

안녕하세요 Bob 입니다.

- 자바스크립트 개발자입니다.
- 오래된 개발자입니다.
- 게임 개발 경험은 없습니다.
- 작은 **게임**들을 연습했습니다.



2019

만드는 사람이 즐거운 게임



2019

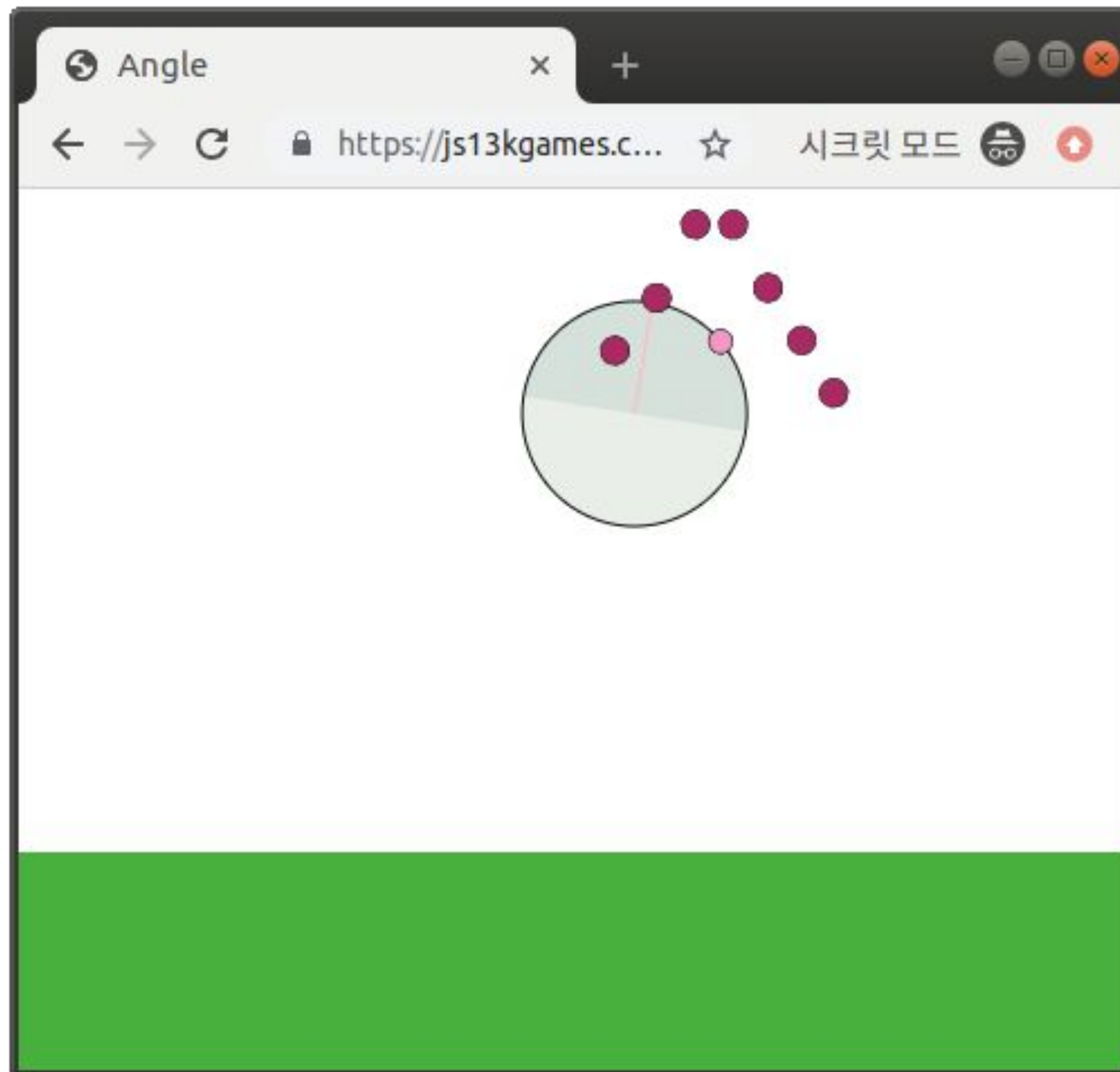
함수를 이어서 사용하면
이해하기 쉬움



2019

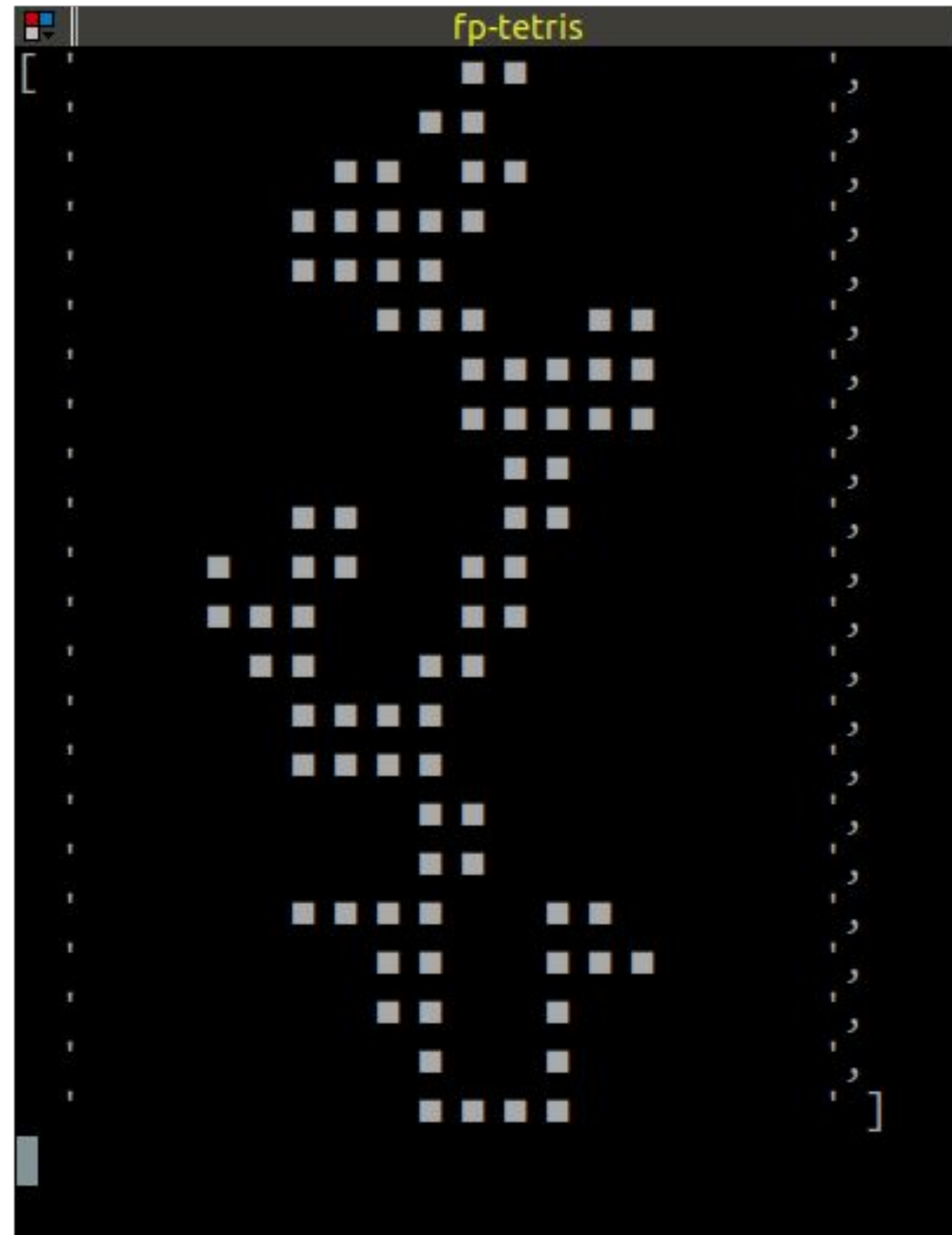
잡초 제거

<http://bit.ly/2rJog0Y>



2019

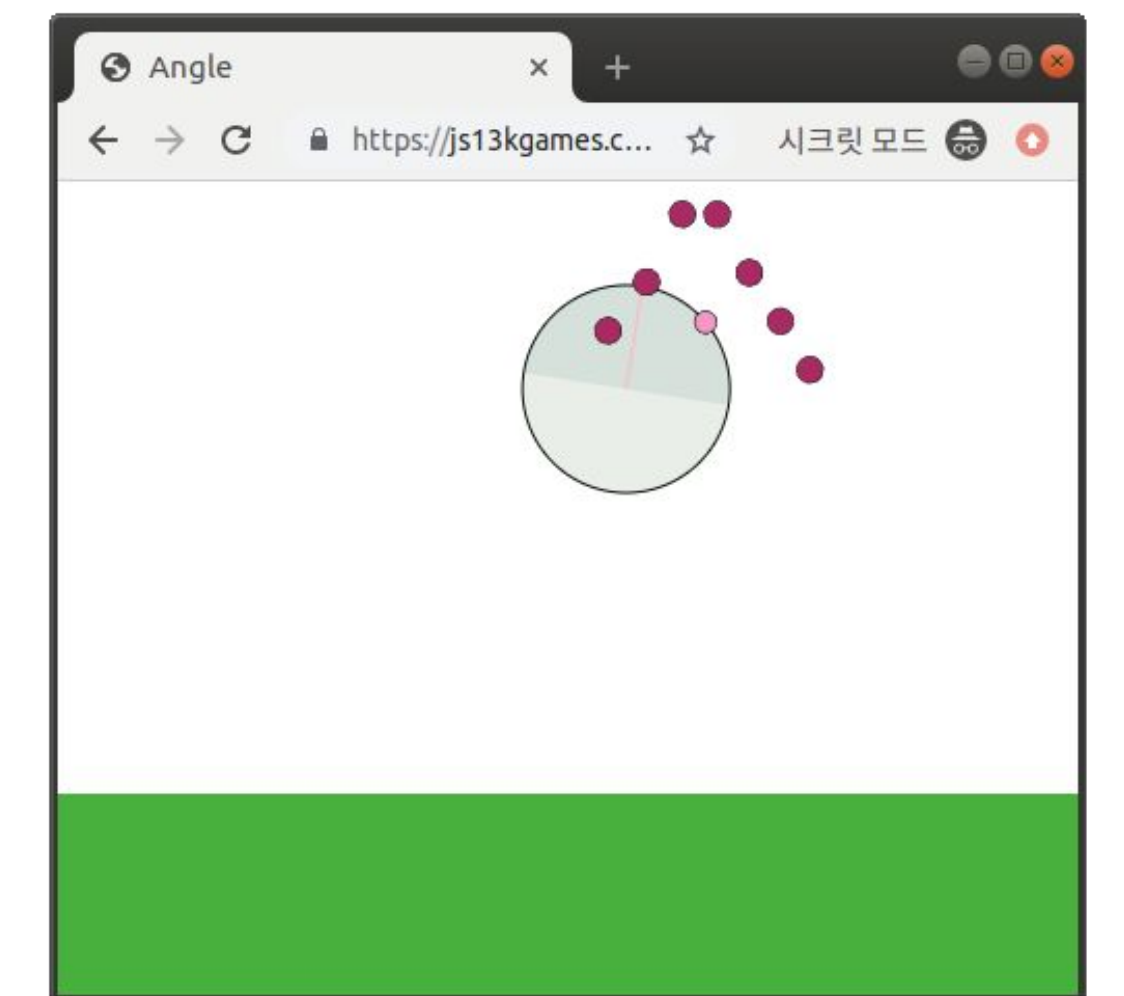
