

# 情報科学プロジェクト実験レポート課題

S142063 佐藤涼亮

平成 29 年 1 月 31 日

## Web アプリケーション

### 1 課題の内容

#### Web アプリケーションの作成

#### 1.1 要点

1. ユーザのログインとクッキーの管理
2. ユーザごとの最高点をデータベースで管理
3. ユーザごとの最高点のうちベスト 10 をゲーム画面の横に表示

### 2 プログラムの説明

ログイン画面とゲーム画面 (アプリ画面) を作成し、それぞれに対応した処理を行う。ログイン画面には、ユーザー名とパスワードの入力欄を設け、ログインに必要な情報の入力を求め、ログインを行う。ログインされたアカウント情報をデータベースで検索し、新しいアカウント情報なら登録、既存のアカウント情報なら更新し、cookie を設定してゲーム画面に移行する。ゲーム画面では、ゲームに加え、既存のアカウントでの得点ランキングやユーザーが今回獲得した得点を表示している。獲得した得点は、猫を倒すごとに加算されるようになっていて、ゲームオーバーになると同時に、データベースへその記録が格納されるようになっている。cookie の有効期限は登録、更新した時からセッションが終了するまでとする。

入力されたユーザー情報は、獲得した最高得点や cookie とともにデータベースに格納される。データベースのテーブル、スキーマは以下の通りである。

```
sqlite> .table
accountScore
sqlite> .schema
CREATE TABLE accountScore(
  user text not null,
  pass text not null,
  score int,
  cookie text
);
sqlite>
```

## 2.1 目的

Web アプリケーションの作成

## 2.2 方法

JavaScript,HTML,CGI,C++を用いる

## 2.3 結果

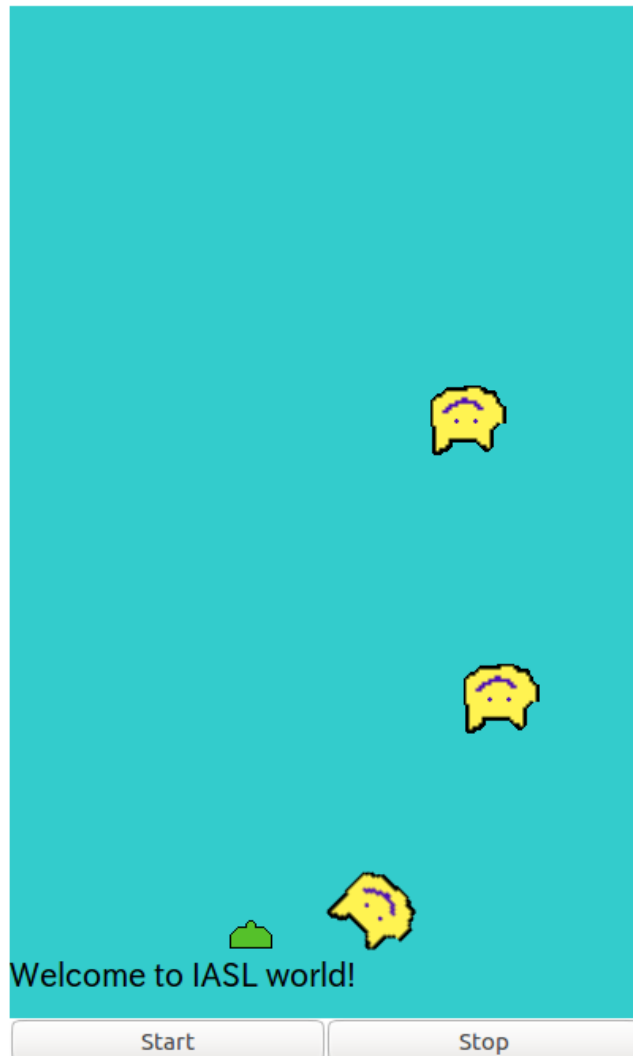
ログイン画面

**ログイン**

ログインしてください

ユーザー名	<input type="text"/>
パスワード	<input type="password"/>
	<input type="button" value="login"/>

## ゲーム画面



your score : 0 point

ランキング		
順位	ユーザー名	スコア
1	user	12
2	hhh	7
3	ddd	5
4	ccc	3
5	aaa	1
6	bbb	1
7	ggg	1
8	abc	0
9	eee	0
10	fff	0

