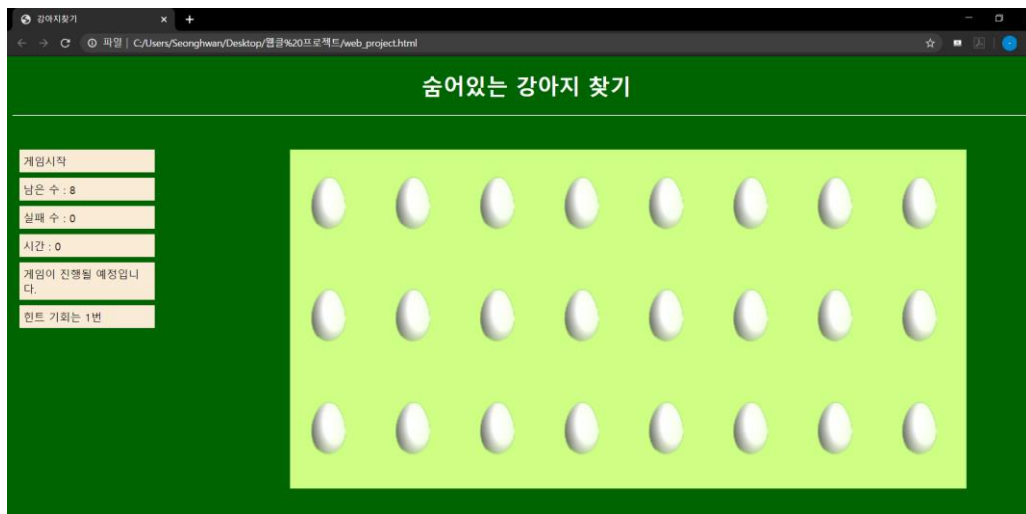


# 프로젝트 “숨어있는 강아지 찾기”

20171615 노 성환

- 게임 기획의도

## <초기화면>



## <게임을 시작한 뒤 알 속에 숨은 강아지를 보여준다.>



<숨어있는 강아지를 모두 찾으면 다음과 같이 끝난다.>



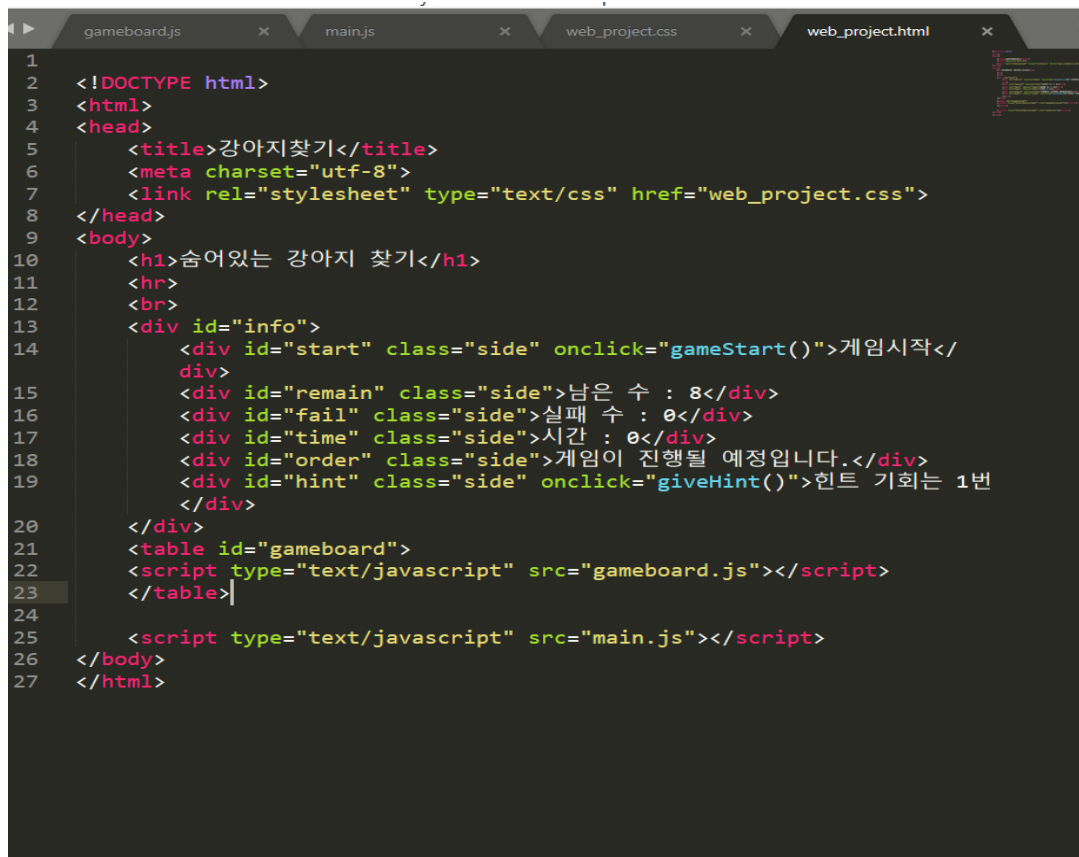
<숨어 있는 강아지를 제한 시간 내 찾지 못하면 다음과 같이 끝난다.>



## ● 업그레이드 상황

1. 게임의 시작과 종료 알림 문구가 띄워진다.
2. 사용자가 오답을 누를 시 제한 시간이 1초 감소된다.
3. 정답과 오답, 게임 성공 시 상황의 각각의 효과음이 나온다.
4. 정답을 모를 때 사용할 수 있는 힌트 기능이 추가되었다.

## ● 게임 설명



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>강아지찾기</title>
5    <meta charset="utf-8">
6    <link rel="stylesheet" type="text/css" href="web_project.css">
7  </head>
8  <body>
9    <h1>숨어있는 강아지 찾기</h1>
10   <hr>
11   <br>
12   <div id="info">
13     <div id="start" class="side" onclick="gameStart()">게임시작</div>
14     <div id="remain" class="side">남은 수 : 8</div>
15     <div id="fail" class="side">실패 수 : 0</div>
16     <div id="time" class="side">시간 : 0</div>
17     <div id="order" class="side">게임이 진행될 예정입니다.</div>
18     <div id="hint" class="side" onclick="giveHint()">힌트 기회는 1번</div>
