

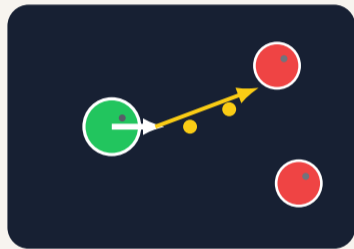
Dal ciclo al videogioco

Laboratorio JavaScript: costruiamo un mini shooter

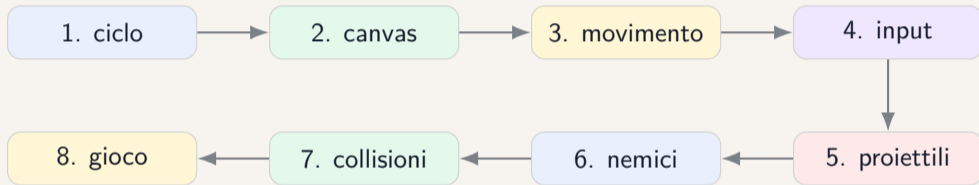
Game Design Summer School 2026

Alla fine sappiamo spiegare

- che cosa e' un ciclo;
- perche' un gioco aggiorna lo schermo molte volte;
- come tastiera, posizione e velocita' diventano regole;
- come array e collisioni creano il gameplay.



La mappa degli step



Step 1: un ciclo nel tempo

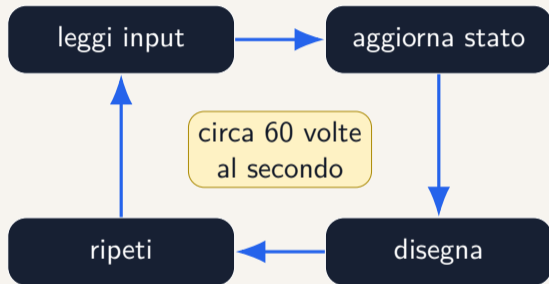
```
1 function setup() {  
2   let volte = 0;  
3  
4   setInterval(() => {  
5     volte = volte + 1;  
6     log("tick " + volte);  
7   }, 500);  
8 }
```

Idea chiave

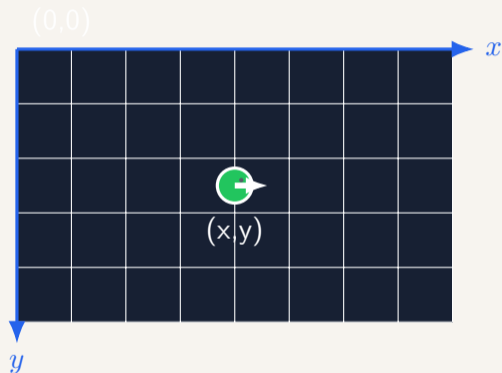
Un gioco non fa una sola cosa: ripete continuamente piccole azioni.



Il game loop



Coordinate: la mappa dello schermo



Da ricordare

- x cresce verso destra.
- y cresce verso il basso.
- ogni oggetto ha una posizione.

Movimento = posizione + velocità

```
1 let x = 80;
2 let velocitaX = 3;
3
4 function draw() {
5   background("#172033");
6   x = x + velocitaX;
7   circle(x, 240, 70);
8 }
```

stessa regola, nuovo frame



Una regola: rimbalza

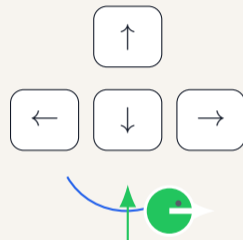
```
1 if (x > width - raggio || x < raggio) {  
2     velocitaX = velocitaX * -1;  
3 }
```

Traduzione

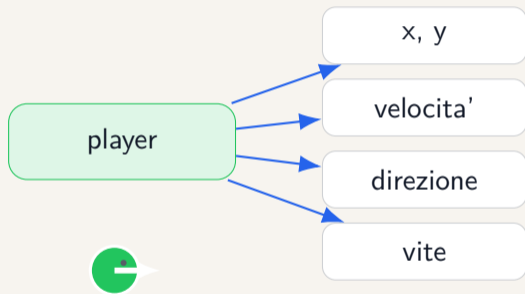
Se arrivi al bordo, vai nella direzione opposta.