## 2.4 考察

以下が、結果で見られるユーザー情報やランキングである。

```
sqlite> select * from accountScore order by score desc;
user|pass|12|82582066
hhh|hhh|7|58150040
ddd|ddd|5|58086813
ccc|ccc|3|58051599
aaa|aaa|1|97125618
bbb|bbb|1|58035592
ggg|ggg|1|58138835
abc|abc|0|97108010
eee|eee|0|58114825
fff|fff|0|58126030
iii|iii|0|58162845
sqlite>
```

これらより、データの格納・更新は成功し、Web アプリケーションを作成することができた。

## 3 感想

今回、今までの実験の内容を応用し、簡単な Web アプリケーションを作成した。時間がなく、レイアウトが拙くなり、デザインが悪くなってしまった。HTML/CSS デザインを学んだが、経験が少ないこともあり、装飾やレイアウトをどのように変え、どのようにデザインをすればいいのか、まだわからないことだらけである。

順位表示は、タイ表示も考えたが、10 位以内にタイが多いとランキングも無駄に大きくなってしまうので、ユーザー名の辞書順に表示するようにした。

## 4 プログラム

ソースコード 1: iasl.cpp

```
1 #include <random>
2 #include <sstream>
3 #include <iomanip>
4 #include <iostream>
5 #include <ctime>
6 #include <string>
7 #include <MyDBS3.hpp>
8 #include <CGInput.hpp>
9 using namespace std;
10
11 int html_login(int s=0);
12 int html_iasl(string user);
13 bool check_cookie(string cookie);
14 bool check_user(string user);
```

```

15 bool check_account(string user,string pass);
16 void create_account(string user,string pass);
17 void update_cookie(string user);
18
19 int main() {
20     CGIInput tbl;
21     cout << "Content-type: \text/html; \charset=UTF-8\r\n";
22     const char *e = getenv("HTTP_COOKIE");
23     if (e != nullptr) {
24         // クッキーが送られてきた
25         string s = e;
26         bool valid = check_cookie(s); // DB を検索して s を探す
27         if (!valid) return html_login(1); // s がない || 期限が無効
28         return html_iasl(tbl["user"]);
29     }
30     e = getenv("REQUEST_METHOD");
31     if ((e != nullptr && string(e) == "POST")) {
32
33         string user = tbl["user"];
34         string pass = tbl["pass"];
35
36         if (user == "" || pass == "") return html_login(2); // form 入力エラー
37
38         if (!check_user(user)) create_account(user,pass); // 新アカウント作成
39
40         else if (check_account(user,pass)) update_cookie(user); // cookie を更新
41
42         else return html_login(2); // pass 間違い
43
44         return html_iasl(tbl["user"]); // 新規またはlogin 成功
45     }
46     return html_login();
47 }
48
49 // login html
50 int html_login(int s){
51     cout << R"(
52     <!doctype html>
53     <html>
54     <head>
55     <meta charset="utf-8">
56     <title>Login</title>
57     </head>
58     <body>
59     <h1 align="center" style="background-color: #F2F2E5;">ログイン</h1>
60     )";
61
62     // 状況に応じた表示
63     s==2?cout << R"(<p align="center" style="color: #ff0000;">入力エラー</p>)":
64     s==1?cout << R"(<p align="center" style="color: #ff0000;">セッションタイムアウト</p>)":
65     cout << R"(<p align="center">>ログインしてください</p>);
66
67     cout << R"(
68     <form action="iasl.cgi" method="post">
69     <table align="center">
70     <tr>
71     <th><label for="user">ユーザー名</label></th>
72     <td><input name="user" size="20" type="text" /></td>
73     </tr>
74     <tr>
75     <th><label for="pass">パスワード</label></th>
76     <td><input name="pass" size="20" type="password" /></td>
77     </tr>
78     <tr>
79     <th></th>
80     <td align="right"><input type="submit" name="OK" value="login" /></td>
81     </tr>