19   </div>
20   <table id="gameboard">
21   <script type="text/javascript" src="gameboard.js"></script>
22   </table>
23
24   <script type="text/javascript" src="main.js"></script>
25 </body>
26 </html>
```

위의 개요에서 나온 게임 화면들을 보면 왼쪽에 나온 정보 블록을 div를 이용해 배치하고, 그 안에 div로 시작, 남은 횟수, 시간 등 여러 정보들을 배치하였다. 그리고 초기화면에 알맞은 정보들을 써 주었다. 그 외에 게임시작을 누르면 gameStart 함수를, hint를 누르면 giveHint 함수가 실행되는 이벤트 리스너를 만들어주었다.

<gameboard.js>

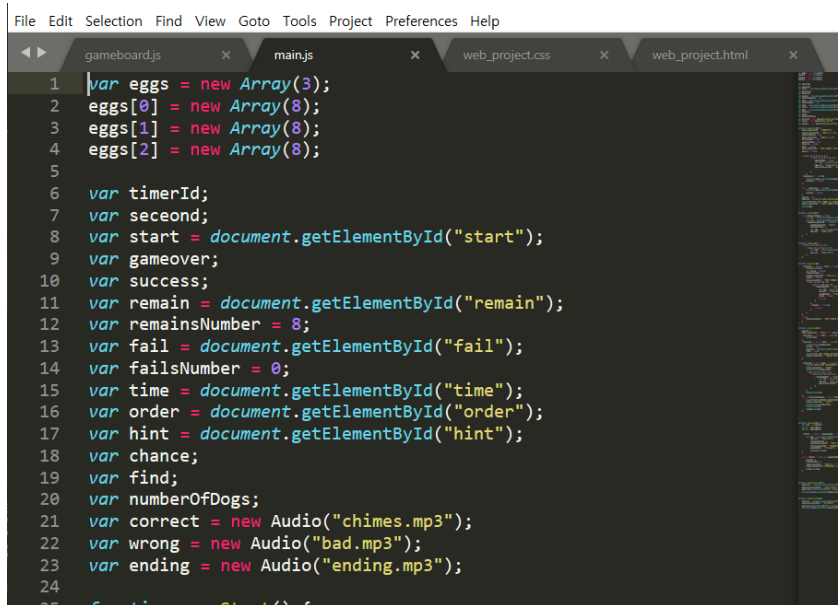
```
gameboard.js  x  main.js  x  web_project.css  x  web_project.html  x
1  for(var i=0; i<3; i++) {
2      document.write("<tr>");
3      for(var j=0; j<8; j++) {
4          document.write("<td><img id='egg' + i + j +
5              |' src='img1.gif' onclick='checkEgg(event)'></td>");
6      }
7      document.write("</tr>");
8  }
```

그리고 html 부분 table 태그 안에 gameboard라는 자바스크립트를 이용해 document.write 안에 3행 8열 공간을 만들어 그 안에 달걀 사진을 넣고 각각에 아이디를 넣어 주었다. 그러면 위의 개요 사진과 같이 달걀들이 나열되어 있는 것을 볼 수 있다. 달걀 이미지를 클릭하면 checkEgg 함수를 실행하는 이벤트 리스너를 만들어 주었다.