Input: il giocatore entra nel gioco

```
1 if (keyIsDown(LEFT_ARROW) || keyIsDown("a")) {
2   player.angle -= player.rotazione;
3 }
4
5 if (keyIsDown(RIGHT_ARROW) || keyIsDown("d")) {
6   player.angle += player.rotazione;
7 }
8
9 if (keyIsDown(UP_ARROW) || keyIsDown("w")) {
10  player.x += Math.cos(player.angle) * velocita;
11  player.y += Math.sin(player.angle) * velocita;
12 }
```



Stato del giocatore



Array: una lista di proiettili

```
1 let bullets = [];  
2 let angle = player.angle;  
3  
4 bullets.push({  
5   x: player.x,  
6   y: player.y,  
7   vx: Math.cos(angle) * 9,  
8   vy: Math.sin(angle) * 9  
9 });
```

0

1

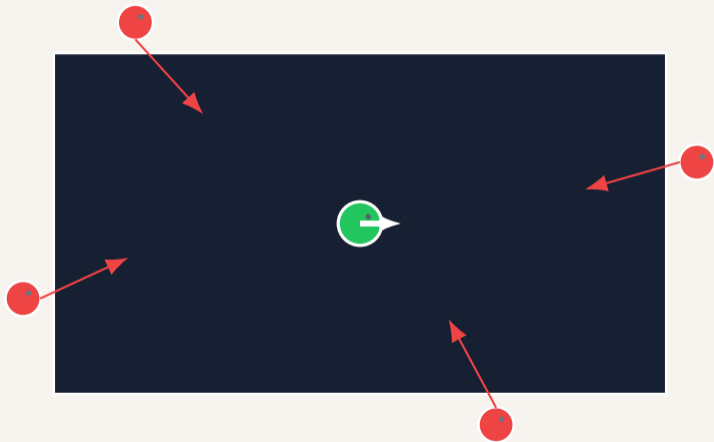
2

3

4

ogni casella e' un proiettile

Nemici: nascono fuori dallo schermo



Collisioni: quando due cerchi si toccano



Regola

Se la distanza tra i centri e' piu' piccola della somma dei raggi, c'e' collisione.

$$distanza < r_1 + r_2$$

Collisione in codice

```
1 if (dist(bullet.x, bullet.y, enemy.x, enemy.y)
2     < bullet.raggio + enemy.raggio) {
3     bullets.splice(b, 1);
4     enemies.splice(e, 1);
5     score += 10;
6 }
```

Qui succedono tre cose

il proiettile sparisce, il nemico sparisce, il punteggio sale.

Feedback: il gioco parla al giocatore



colore

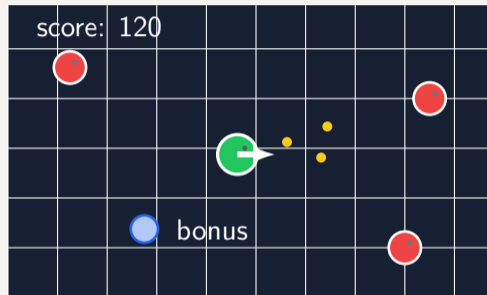


movimento esplosione

Mini shooter finale

Cosa c'e' dentro

- game loop;
- arena 1440x960;
- controllo a rotazione;
- sparo automatico singolo o triplo;
- bonus arma e riparazione;
- due tipi di nemici;
- collisioni;
- punteggio, vite e restart.



Domande per far ragionare

- Che cosa succede se aumento la velocità del player?
- Perché i proiettili fuori dallo schermo vanno cancellati?
- Come potremmo aggiungere un boss?
- Come cambierebbe il gioco se i nemici scappassero invece di inseguire?
- Quale variabile controlla la difficoltà?

Facile

Cambia colori, dimensioni e velocita'.

Media

Aggiungi un nemico piu' lento ma piu' grande.

Difficile

Aggiungi power-up o livelli.

Un gioco e' un ciclo

che legge input, aggiorna regole e disegna conseguenze.

