

Let's War Game

Made by neko_hat

1

두 개의 자바 스크립트가 있습니다.

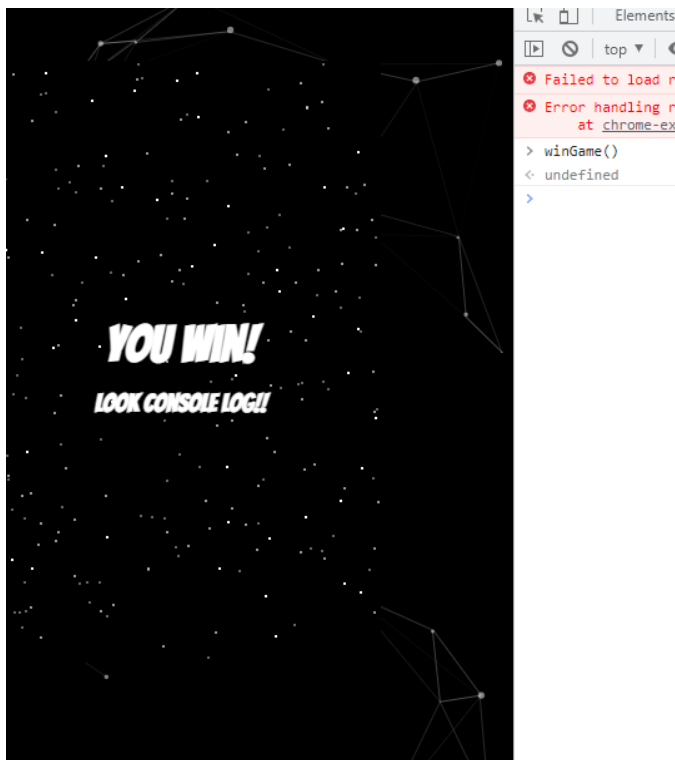
engine.js

game.js

다음 자바 스크립트를 난독화 해제 사이트를 통해 해제하면, 여전히 engine은 알아보기 힘들지만 game에는 눈에 띄는 친구가 있습니다.

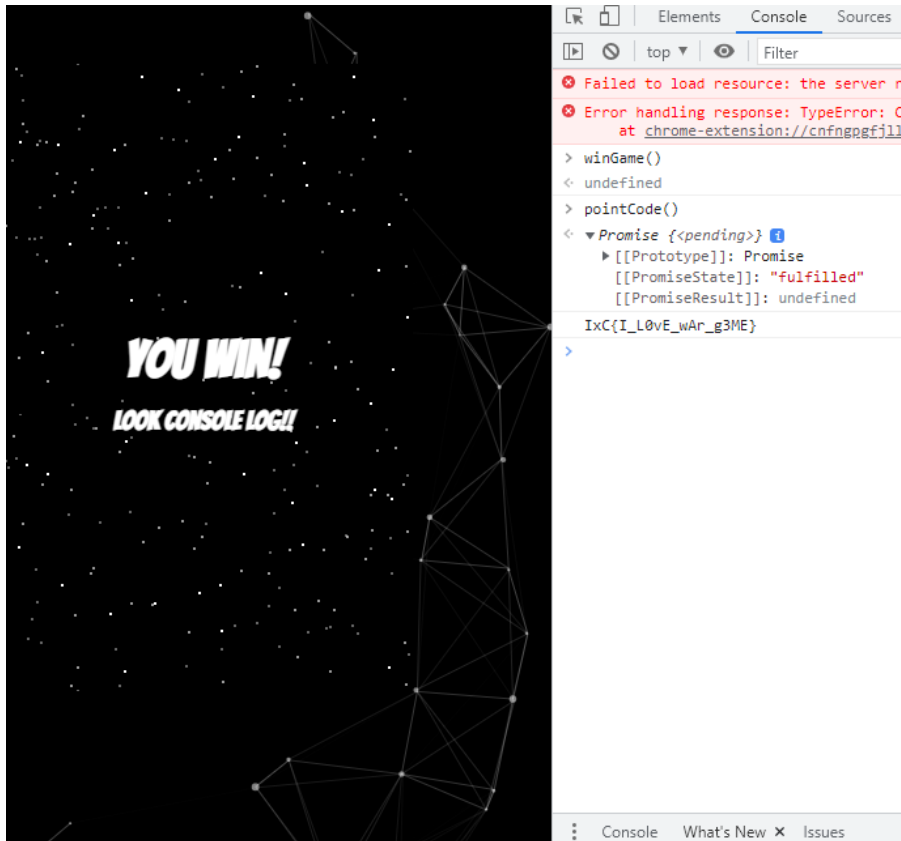
```
    }  
  },  
  playGame = function() {  
    var _0x2d9f18 = _0x231b7d,  
        _0x31f097 = new GameBoard();  
    _0x31f097[_0x2d9f18(0xa0)](new PlayerShip()), _0x31f097['add'](new Level(level1, winGame, pointCode)), Game[_0x2d9f18(0xa1)](new TitleScreen('You win!', atob(_0x44fffa(0xc8)), playGame));  
  },  
  winGame = function() {  
    var _0x44fffa = _0x231b7d;  
    Game['setBoard'](0x3, new TitleScreen('You win!', atob(_0x44fffa(0xc8)), playGame));  
  },  
  loseGame = function() {  
    Game['setBoard'](0x4, new TitleScreen('You lose!', atob(_0x44fffa(0xc9)), playGame));  
  }  
}
```

winGame이라는 함수입니다.



