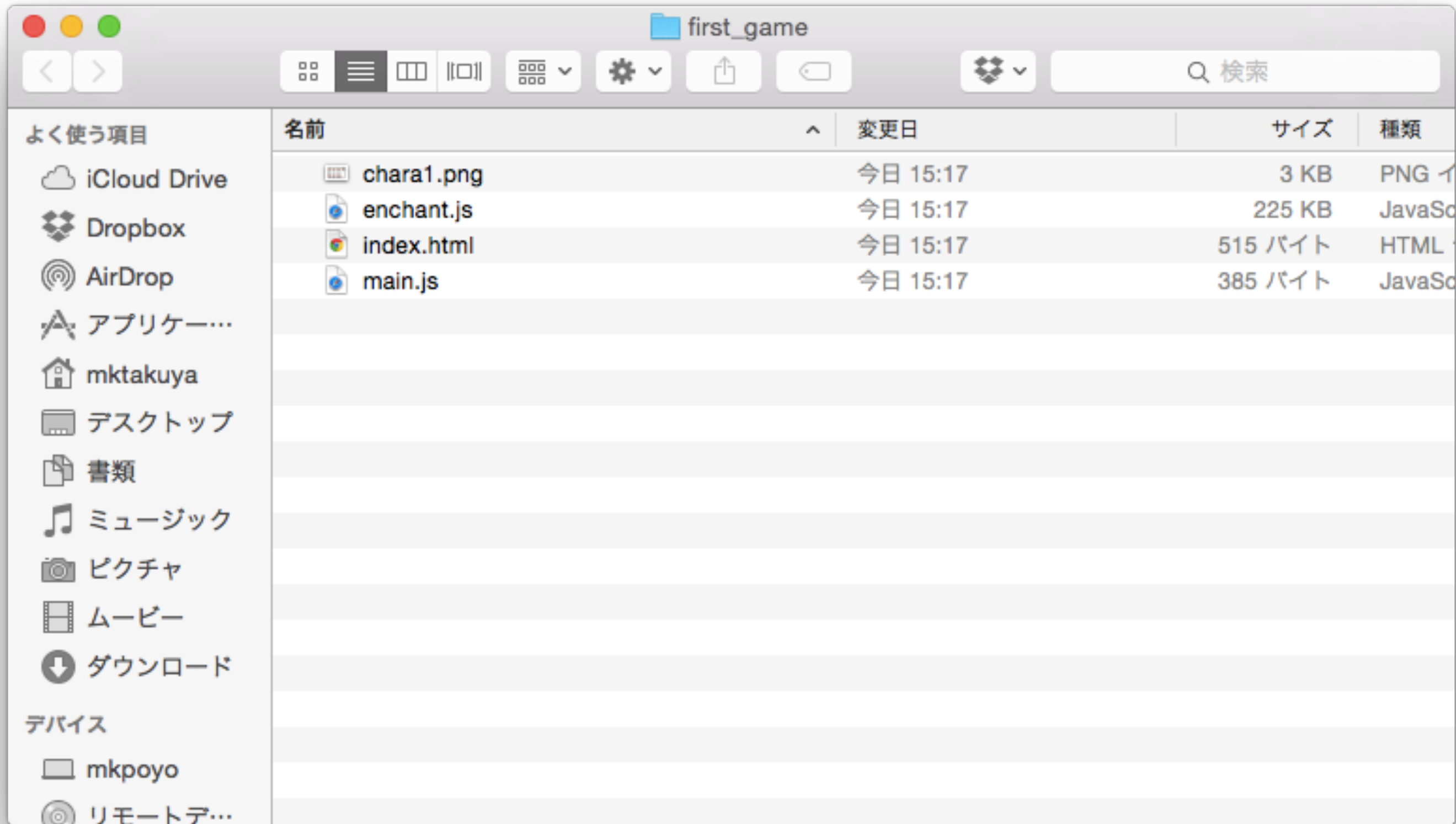
「画像を動かしてみる」

必要なものの配布

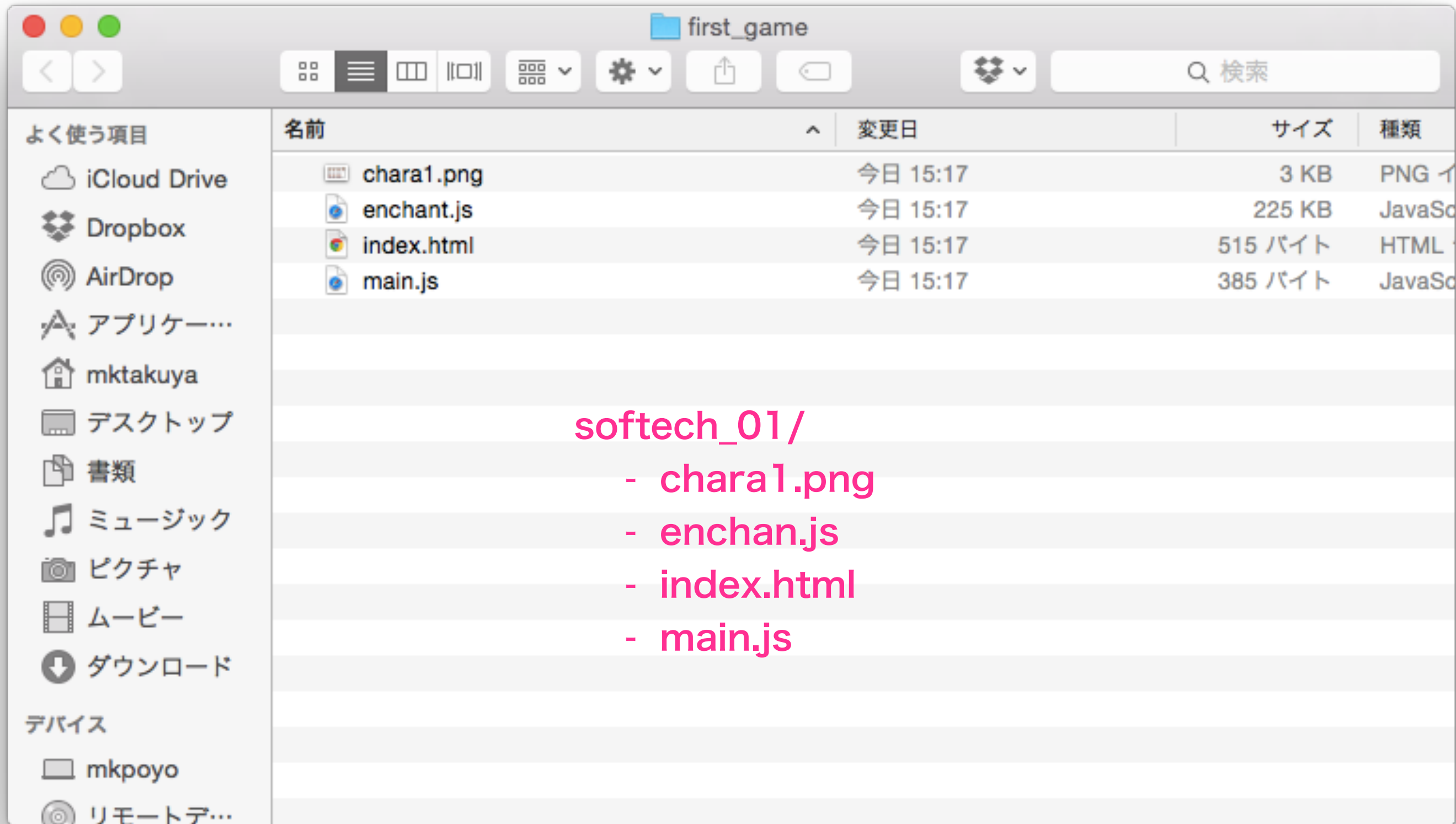
必要なものの配布

- ・ USBメモリに入っている、
softech_01 というフォルダをコピー
