

Genova, 18-20 ottobre 2024

[ITADINFO]

2° CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

Come nasce un videogioco: aspetti di progettazione multimediale audio-video ed interfacce

Adriano Mancini¹, Leonardo Gabrielli¹, Maura Mengoni², Gionata Massi³ e Federico Robuffo⁴

¹Dipartimento di Ingegneria dell'Informazione (DII), Università Politecnica delle Marche, Ancona, IT

²Dipartimento di Ingegneria Industriale e Scienze Matematiche (DIISM), Università Politecnica delle Marche, Ancona, IT

³Istituto Istruzione Superiore Savoia Benincasa, Ancona, IT

⁴Istituto Istruzione Superiore Cambi Serrani, Falconara M.ma, Ancona, IT

■ Struttura della presentazione

- Contesto di riferimento
- Come nasce un videogame
 - UI/UX
 - Audio
 - Coding
- Il gioco sviluppato
- Focus finale su Competenze, Abilità e Conoscenze
- Considerazioni finali



■ Mini-corsi PNRR (DM 934 del 3 agosto 2022)

- **Orientamento attivo** nella transizione scuola-università” – nell’ambito del Piano Nazionale di Ripresa e Resilienza, Missione 4 “Istruzione e ricerca
- **Progetto** finanziato dall’Unione europea – **NextGenerationEU**



<https://www.mur.gov.it/it/pnrr/misure-e-componenti/m4c1/investimento-16-orientamento-attivo-nella-transizione>



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

[ITADINFO]

2° CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

■ Mini-corsi PNRR (DM 934 del 3 agosto 2022)

- **Requisiti:**

- Durata 15h
- Almeno 2/3 in presenza
- 1h di orientamento universitario
- Frequenza minima: 70%



Contesto

- Laurea Triennale in Ingegneria dell'Informazione per Videogame e Realtà Virtuale
- Corso unico nel suo genere in Italia
- Attivo il terzo anno (A.A.24/25)
- 100+ iscritti in media
- Primi/e laureati/e a Luglio 2025