```
File Edit Selection Find View Goto Tools Project Preferences Help
gameboard.js  x  main.js  x  web_project.css  x  web_project.html  x
1  {
2  body{
3      background-color: darkgreen;
4  }
5
6  #info {
7      width: 200px; height: 300px;
8      float: left;
9      margin-left: 10px;
10     margin-top: 20px;
11 }
12
13 #gameboard {
14     width: 1000px; height: 500px;
15     background: #CEFF84;
16     float: left;
17     margin-left: 200px;
18     margin-top: 20px;
19 }
20
21
22 .side {
23     background: antiquewhite;
24     margin-bottom: 4%;
25     padding: 3%;
26 }
27
28
29 td {
30     padding: 3%;
31 }
32
33 h1{
34     text-align: center;
35     color: white;
36 }
```

위 css는 각 id, class, 태그에 해당되는 요소들을 개요 사진처럼 알맞게 꾸며주었다.

## <변수 설정>

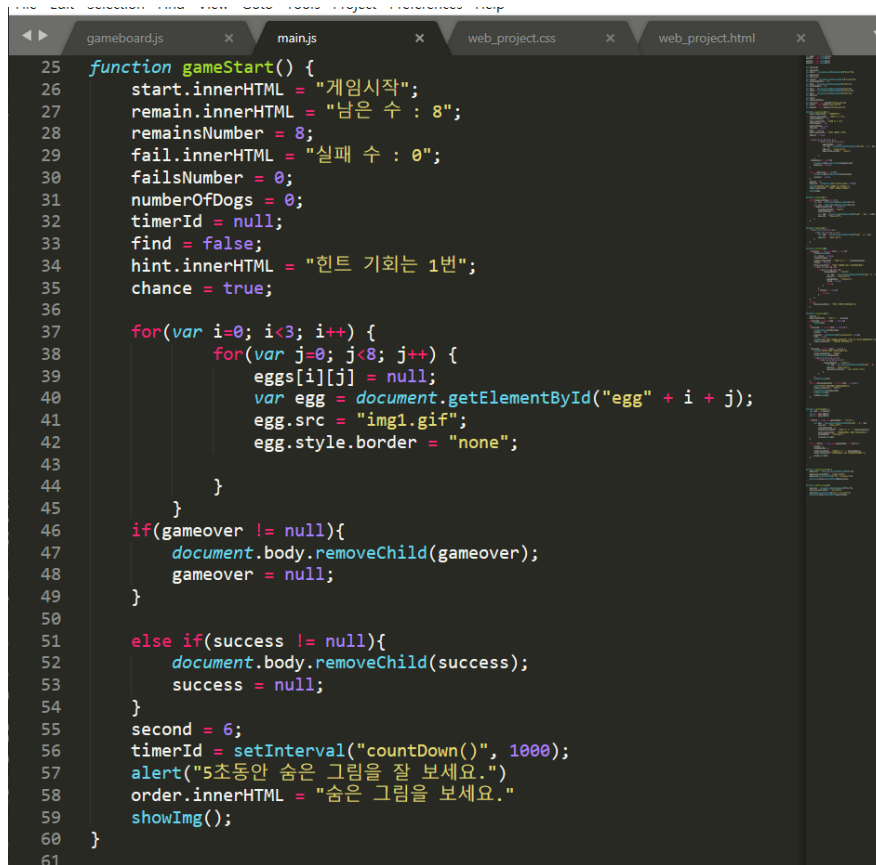