- ・ 自分のをコピーしたら隣の人へ。

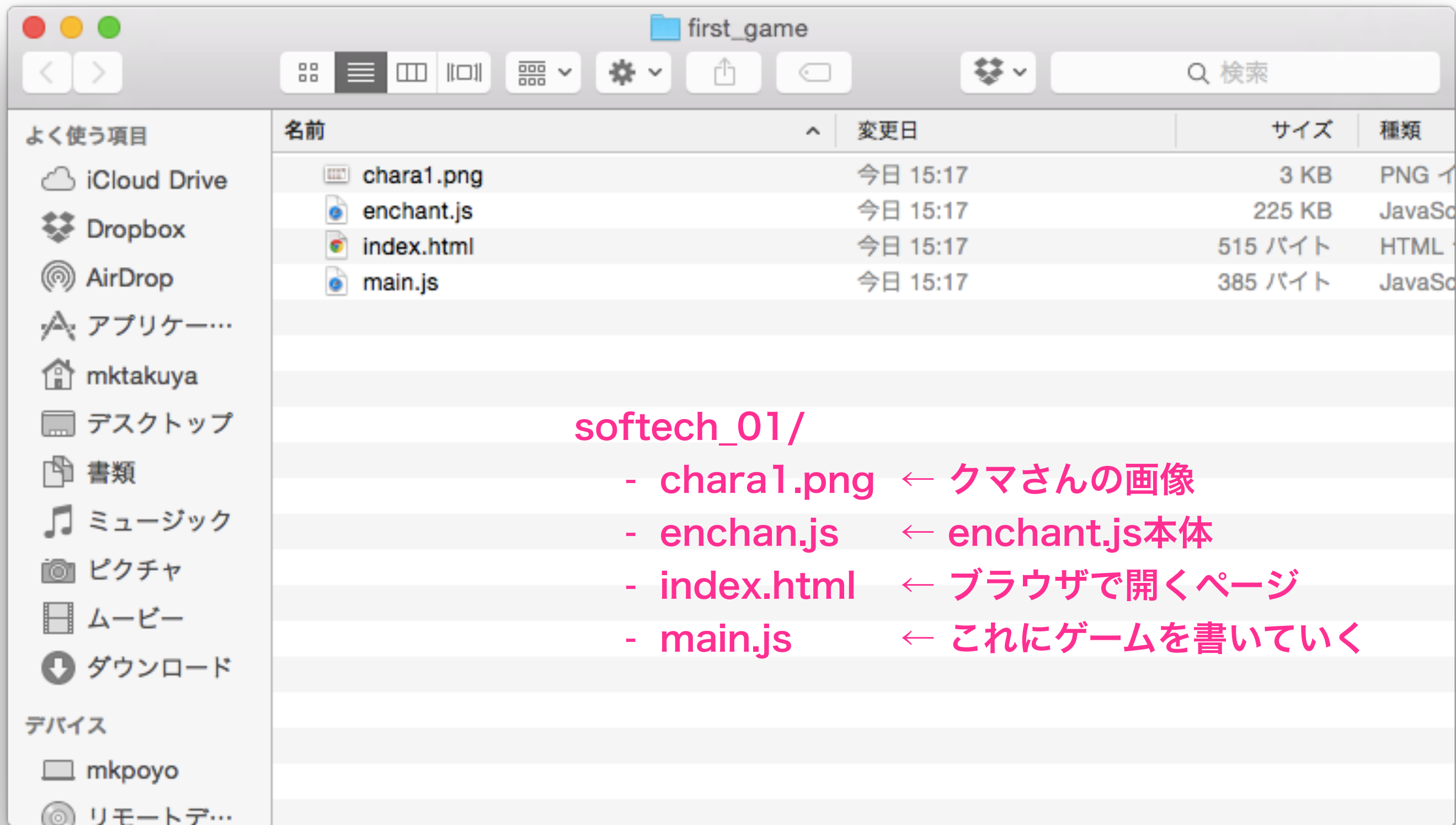
必要なものの配布



必要なものの配布



必要なものの配布



The screenshot shows a macOS Finder window titled 'first_game'. The left sidebar contains 'よく使う項目' (Favorites) and 'デバイス' (Devices). The main pane displays a list of files in the 'first_game' folder.

