

풍선 터트리기 게임



Balloon Pop Game

목 차



기획



게임소개



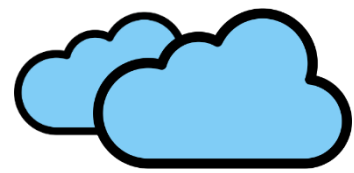
디자인



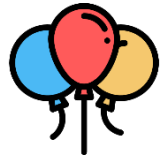
코딩



에필로그



기획



기획의도



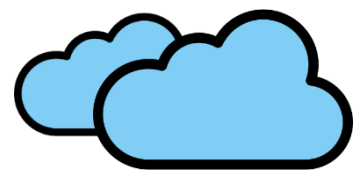
어린 아이도 즐길 수 있는 쉽고 재미있는 게임!



팀을 구성 함으로써 개인 작업 할때는 느낄 수 없는 협동심



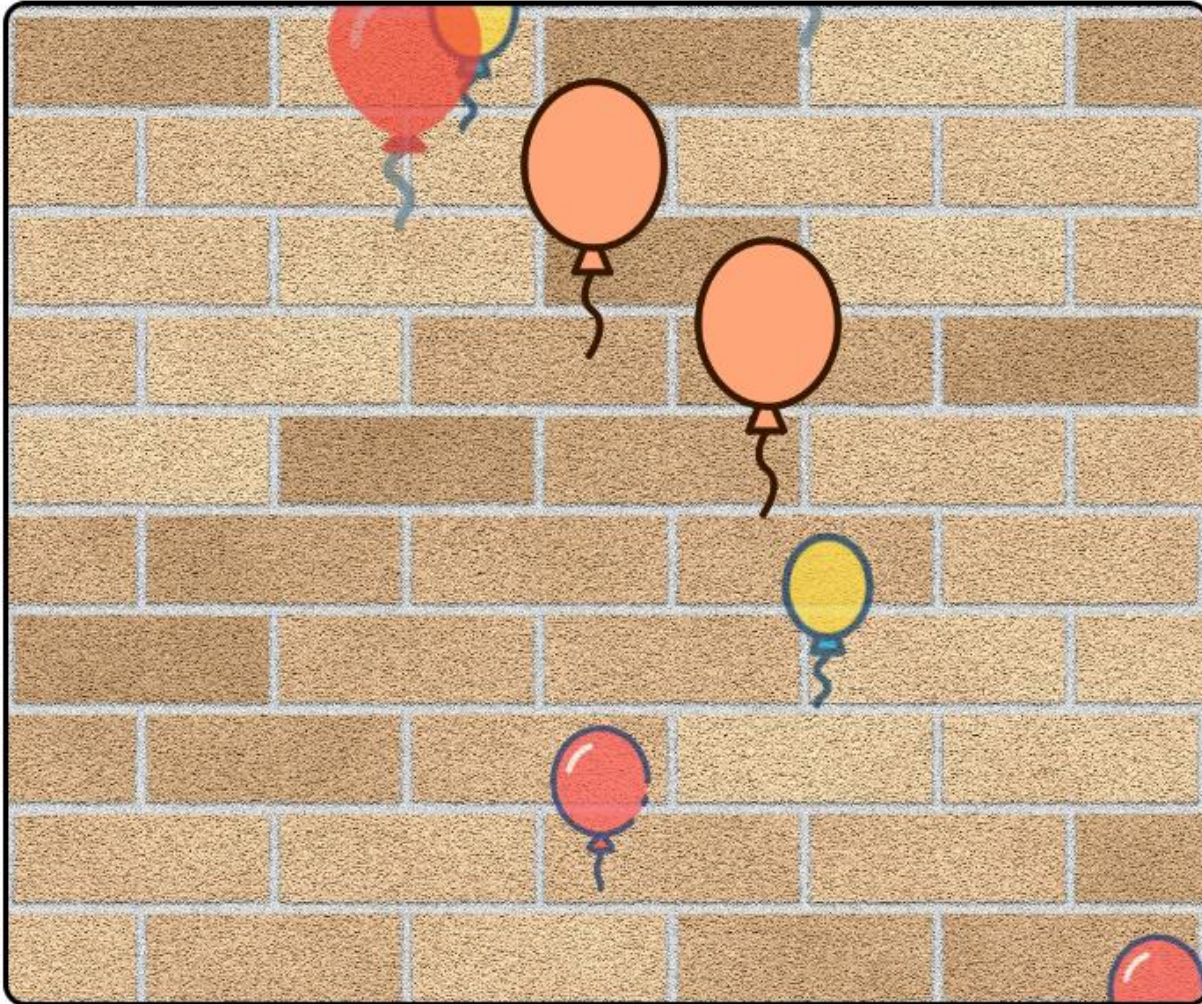
자바스크립트 능력 향상 ↑



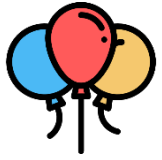
게임소개



게임소개



- ☁ 30초동안 올라오는 여러 개의 풍선 중
색깔이 뚜렷한 풍선만 터트릴 수 있어요!
- ☁ 풍선 한개당 점수가 100점씩 올라 갑니다



게임소개

풍선터트리기

EASY

NORMAL

HARD

START 버튼을 누르면 게임이 시작됩니다.



난이도는 총 세 단계로 구성 됐습니다.



NORMAL부터 속도가 빨라지며 HARD는 꽤 어려워지니 시작전에

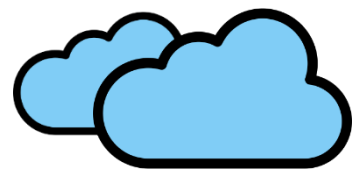
심호흡 한번 해주세요 😊



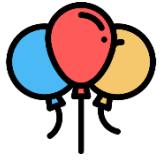
게임소개



30초가 되면 귀여운 곰돌이 인형이
등장하면서
게임이 종료 됩니다.