```
File Edit Selection Find View Goto Tools Project Preferences Help
gameboard.js x main.js x web_project.css x web_project.html x
1 var eggs = new Array(3);
2 eggs[0] = new Array(8);
3 eggs[1] = new Array(8);
4 eggs[2] = new Array(8);
5
6 var timerId;
7 var second;
8 var start = document.getElementById("start");
9 var gameOver;
10 var success;
11 var remain = document.getElementById("remain");
12 var remainsNumber = 8;
13 var fail = document.getElementById("fail");
14 var failsNumber = 0;
15 var time = document.getElementById("time");
16 var order = document.getElementById("order");
17 var hint = document.getElementById("hint");
18 var chance;
19 var find;
20 var numberOfDogs;
21 var correct = new Audio("chimes.mp3");
22 var wrong = new Audio("bad.mp3");
23 var ending = new Audio("ending.mp3");
24
25 function gameStart() {
```

eggs라는 3행 8열 배열을 선언해 주었다. 이 배열은 아까 위에서 만든 달걀들의 아이디를 가져와 쓰게 될 용도이다.

- timerId: 게임 중 타이머를 설정하기 위한 변수
- second: 타이머에 초를 설정하는 변수
- start, remain, fail, time, order, hint: 각 항목의 아이디 속성값을 가진 객체를 저장하는 변수 (info 부분)
- remainNumber, failsNumber: 남은 횟수, 실패 횟수를 나타내는 변수
- gameOver, success: 게임의 엔딩에 따라 사용될 변수
- chance, find: 찬스와 찾고 있는 상황임을 나타내는 bool형 변수
- numberOfDogs: 강아지의 수를 나타내는 변수
- correct, wrong, ending: 각 상황별로 출력될 오디오 변수

### <초기 변수 초기화>