테트리스



2019

잡초 제거의 루프

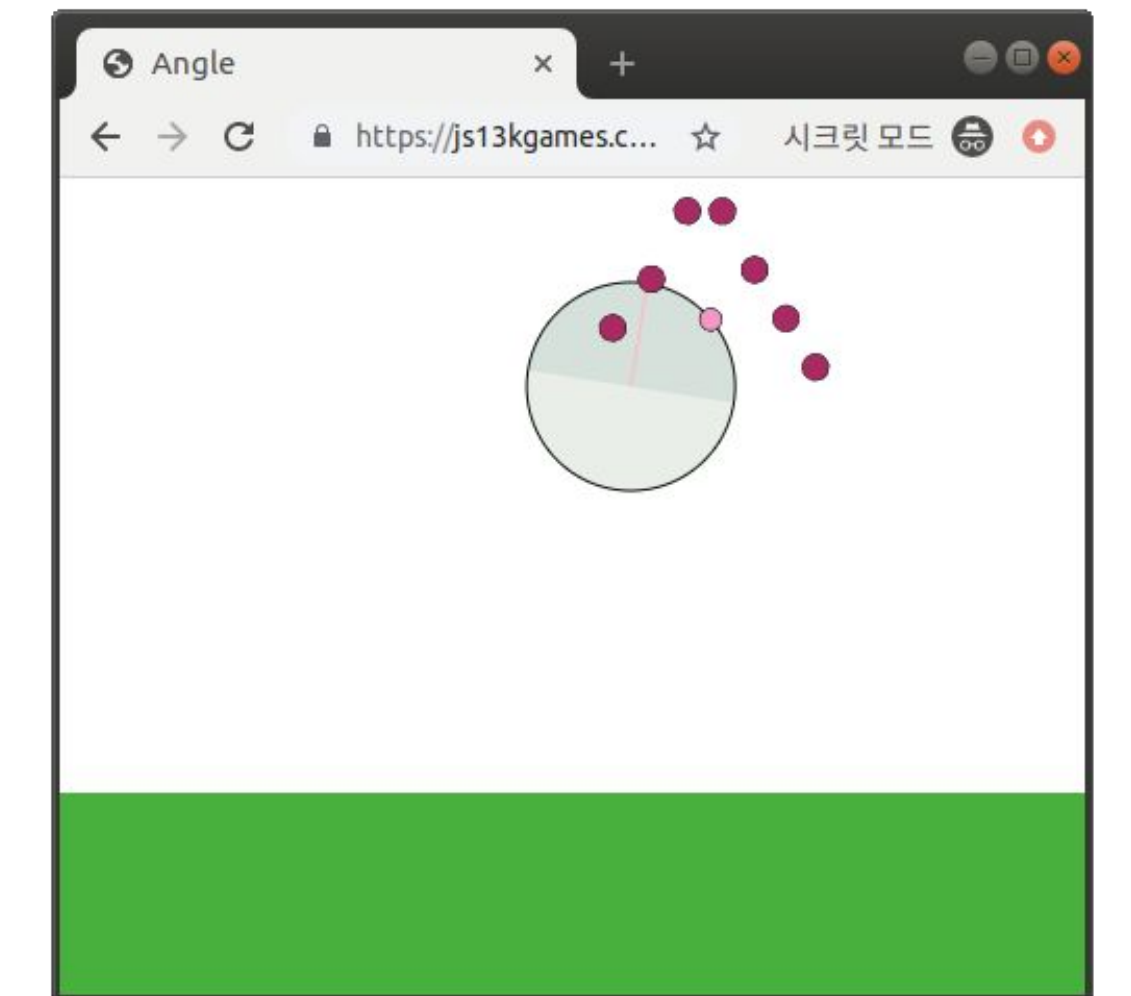
```
function animate() { http://bit.ly/354VJ4c  
  objs = update(objs);  
  clearScreen(ctx);  
  objs = draw(objs);  
}  
GLOBAL.timer = setInterval(function() {  
  animate();  
}, 30);
```



2019

함수를 이어서 사용하기

```