디자인

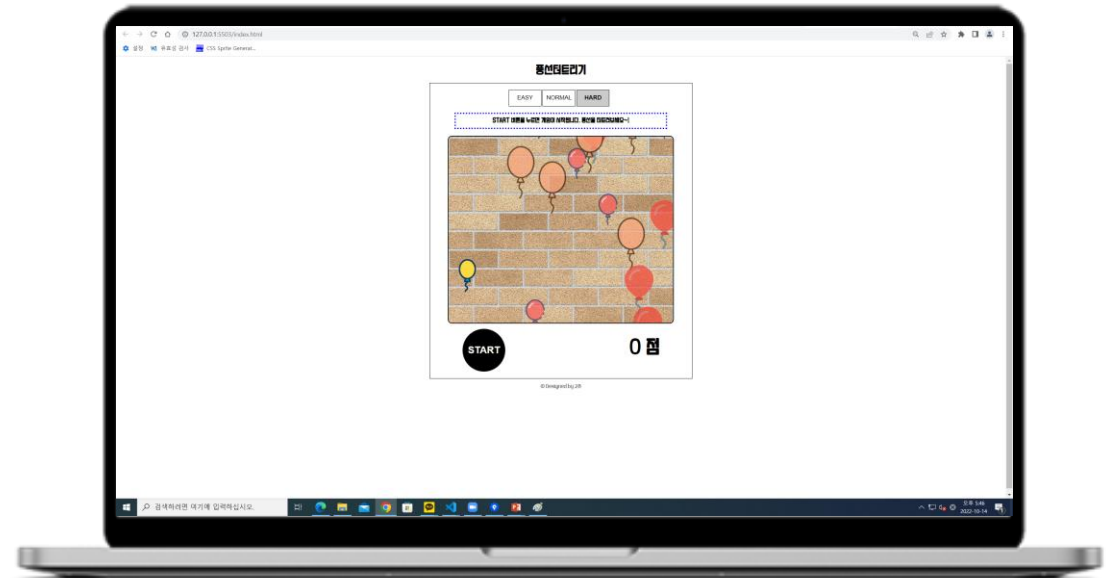


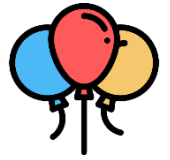
디자인

풍선터트리기 게임

제작기간 : 2022.09.27~2022.10.05

사용 프로그램 : HTML, CSS, JAVASCRIPT

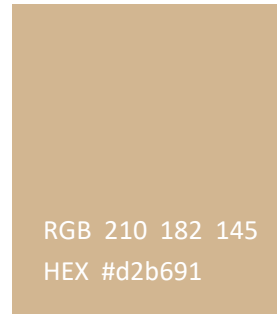




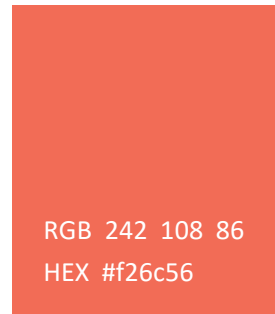
디자인

Colors

Background Colors



Balloon Colors



Icons





디자인

Typography

Aa

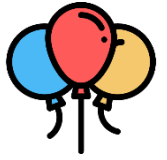