```
25 function gameStart() {
26     start.innerHTML = "게임시작";
27     remain.innerHTML = "남은 수 : 8";
28     remainsNumber = 8;
29     fail.innerHTML = "실패 수 : 0";
30     failsNumber = 0;
31     numberOfDogs = 0;
32     timerId = null;
33     find = false;
34     hint.innerHTML = "힌트 기회는 1번";
35     chance = true;
36
37     for(var i=0; i<3; i++) {
38         for(var j=0; j<8; j++) {
39             eggs[i][j] = null;
40             var egg = document.getElementById("egg" + i + j);
41             egg.src = "img1.gif";
42             egg.style.border = "none";
43         }
44     }
45
46     if(gameover != null){
47         document.body.removeChild(gameover);
48         gameOver = null;
49     }
50
51     else if(success != null){
52         document.body.removeChild(success);
53         success = null;
54     }
55     second = 6;
56     timerId = setInterval("countDown()", 1000);
57     alert("5초동안 숨은 그림을 잘 보세요.")
58     order.innerHTML = "숨은 그림을 보세요."
59     showImg();
60 }
61
```

게임 시작을 누르게 되면 위의 gameStart 함수가 실행된다. Info 칸에 들어갈 정보들을 innerHTML로 넣어주는 등 알맞게 초기화 한다.

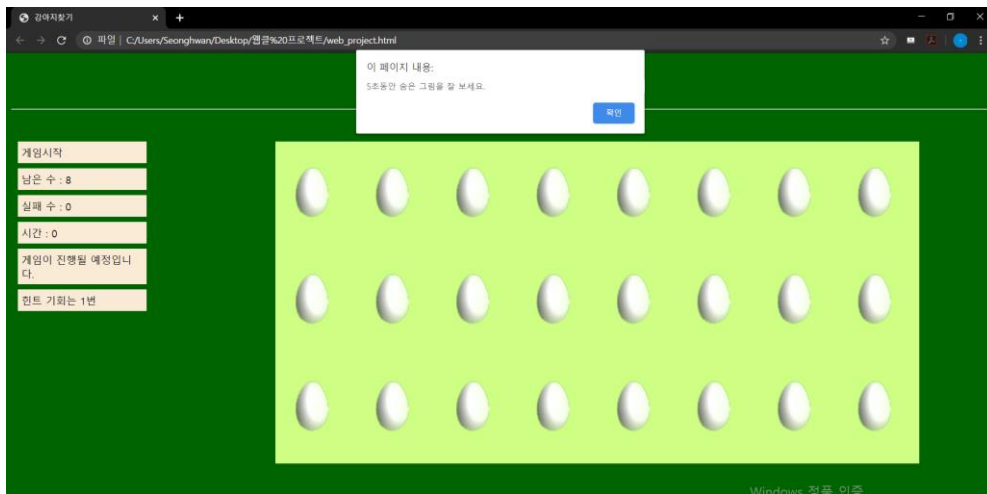
3행 8열 달걀 배열에서 반복문을 통해 하나씩 접근해 해당 아이디를 받아 달걀 사진으로 초기화를 시켜준다.

게임을 성공했거나 실패했을 때 사용될 gameOver, success을 body 태그에서 떼어내어 초기화를 시켜준다.

countDown을 통해 사용자가 5초동안 showImg를 통한 그림들을 볼 수 있도록 한다. 이 때 보여주기 전에 알림을 보여주고, 정보들을 알맞게 바꿔준다.

그 결과 아래 사진과 같이 나타난다.

### <숨은 사진을 보여주기 전 화면>



### <showImg 함수>

```
61
62 function showImg() {
63     while(numberOfDogs != 8) {
64         var row = Math.floor(Math.random()*3);
65         var col = Math.floor(Math.random()*8);
66         if(eggs[row][col] == null) {
67             eggs[row][col] = "dog";
68             numberOfDogs++;
69             var egg = document.getElementById("egg" + row + col);
70             egg.src = "img2.gif";
71         }
72     }
73 }
```

행과 열에서 정해진 범위만큼 랜덤으로 수를 뽑아 해당 칸이 비어 있으면 dog으로 설정해 주고 이를 실행할 때 마다 numberOfDogs가 8개가 될 때까지 증가시킨다. 그리고 해당 칸의 아이디를 받아 개요 사진에서 숨은 강아지를 보여줬듯이 이미지를 달걀에서 강아지 사진으로 바꿔준다.

### <countDown 함수>