const update = compose(  
  moveCircles,  
  moveSatellites,  
  updateBall,  
  .....  
);  
const draw = compose(  
  drawCircles(ctx),  
  drawBlocks(ctx)  
);
```

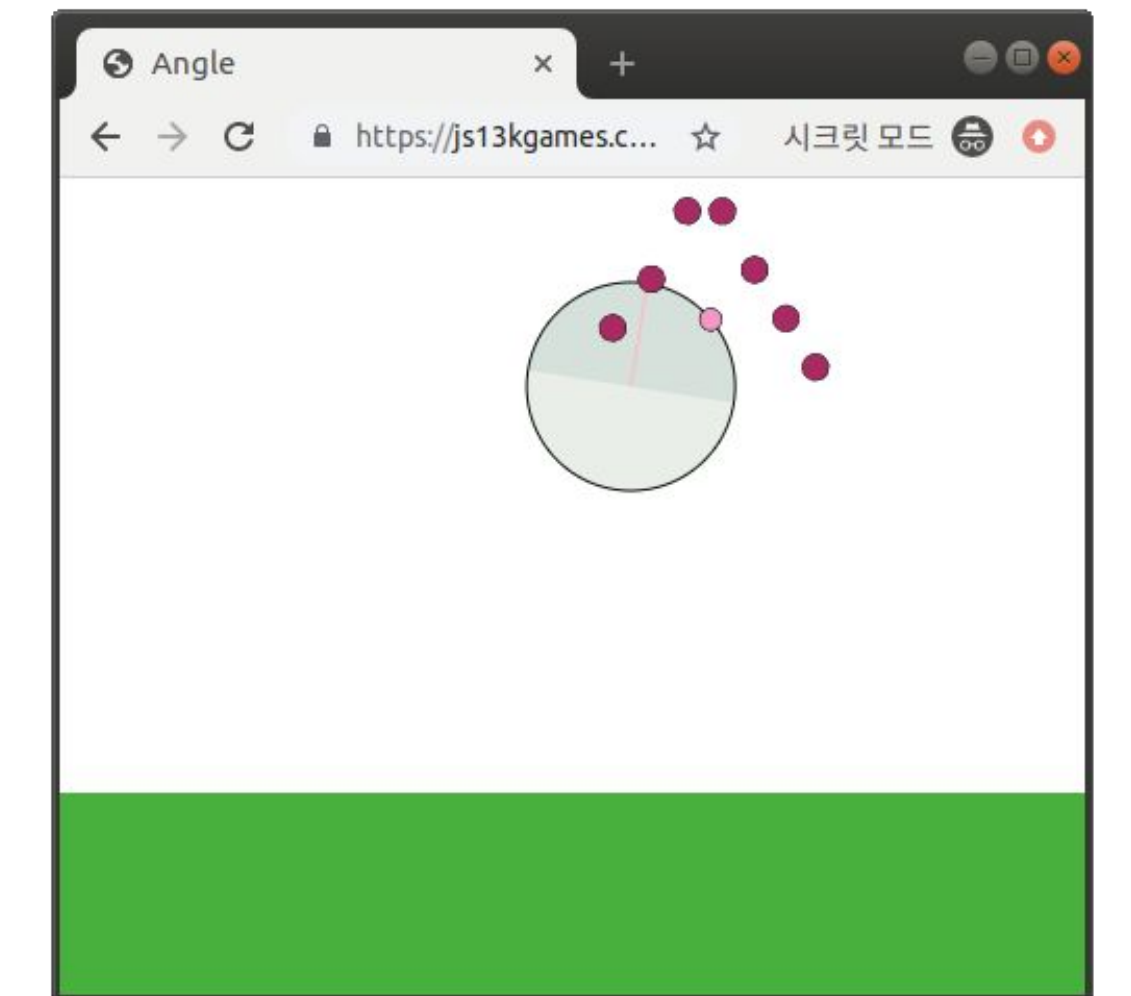


2019

같은 인자를 사용하는 여러 함수들

```
const updateBall = update(isUnderCount) (applyFreely);
```