Aa

양진체

가나다라마바사아자차카타파하
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

돋움체

가나다라마바사아자차카타파하
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

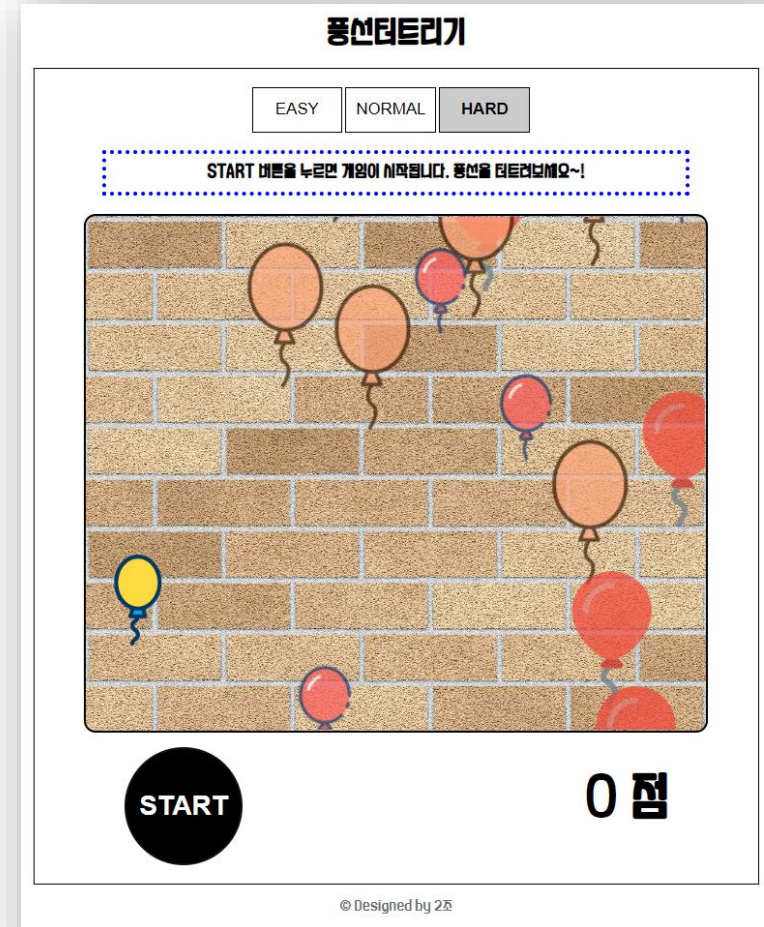


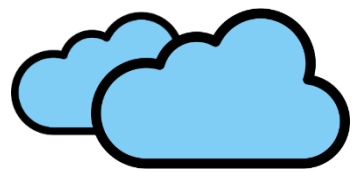
디자인

Description

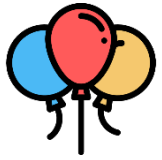
풍선 터트리기 게임은 동심으로 돌아가 게임을 하는 듯한
기분이 들도록 다채로운 색감과 귀여운 아이콘들로 구성
하였습니다.

