

```
1 /**
2  * グローバルオブジェクトを取得
3  */
4 var stage = null;
5 var keyEvent = {};
6 var enemy = [];
7
8 //スコアサービスの準備
9 var score = new lib.score({
10     type: ['hit', 'clear', 'bonus']
11 });
12
13
14
15 /**
16  * アプリエントリポイント (ここから書き始めます)
17  */
18 $(document).ready(function(){
19     setStartboard();
20 });
21
22
23
24 /**
25  * ゲーム画面を作成して表示
26  */
27 var setPlay = function(){
28
29     if($('#stage')){
30         $('#stage').remove();
31     }
32
33     //ステージのセッティング
34     var width = $(window).width();
35     var height = $(window).height();
36
37     //キャンバスの作成
38     stage = $('<canvas id="stage" />').appendTo($(document.body));
39     stage.attr('width', width);
40     stage.attr('height', height);
41
42     //CreateJSで使うステージを作成
43     stage = new createjs.Stage("stage");
44     createjs.Ticker.addListener(stage);
45
46     //バックグラウンドの描画
47     var bg = new app.background({
48         url: app.config.baseUrl + "/images/set1/background/background_fr
49     });
50     stage.addChild(bg.getObject());
51
52     //主人公を描画
```

```
53     var mc = new app.charactor({
54         baseUrl: app.config.baseUrl + "/images/set1/bear"
55     });
56
57     //主人公の初期位置を設定
58     var chara = mc.getObject();
59     chara.x = 200;
60     chara.y = $(window).height() - 200;
61     chara.scaleX = .3;
62     chara.scaleY = .3;
63     stage.addChild(chara);
64
65     //キーイベントを一旦解除
66     $(window).off('keydown', keyEvent.keydown);
67
68     //キーイベントを追加
69     keyEvent.keydown = function(e){
70         switch(e.keyCode){
71
72             //スペースをクリックしたとき
73             case 32:
74                 if(mc.getStatus() === 'normal'){
75                     //万歳させる
76                     mc.banzai();
77
78                     //武器を生成
79                     var wp = new app.weapon({
80                         url: app.config.baseUrl + "/images/set1/accesarry/ap
81                         scale: .3,
82                         target: mc,
83                         enemy: enemy
84                     });
85                     stage.addChild(wp.getObject());
86
87                     setTimeout(function(){
88
89                         //武器を一定方向に投げる
90                         wp.shoot({
91                             y: 0,
92                             time: 1000
93                         });
94                     }, 500);
95
96                 }else{
97                     mc.clear();
98                 }
99                 break;
100
101             //右キーを押したときに実行
102             case 39:
103                 mc.walk(10);
104                 break;
```

```
105
106     //左キーを押したときに実行
107     case 37:
108         mc.walk(-10);
109         break;
110     }
111
112     return;
113 };
114 $(window).keydown(keyEvent.keydown);
115
116 //キーパッドの操作
117 var keyPad = new app.keypad({
118     listeners: {
119         click: function(code){
120             keyEvent.keydown({keyCode: code});
121         }
122     },
123     pad: {
124         red: {
125             name: 'A',
126             top: $(window).height() - 200,
127             left: 10,
128             code: 32
129         },
130         blue: {
131             name: 'B',
132             top: $(window).height() - 200,
133             left: 130,
134             code: 32
135         },
136         cross: {
137             top: $(window).height() - 250,
138             left: $(window).width() - 250
139         }
140     }
141 });
142
143 //敵を作成するためのまとめ関数 (この関数を実行すると1体できる)
144 enemy = [];
145 var setEn = function(initx, inity, scale){
146
147     //敵を作成 (常に横に動いている)
148     var en = new app.charactor({
149         baseUrl: app.config.baseUrl + "/images/set1/bear"
150     });
151
152     //武器が敵に当たったときに実行
153     en.onHit = function(){
154         en.destroy();
155         bg.shake();
156     }
```

```
157
158     //加点する
159     score.set(100, 'hit');
160
161     $.each(enemy, function(k, v){
162         if(en === v){
163             enemy.splice(k, 1);
164             return false;
165         }
166
167         return true;
168     });
169
170     if(enemy.length === 0){
171         keyPad.hide();
172         setScoreboard();
173     }
174     return;
175 }
176
177 //敵オブジェクトを取得してステージに表示
178 var emy = en.getObject();
179 emy.x = initx || 500;
180 emy.y = inity || 200;
181 emy.scaleX = scale;
182 emy.scaleY = scale;
183 stage.addChild(emy);
184
185 //敵は自動的に動く様に
186 var move = function(amount){
187     en.walk(amount, 1000, function(){
188         var a = Math.random() * ($(window).width() / 3) * (Math.
189             if($(window).width() < emy.x + a){
190                 a = a * -1;
191             }else if(emy.x + a < 0){
192                 a = a * -1;
193             }
194
195             if(Math.random() > .5){
196                 en.banzai();
197             }else{
198                 en.clear();
199             }
200
201             move(a);
202         });
203     };
204     move(Math.random() * 100);
205
206     return en;
207 }
208 }
```

```
209 //敵を10体作成
210 for(var i = 0; i < 10; i++){
211     enemy.push(
212         setEn($(window).width() * Math.random(), 300 * Math.random()
213     );
214 }
215 return;
216 };
217
218
219
220
221 /**
222  * スタート画面を表示
223  */
224 var setStartboard = function(){
225     var m = new lib.modal();
226     m.show();
227
228     var s = new app.startboard({
229         title: 'いじわる熊をやっつける!',
230         desc: '森にあらわれたいじわる熊にリンゴをなげつけて、森から追い出せ!',
231         listeners: {
232             close: function(){
233                 m.hide();
234                 s.hide();
235
236                 setTimeout(function(){
237                     setPlay();
238                 }, 1000);
239                 return;
240             }
241         }
242     });
243     s.show();
244     return;
245 };
246
247
248
249
250 /**
251  * ゲーム終了後のスコアボードの表示
252  */
253 var setScoreboard = function(){
254     var m = new lib.modal();
255     m.show();
256
257     var d = score.getDetail();
258     var s = new app.scoreboard({
259         listeners: {
260             close: function(){
```

```
261         m.hide();
262         s.hide();
263
264         setPlay();
265         return;
266     }
267 }
268 });
269 s.show({
270     total: score.get(),
271     detail: [
272         {
273             name: 'ヒット',
274             value: d.hit
275         },
276         {
277             name: 'クリア',
278             value: d.clear
279         },
280         {
281             name: 'ボーナス',
282             value: d.bonus
283         }
284     ]
285 });
286
287 return;
288 };
289
```