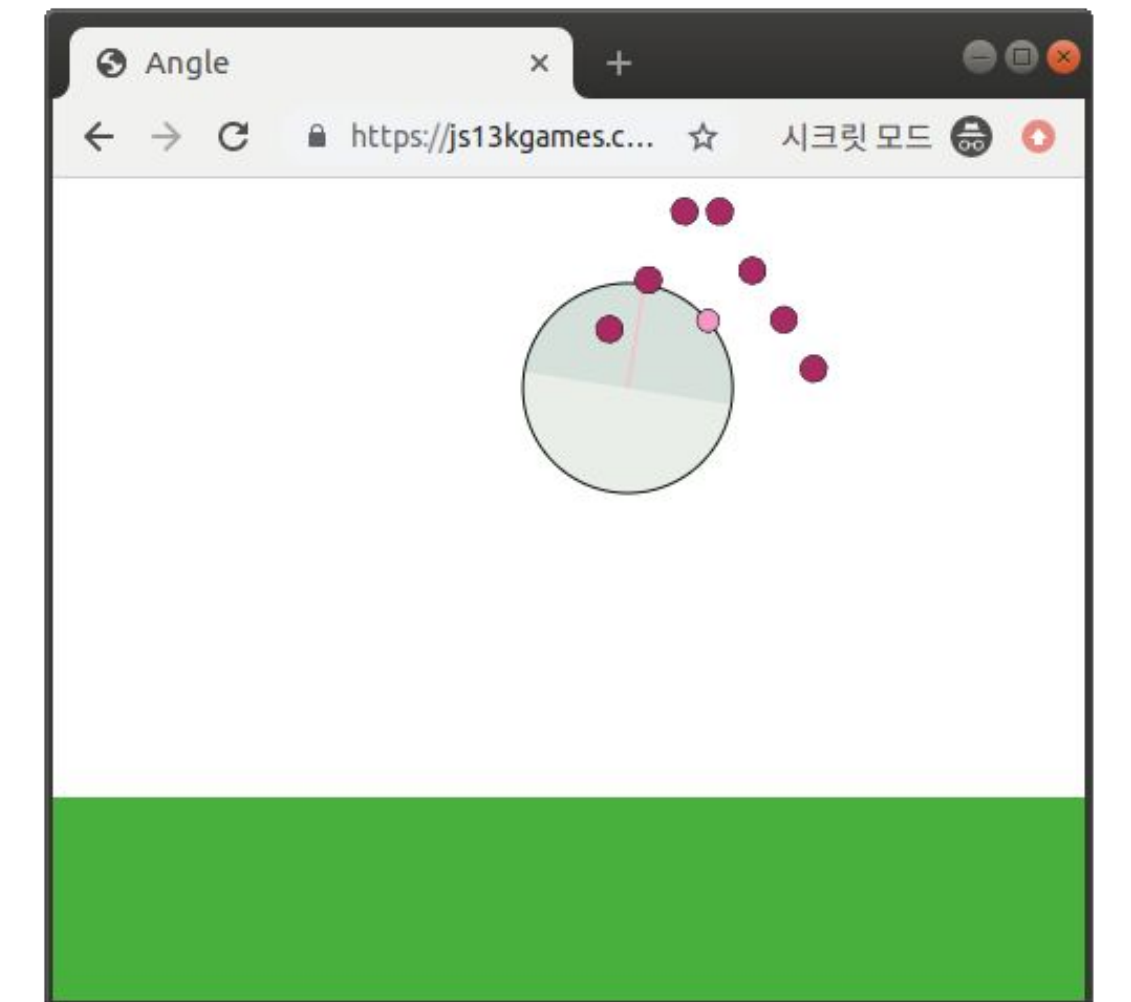
```
const applyFreely = compose(  
  incrementLifeCount,  
  applyLeftOrRight,  
  applyUpOrDown  
)
```



2019

함수를 이어주는 함수

```
function compose() {  
  var fns = arguments;  
  
  return function (result) {  
    for (var i = fns.length - 1; i > -1; i--) {  
      result = fns[i].call(this, result);  
    }  
  
    return result;  
  };  
};
```



2019

함수를 함수로 설명



2019

```

const update = compose(
  moveCircles,
  moveSatellites,
  checkBallOnTheBottom,
  checkBlockOnTheBottom,
  checkBlockOnTheTop,
  addBall,
  updateBall,
  checkCollisionBall,
  addBlock,
  moveBlocks,
  checkCollisionBlock,
  moveLeftOrRight,
  gravityBall,
  removeNoneType,
  startDemoKey
);

```

```

const updateBall =
update(isUnderCount) (applyFreely);