名前	変更日	サイズ	種類
chara1.png	今日 15:17	3 KB	PNG イメージ
enchant.js	今日 15:17	225 KB	JavaScript
index.html	今日 15:17	515 バイト	HTML
main.js	今日 15:17	385 バイト	JavaScript

よく使う項目

- iCloud Drive
- Dropbox
- AirDrop
- アプリケーション...
- mktakuya
- デスクトップ
- 書類
- ミュージック
- ピクチャ
- ムービー
- ダウンロード

デバイス

- mkpoyo
- リモートデ...

softech_01/

- chara1.png ← クマさんの画像
- enchant.js ← enchant.js本体
- index.html ← ブラウザで開くページ
- main.js ← これにゲームを書いていく

ゲームの構造

ブラウザで開くファイル

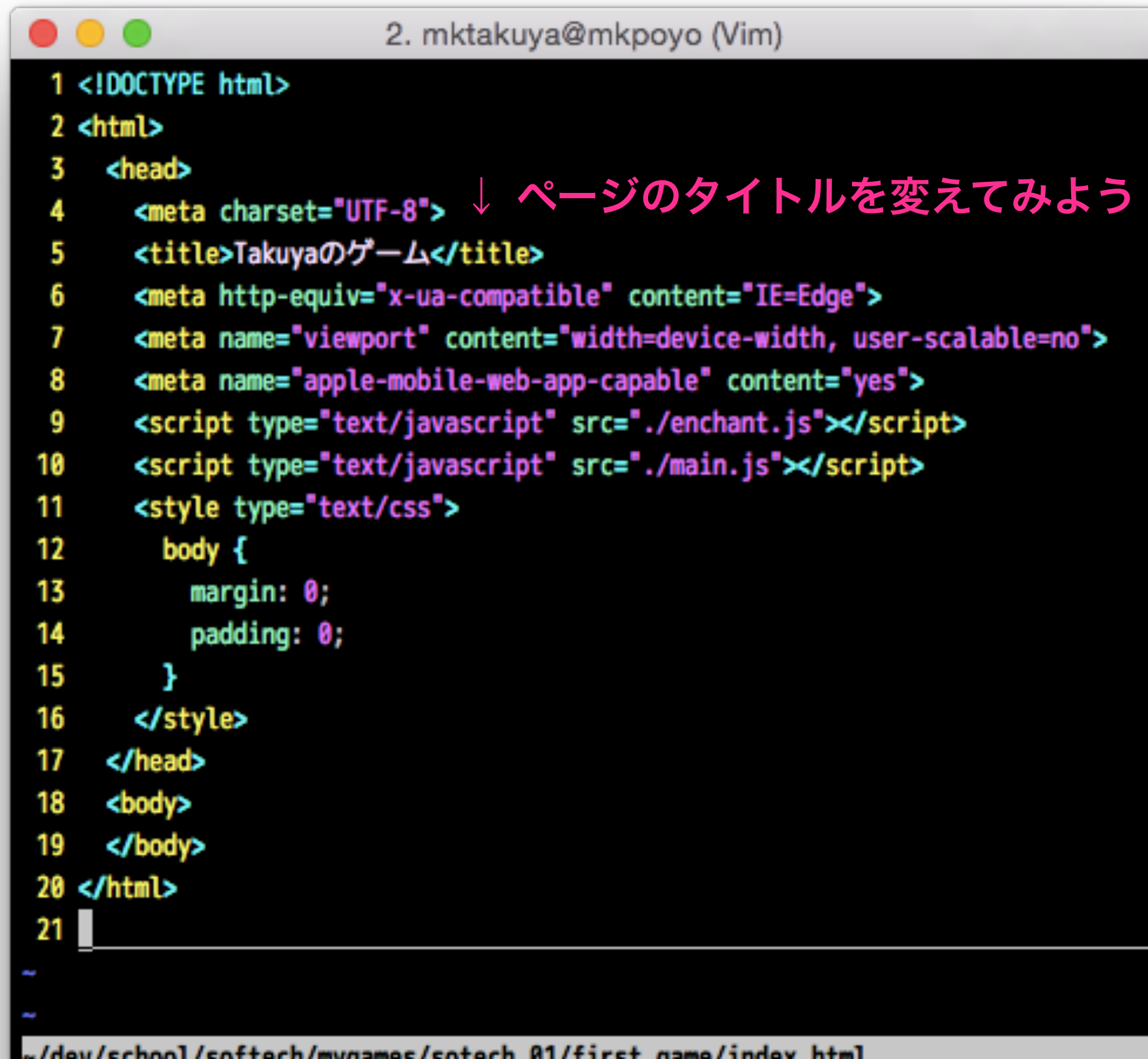
index.html

```
2. mktakuya@mkpoyo (Vim)
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Takuyaのゲーム</title>
6     <meta http-equiv="x-ua-compatible" content="IE=Edge">
7     <meta name="viewport" content="width=device-width, user-scalable=no">
8     <meta name="apple-mobile-web-app-capable" content="yes">
9     <script type="text/javascript" src="./enchant.js"></script>
10    <script type="text/javascript" src="./main.js"></script>
11    <style type="text/css">
12      body {
13        margin: 0;
14        padding: 0;
15      }
16    </style>
17  </head>
18  <body>
19  </body>
20 </html>
21
```

