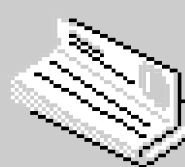
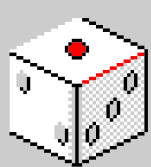


Programación Reactiva

Grupo 5

Nicolás Barría
Sebastián Carrasco
Juan Vargas



11:11 PM



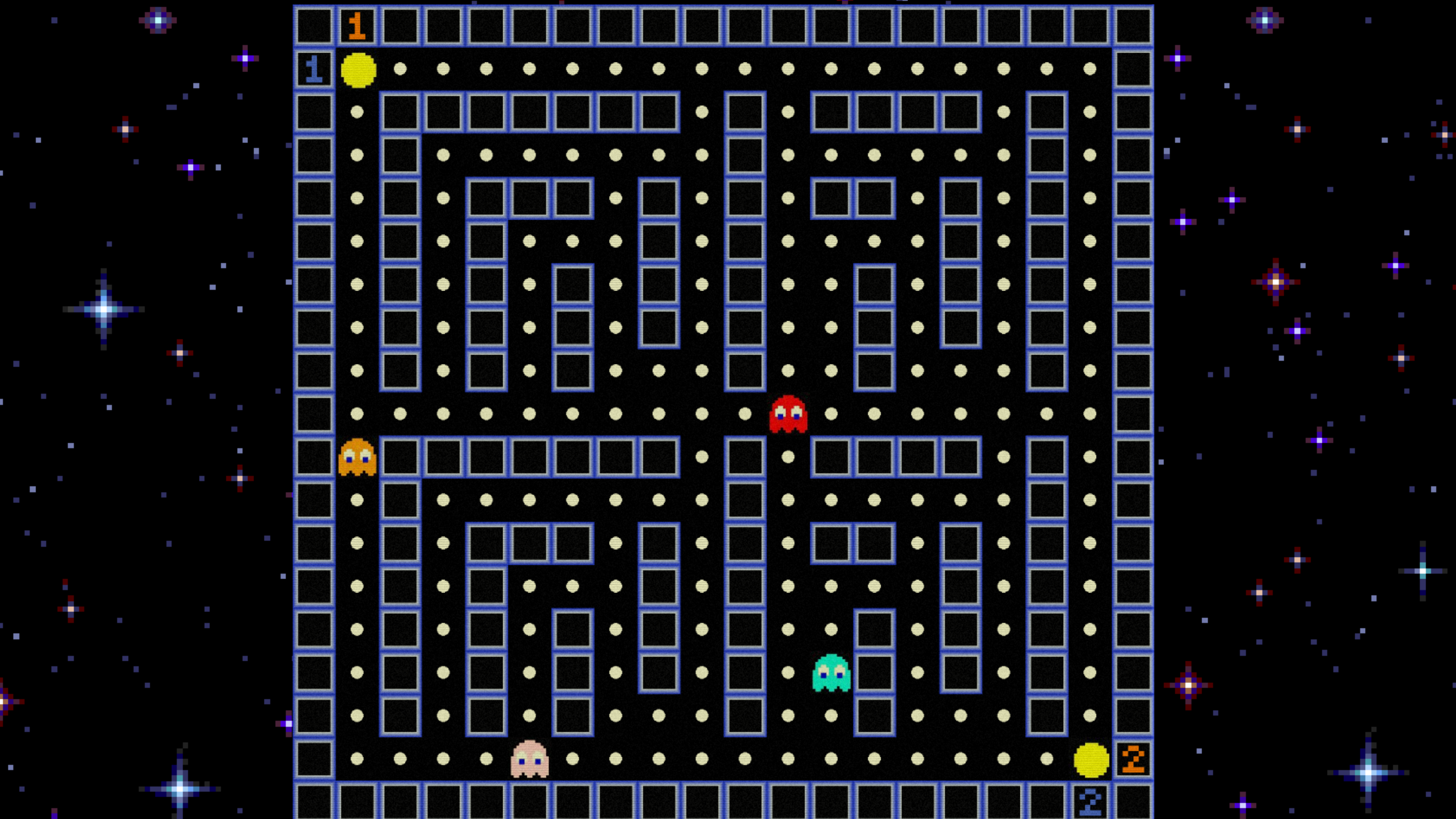
START

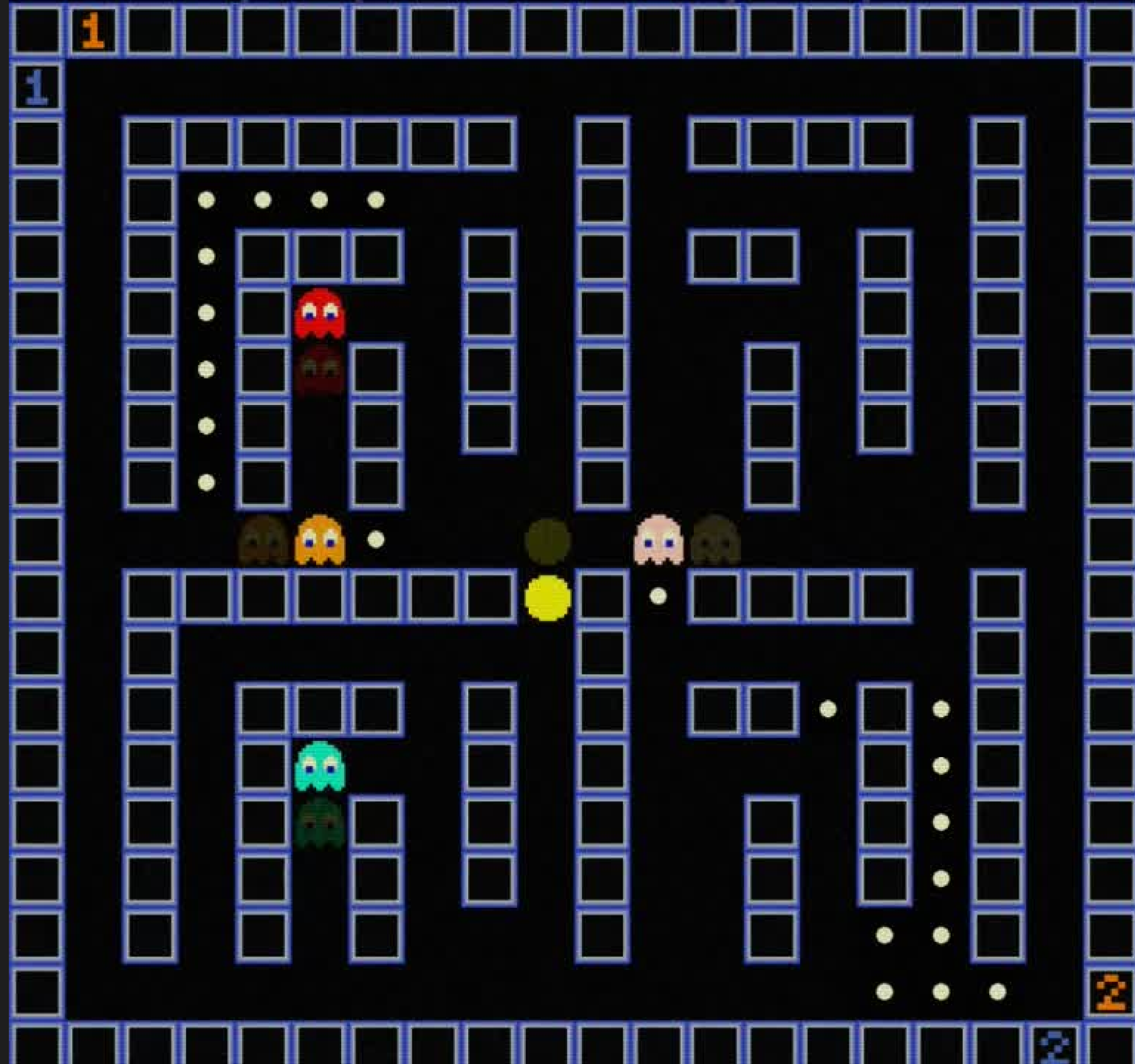
P1: WASD

P2: ↑←↓→

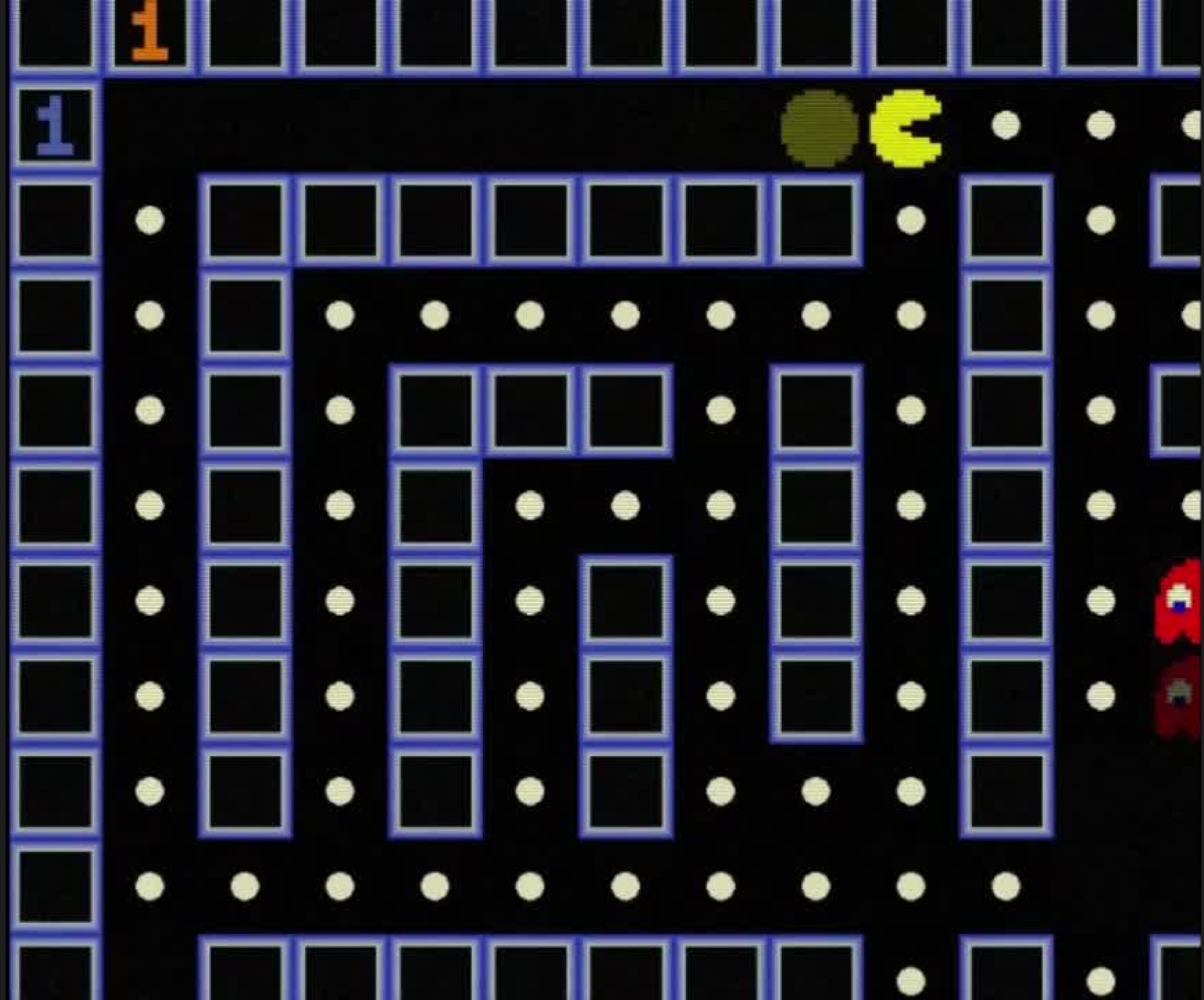
Portal: CV

Portal: OP