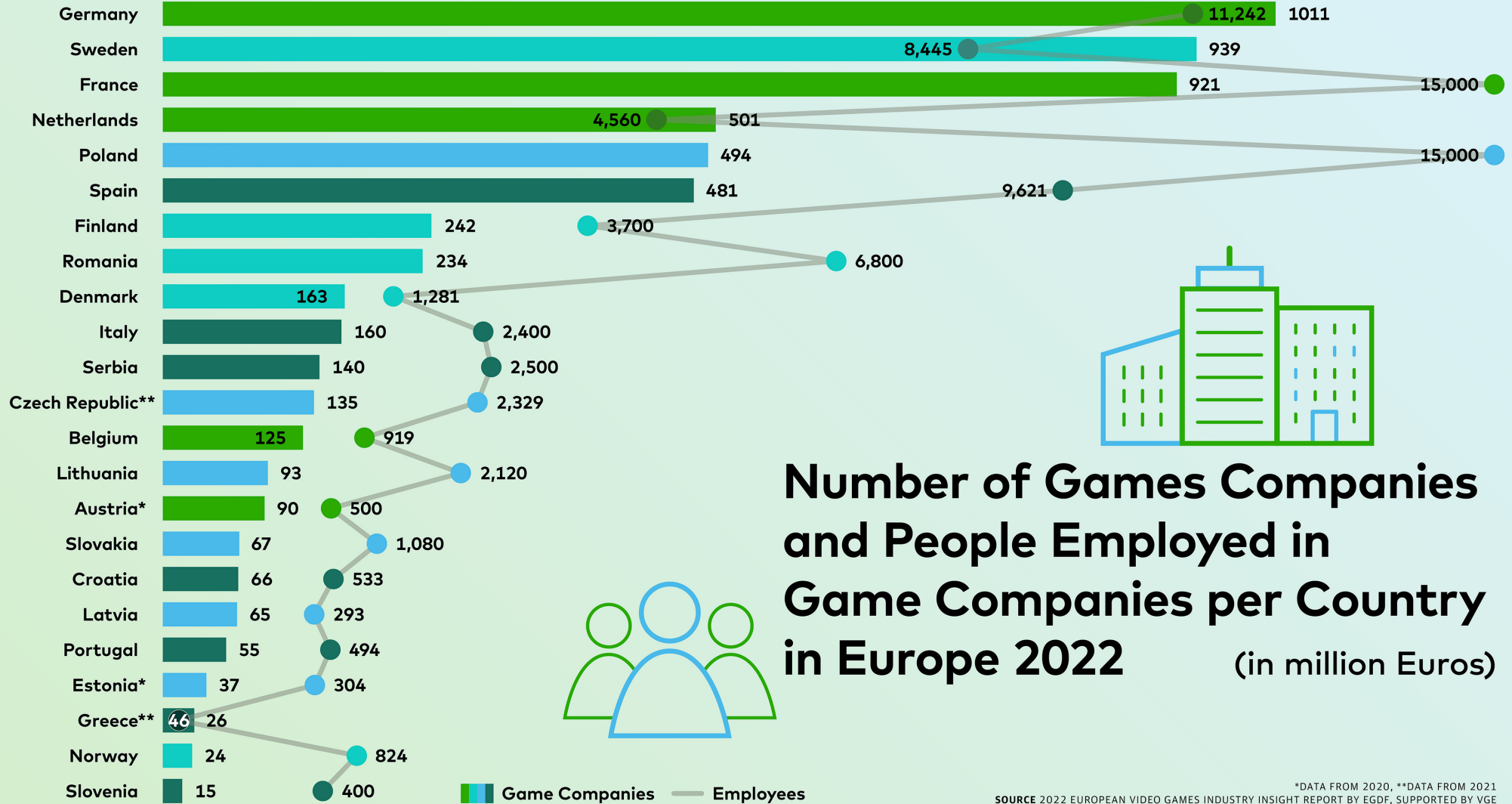
UNIVERSITÀ
POLITECNICA
DELLE MARCHE



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

*Ingegneria
dell'
Informazione
per
Videogame e
Realtà
Virtuale*

Contesto EU...



Number of Games Companies and People Employed in Game Companies per Country in Europe 2022
(in million Euros)



Progettazione del mini-corso

- Il corso mira a far comprendere come dietro alla **creazione** di un **videogame** vi sia un **complesso insieme di competenze specializzate**, dalla progettazione grafica all'ingegneria del software, passando per la composizione musicale e la progettazione dell'interfaccia utente.
- Il corso vuole mostrare come la **collaborazione** tra **esperti** in questi campi sia **essenziale** per **creare un videogioco di successo**.



Moduli

- Il mini-corso è stato strutturato in tre fasi principali.
- Ha offerto una panoramica delle competenze necessarie per creare un videogioco

Modulo #1 Introduzione alla UI e UX con cenni di game design

- 4h (1h30min lezione frontale, 2h30 attività laboratorio)

Modulo #2 Audio e post-processamento

- 5h (2h lezione frontale, 3h attività laboratorio)