```

```

82 </table>
83 </form>
84 </body>
85 </html>
86 );
87 return 0;
88 }
89
90 // search.html and cgi
91 int html_iasl(string user){
92     cout << R"(
93     <!DOCTYPE html><html>
94     <head>
95     <meta charset="UTF-8">
96     <title>iasl app</title>
97     <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
98     <script src="/iasl.js"></script>
99     <script src="/iasl.html.js"></script>
100    </head>
101    <body onLoad='iasl.start()'>
102    <div style="position:absolute;left: 500px;">
103    <p>your score : <span id='score'>0</span> point</p>
104    <table border=1 cellspacing=0 style="border-collapse: collapse;">
105    <tr style="background: #fff99;font-size: 30px;">
106    <th colspan=3>ランキング</th>
107    </tr>
108    <tr>
109    <th style="background: #ffcccc;">順位</th>
110    <th style="background: #ffcccc;">ユーザー名</th>
111    <th style="background: #ffcccc;">スコア</th>
112    </tr>
113    )";
114
115    MyDBS d("db/accountScore.db");
116    if(!d) return 1;
117
118    string sql = "select user,score from accountScore order by score desc";
119    if(d.prepare(sql) != SQLITE_OK){
120        return 1;
121    }
122    int count=1;
123    string username;
124    int score;
125    while(d.step(&username,&score) == SQLITE_ROW){
126        cout << "<tr>"
127             "<td>" << count << "</td>"
128             "<td>" + username + "</td>"
129             "<td>" << score << "</td>"
130             "</tr>";
131        if(count++ == 10) break;
132    }
133
134    cout << R"(
135    </table>
136    </div>
137    </body>
138    </html>
139    )";
140
141    return 0;
142 }
143
144 // cookieの有無と期限の確認
145 bool check_cookie(string cookie){
146     MyDBS d("db/accountScore.db");
147     if(!d) return 0;
148
149     cookie = cookie.substr(5);
150

```

```

151 string sql = "select count(*) from accountScore where cookie=?";
152 if(d.prepare(sql,cookie) != SQLITE_OK){
153     return 0;
154 }
155 int count;
156 if(d.step(&count) == SQLITE_ROW){
157     if(count > 0) return 1;
158 }
159 return 0;
160 }
161
162 // userの有無
163 bool check_user(string user){
164     MyDBS d("db/accountScore.db");
165     if(!d) return 0;
166
167     string sql = "select count(*) from accountScore where user=?";
168     if(d.prepare(sql,user) != SQLITE_OK){
169         return 0;
170     }
171     int count=0;
172     if(d.step(&count) == SQLITE_ROW){
173         if(count > 0) return 1;
174     }
175     return 0;
176 }
177
178 // user と pass 対応する 2 つの有無
179 bool check_account(string user,string pass){
180     MyDBS d("db/accountScore.db");
181     if(!d) return 0;
182
183     string sql = "select count(*) from accountScore where user=? and pass=?";
184     if(d.prepare(sql,user,pass) != SQLITE_OK){
185         return 0;
186     }
187     int count=0;
188     if(d.step(&count) == SQLITE_ROW){
189         if(count > 0) return 1;
190     }
191     return 0;
192 }
193
194 // 新しいuserの設定
195 void create_account(string user,string pass){
196     MyDBS d("db/accountScore.db");
197     if(!d) exit(2);
198
199     string sql = "insert into accountScore values(?,?,?,?)";
200
201     static default_random_engine dre((unsigned)time(0));
202     uniform_int_distribution<int> di(1,99999999);
203     stringstream ss;
204     ss << setfill('0') << setw(8) << di(dre);
205     string cookie = ss.str();
206
207     if(d.exec(sql,user,pass,0,cookie) == SQLITE_OK){
208         // cookieの設定
209         cout << "Set-Cookie: name=" + cookie + ";"
210              << "Path=/cgi/"
211              << "\r\n";
212     }
213 }
214
215 // 既存userの更新
216 void update_cookie(string user){
217     MyDBS d("db/accountScore.db");
218     if(!d) return;
219

```

```

220 string sql = "update_accountScore_set_cookie=?where_user=?";
221
222 static default_random_engine dre((unsigned)time(0));
223 uniform_int_distribution<int> di(1,99999999);
224 stringstream ss;
225 ss << setfill('0') << setw(8) << di(dre);
226 string cookie = ss.str();
227
228 if(d.exec(sql,cookie,user) == SQLITE_OK){
229     // cookie の設定
230     cout << "Set-Cookie: name=" + cookie + "; "
231           "Path=/cgi/"
232           << "\r\n";
233 }
234 }

```

## ソースコード 2: store.cpp

```

1  #include <random>
2  #include <sstream>
3  #include <iomanip>
4  #include <iostream>
5  #include <ctime>
6  #include <string>
7  #include <MyDBS3.hpp>
8  #include <CGInput.hpp>
9  using namespace std;
10
11 int main() {
12     CGInput tbl;
13     cout << "Content-type: text/plain; charset=UTF-8\r\n\r\n";
14     const char *e = getenv("HTTP_COOKIE");
15     if (e != nullptr) {
16         // クッキーが送られてきた
17         string s = e;
18         s = s.substr(5);
19         cout << s;
20         MyDBS d("db/accountScore.db");
21         if(!d) { cout << "open"; return 1; }
22
23         string sql = "select_score_from_accountScore_where_cookie=?";
24         if(d.prepare(sql,s) != SQLITE_OK){
25             cout << "prepare";
26             return 1;
27         }
28         int score;
29         if(d.step(&score) != SQLITE_ROW){
30             cout << "step";
31             return 1;
32         }
33
34         if(score<stoi(tbl["score"])){
35             string sql = "update_accountScore_set_score=?where_cookie=?";
36             if(d.exec(sql,stoi(tbl["score"]),s) != SQLITE_OK){
37                 cout << "exec";
38                 return 1;
39             }
40             cout << "ok";
41         }
42         return 0;
43     }
44     cout << "not";
45     return 0;
46 }