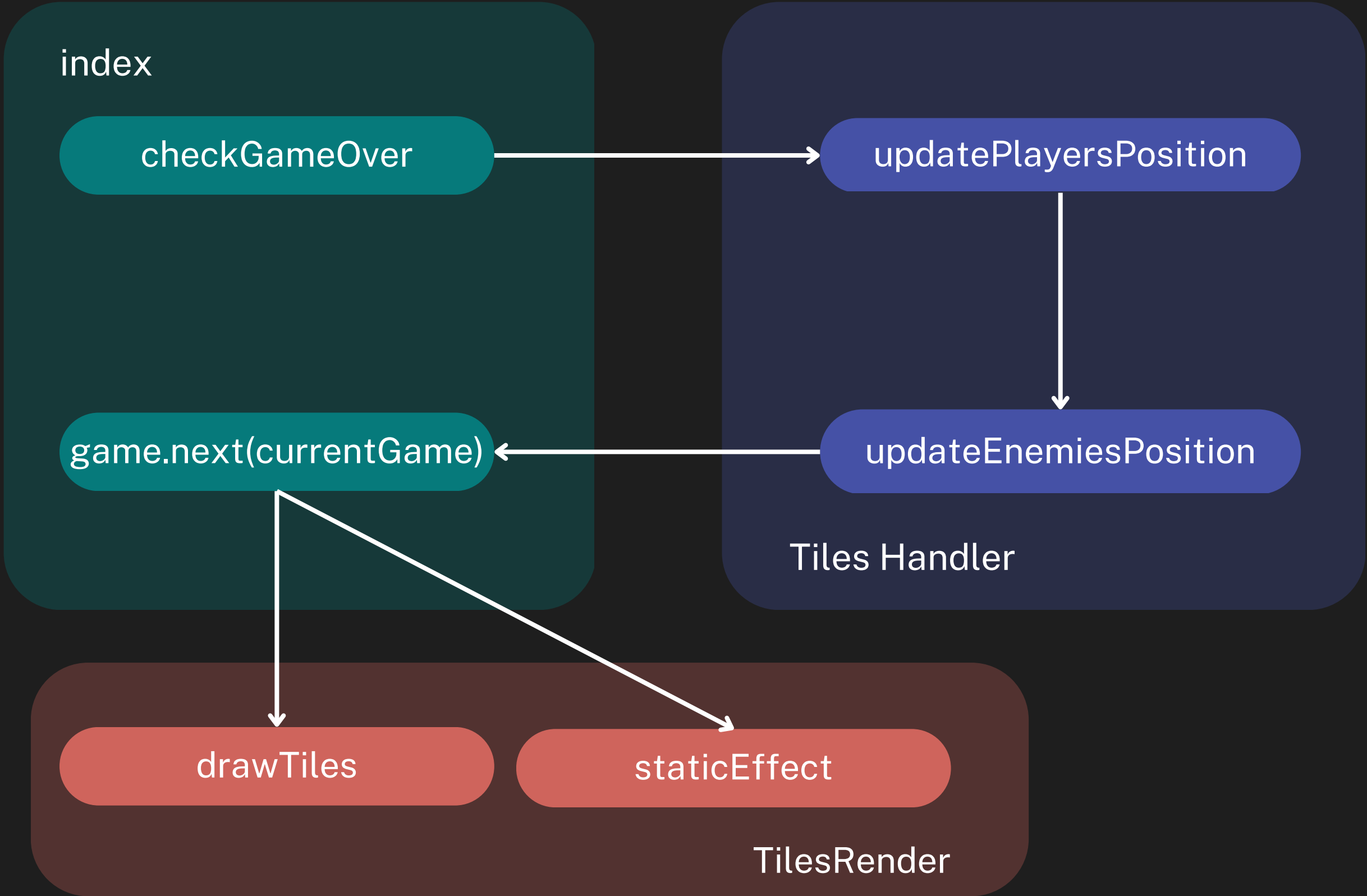




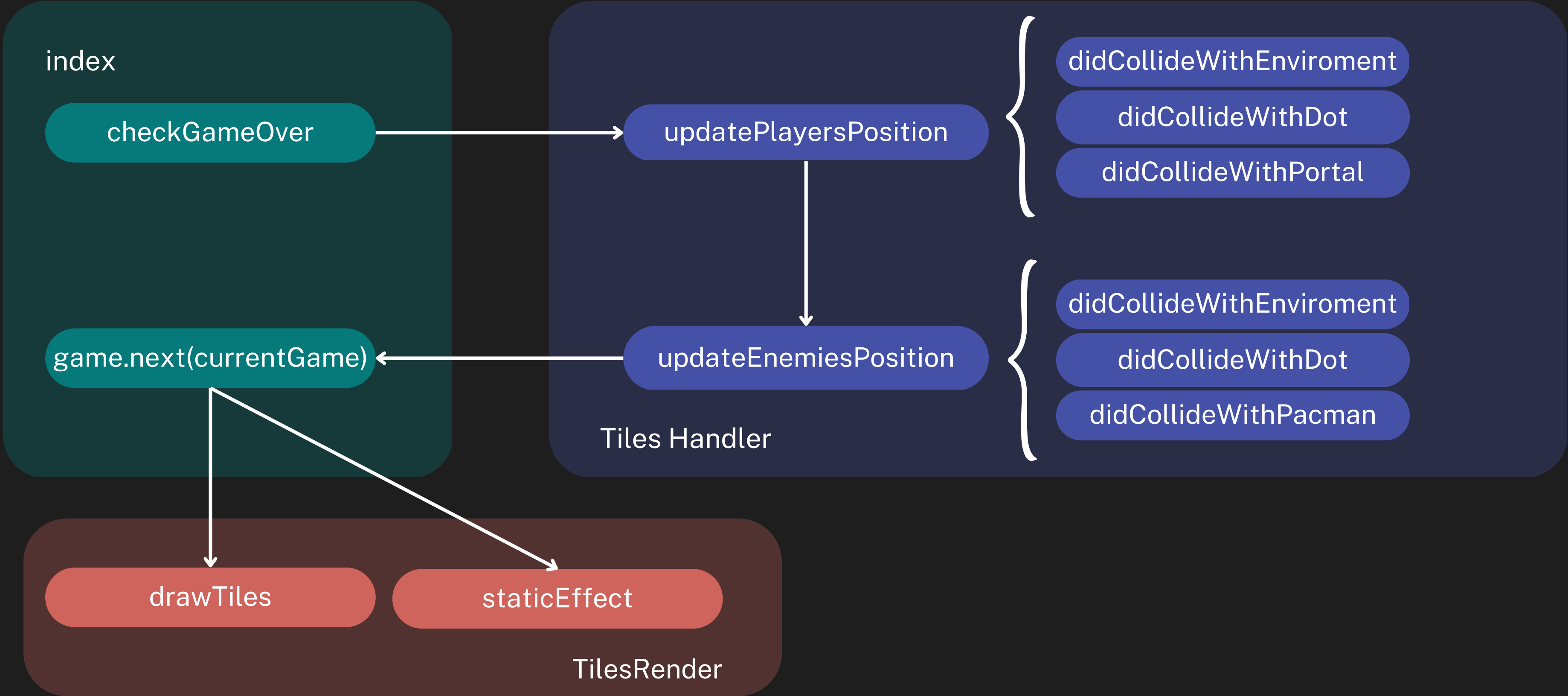
Demo

v4rgas.github.io/reactive-js-grupo-05

Enfoque funcional



Enfoque funcional



BehaviorSubject Game

Requiere un valor inicial y emite su valor actual a sus suscriptores



```
1  const game = new BehaviorSubject({  
2    tiles,  
3    p1: P1_START,  
4    p2: P2_START,  
5    enemies: ENEMIES,  
6    started: false,  
7    over: false,  
8  });
```

Inicio del juego



```
1  fromEvent(canvas, 'click')
2    .pipe(
3      takeUntil(game.pipe(filter((currentGame) => currentGame.started))),
4      withLatestFrom(game),
5      map([, currentGame]) => ({
6        ...currentGame,
7        started: true,
8      })),
9    )
10   .subscribe((newGame) => {
11     game.next(newGame);
12     clearCanvas();
13   });
```

Eventos de presión de teclas



```
1  const keydown = fromEvent(document, 'keydown')
2      .pipe(
3          takeUntil(game.pipe(filter((currentGame) => currentGame.over))),
4          skipUntil(game.pipe(filter((currentGame) => currentGame.started))),
5          withLatestFrom(game));
```


Cambio de dirección



```
1  keydown.pipe(filter(([ { key }, _ ]) => Object.keys(DIRECTIONSP1).includes(key)))
2      .subscribe([currentKeydown, currentGame]) => {
3      const { p1 } = currentGame;
4      game.next({
5          ...currentGame,
6          p1: { ...p1, direction: DIRECTIONSP1[currentKeydown.key] },
7      });
8  })
```