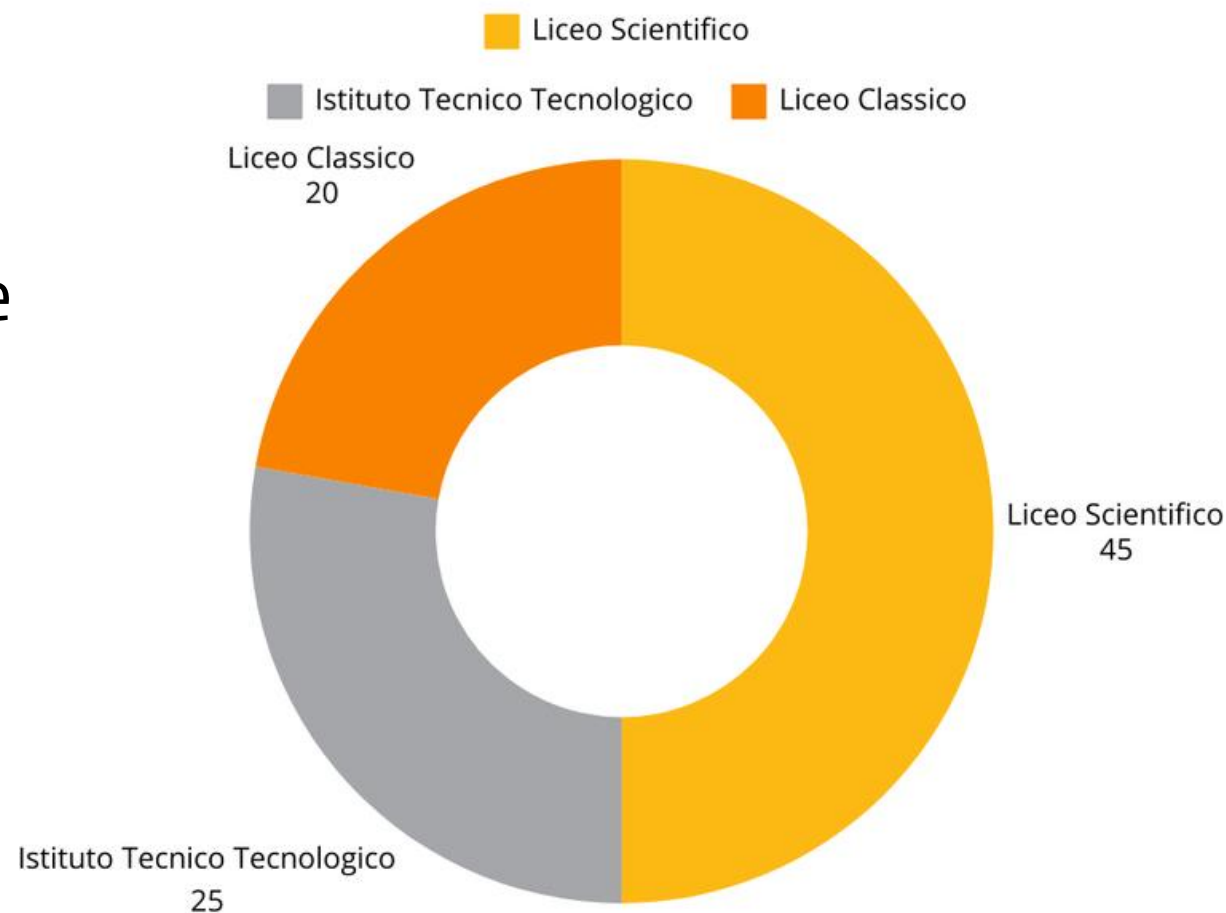
Modulo #3 Programmazione di un gioco in Python con Pygame

- 5h (5h coding)