```

```

const applyFreely = compose(
  incrementLifeCount,
  applyLeftOrRight,
  applyUpOrDown
)

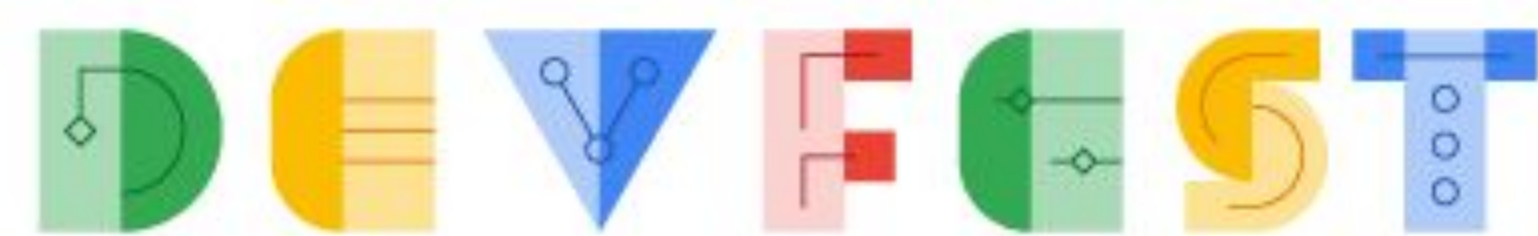
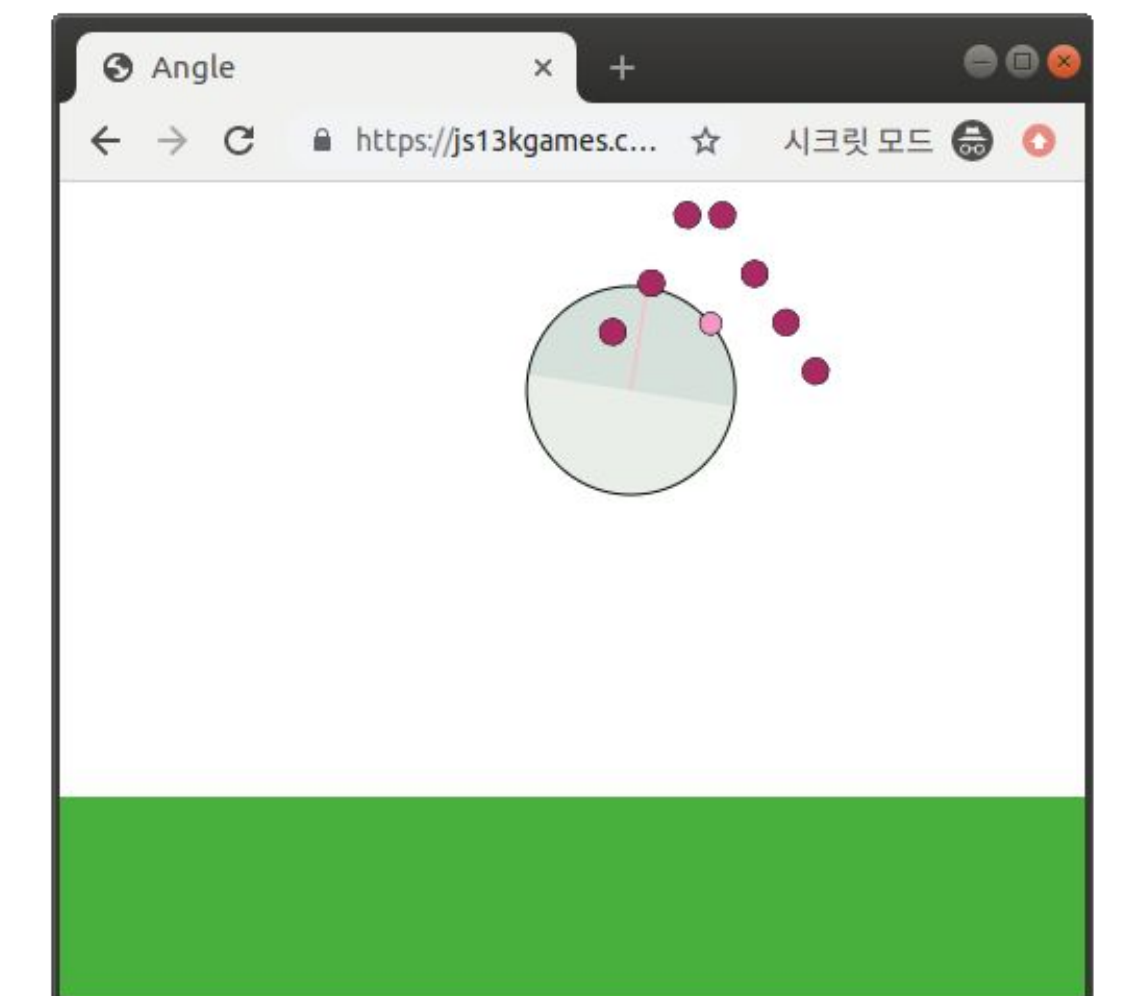
```

```

function applyLeftOrRight(circle) {
  .....
}

