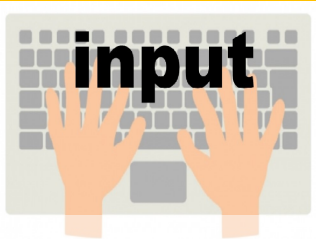


- ① 「05_tic_tac_toe」をダウンロードする。
 - ・解凍して、デスクトップに置く。



- ② 「tic_tac_toe.html」をエディタ（VScode）で開き、以下のコーディングをする。

赤枠のコードをすべて入力すること。

```
26 <div class="squares-container">
27   <div class="squares-box">
28     <div class="square" id="a_1"></div>
29     <div class="square" id="a_2"></div>
30     <div class="square" id="a_3"></div>
31     <div class="square" id="b_1"></div>
32     <div class="square" id="b_2"></div>
33     <div class="square" id="b_3"></div>
34     <div class="square" id="c_1"></div>
35     <div class="square" id="c_2"></div>
36     <div class="square" id="c_3"></div>
37   </div>
38 </div>
39
40 <div class="btn-container">
41   <div class="js-hidden" id="newgame-btn">
42     <button id="btn90" class="btn" type="button">New Game</button>
43   </div>
44 </div>
45
46 </div>
47 </div>
48
49 <!-- フッター -->
```

ここはコーディング済

ここはコーディング済



③ 「tic_tac_toe.css」をエディタ（VScode）で開き、以下のコーディングをする。

赤枠のコードをすべて入力すること。

```
155  /* backボタン */
156  /*****
157  /* class = "back" */
158  .back{
159      margin-top: 8px;
160      text-align: left;
161  }
162  /*****
163  /* New Game ボタン */
164  /*****
165  /* class = ".btn-container" */
166  .btn-container {
167      padding-top: 40px;
168  }
169
170  /* id = "btn90" */
171  #btn90:hover {
172      background-color: #ffd347;
173      transition-duration: 0.4s;
174  }
175  /*****
176  /* footer
```

ここはコーディング済

ここはコーディング済



④ 「tic_tac_toe.js」をエディタ（VSCode）で開き、以下のコーディングをする。



赤枠のコードをすべて入力すること。
「A」～「C」のコードは自分で考えること。

```
10 // squaresの要素（ボックス）を取得（しゅとく）
17 const a_1 = document.getElementById("a_1");
18 const a_2 = document.getElementById("a_2");
19 const a_3 = document.getElementById("a_3");
20 const b_1 = document.getElementById("b_1");
21 const b_2 = document.getElementById("b_2");
22 const b_3 = document.getElementById("b_3");
23 const c_1 = document.getElementById("c_1");
24 const c_2 = document.getElementById("c_2");
25 const c_3 = document.getElementById("c_3");
26
27 // NewGameボタン取得（しゅとく）
28 A
29 B
30
31 // Win or Lose Judgment Line
32 const line1 = JudgLine(squaresArray, ["a_1", "a_2", "a_3"]);
33 const line2 = JudgLine(squaresArray, ["b_1", "b_2", "b_3"]);
```

ここはコーディング済

ここはコーディング済

A...「New Game」ボタンエリアの要素を取得。
定数の定義：newgamebtn_display
HTMLのid名「newgame-btn」の要素を取得しセット

B...「New Game」ボタンの要素を取得。
定数の定義：newgamebtn
HTMLのid名「btn90」の要素を取得しセット

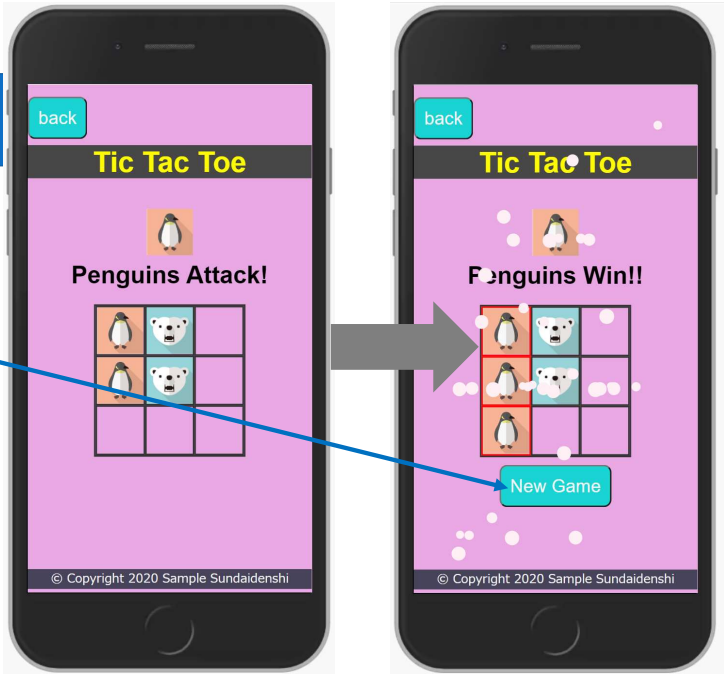
```
210 // ゲーム終了時の処理
211 // *****
212 // 実際に書いてみる！JavaScriptでforEachメソッドを使う方法【初心者向け】：https://t
213 // classListの使い方まとめ：https://qiita.com/tomokichi_ruby/items/2460c5902d19b
214 function gameOver(status) {
215 // all square unclickable
216 squaresArray.forEach(function (square) {
217 square.classList.add("js-unclickable");
218 });
219
220 // display New Game button : display
221 C
222
223 // winEffect
224 if(status==="penguins") {
225 // winner-line penguins high-light
226 if (winningLine) {
227 winningLine.forEach(function (square) {
228 square.classList.add("js-pen_highLight");
```