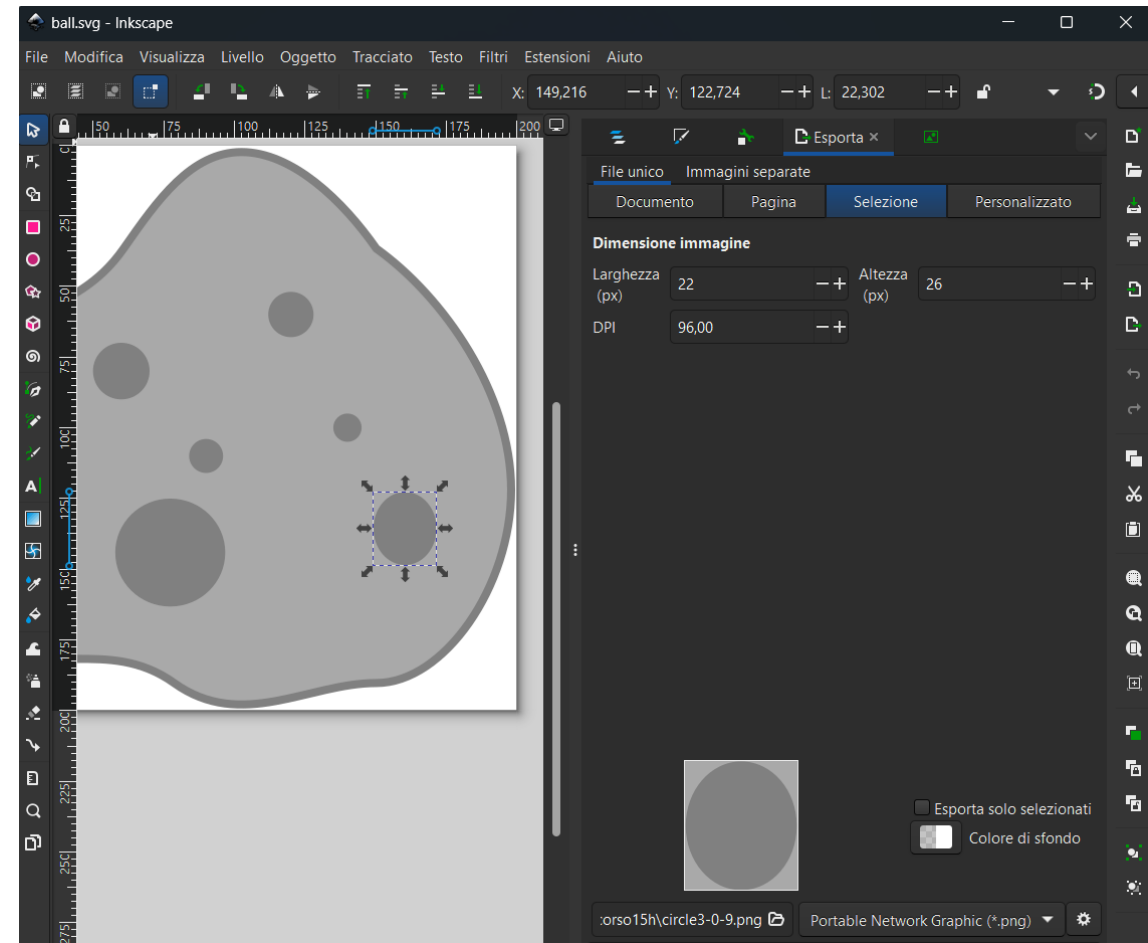
Alcuni numeri

- 5 istituti coinvolti
- 90 studenti/studentesse coinvolti/e
- Per ogni «mini-corso»:
 - 5h di lezioni frontali
 - 10h di laboratorio
- Classi 1[^], 3[^], 4[^] e 5[^]
- 100% in presenza
- Frequenza >90%



Modulo #1 - Introduzione alla UI e UX con cenni di game design

- Sono stati introdotti i concetti base del game design, enfatizzando l'importanza di creare esperienze di gioco fluide e gratificanti.
- Gli studenti hanno scoperto come le decisioni di design possono influenzare l'interazione del giocatore con il gioco e come un buon design possa migliorare l'immersività e l'engagement del gioco stesso.
- **In laboratorio:** apprendimento base di *Inkscape*, un potente strumento di grafica vettoriale open-source.



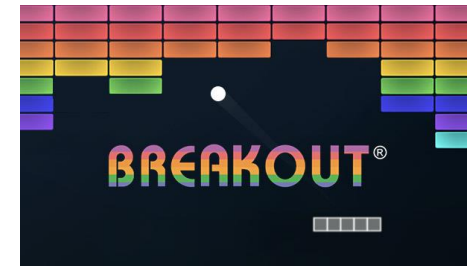
Modulo #2 Audio e post-processamento

- Gli **studenti** hanno sviluppato una **fruizione** più **consapevole** del **sonoro** attraverso **ascolto critico** per discriminare **elementi musicali** ed **effetti speciali** e riconoscerne l'**uso** da un punto di vista **emotivo** o come **vettore** di **informazione** e **contesto**.
- È stato chiesto di **sonorizzare** il **gameplay** di un **videogioco**.
- Gli studenti hanno **selezionato** un **breve clip video** tratto da un **videogioco**, al quale è stato tolto l'audio originale. Per **stimolare** la **creatività** gli studenti sono poi stati divisi in **gruppi**; hanno **creato** a partire da **semplici oggetti quotidiani** o attraverso la **vocalizzazione** e l'**uso del corpo suoni** da integrare nel gioco.



Modulo #3 Programmazione di un gioco in Python con Pygame

- La **fase finale** del corso `e stata dedicata alla **programmazione**, utilizzando **Python** e il framework **Pygame** per creare un gioco simile ad Atari Breakout con un approccio ispirato al Necessity Learning Design.
- Questa fase ha rappresentato il **culmine** delle **conoscenze teoriche** e **pratiche acquisite** nelle **fasi precedenti**.
- **Hanno** imparato a **implementare** la **fisica di base del gioco**, **gestire** semplici **collisioni** (pallina-parete) e **integrare** gli **effetti sonori contestuali** sviluppati nella fase precedente.