Disparo de portales



```
1  keydown.pipe(filter(({ key }, _) => Object.keys(POWERSP1).includes(key)))
2    .subscribe(([currentKeydown, currentGame]) => {
3      const { p1, tiles } = currentGame;
4      let newTiles, newPortal1, newPortal2;
5      if (POWERSP1[currentKeydown.key] === 'portal1') {
6        [newTiles, newPortal1] = shootPortal(p1, p1.portal1, tiles);
7      }
8      if (POWERSP1[currentKeydown.key] === 'portal2') {
9        [newTiles, newPortal2] = shootPortal(p1, p1.portal2, tiles);
10     }
11     game.next({
12       ...currentGame,
13       p1: { ...p1, portal1: newPortal1, portal2: newPortal2 },
14       tiles: newTiles,
15     });
16   });
```



Ticks del juego

```
1 timer(0, 500)
2   .pipe(
3     skipUntil(game.pipe(filter((currentGame) => currentGame.started))),
4     takeUntil(game.pipe(filter((currentGame) => currentGame.over))),
5     withLatestFrom(game)
6   ).subscribe([, currentGame]) => {
7     if (checkGameOver(currentGame)) {
8       game.next({ ...currentGame, over: true });
9       return;
10    }
11
12    let [newTiles, newPlayer1, newPlayer2] = updatePlayersPosition(currentGame);
13    let newEnemies = [];
14
15    [newTiles, newEnemies, newPlayer1, newPlayer2] = updateEnemiesPosition({
16      ...currentGame,
17      tiles: newTiles,
18      p1: newPlayer1,
19      p2: newPlayer2,
20    });
21
22    game.next({
23      ...currentGame,
24      tiles: newTiles,
25      p1: newPlayer1,
26      p2: newPlayer2,
27      enemies: newEnemies,
28    });
29  });
```

Animación de pacman



```
1  interval(100)
2    .pipe(
3      skipUntil(game.pipe(filter((currentGame) => currentGame.started))),
4      takeUntil(game.pipe(filter((currentGame) => currentGame.over))),
5      withLatestFrom(game),
6    )
7    .subscribe(([time, currentGame]) => {
8      const { p1, p2 } = currentGame;
9      if (time % 2 === 0) {
10        p1.image = PACMAN_I[p1.direction];
11        p2.image = PACMAN_I[p2.direction];
12      } else {
13        p1.image = PACMAN_I.circle;
14        p2.image = PACMAN_I.circle;
15      }
16      game.next({ ...currentGame, p1, p2 });
17    });
```


Dibujo del juego

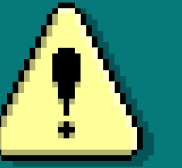
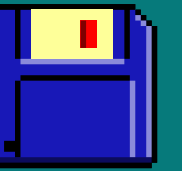
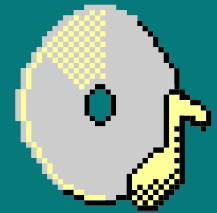
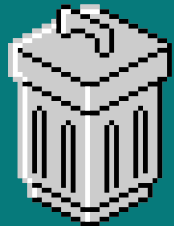
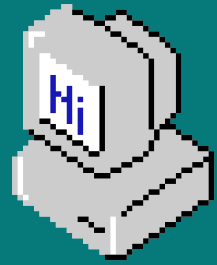



```
1  game.subscribe((currentGame) => {  
2    if(!currentGame.over){  
3      drawTiles(currentGame, ctx);  
4    }  
5    staticEffect(ctx, canvas.width, canvas.height);  
6  });
```

Fin del juego




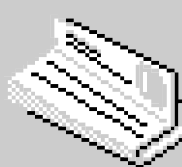
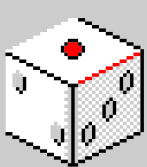
```
1  game
2    .pipe(filter((currentGame) => currentGame.over))
3    .subscribe((currentGame) => {
4      clearCanvas();
5      drawScores(currentGame.p1, currentGame.p2);
6      staticEffect(ctx, canvas.width, canvas.height);
7    });
8
```



 Programación Reactiva

Grupo 5


Nicolás Barría
Sebastián Carrasco
Juan Vargas



11:11 PM