너무 쉽다는 마음에 바로 winGame 함수를 입력하면 실패합니다. 그렇게 쉽지는 않죠.

하지만 코드 상에 그 옆에 있던 pointCode 함수를 실행하면 너무 손쉽게 플래그를 얻을 수 있습니다. 다행히도 이런 쉬운 방법이 있습니다!



사실 원래 바라던 풀이는 playGame 함수 속의 Level이라는 함수를 확인하고 engine.js에서 찾아보는 거였는데요.

```
Level[_0x1542f0(0xea)][_0x1542f0(0x108)] = function(_0x5f233f) {
  var _0x5a079e = _0x1542f0,
      _0x5679fc = 0x0,
      _0x27c18c = [],
      _0x2264d5 = null;
  this[_0x1542f0(0x108)] += _0x5f233f * 0x3e8;
  while ((_0x2264d5 = this[_0x1542f0(0x108)]][_0x5679fc]) && _0x2264d5[0x0] < this[_0x1542f0(0x108)] + 0x7d0) {
    if (this[_0x1542f0(0x108)] > _0x2264d5[0x1]) _0x27c18c[_0x5a079e(0x11e)](_0x2264d5);
    else {
      if (_0x2264d5[0x0] < this[_0x1542f0(0x108)]) {
        var _0x4906cd = enemies[_0x2264d5[0x3]],
            _0x591a2b = _0x2264d5[0x4];
        this[_0x5a079e(0xf4)][_0x5a079e(0xf5)](new Enemy(_0x4906cd, _0x591a2b), _0x2264d5[0x0] += _0x2264d5[0x2]
        );
      }
      _0x5679fc++;
    }
  }
  for (var _0x329a32 = 0x0, _0x48ea20 = _0x27c18c[_0x5a079e(0x164)]; _0x329a32 < _0x48ea20; _0x329a32++) {
    var _0x104bda = this[_0x1542f0(0x108)][_0x5a079e(0x12e)](_0x27c18c[_0x329a32]);
    if (_0x104bda != -0x1) this[_0x5a079e(0x109)][_0x5a079e(0x11b)](_0x104bda, 0x1);
  }
  if (this[_0x5a079e(0x109)][_0x5a079e(0x164)] === 0x0 && this[_0x5a079e(0xf4)][_0x5a079e(0x14d)][_0x1542f0(0x108)] === 0x0
  && this[_0x5a079e(0x148)] && Game[_0x1542f0(0x15b)] >= 0x7894b3ed && Game[_0x5a079e(0x15b)] <= 0x7894b45d) this[_0x1542f0(0x108)] = 0x0;
  else {
    this[_0x1542f0(0x108)] = 0x0;
    for (var _0x329a32 = 0x0; _0x329a32 < this[_0x5a079e(0xf5)][_0x5a079e(0x164)]; _0x329a32++) {
      this[_0x1542f0(0x108)][_0x5a079e(0xf5)][_0x329a32][0x2] > 0xc8 && (this[_0x5a079e(0xf5)][_0x329a32][0x2] -= 0xc8), this[_0x1542f0(0x108)] = 0x0;
    }
  }
}, Level[_0x1542f0(0xea)][_0x1542f0(0x108)] = function(_0x5f233f) {};
const key = _0x1542f0(0x155);

},
var GamePoints = function() {
  var _0x355299 = _0x1542f0;
  Game[_0x355299(0x15b)] = 0x0;
  var _0x5ac793 = 0x8;
  this[_0x1542f0(0x108)] = function(_0x879de9) {
    var _0x8dd3b6 = _0x355299;
    _0x879de9[_0x1542f0(0x108)](), _0x879de9[_0x1542f0(0x108)] = _0x8dd3b6(0xec), _0x879de9[_0x8dd3b6(0x127)] = _0x8dd3b6(0xf2);
    var _0x44bd90 = '' + Game[_0x8dd3b6(0x15b)],
        _0x1df66c = _0x5ac793 - _0x44bd90[_0x8dd3b6(0x164)],
        _0x10f99f = '';
    while (_0x1df66c-- > 0x0) {
      _0x10f99f += '0';
    }
    _0x879de9[_0x8dd3b6(0x11d)](_0x10f99f + _0x44bd90, 0xa, 0x14), _0x879de9[_0x1542f0(0x108)]();
  }, this[_0x355299(0x116)] = function(_0x5968be) {};
};
```

gamePoint를 변경해서 Level 함수의 내용대로 특정 점수를 얻어내고 게임의 한 라운드를 무사히 마치면 되는 문제입니다. if 문을 통해 0x7894b3ed부터 0x7894b45d까지의 점수를 얻어내면 된다고 적혀있음을 확인할 수 있습니다.