UNIVERSITÀ
POLITECNICA
DELLE MARCHE



[ITADINFO]

2° CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

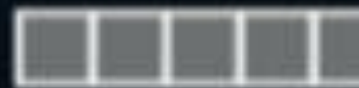
Problema...differenti livelli di conoscenze, abilità e competenze

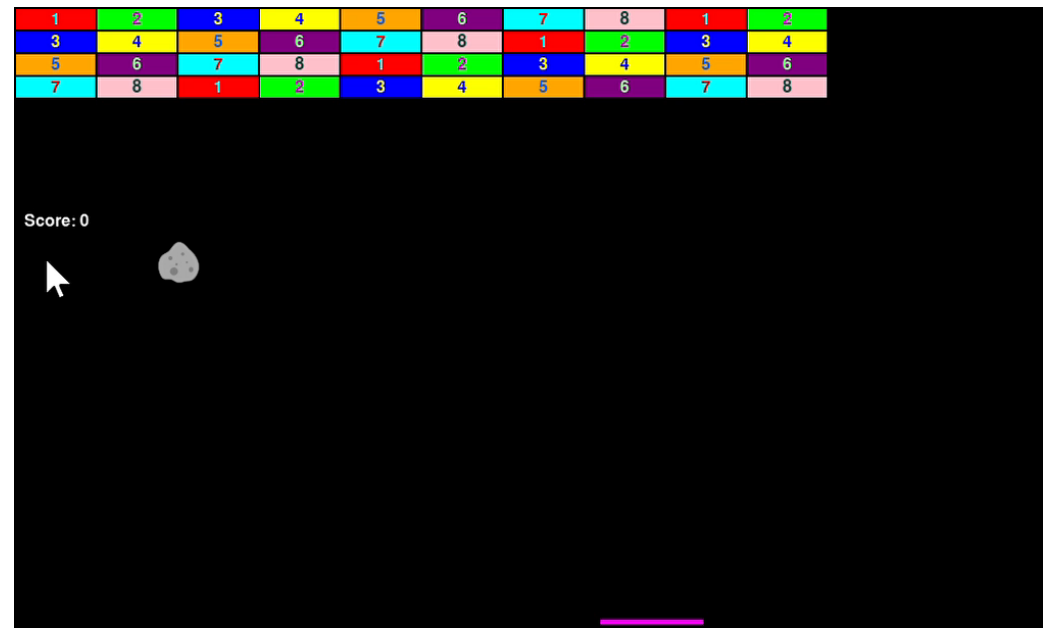
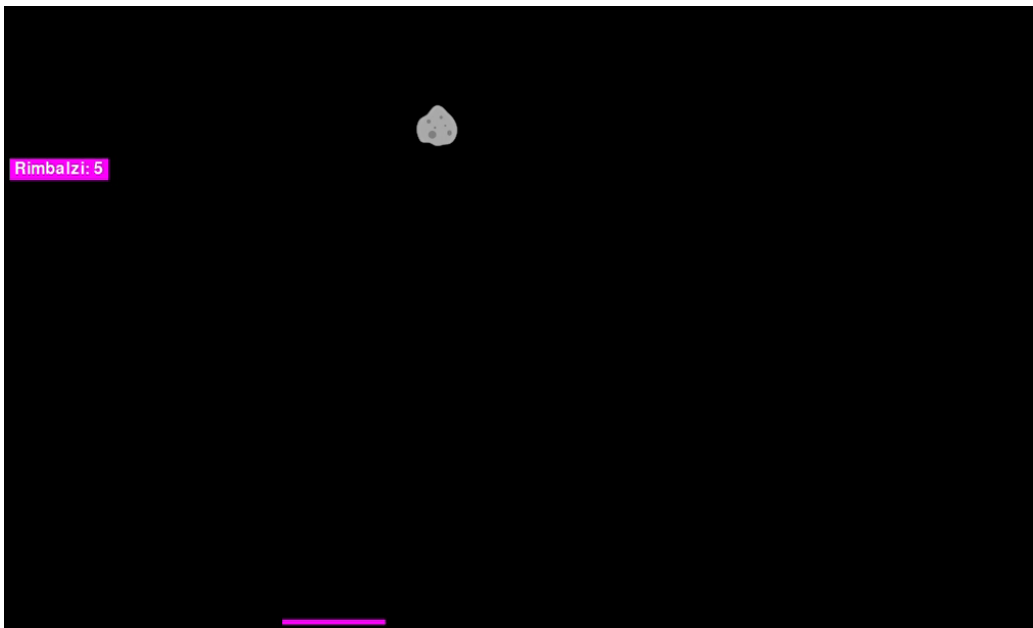
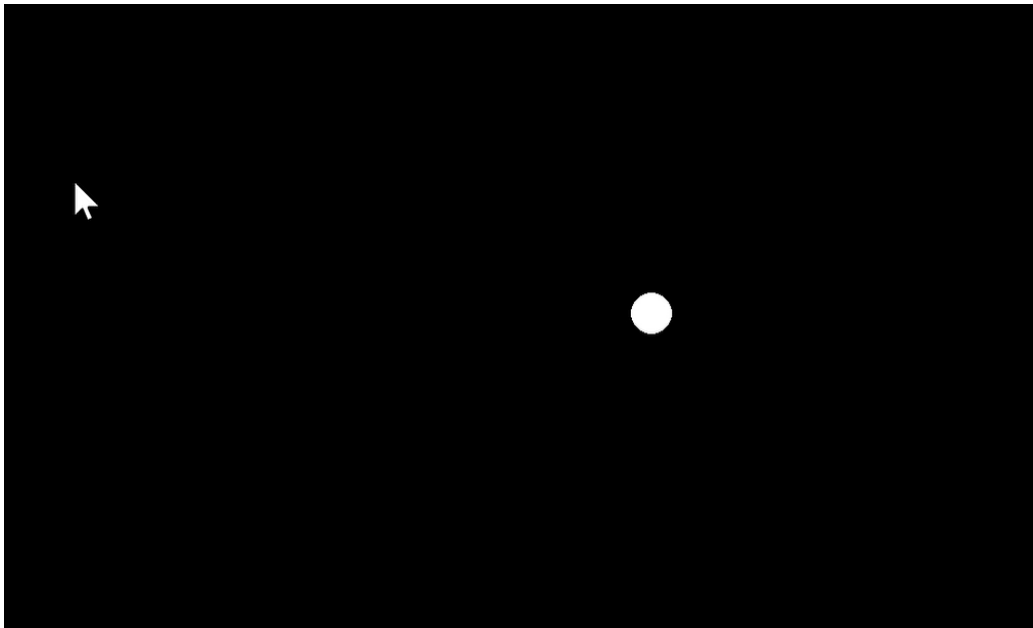
- Fase 0: warm-up Python
- Fase 1: introduzione a pygame...black screen
- Fase 2: primitive grafiche di disegno e sistemi di coordinate
- Fase 3: animazione di un cerchio e frame per second
- Fase 4: rimbalzo (`if-elif-else`)
- Fase 5: integrazione immagini + audio
- Fase 6: interazione uomo-macchina tramite arrow keys (`for`)
- Fase 7: logico di gioco
- Fase 8: logica avanzata di gioco