```

<http://bit.ly/2Kq6ial>

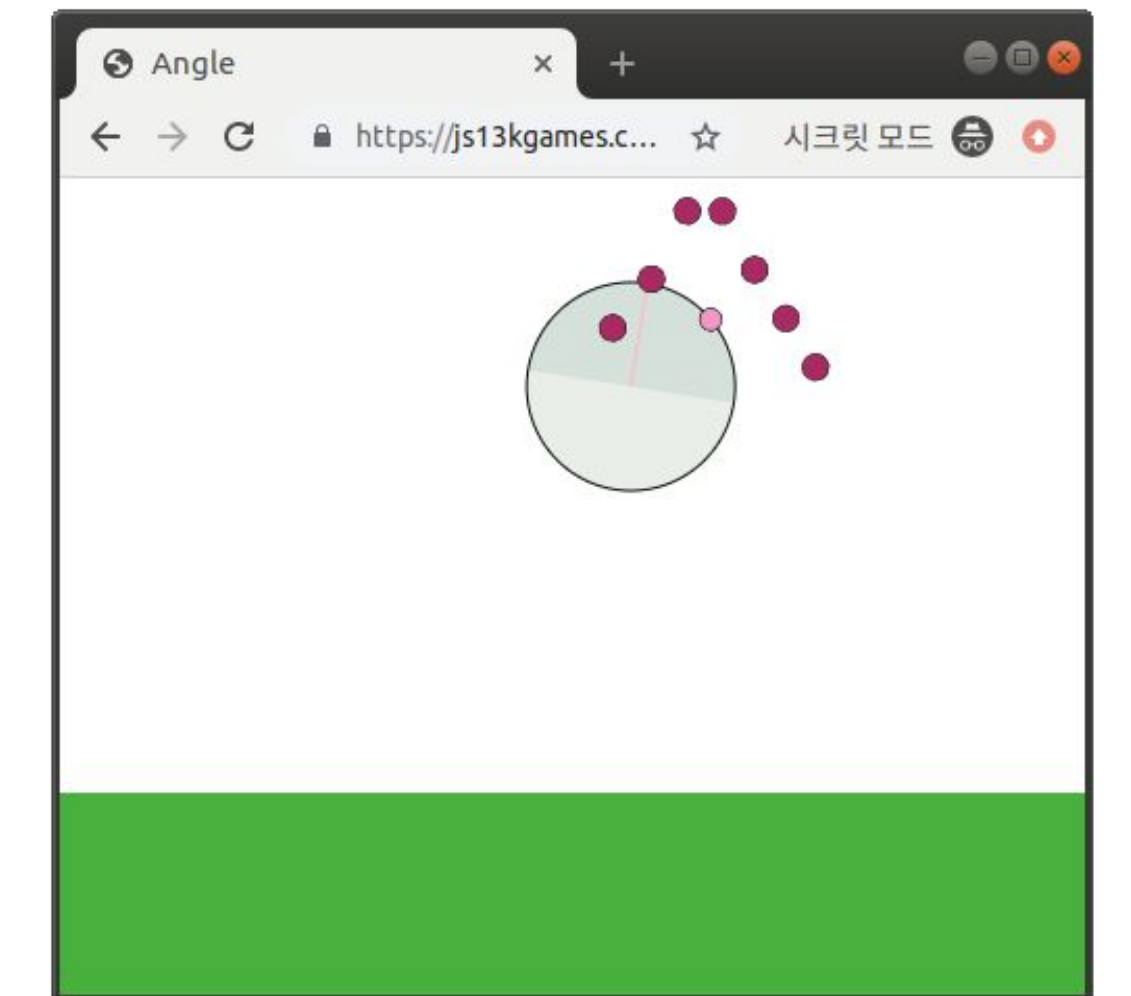


2019



더욱 구체적인 동작은 마지막 함수에서 볼 수 있다

```
function applyLeftOrRight(circle) {  
  if (circle.dx === undefined) return circle;  
  const c = clone(circle);  
  if (isInRange(c.x + c.dx, c.xRange)) {  
    c.x += c.dx;  
  } else {  
    c.angle = 180 - c.angle;  
    const { dx } = getDxDy(c.angle, c.speed);  
    c.dx = dx;  
  }  
  c.testAngle = getTestAngle(c.angle, c.testAngle)  
  return c;  
}
```



함수가 함수로 설명되면
책의 차례처럼 살펴볼 수 있다.



2019

장점

- 동일한 인자를 받고 동일한 인자를 리턴하는 함수들은 이해하기 쉽다.
- 함수들은 내부 상태를 가지지 않음으로 이해하기 쉽다.
- 함수 각각은 작은 크기를 유지함으로 이해하기 쉽다.



이해하기 쉬운 코드는

"어머 이게 뭐야?"

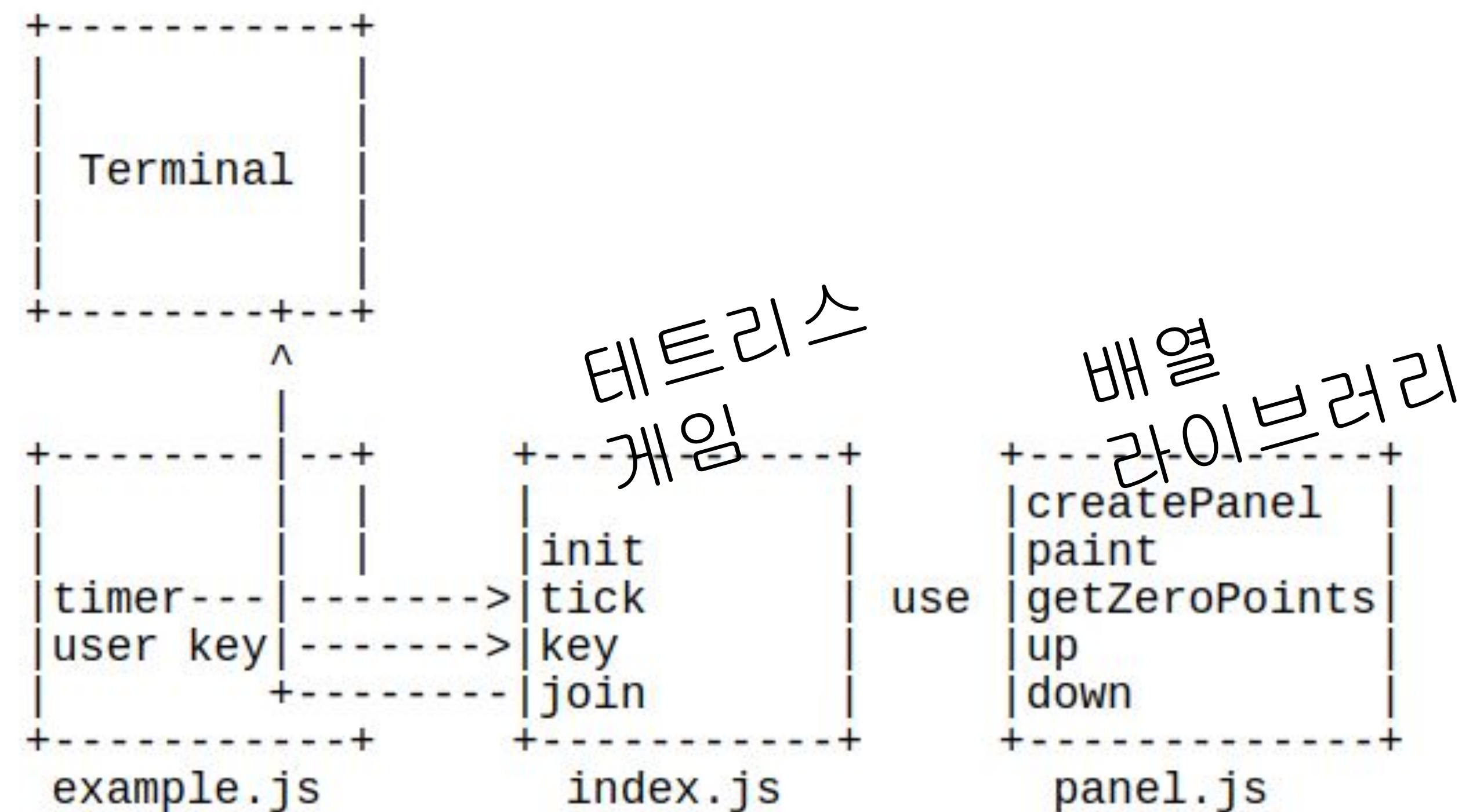
이렇게 놀라지 않는 코드라고
생각합니다.