~/dev/school/softtech/mygames/softtech_01/first_game/index.html

ブラウザで開くファイル

index.html



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8"> ↓ ページのタイトルを変えてみよう
5     <title>Takuyaのゲーム</title>
6     <meta http-equiv="x-ua-compatible" content="IE=Edge">
7     <meta name="viewport" content="width=device-width, user-scalable=no">
8     <meta name="apple-mobile-web-app-capable" content="yes">
9     <script type="text/javascript" src="./enchant.js"></script>
10    <script type="text/javascript" src="./main.js"></script>
11    <style type="text/css">
12      body {
13        margin: 0;
14        padding: 0;
15      }
16    </style>
17  </head>
18  <body>
19  </body>
20 </html>
21
```

~/dev/school/softtech/mygames/softtech_01/first_game/index.html

**さっそくコードを
書いていきます**

さっそく コードを書いています

- ・ 「習うより慣れる」ということで、まずはスライドのコードを写す。
- ・ 必要なところは説明するし、わからなくてもゲーム作りを楽しめそうなところについては説明しません。
- ・ 難しい文法がどうこうよりも、まずは「手を動かしてゲームを作る」ことを楽しもう！

ゲーム本体

main.js

```
enchant();

window.onload = function() {
    var core = new Core(320, 320);

    core.onload = function() {

    }
    core.start();
}
```

ゲーム本体

main.js

`enchant();` ← `enchant.js` を使うためのおまじない。

```
window.onload = function() {  
    var core = new Core(320, 320); ← ゲームのCore（核）となる部分  
  
    core.onload = function() {  
        ここに処理を書いていく。  
    }  
    core.start(); ← ゲームスタート！  
}
```

画像を表示する

～クマさん現る～

main.js

```
enchant();

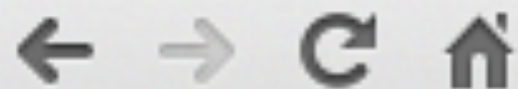
window.onload = function() {
  var core = new Core(320, 320);
  core.preload('chara1.png');

  core.onload = function() {
    var bear = new Sprite(32, 32);
    bear.image = core.assets['chara1.png'];
    bear.x = 0;
    bear.y = 0;

    core.rootScene.addChild(bear);
  }
  core.start();
}
```



index.html



file:///Users/mktakuya/dev/school/softech/mygames/firs



アプリ



みてるなう



amazlet it!



enchantjs



tab



クマさん現る！

main.js

```
enchant();

window.onload = function() {
  var core = new Core(320, 320);
  core.preload('chara1.png'); ← ①

  core.onload = function() {
    var bear = new Sprite(32, 32); ← ②
    bear.image = core.assets['chara1.png']; ← ③
    bear.x = 0; ← ④
    bear.y = 0;

    core.rootScene.addChild(bear); ← ⑤
  }
  core.start();
}
```

画像を表示するには

1. 画像を読み込む。

```
core.preload('chara1.png');
```

2. 変数を定義する。

```
var bear = Sprite();
```

3. ファイル名を指定する。

```
bear.image = core.assets['chara1.png'];
```

4. 座標を指定する。

```
bear.x = 0;
```

```
bear.y = 0;
```

5. bearをシーンに追加する。

```
core.rootScene.addChild(bear);
```

1. 画像を読み込む

- ・ enchant.jsのゲームで画像を使うには、先に読み込んでおく（→ preloadする）必要がある。