상단에는 난이도별로 모드를 선택할 수 있는 버튼, 그 밑
에는 게임안내가 쓰여있고 중앙에는 벽돌배경의 메인화
면과 다채로운 색상의 풍선들이 나타나며 다트로 풍선을
터트리면 터지는 아이콘이 나타나 역동감을 더해줍니다.
하단에는 시작버튼과 점수가 표시되어 실시간으로 자신
의 점수를 확인할 수 있습니다.





코딩



코딩

HTML

```
<div class="loading">
  <p>로딩 중</p>
</div>

<header>
  <h1>풍선 터트리기</h1>
</header>

<section class="game">
  <div class="mode-box">
    <button class="modeBtn easy on" onclick="levelSetting('easy',this)">EASY</button>
    <button class="modeBtn normal" onclick="levelSetting('normal',this)">NORMAL</button>
    <button class="modeBtn hard" onclick="levelSetting('hard',this)">HARD</button>
  </div>

  <h2>START 버튼을 누르면 게임이 시작됩니다. 풍선을 터트려 보세요~!</h2>

  <div class="game-box">
    <div id="ballBox"></div>
    <div class="present"></div>
  </div>

  <button class="start" onclick="init()">START</button>
  <strong class="score"><em id="scoreNum">0</em> 점</strong>
</section>

<footer>
  <address>&copy; Designed by 2조</address>
</footer>
```

HEADER / SECTION / FOOTER 로
나누어 구성

게임기의 느낌이 나도록
본문영역에 game-box 영역을 넣어
게임기의 화면 형태로 작업



코딩

JAVA SCRIPT

```
const $loading = $('.loading');
// 로딩 중
$loading.children('p').fadeOut(5000);
$loading.delay(350).fadeOut(800, function(){
  $(this).remove();
});

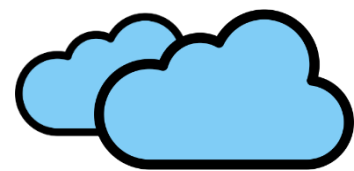
//게임 모드 설정
function levelSetting(type , obj){
  if(gameStatus){
    console.log('게임진행중');
    return false;
    // 게임이 끝나기 전에는 다른모드 작동안함
  }
  const $modeBtn = $('.modeBtn');

  // 버튼 클릭 활성화처리
  $modeBtn.removeClass('on'); //모든버튼 on제거
  obj.classList.add('on'); //클릭한 버튼 on처리

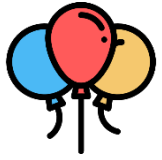
  if(type == 'easy'){
    ballType = 4; //4번째 풍선만
    interval = 300;
    ani = '10s'
  }
  if(type == 'normal'){
    ballType = 2; //2번째 풍선만
    maxLength = 20 //풍선갯수 늘리기
    interval = 200;
    ani = '7s'
  }
  if(type == 'hard'){
    ballType = 1; //1번째 풍선만
    maxLength = 30 //풍선갯수 늘리기
    interval = 100;
    ani = '3s'
  }
}

49 // 스타트버튼에 대한 실행함수
50 function init(){
51   if(gameStatus){
52     console.log('게임진행중');
53     return false;
54   }
55   let present = $('.present');
56
57   gameStatus = true; //게임이 시작되면
58   present.removeClass("on"); //곰돌이 없어짐
59   rootScore = 0; //점수 0으로 리셋
60   $("#scoreNum").html(rootScore);
61
62   let wrap = $("#ballBox");
63   let ball = $("<span class='ball'>");
64   let ballIdx = 0;
65   timer = setInterval(function(){
66     if(ballIdx >= maxLength) {
67       clearInterval(timer);
68       setTimeout(function(){
69         present.addClass("on");
70         gameStatus = false;
71       },3000);
72       return false;
73     }
74
75     // 풍선복제
76     let item = ball.clone(true);
77     let ran1 = Math.ceil(Math.random()*4);
78     let ran2 = Math.ceil(Math.random()*4);
79     let ran3 = Math.ceil(Math.random()*587);
80     item.addClass("size-"+ ran1);
81     item.addClass("ani-"+ ran2);
82     item.css({left:ran3,animationDuration:ani});
83     // item.click(clickEvt)
84     if(ran1 == ballType){
85
86       let ran2 = Math.ceil(Math.random()*4);
87       let ran3 = Math.ceil(Math.random()*587);
88       item.addClass("size-"+ ran1);
89       item.addClass("ani-"+ ran2);
90       item.css({left:ran3,animationDuration:ani});
91       // item.click(clickEvt)
92       if(ran1 == ballType){
93         ballIdx++;
94         item.css({zIndex:2});
95         item.addClass("on");
96         item.on("mousedown",clickEvt);
97       }
98       wrap.append(item);
99     },interval)
100   }
101
102   // 풍선을 클릭했을때 실행됨. 스코어처리
103   let rootScore = 0;
104   function clickEvt(evt,obj){
105     if(evt.target.chk) return false; //더블클릭 예외
106     evt.target.chk = true;
107     let scoreEl = $("#scoreNum")
108     rootScore+=100;
109     scoreEl.html(rootScore);
110     evt.target.classList.add('end');
111
112     let present = $('.present');
113     if(rootScore == maxLength*100) {
114       clearInterval(timer);
115       present.addClass("on");
116       gameStatus = false;
117     }
118   }
119 }
```