```

## ソースコード 3: iasl.html.js



```

1 obj_nnn10 = new Array('nnn2.png', 'nnn0.png', 'nnn1.png', ''); // cat
2 obj_nnn18 = new Array('nnn3.png', 'nnn4.png', 'nnn5.png', 'nnn6.png', ''); // end
3 obj_nnn14 = new Array('nnn7.png', 'nnn8.png', 'nnn2.png', 'nnn9.png', ''); // kill
4
5 function trans_nnn10_nnn12(e, out, delayflg){
6     if (out) { // action
7         setImage(e, e.iasl_img[0], 45, 0);
8         return;
9     }
10
11     // transition
12     if (e.iasl_pending) { e.iasl_pending = 0; return; }
13     if (e.iasl_count+1 >= 10) { e.iasl_func = trans_nnn10_nnn11; return; }
14     { e.iasl_func = trans_nnn10_nnn12; return; }
15 }
16
17 function trans_nnn10_nnn15(e, out, delayflg){
18     if (out) { // action
19         setImage(e, e.iasl_img[0], 45, 0);
20         move_rel(e, 8, 8);
21         return;
22     }
23
24     // transition
25     if (e.iasl_pending) { e.iasl_pending = 0; return; }
26     if (e.iasl_bumped & (1<<iasl_bumped_d)) { e.iasl_func = trans_nnn10_nnn12; return; }
27     if (e.iasl_bumped & (1<<iasl_bumped)
28         && isBumpedObj(e, obj_nnn14) == true) { scoreAdd(); e.iasl_func = trans_nnn10_nnn11;
29         return; }
30     if (e.iasl_count+1 >= 10) { e.iasl_func = trans_nnn10_nnn13; return; }
31     { e.iasl_func = trans_nnn10_nnn15; return; }
32 }
33
34 function trans_nnn10_nnn13(e, out, delayflg){
35     if (out) { // action
36         setImage(e, e.iasl_img[1], 0, 0);
37         move_rel(e, 0, 8);
38         return;
39     }
40
41     // transition
42     if (e.iasl_pending) { e.iasl_pending = 0; return; }
43     if (gamble(50)) { e.iasl_func = trans_nnn10_nnn16; return; }
44     { e.iasl_func = trans_nnn10_nnn15; return; }
45 }
46
47 function trans_nnn10_nnn17(e, out, delayflg){
48     if (out) { // action
49         setImage(e, e.iasl_img[2], 0, 0);
50         move_abs(e, getRandX(), e.iasl_y);
51         return;
52     }
53
54     // transition
55     if (e.iasl_pending) { e.iasl_pending = 0; return; }
56     if (gamble(10)) { e.iasl_func = trans_nnn10_nnn16; return; }
57     { e.iasl_func = trans_nnn10_nnn17; return; }
58 }
59
60 function trans_nnn10_nnn16(e, out, delayflg){
61     if (out) { // action
62         setImage(e, e.iasl_img[0], -45, 0);
63         move_rel(e, -8, 8);
64         return;
65     }
66
67     // transition
68     if (e.iasl_pending) { e.iasl_pending = 0; return; }