2019

테트리스

- 사용자 키 입력과 타이머 이렇게 두 개의 이벤트를 처리
- 이벤트가 발생하면 단순한 데이터 구조를 여러 함수가 변경
- 아래 테트리스 게임과 배열 라이브러리에서는 일부 키워드를 사용하지 않음 (if, for, while, switch, class, this, i++)



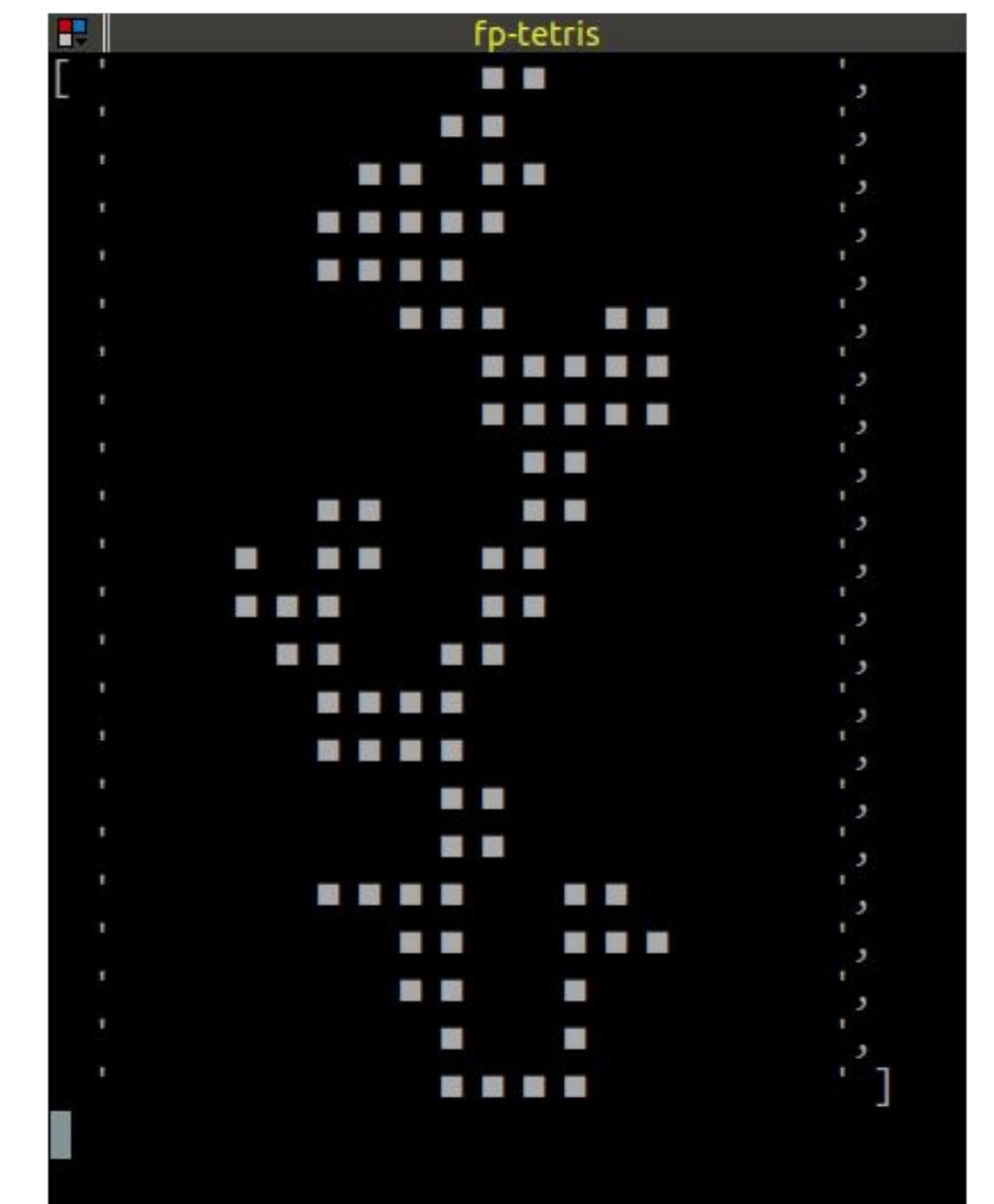
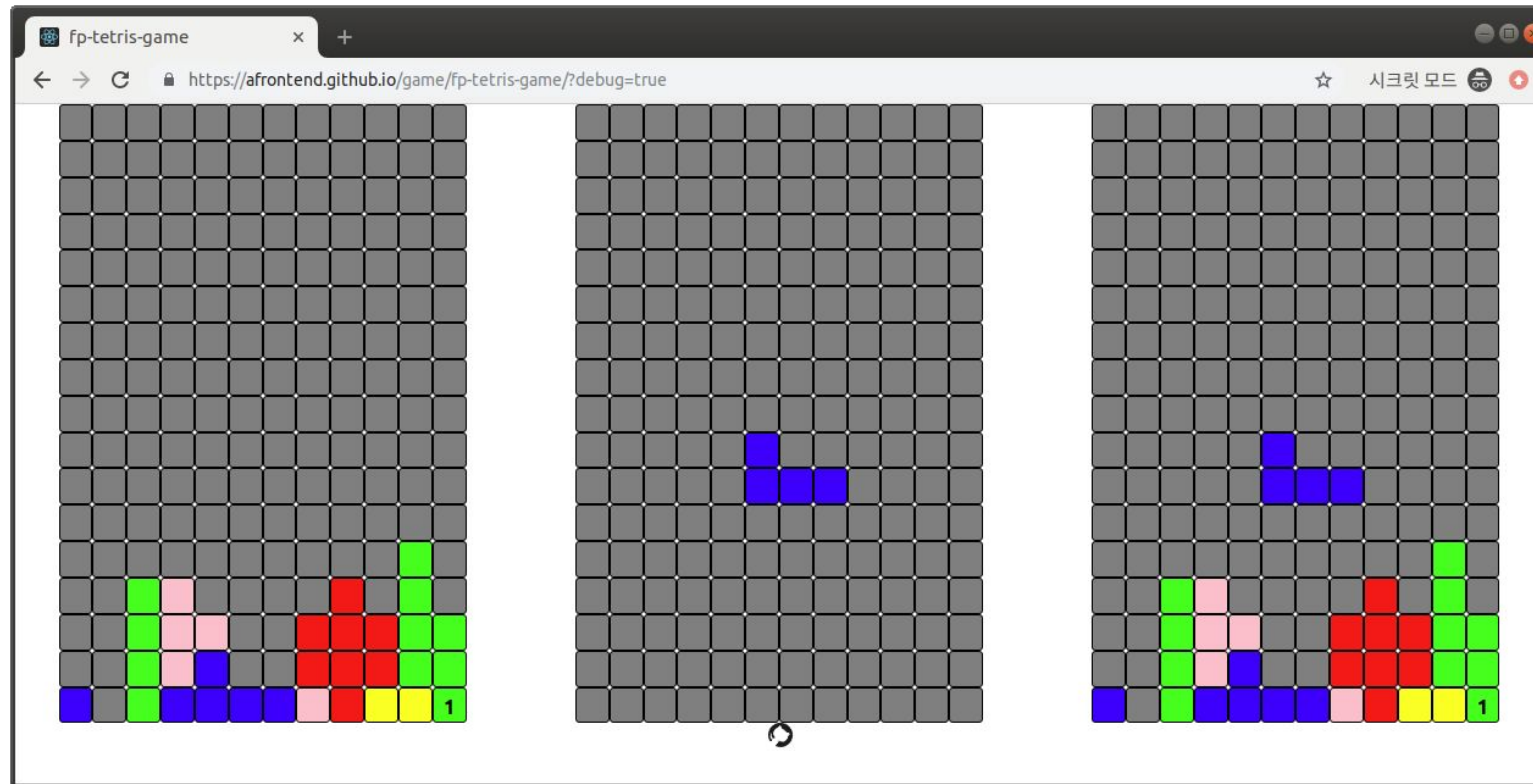
<http://bit.ly/2NNn318>