ここはコーディング済

ここはコーディング済

Cは
要素.classList.remove("クラス名")を使う

C...「New Game」ボタンエリアを表示する。
以下の要素から、クラスを削除する。
要素： newgamebtn_display
クラス名： "js-hidden"



```

262 // NewGameボタン クリック時、ゲーム初期化
263 // *****
264 // classListの使い方まとめ: https://qiita.com/tomokichi_ruby/items/2460c5902d19b81cace5
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284

```

D ... New Gameボタンを押したときの処理を作ってみよう！
ゲームを起動したときの表示状態に戻すよ！

- ① イベントリスナーを追加
ターゲット: 定数 **newgamebtn**
イベントの種類: "click"
- ② クリックした時に、実行する処理
 - ②-1. **flag**を初期化する。...新しいゲームは、ペンギンのターンから始める。
 - ②-2. ターン数カウンター(**counter**)を初期化(9) にする。
 - ②-3. **winningLine** を初期化(null)する。
 - ②-4. 9個のマスを初期化する。

配列: **squaresArray**
forEachの引数: **square**

以下、5つのクラスを削除する。

要素: **square**
"js-pen-checked" ...マス目にペンギンを表示
"js-bear-checked" ...マス目にクマを表示
"js-unclickable" ...マス目をプロテクト
"js-pen_highLight" ...winningLineのマス目を赤い線で囲っている
"js-bear_highLight" ...winningLineのマス目を青い線で囲っている

②-4
配列.forEach と
要素.classList.remove("クラス名") を使う

- ②-5. メッセージのセット。
「メッセージ切り替え」の関数呼び出し
関数名: **setMessage**
引数: **"pen-turn"**
- ②-6. 「NewGame」のボタンエリアを非表示にする。
以下の要素に、クラスを追加する。
要素: **newgamebtn_display**
クラス名: **"js-hidden"**
- ②-7. **snowfall(jQuery plugin)**をストップする。