```

```

68     if (e.iasl_bumped & (1<<iasl_bumped_d)) { e.iasl_func = trans_nnn10_nnn12; return; }
69     if (e.iasl_bumped & (1<<iasl_bumped)
70         && isBumpedObj(e, obj_nnn14) == true) { scoreAdd(); e.iasl_func = trans_nnn10_nnn11;
        return; }
71     if (e.iasl_count+1 >= 10) { e.iasl_func = trans_nnn10_nnn13; return; }
72     { e.iasl_func = trans_nnn10_nnn16; return; }
73 }
74
75 function trans_nnn10_nnn11(e, out, delayflg){
76     if (out) { // action
77         setImage(e, e.iasl_img[2], 0, 0);
78         move_abs(e, 200, 0);
79         return;
80     }
81
82     // transition
83     if (e.iasl_pending) { e.iasl_pending = 0; return; }
84     { e.iasl_func = trans_nnn10_nnn17; return; }
85 }
86
87 function trans_nnn18_nnn19(e, out, delayflg){
88     if (out) { // action
89         setImage(e, e.iasl_img[0], 0, 0);
90         return;
91     }
92
93     // transition
94     if (e.iasl_pending) { e.iasl_pending = 0; return; }
95     { e.iasl_func = trans_nnn18_nnn19; return; }
96 }
97
98 function trans_nnn18_nnn22(e, out, delayflg){
99     if (out) { // action
100         setImage(e, e.iasl_img[1], 0, 0);
101         follow_mouse(e);
102         return;
103     }
104
105     // transition
106     if (e.iasl_pending) { e.iasl_pending = 0; return; }
107     if (e.iasl_bumped & (1<<iasl_bumped)
108         && isBumpedObj(e, obj_nnn10) == true) { e.iasl_func = trans_nnn18_nnn20; return; }
109     if (e.iasl_bumped & (1<<iasl_bumped)
110         && isBumpedObj(e, obj_nnn14) == true) { /*scoreAdd()*/; e.iasl_func =
        trans_nnn18_nnn20; return; }
111     if (e.iasl_count+1 >= 7) { e.iasl_func = trans_nnn18_nnn21; return; }
112     { e.iasl_func = trans_nnn18_nnn22; return; }
113 }
114
115 function trans_nnn18_nnn21(e, out, delayflg){
116     if (out) { // action
117         setImage(e, e.iasl_img[2], 0, 0);
118         follow_mouse(e);
119         return;
120     }
121
122     // transition
123     if (e.iasl_pending) { e.iasl_pending = 0; return; }
124     if (e.iasl_clicked == iasl_clicked_l) { e.iasl_func = trans_nnn18_nnn23; return; }
125     if (e.iasl_bumped & (1<<iasl_bumped)
126         && isBumpedObj(e, obj_nnn10) == true) { e.iasl_func = trans_nnn18_nnn20; return; }
127     if (e.iasl_bumped & (1<<iasl_bumped)
128         && isBumpedObj(e, obj_nnn14) == true) { e.iasl_func = trans_nnn18_nnn20; return; }
129     { e.iasl_func = trans_nnn18_nnn21; return; }
130 }
131
132 function trans_nnn18_nnn23(e, out, delayflg){
133     if (out) { // action

```

```

134     setImage(e, e.iasl_img[2], 0, 0);
135     follow_mouse(e);
136     if (delayflg) { e.iasl_pending = 1; return; }
137     { var sibling = new Array();
138       iasl_fork(e, obj_nnn14, trans_nnn14_nnn24, sibling);
139       e.iasl_children.push(sibling); }
140     return;
141   }
142
143   // transition
144   if (e.iasl_pending) { e.iasl_pending = 0; return; }
145   if (e.iasl_bumped & (1<<iasl_bumped)
146     && isBumpedObj(e, obj_nnn10) == true) { e.iasl_func = trans_nnn18_nnn20; return; }
147   { e.iasl_func = trans_nnn18_nnn22; return; }
148 }
149
150 function trans_nnn18_nnn20(e, out, delayflg){
151   if (out) { // action
152     setImage(e, e.iasl_img[3], 0, 0);
153     rove(e, 1, 0);
154     return;
155   }
156
157   // transition
158   if (e.iasl_pending) { e.iasl_pending = 0; return; }
159   if (e.iasl_count+1 >= 9) { scoreStore(); e.iasl_func = trans_nnn18_nnn19; return; }
160   { e.iasl_func = trans_nnn18_nnn20; return; }
161 }
162
163 function trans_nnn14_nnn24(e, out, delayflg){
164   if (out) { // action
165     setImage(e, e.iasl_img[0], 0, 0);
166     move_rel(e, 12, -12);
167     return;
168   }
169
170   // transition
171   if (e.iasl_pending) { e.iasl_pending = 0; return; }
172   { e.iasl_func = trans_nnn14_nnn25; return; }
173 }
174
175 function trans_nnn14_nnn27(e, out, delayflg){
176   if (out) { // action
177     setImage(e, e.iasl_img[1], 15, 0);
178     return;
179   }
180
181   // transition
182   if (e.iasl_pending) { e.iasl_pending = 0; return; }
183   if (e.iasl_count+1 >= 8) { e.iasl_func = trans_nnn14_nnn26; return; }
184   { e.iasl_func = trans_nnn14_nnn27; return; }
185 }
186
187 function trans_nnn14_nnn25(e, out, delayflg){
188   if (out) { // action
189     setImage(e, e.iasl_img[0], 0, 0);
190     move_rel(e, 0, -5);
191     return;
192   }
193
194   // transition
195   if (e.iasl_pending) { e.iasl_pending = 0; return; }
196   if (e.iasl_bumped & (1<<iasl_bumped)
197     && isBumpedObj(e, obj_nnn10) == true) { e.iasl_func = trans_nnn14_nnn27; return; }
198   if (e.iasl_bumped & (1<<iasl_bumped)
199     && isBumpedObj(e, obj_nnn18) == true) { freeInstance(e); return; }
200   if (e.iasl_bumped & (1<<iasl_bumped_u)) { freeInstance(e); return; }
201   { e.iasl_func = trans_nnn14_nnn25; return; }