2019

단순한 데이터 구조를 사용

<http://bit.ly/2QmfjF0>



2019

단순한 데이터 구조의 장점

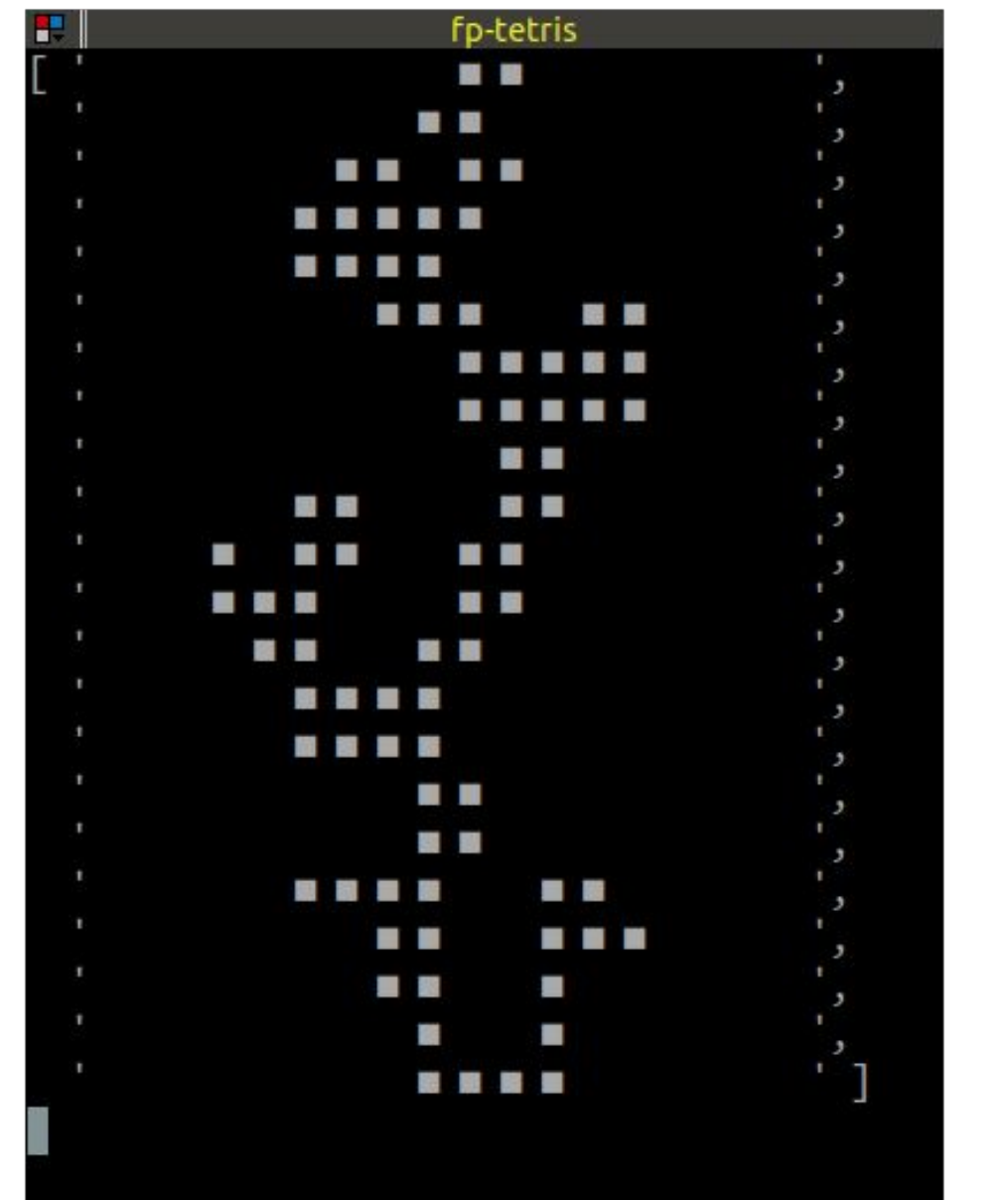
- 2차원 배열 2개는 특별하지 않아서 이해하기 쉬움
- 저장이 쉬움
- 좌표 계산 하지 않음



함수 이어서 사용하기의 또 다른 예

<http://bit.ly/2NMt4ey>, <http://bit.ly/2ppPrx6>

```
const lastRowIsBlank = fp.pipe(  
  fp.last,  
  fp.filter(isItem),  
  fp.size,  
  Boolean  
);  
const down = fp.pipe(  
  fp.cloneDeep,  
  p => [getEmptyRow(p[0].length), ...p.slice(0, -1)]  
);
```



2019

그래서 결론은?



2019

함수를 이어서 사용하면
이해하기 쉬움



2019

A collection of various colorful geometric shapes and icons scattered across the page, including circles, triangles, squares, rectangles, and lines in shades of blue, yellow, green, red, and pink. Some icons resemble hands or abstract symbols.

감사합니다.

작은 게임들

<http://bit.ly/2yqBRuR>

블로그 글

<http://bit.ly/2NNn318>