- 1.로드이벤트 : 로딩중 이미지 노출작업
- 2.게임 모드 (easy, normal, hard) 에 대한 세부설정작업 : 풍선의 종류&갯수&노출 및 ani타임.
- 3.스타트버튼에 대한 실행함수 init()에 게임진행중 풍선복제. 복제값에 size와 animate설정.
- 4.클릭에 대한 실행함수 clickEvt() : 풍선을 클릭성공했을때 점수가 100점씩 올라가도록 설정 / 최종점수가 풍선수*100과 같아지는 시점에 게임이 종료되고 곰돌이가 실행되도록 설정 => init() 함수 안에 풍선에 대한 clickEvt를 호출함.



에필로그



느낀 점

다빈

하고 싶은게 많았으나 구현을 하기에는
현실적으로 어렵기에

실행 못한게 아쉽지만 좋은 결과물이
만들어져서 기쁜 맘도 있다.

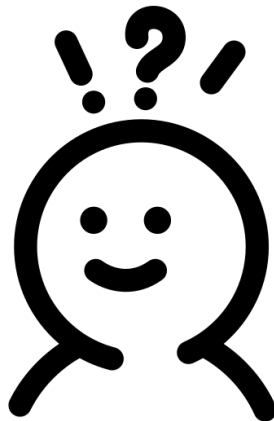
태은

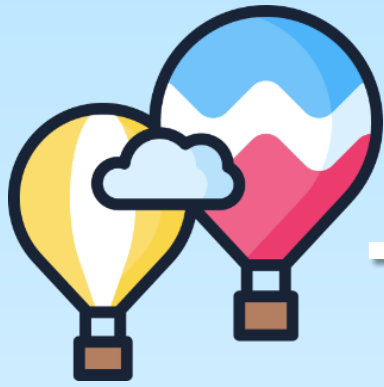
정해진부분에 대해 마무리를 해야한다는것
에 부담을 느꼈고,

구상한대로 구동되도록 코드를 짜는것이 아
주 어려웠다.

정연

팀프로젝트를 진행하면서 정확히 역할을 분
배하는 것에 어려움을 느꼈고 상상하던 것을
실제로 구현하는 것이 만만치 않았다.
그래도 혼자하는 것보다 팀원들과 같이 해서
더 수월하게 프로젝트를 마칠 수 있었던 것
같다.





Thank You