```

```

202 }
203
204 function trans_nnn14_nnn28(e, out, delayflg){
205     if (out) { // action
206         setImage(e, e.iasl_img[2], 0, 0);
207         return;
208     }
209
210     // transition
211     if (e.iasl_pending) { e.iasl_pending = 0; return; }
212     { freeInstance(e); return; }
213 }
214
215 function trans_nnn14_nnn26(e, out, delayflg){
216     if (out) { // action
217         setImage(e, e.iasl_img[3], 0, 0);
218         move_abs(e, 380, 580);
219         return;
220     }
221
222     // transition
223     if (e.iasl_pending) { e.iasl_pending = 0; return; }
224     { freeInstance(e); return; }
225 }
226
227 function group_main(e) {
228     var sibling = new Array();
229     iasl_fork(e, obj_nnn18, trans_nnn18_nnn21, sibling);
230     iasl_fork(e, obj_nnn10, trans_nnn10_nnn11, sibling);
231     iasl_fork(e, obj_nnn10, trans_nnn10_nnn11, sibling);
232     iasl_fork(e, obj_nnn10, trans_nnn10_nnn11, sibling);
233     e.iasl_children.push(sibling);
234 }
235
236 function iasl_start(){
237     iasl_timeritvl = 100;
238     newboard(20,60,400,600,'#33CCCC');
239 }
240
241 // 得点の加算
242 function scoreAdd(){
243     var score = Number($('#score').text());
244     score = score+1;
245     //console.log(score);
246     $('#score').text(score);
247 }
248
249 function scoreStore(){
250     var score = $('#score').text();
251     //console.log(score);
252     $.post("/cgi/store.cgi", "score="+score, function(result){
253         //console.log(result);
254     });
255 }

```

#### ソースコード 4: iasl\_ja.js

```

1 obj_nnn10 = new Array('nnn2.png', 'nnn0.png', 'nnn1.png', ''); // cat
2 obj_nnn18 = new Array('nnn3.png', 'nnn4.png', 'nnn5.png', 'nnn6.png', ''); // end
3 obj_nnn14 = new Array('nnn7.png', 'nnn8.png', 'nnn2.png', 'nnn9.png', ''); // kill
4
5 function trans_nnn10_nnn12(e, out, delayflg){
6     if (out) { // action
7         setImage(e, e.iasl_img[0], 45, 0);
8         return;
9     }
10
11     // transition