Gioco

BREAKOUT





Considerazioni

- Per gli studenti con livello di conoscenza di programmazione base sono stati forniti *snippet* di codice da «adattare»
- Lo sviluppo è stato guidato dal docente stimolando l'interazione con quesiti semplici volti a sbloccare lo sviluppo della «feature» successiva...
- Fase 0 – Fase 4 tutti
- Fase 5 – Fase 7 (scientifico + itt)
- Fase 8 (itt solo poche unità)

```
PLATFORM_HEIGHT = 5
platform_pos = [WIDTH//2 - PLATFORM_WIDTH//2, HEIGHT-PLATFORM_HEIGHT]
hit_wall_count = 0
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_r:
                background = RED
            elif event.key == pygame.K_g:
                background = GREEN

    screen.fill(background)
    ball_pos[0] += speed[0] # a += 1 --> a = a + 1
    ball_pos[1] += speed[1]
    if ball_pos[0] >= (WIDTH - BALL_RADIUS):
        speed[0] = -speed[0]
        hit_wall_count+=1
    if ball_pos[0] <= BALL_RADIUS:
        speed[0] = -speed[0]
        hit_wall_count+=1
    if ball_pos[1] >= (HEIGHT - BALL_RADIUS):
        speed[1] = -speed[1]
    if ball_pos[1] <= BALL_RADIUS:
        speed[1] = -speed[1]
        hit_wall_count+=1
    #disegno cerchio
    pygame.draw.circle(screen, WHITE, ( ball_pos[0], ball_pos[1] ), BALL_RADIUS)
```

```
hit_wall_count+=1
#disegno cerchio
pygame.draw.circle(screen, WHITE, ( ball_pos[0], ball_pos[1] ), BALL_RADIUS)
score_text = font.render(f"Hit: {hit_wall_count}", True, WHITE)
score_rect = score_text.get_rect(topleft=(10, 10))
#disegno rettangolo
pygame.draw.rect(screen, MAGENTA, score_rect.inflate(10, 5))
pygame.draw.rect(screen, MAGENTA, (int(platform_pos[0]), int(platform_pos[1]), PLATFORM_WIDTH, PLATFORM_HEIGHT))
screen.blit(score_text, score_rect)
pygame.display.flip()
clock.tick(30)
```