```
core.preload('file_name.png');
```

- ・ 読み込んだ画像は、

```
core.assets['file_name.png'];
```


で使うことができる。

2. 変数を定義する

- ・ 変数とは、プログラムの中で値（数、文字、文字列、キャラクターなど）を入れておくための箱のようなもの。
- ・ Javascriptでは、varを使って定義する。
var name = “Mukohira Takuya”;
var age = 19;
- ・ 値には種類があり、プログラムでは型という。
19 → 整数型（int型）
“Takuya Mukohira” → 文字列型
他にもいくつかある。

2. 変数を定義する

- ・ enchant.jsでは、
画像を扱うために少し変わった変数の定義をする。
`var bear = Sprite();`
- ・ これで、
Sprite型の変数bearが定義されたことになる。
- ・ 詳しいことは学習していくうちにわかる（と思う）
ので、今はあまり深く突っ込まないことにする。

2. 変数を定義する

- ・ Sprite型の変数は、内部に様々なデータを持っている。
- ・ そのデータには、
 bear.image, bear.x
などと記述して操作することができる。

3. ファイル名を指定する

- ・ Sprite型の変数bearに、どの画像を割り当てるかを指定する。

```
bear.image = core.assets['chara1.png'];
```

- ・ 使用したい画像は必ず preload しておくこと。

4. 座標を指定する

- Sprite型の変数bearは、座標のデータを持っていて、X座標、Y座標をそれぞれ
bear.x, bear.y と格納されている。
- bear.x = 0; bear.y = 0; とコードを書くことで、
bearの座標を(0, 0)に設定できる。

5. シーンにbearを追加する

- ・ enchant.jsには、シーンという概念がある。
- ・ 定義した変数を表示するには、画像や座標を指定したあとに、シーンに追加しなければならない。

```
core.rootScene.addChild(bear);
```

main.js

```
enchant();
```

```
window.onload = function() {  
  var core = new Core(320, 320);  
  core.preload('chara1.png');
```

← ① 画像を読み込む

```
  core.onload = function() {
```

```
    var bear = new Sprite(32, 32);
```

← ② 変数を定義する

```
    bear.image = core.assets['chara1.png'];
```

← ③ ファイル名を指定する

```
    bear.x = 0;
```

← ④ 座標を指定する

```
    bear.y = 0;
```

```
    core.rootScene.addChild(bear);
```

← ⑤ シーンにbearを追加する

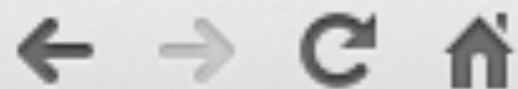
```
  }
```

```
  core.start();
```

```
}
```



index.html



file:///Users/mktakuya/dev/school/softech/mygames/firs



アプリ



みてるなう



amazlet it!



enchantjs



tab



クマさん現る！

課題1

課題1

1. クマさんの位置をずらしてみよう。
2. クマさんをもう一体表示してみよう。

まとめ

- ・ ゲームは、
- ・ ゲーム画面に画像を表示するには、Sprite()を使って変数を作る。
- ・ Sprite() で作った変数は

画像を動かす

こいつ…動くぞ!?



「イベント」という 概念について

プログラミングにおける 「イベント」とは

- ・ プログラム中で起こる何らかの出来事。
- ・ マウスが動いた / マウス左クリックされた / ボタンが押された / 矢印キーが押された / 15時になった … などなど

ゲームプログラミングでの イベント処理

- ・ 普通、プログラムは上から順に実行される。
- ・ 加えてゲームプログラミングでは、「あるイベントが発生した時のみ実行する処理」がよく出てくる。
- ・ 今回は、「フレームが進んだ時」というイベントを検知し、処理を行う。

main.js

```
enchant();

window.onload = function() {
  var core = new Core(320, 320);
  core.preload('chara1.png');

  core.onload = function() {
    var bear = new Sprite(32, 32);
    bear.image = core.assets['chara1.png'];
    bear.x = 0;
    bear.y = 0;

    core.rootScene.addChild(bear);

    bear.addEventListener('enterframe', function() {
      this.x += 3;
    });
  }
  core.start();
}
```

main.js

```
enchant();

window.onload = function() {
  var core = new Core(320, 320);
  core.preload('chara1.png');

  core.onload = function() {
    var bear = new Sprite(32, 32);
    bear.image = core.assets['chara1.png'];
    bear.x = 0;
    bear.y = 0;

    core.rootScene.addChild(bear);

    bear.addEventListener('enterframe', function() {
      this.x += 3; ← ②
    }); ← ①
  }
  core.start();
}
```

イベントに対する処理 の書き方

- ・ ゲームが開始されてから、フレームが1秒に60回進んでいく（60fps）
- ・ 今回のイベントは、「フレームが進んだ時」。
- ・ `bear.addEventListener('enterframe', function(){
 bear.x += 3;
});`

`enterframe`（フレームが進む）ごとに、
`function(){}`の中身を実行するという意味。

- ・ `function(){}`の中には、`bear.x += 3;`が書いてある。
`bear.x += 3;`
は、「`bear.x`（bearのX座標）へ3加える。」という意味。
- ・ `bear.x += 3;`以外の部分はおまじないとして覚えておくが吉。

main.js