```

```

12     if (e.iasl_pending) { e.iasl_pending = 0; return; }
13     if (e.iasl_count+1 >= 10) { e.iasl_func = trans_nnn10_nnn11; return; }
14     { e.iasl_func = trans_nnn10_nnn12; return; }
15 }
16
17 function trans_nnn10_nnn15(e, out, delayflg){
18     if (out) { // action
19         setImage(e, e.iasl_img[0], 45, 0);
20         move_rel(e, 8, 8);
21         return;
22     }
23
24     // transition
25     if (e.iasl_pending) { e.iasl_pending = 0; return; }
26     if (e.iasl_bumped & (1<<iasl_bumped_d)) { e.iasl_func = trans_nnn10_nnn12; return; }
27     if (e.iasl_bumped & (1<<iasl_bumped)
28         && isBumpedObj(e, obj_nnn14) == true) { scoreAdd(); e.iasl_func = trans_nnn10_nnn11;
29         return; }
30     if (e.iasl_count+1 >= 10) { e.iasl_func = trans_nnn10_nnn13; return; }
31     { e.iasl_func = trans_nnn10_nnn15; return; }
32 }
33
34 function trans_nnn10_nnn13(e, out, delayflg){
35     if (out) { // action
36         setImage(e, e.iasl_img[1], 0, 0);
37         move_rel(e, 0, 8);
38         return;
39     }
40
41     // transition
42     if (e.iasl_pending) { e.iasl_pending = 0; return; }
43     if (gamble(50)) { e.iasl_func = trans_nnn10_nnn16; return; }
44     { e.iasl_func = trans_nnn10_nnn15; return; }
45 }
46
47 function trans_nnn10_nnn17(e, out, delayflg){
48     if (out) { // action
49         setImage(e, e.iasl_img[2], 0, 0);
50         move_abs(e, getRandX(), e.iasl_y);
51         return;
52     }
53
54     // transition
55     if (e.iasl_pending) { e.iasl_pending = 0; return; }
56     if (gamble(10)) { e.iasl_func = trans_nnn10_nnn16; return; }
57     { e.iasl_func = trans_nnn10_nnn17; return; }
58 }
59
60 function trans_nnn10_nnn16(e, out, delayflg){
61     if (out) { // action
62         setImage(e, e.iasl_img[0], -45, 0);
63         move_rel(e, -8, 8);
64         return;
65     }
66
67     // transition
68     if (e.iasl_pending) { e.iasl_pending = 0; return; }
69     if (e.iasl_bumped & (1<<iasl_bumped_d)) { e.iasl_func = trans_nnn10_nnn12; return; }
70     if (e.iasl_bumped & (1<<iasl_bumped)
71         && isBumpedObj(e, obj_nnn14) == true) { scoreAdd(); e.iasl_func = trans_nnn10_nnn11;
72         return; }
73     if (e.iasl_count+1 >= 10) { e.iasl_func = trans_nnn10_nnn13; return; }
74     { e.iasl_func = trans_nnn10_nnn16; return; }
75 }
76
77 function trans_nnn10_nnn11(e, out, delayflg){
78     if (out) { // action
79         setImage(e, e.iasl_img[2], 0, 0);

```

```

78     move_abs(e, 200, 0);
79     return;
80 }
81
82 // transition
83 if (e.iasl_pending) { e.iasl_pending = 0; return; }
84 { e.iasl_func = trans_nnn10_nnn17; return; }
85 }
86
87 function trans_nnn18_nnn19(e, out, delayflg){
88     if (out) { // action
89         setImage(e, e.iasl_img[0], 0, 0);
90         return;
91     }
92
93     // transition
94     if (e.iasl_pending) { e.iasl_pending = 0; return; }
95     { e.iasl_func = trans_nnn18_nnn19; return; }
96 }
97
98 function trans_nnn18_nnn22(e, out, delayflg){
99     if (out) { // action
100         setImage(e, e.iasl_img[1], 0, 0);
101         follow_mouse(e);
102         return;
103     }
104
105     // transition
106     if (e.iasl_pending) { e.iasl_pending = 0; return; }
107     if (e.iasl_bumped & (1<<iasl_bumped)
108         && isBumpedObj(e, obj_nnn10) == true) { e.iasl_func = trans_nnn18_nnn20; return; }
109     if (e.iasl_bumped & (1<<iasl_bumped)
110         && isBumpedObj(e, obj_nnn14) == true) { /*scoreAdd()*/; e.iasl_func =
111         trans_nnn18_nnn20; return; }
112     if (e.iasl_count+1 >= 7) { e.iasl_func = trans_nnn18_nnn21; return; }
113     { e.iasl_func = trans_nnn18_nnn22; return; }
114 }
115
116 function trans_nnn18_nnn21(e, out, delayflg){
117     if (out) { // action
118         setImage(e, e.iasl_img[2], 0, 0);
119         follow_mouse(e);
120         return;
121     }
122
123     // transition
124     if (e.iasl_pending) { e.iasl_pending = 0; return; }
125     if (e.iasl_clicked == iasl_clicked_1) { e.iasl_func = trans_nnn18_nnn23; return; }
126     if (e.iasl_bumped & (1<<iasl_bumped)
127         && isBumpedObj(e, obj_nnn10) == true) { e.iasl_func = trans_nnn18_nnn20; return; }
128     if (e.iasl_bumped & (1<<iasl_bumped)
129         && isBumpedObj(e, obj_nnn14) == true) { e.iasl_func = trans_nnn18_nnn20; return; }
130     { e.iasl_func = trans_nnn18_nnn21; return; }
131 }
132
133 function trans_nnn18_nnn23(e, out, delayflg){
134     if (out) { // action
135         setImage(e, e.iasl_img[2], 0, 0);
136         follow_mouse(e);
137         if (delayflg) { e.iasl_pending = 1; return; }
138         { var sibling = new Array();
139           iasl_fork(e, obj_nnn14, trans_nnn14_nnn24, sibling);
140           e.iasl_children.push(sibling); }
141         return;
142     }
143
144     // transition
145     if (e.iasl_pending) { e.iasl_pending = 0; return; }