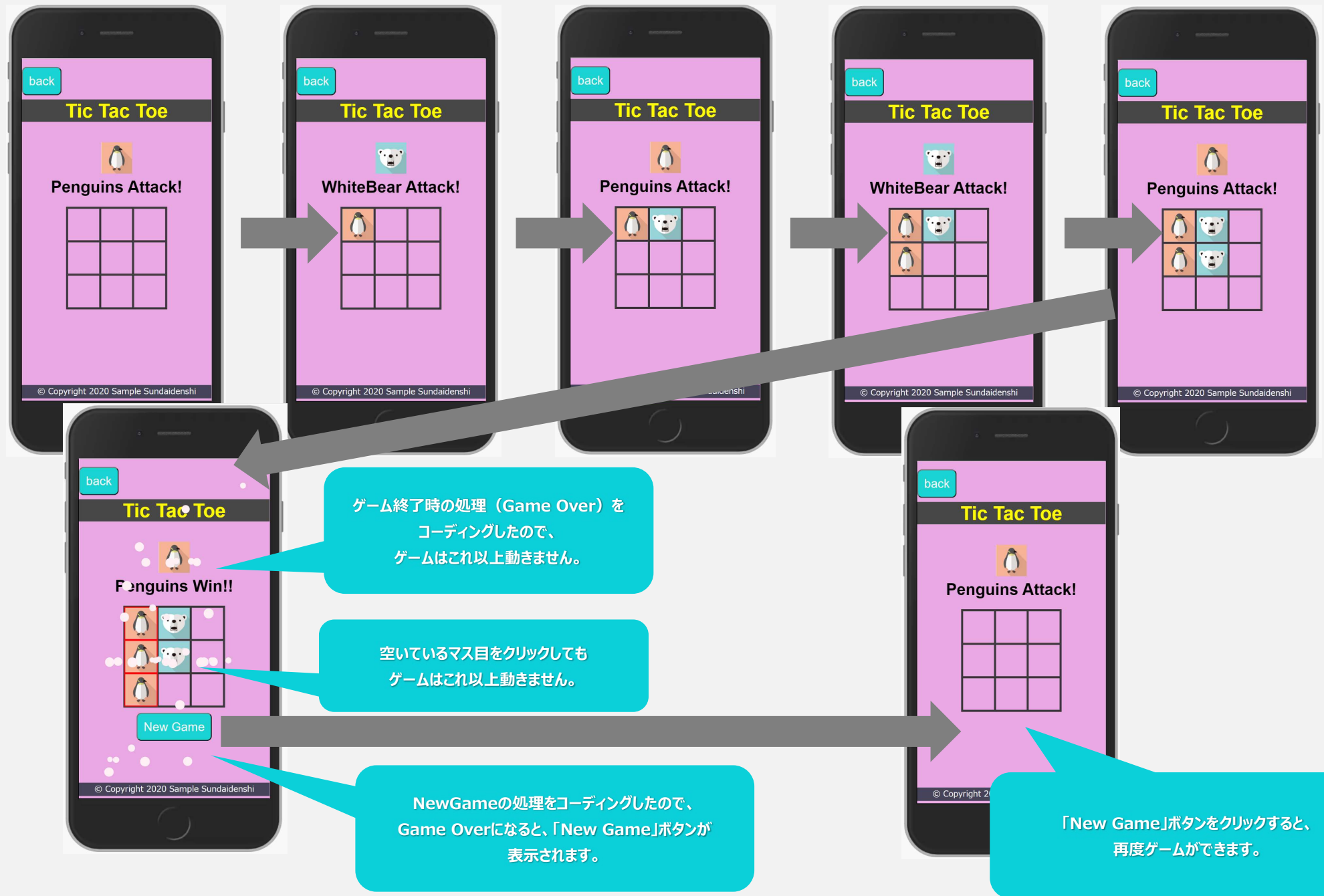
②-6
要素.classList.add("クラス名") を使う

```
// snowfall stop
$(document).snowfall("clear");
```



⑤ 【Chrome で表示して確認】

⑤-1. 同じように画面が表示されているかを確認する。



⑤-2. winningLineは8つあるので、

ペンギンが勝つ 8パターン (①~⑧) すべてをテストすること。

クマが勝つ 8パターン (①~⑧) すべてをテストすること。

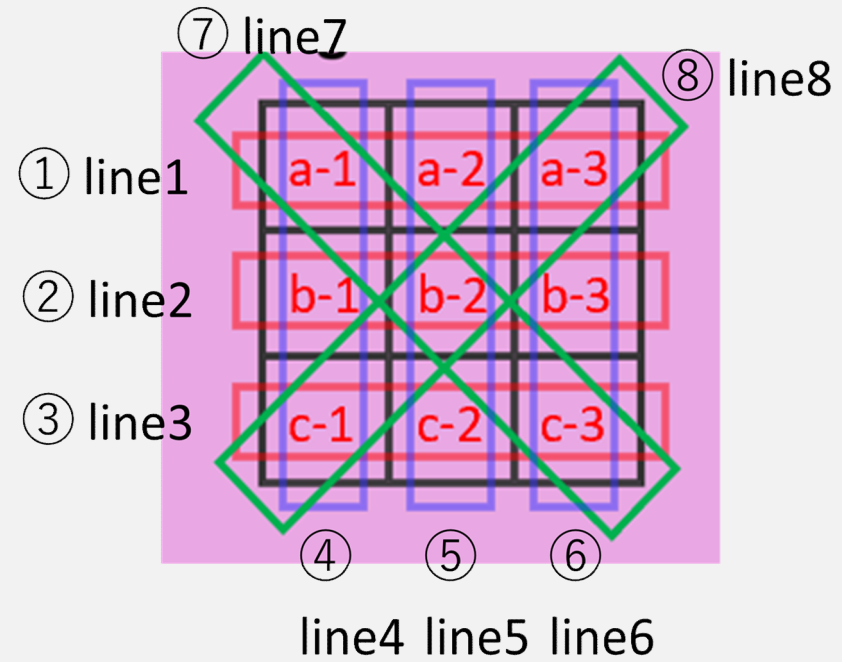
⑤-3. 引き分け (Draw) になるパターン をテストすること。

⑤-4. 「New Game」ボタン をクリックして、再度ゲームができるか確認すること。

ペンギンが勝って「New Game」ボタンを押すパターン をテストすること。

クマが勝って「New Game」ボタンを押すパターン をテストすること。

引き分け (Draw) になって「New Game」ボタンを押すパターン をテストすること。



授業の最後に必ずやること。

・ Classroomに「05_tic_tac_toe」のフォルダを圧縮 (zip) して、提出する。

※絶対に作ったアプリをデスクトップに置いたままにはしないこと!!

オンライン授業になったときに困るから!