```
enchant();

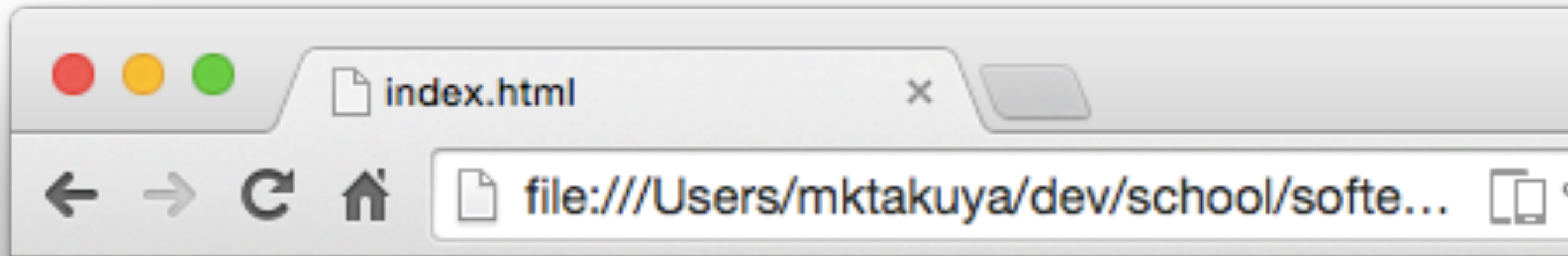
window.onload = function() {
  var core = new Core(320, 320);
  core.preload('chara1.png');

  core.onload = function() {
    var bear = new Sprite(32, 32);
    bear.image = core.assets['chara1.png'];
    bear.x = 0;
    bear.y = 0;

    core.rootScene.addChild(bear);

    bear.addEventListener('enterframe', function() {
      this.x += 3; ← ② bearのX座標を
    });              3進める
  }
  core.start();
}
```

← ① フレームが進むごとに実行する
(おまじない)



クマさん動く！

画像をいろいろ動かす

回転させたり、拡大させたり

bearを 操作することができる

- ・ bear.scaleや、bear.rotateのように、bear.○○ で色々な動作を実現できる。

main.js

```
enchant();

window.onload = function() {
  var core = new Core(320, 320);
  core.preload('chara1.png');

  core.onload = function() {
    var bear = new Sprite(32, 32);
    bear.image = core.assets['chara1.png'];
    bear.x = 0;
    bear.y = 0;

    core.rootScene.addChild(bear);

    bear.addEventListener('enterframe', function() {
      this.x += 3; ← フレームが進むごとに実行したい処理は
    });           この中に書いていく。
  }
  core.start();
}
```

クマさん回転

bear.rotate(2);

main.js

```
enchant();

window.onload = function() {
  var core = new Core(320, 320);
  core.preload('chara1.png');

  core.onload = function() {
    var bear = new Sprite(32, 32);
    bear.image = core.assets['chara1.png'];
    bear.x = 0;
    bear.y = 0;

    core.rootScene.addChild(bear);

    bear.addEventListener('enterframe', function() {
      this.x += 3;

      this.rotate(2);
    });
  }
  core.start();
}
```

クマさん巨大化

bear.scale(1.01, 1.01);

main.js