Competenze tecniche

- **Design grafico:** utilizzo di Inkscape per creare grafica vettoriale, progettazione di interfacce utente intuitive e visivamente accattivanti.
- **Acquisizione ed elaborazione numerica del suono:** tecniche di registrazione audio e possibilità creative fornite dai software utilizzati nel settore.
- **Programmazione:** conoscenze di base della programmazione in Python, utilizzo del framework Pygame per lo sviluppo di giochi, gestione delle collisioni e implementazione della fisica di base nel gioco.



Competenze Trasversali

- **Problem-solving:** abilità nel risolvere problemi complessi durante lo sviluppo del gioco, applicazione del pensiero logico e critico.
- **Collaborazione:** capacità di lavorare in gruppo, comunicare efficacemente con i compagni di squadra e condividere idee e soluzioni.
- **Creatività:** sviluppo della creatività attraverso la progettazione di interfacce, la creazione di grafica e la realizzazione di effetti sonori a partire dall'uso del corpo e da oggetti quotidiani.



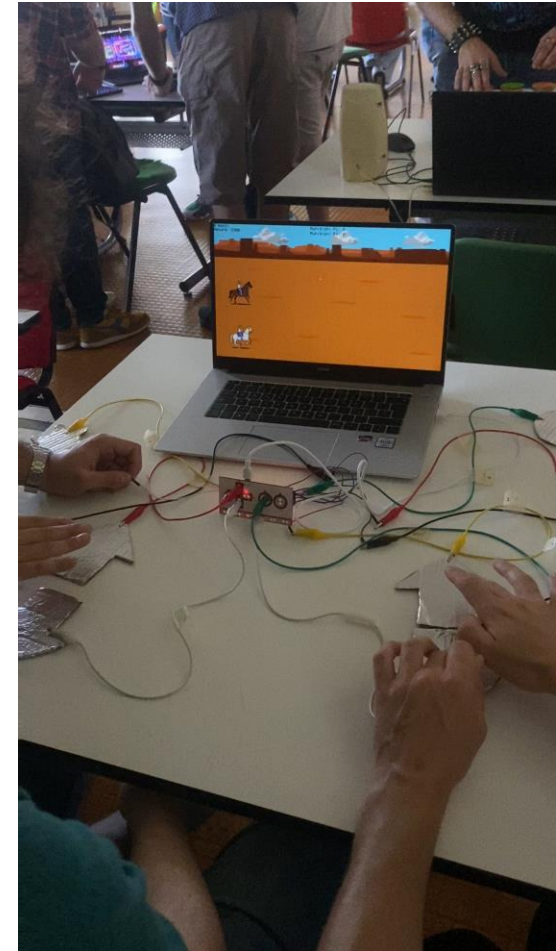
Conoscenze Teoriche

- **Principi di game design:** comprensione delle basi del design di videogiochi e dell'importanza della UI e UX.
- **Fondamenti di audio per videogiochi:** conoscenza degli aspetti chiave dell'audio nel game design e orientamento rispetto alle esistenti tecniche di acquisizione e post-processing.
- **Fondamenti di programmazione di videogiochi:** conoscenza di concetti base come rendering ed animazione di una scena, frame buffer e frame per second anche in ambiente Linux.



Considerazioni

- **15h sono poche** per un **tema** così **ampio** e **complesso**, ma.....la **conoscenza pregressa** di un **linguaggio** di **programmazione** si dimostra di **notevole utilità** aumentando il grado di **raggiungimento** degli **obiettivi iniziali**;
- vale la pena rimarcare comunque che l'utilizzo di un linguaggio come **Python** **semplifichi** il **processo** di **codifica**
- Sarebbe **ideale espandere** ed **approfondire diversi aspetti** durante le ore di lezione cercando di far progredire il progetto (es. interfaccia uomo-macchina con Arduino...)...**molti** infatti sono i **legami** con **diverse materie** (matematica, fisica, lingua straniera, ...)



Genova, 18-20 ottobre 2024

[ITADINFO]

**2° CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA**

**Come nasce un videogioco: aspetti di progettazione multimediale
audio-video ed interfacce**

**Grazie per l'attenzione !
Domande?**

Adriano Mancini a.mancini@staff.univpm.it