```
File Edit Selection Find View Goto Tools Project Preferences Help
gameboard.js x main.js x web_project.css x web_project.html x
112 function countDown() {
113     second--;
114     time.innerHTML = "시간 : " + second;
115     if(second == 0 && find == false){
116         hideImg();
117     }
118     if(second == -1 && find == false) {
119         clearInterval(timerId);
120         second = 21;
121         timerId = setInterval("countDown()", 1000);
122         find = true;
123         alert("20초 동안 정답을 찾으세요. 오답 시 1초씩 줄어듭니다.");
124         order.innerHTML = "정답을 찾으세요.";
125     }
126
127     if(second <= 0 && find == true) {
128         alert("시간이 종료 되었습니다.");
129         order.innerHTML = "실패";
130         clearInterval(timerId);
131         for(var i=0; i<3; i++) {
132             for(var j=0; j<8; j++) {
133                 if(eggs[i][j] == "dog") {
134                     var egg = document.getElementById("egg" + i + j);
135                     egg.src = "img2.gif";
136                     egg.style.border = "1px solid red";
137                 }
138             }
139         }
140         makeGameOver();
141     }
142     else if(remainsNumber == 0 && find == true) {
143         alert("모든 강아지를 찾았습니다!");
144         order.innerHTML = "성공";
145         clearInterval(timerId);
146         makeSuccess();
147         ending.play();
148     }
149 }
```

숨은 강아지를 보여주는 시간 동안 초를 계속 보여주고, 시간이 다 되었으면 다시 그림을 숨기는 `hideImg` 함수를 실행한다.

### <hideImg 함수>

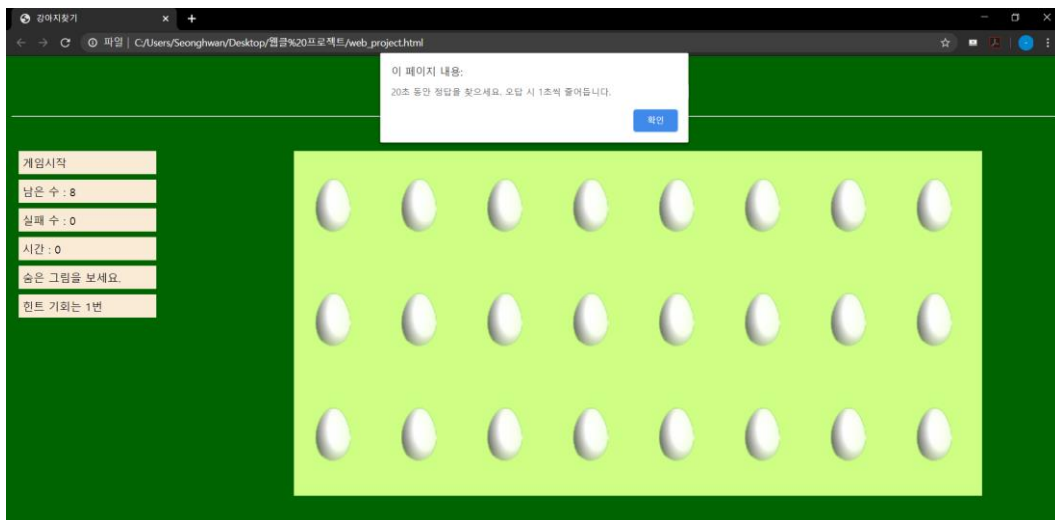
```
74
75 function hideImg() {
76     for(var i=0; i<3; i++) {
77         for(var j=0; j<8; j++) {
78             var egg = document.getElementById("egg" + i + j);
79             egg.src = "img1.gif";
80         }
81     }
82 }
83
```

다시 달걀 전체를 돌면서 모든 사진을 달걀 사진으로 바꾼다.



다시 원래대로 사진을 달걀로 모두 바꾼 후 타이머를 해제하고, 다시 초를 20초로 맞추고 find를 true로 바꿔 숨은 강아지를 찾기 시작한다. 이에 맞게 아래 사진과 같이 알림을 보여주고, 정보들을 알맞게 바꿔준다.

#### <찾기 시작을 알리는 화면>



그리고 이제 달걀을 클릭하면서 게임을 시작한다. 아까 위에서 달걀을 클릭하면 checkEgg 함수를 실행하는데 이는 아래와 같다.

#### <checkEgg 함수>

```
153 function checkEgg(e) {
154     var egg = e.target;
155     var i = egg.id[3];
156     var j = egg.id[4];
157
158     if(find == true && eggs[i][j] == "dog") {
159         var egg = document.getElementById("egg" + i + j);
160         egg.src = "img2.gif";
161         remainsNumber--;
162         remain.innerHTML = "남은 수 : " + remainsNumber;
163         order.innerHTML = "정답입니다. 계속 찾아보세요.";
164         eggs[i][j] = "checked";
165         correct.play();
166     }
167
168     else if(find == true && eggs[i][j] != "dog") {
169         second--;
170         failsNumber++;
171         fail.innerHTML = "실패 수 : " + failsNumber;
172         order.innerHTML = "오답입니다. 1초 줄어듭니다.";
173         wrong.play();
174     }
175 }
```

egg라는 변수로 event이름으로 이벤트 객체를 전달받는다. 그리고 egg 아이디의 3번째, 4번째를 i, j에 저장하고, 해당 인덱스가 dog인지 확인하는데 이는 3행 8열 안에 있는 egg의 위치이다. 해당 자리가 dog이면 달걀 사진에서 강아지 사진으로 바꾸고, 남은 수가 줄어들면서, 정보들이 알맞은 문구를 출력하고, dog에서 checked로 바뀐다. 동시에 정답 효과음도 출력한다. 해당 화면은 아래와 같다.

#### <정답 시 화면>



해당 달걀이 dog이 아니었을 경우에는 1초 감소되고, 실패횟수는 증가시키면서 이에 알맞은 문구를 정보에 출력시킨다. 동시에 오답 효과음도 출력한다. 해당 화면은 아래와 같다.

#### <오답 시 화면>



### <giveHint 함수>