```
enchant();

window.onload = function() {
  var core = new Core(320, 320);
  core.preload('chara1.png');

  core.onload = function() {
    var bear = new Sprite(32, 32);
    bear.image = core.assets['chara1.png'];
    bear.x = 0;
    bear.y = 0;

    core.rootScene.addChild(bear);

    bear.addEventListener('enterframe', function() {
      this.x += 3;

      this.scale(1.01, 1.02);
    });
  }
  core.start();
}
```

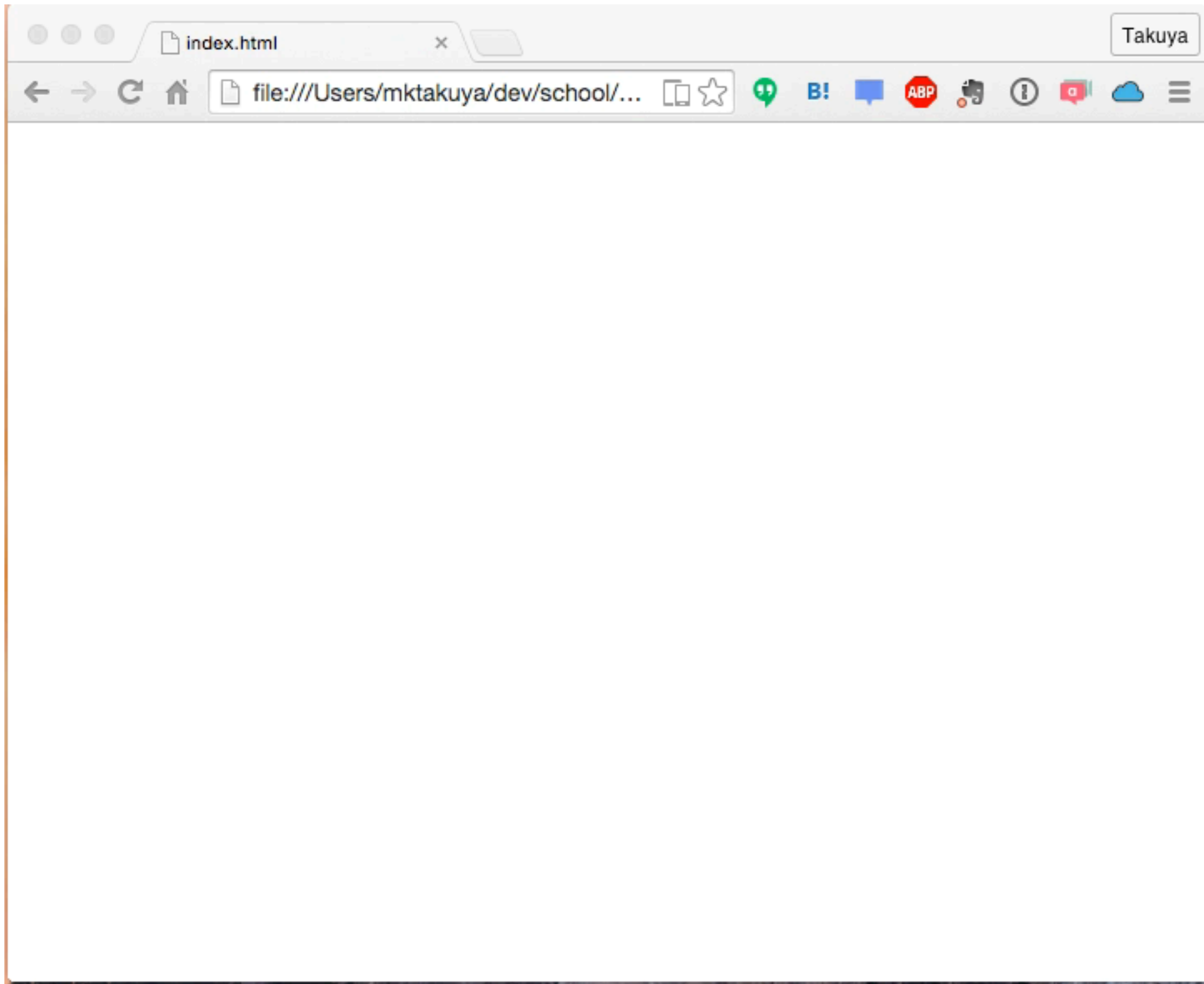
file:///Users/mktakuya/dev/school/softech/mygames/first_game/index.html



図1. 回転しながら巨大化していく不審物

ここで問題

クマさん端っここで消える問題



クマさん端っここで消える問題

- ・ core、つまりゲームの画面サイズは 320 x 320
- ・ bear.x が 320 を超えると、消えてしまう。

どうすれば良いと思う？

クマさん端っここで消える問題

- ・ bear.x が 320を超えた時、
bear.xを0にリセットすれば良いのでは？

続きは次回！