```

```

145     if (e.iasl_bumped & (1<<iasl_bumped)
146         && isBumpedObj(e, obj_nnn10) == true) { e.iasl_func = trans_nnn18_nnn20; return; }
147     { e.iasl_func = trans_nnn18_nnn22; return; }
148 }
149
150 function trans_nnn18_nnn20(e, out, delayflg){
151     if (out) { // action
152         setImage(e, e.iasl_img[3], 0, 0);
153         rove(e, 1, 0);
154         return;
155     }
156
157     // transition
158     if (e.iasl_pending) { e.iasl_pending = 0; return; }
159     if (e.iasl_count+1 >= 9) { scoreStore(); e.iasl_func = trans_nnn18_nnn19; return; }
160     { e.iasl_func = trans_nnn18_nnn20; return; }
161 }
162
163 function trans_nnn14_nnn24(e, out, delayflg){
164     if (out) { // action
165         setImage(e, e.iasl_img[0], 0, 0);
166         move_rel(e, 12, -12);
167         return;
168     }
169
170     // transition
171     if (e.iasl_pending) { e.iasl_pending = 0; return; }
172     { e.iasl_func = trans_nnn14_nnn25; return; }
173 }
174
175 function trans_nnn14_nnn27(e, out, delayflg){
176     if (out) { // action
177         setImage(e, e.iasl_img[1], 15, 0);
178         return;
179     }
180
181     // transition
182     if (e.iasl_pending) { e.iasl_pending = 0; return; }
183     if (e.iasl_count+1 >= 8) { e.iasl_func = trans_nnn14_nnn26; return; }
184     { e.iasl_func = trans_nnn14_nnn27; return; }
185 }
186
187 function trans_nnn14_nnn25(e, out, delayflg){
188     if (out) { // action
189         setImage(e, e.iasl_img[0], 0, 0);
190         move_rel(e, 0, -5);
191         return;
192     }
193
194     // transition
195     if (e.iasl_pending) { e.iasl_pending = 0; return; }
196     if (e.iasl_bumped & (1<<iasl_bumped)
197         && isBumpedObj(e, obj_nnn10) == true) { e.iasl_func = trans_nnn14_nnn27; return; }
198     if (e.iasl_bumped & (1<<iasl_bumped)
199         && isBumpedObj(e, obj_nnn18) == true) { freeInstance(e); return; }
200     if (e.iasl_bumped & (1<<iasl_bumped_u)) { freeInstance(e); return; }
201     { e.iasl_func = trans_nnn14_nnn25; return; }
202 }
203
204 function trans_nnn14_nnn28(e, out, delayflg){
205     if (out) { // action
206         setImage(e, e.iasl_img[2], 0, 0);
207         return;
208     }
209
210     // transition
211     if (e.iasl_pending) { e.iasl_pending = 0; return; }
212     { freeInstance(e); return; }

```

```

213 }
214
215 function trans_nnn14_nnn26(e, out, delayflg){
216     if (out) { // action
217         setImage(e, e.iasl_img[3], 0, 0);
218         move_abs(e, 380, 580);
219         return;
220     }
221
222     // transition
223     if (e.iasl_pending) { e.iasl_pending = 0; return; }
224     { freeInstance(e); return; }
225 }
226
227 function group_main(e) {
228     var sibling = new Array();
229     iasl_fork(e, obj_nnn18, trans_nnn18_nnn21, sibling);
230     iasl_fork(e, obj_nnn10, trans_nnn10_nnn11, sibling);
231     iasl_fork(e, obj_nnn10, trans_nnn10_nnn11, sibling);
232     iasl_fork(e, obj_nnn10, trans_nnn10_nnn11, sibling);
233     e.iasl_children.push(sibling);
234 }
235
236 function iasl_start(){
237     iasl_timeritvl = 100;
238     newboard(20,60,400,600,'#33CCCC');
239 }
240
241 // 得点の加算
242 function scoreAdd(){
243     var score = Number($('#score').text());
244     score = score+1;
245     //console.log(score);
246     $('#score').text(score);
247 }
248
249 function scoreStore(){
250     var score = $('#score').text();
251     //console.log(score);
252     $.post("/cgi/store.cgi", "score="+score, function(result){
253         //console.log(result);
254     });
255 }

```