```
82 }
83
84 function giveHint(){
85     if(chance == true && find == true){
86         correct.play();
87         var check = false;
88         remainsNumber--;
89         remain.innerHTML = "남은 수 : " + remainsNumber;
90         chance = false;
91         hint.innerHTML = "힌트 기회를 모두 사용했습니다."
92         for(var i=0;i<3;i++){
93             for(var j=0;j<8;j++){
94                 if(eggs[i][j] == "dog"){
95                     var egg = document.getElementById("egg"+ i + j);
96                     egg.src = "img2.gif";
97                     eggs[i][j] = "checked";
98                     check = true;
99                     break;
100                 }
101             }
102             if(check == true){
103                 break;
104             }
105         }
106     }
107     else{
108         hint.innerHTML = "힌트 기회가 없습니다.";
109     }
110 }
111
```

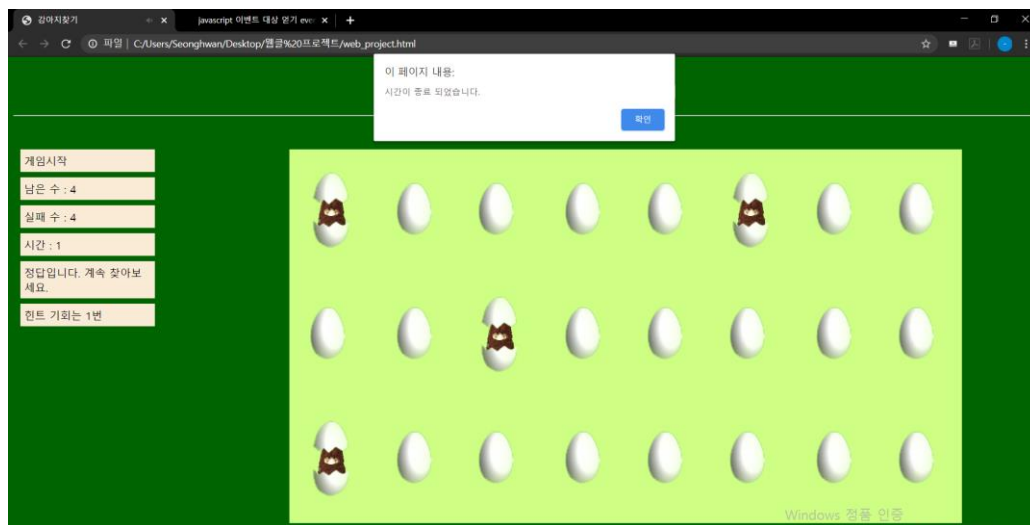
힌트를 클릭하면 1번의 힌트가 제공되는데 이를 실행하는 함수는 위와 같다. 힌트를 사용하면 chance를 소비했으므로 false로 바꿔주고, 달걀들을 돌면서 해당 달걀이 dog이면 그 달걀의 이미지를 강아지로 바꿔주고, 달걀을 클릭해서 찾았을 때와 마찬가지로 checked로 바꿔주고, 남은 횟수를 줄여주고, 효과음을 출력한다. 그리고 정보 문구에도 알맞게 출력시킨다. 해당 화면은 아래와 같다.

### <힌트 사용 시 화면>



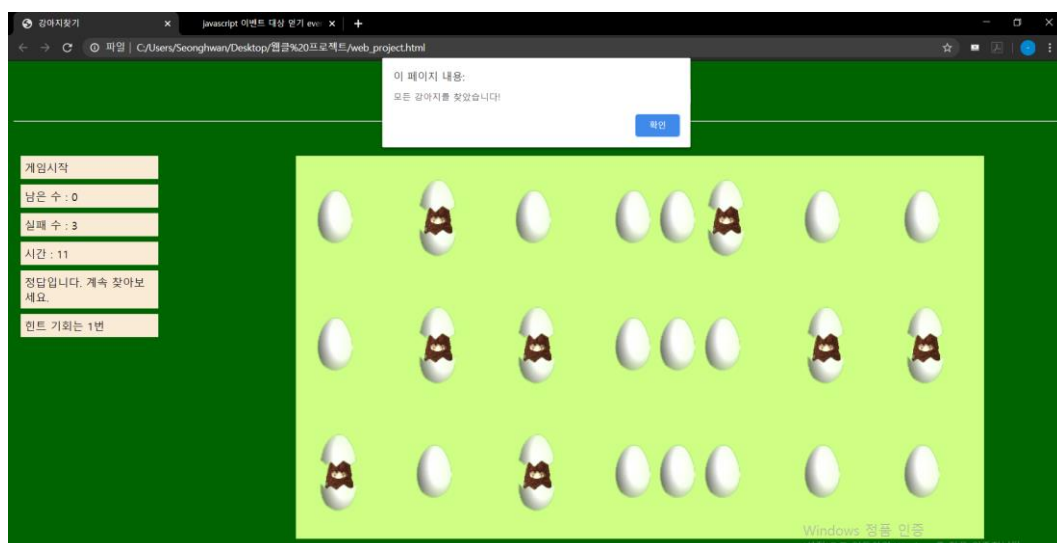
countDown 함수에서 시간이 다 소모되었으면 이를 알려주는 알림과 정보에 알맞은 문구를 출력하고, 타이머를 해제한다. 그리고 달걀들을 돌면서 아직 dog으로 남아있는 달걀들의 이미지를 강아지로 바꿔주고, 해당 달걀의 테두리를 표시해 찾지 못한 부분들을 알려준다. 그리고 makeGameOver 함수를 실행해 gameOver 문구를 출력한다. (gameover을 출력시키는 화면은 개요 사진과 같다.)

### <게임 실패 시 알림 화면>



countDown 함수에서 남은 수가 0이 되면, 이를 알려주는 알림과 정보에 알맞은 문구를 출력하고, 타이머를 해제한다. 그리고 makeSuccess 함수를 실행해 success 문구를 출력한다. (success를 출력시키는 화면은 개요 사진과 같다.)

### <게임 성공 시 알림 화면>



<makeGameOver, makeSuccess 함수>

```
178
179  function makeGameOver() {
180      gameOver = document.createElement("div");
181      gameOver.innerHTML = "GAME OVER";
182      gameOver.setAttribute("id", "gameover");
183      document.body.appendChild(gameover);
184  }
185
186  function makeSuccess(){
187      success = document.createElement("div");
188      success.innerHTML = "SUCCESS";
189      success.setAttribute("id", "success");
190      document.body.appendChild(success);
191  }
192
```

gameover 변수에 div 태그의 객체를 동적으로 생성해 아이디를 가지게 하고, "Game Over"라는 문구를 가지고 body 태그에 삽입해 화면에 띄운다.

success도 "SUCCESS"라는 문구를 가지고 같은 기능을 한다.

```
36 }
37 #gameover{
38     position: absolute;
39     font-weight: 600;
40     font-size: 60px;
41     color: red;
42     margin-left: 700px;
43     margin-top: 200px;
44 }
45 #success{
46     position: absolute;
47     font-weight: 600;
48     font-size: 60px;
49     color: blue;
50     margin-left: 800px;
51     margin-top: 200px;
52 }
```

위 css는 출력될 "Game Over", "SUCCESS"를 해당 id에 맞는 요소들을 개요 화면과 같이 꾸며주었다.