



# Laboratorio di Programmazione 1

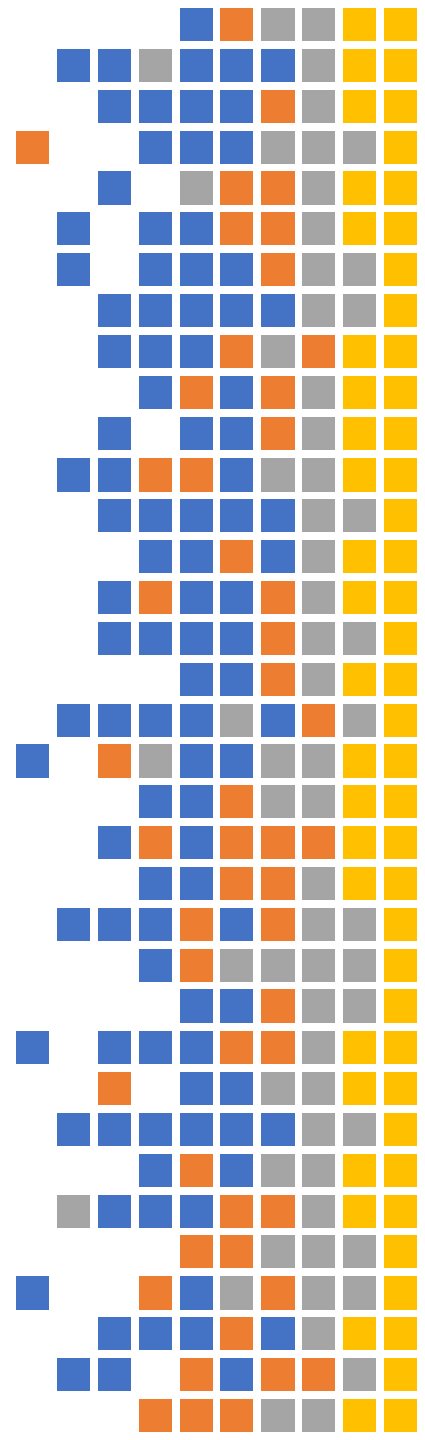
## CdL in Informatica – A.A. 2023 – 2024

Progetto finale  
Specifiche

**Federico Meloni**

Andrea Loddo

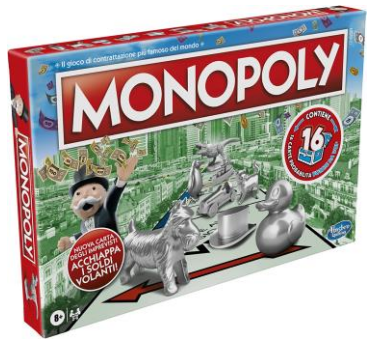
Luca Zedda



# Il progetto negli anni

Storicamente il progetto di PR1 prevede un gioco di carte.

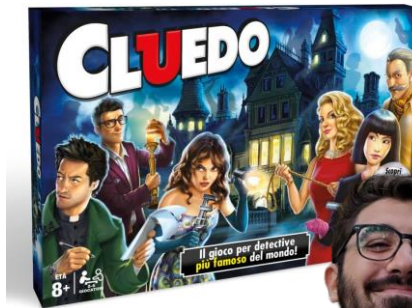
Quest'anno non fa eccezione!



2016



2017



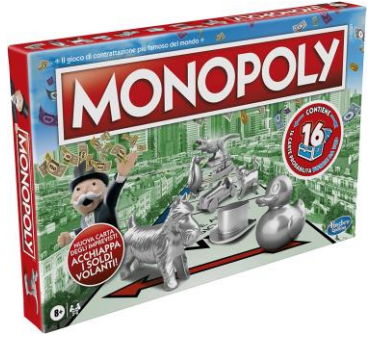
2018



2019



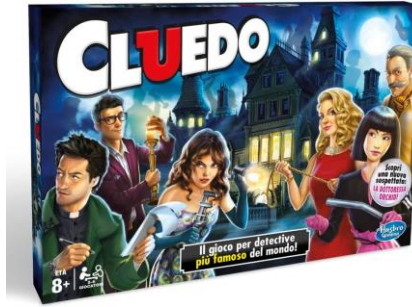
2023



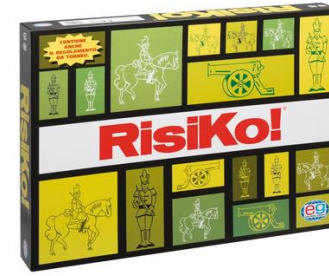
2016



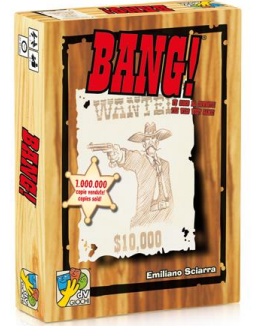
2017



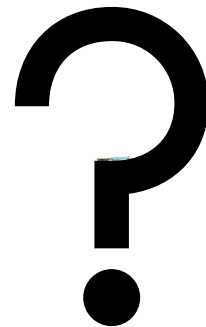
2018



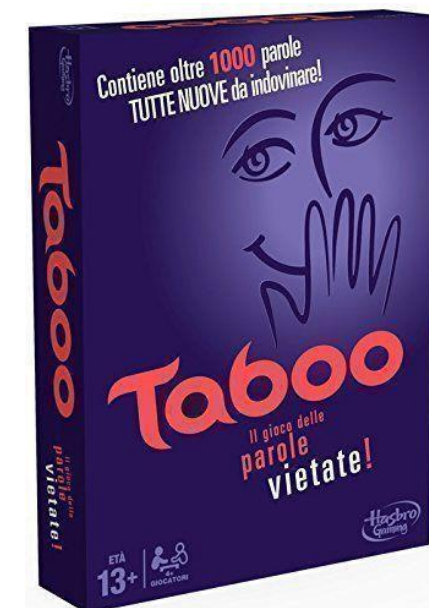
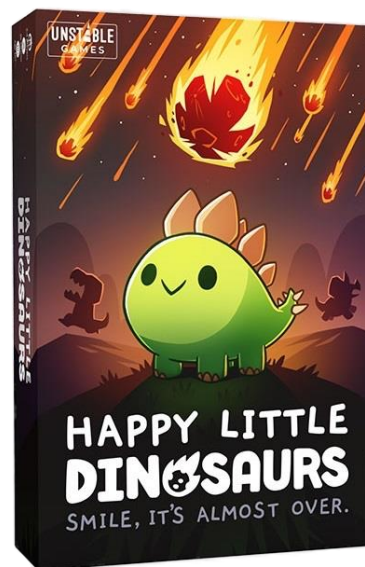
2019



2023

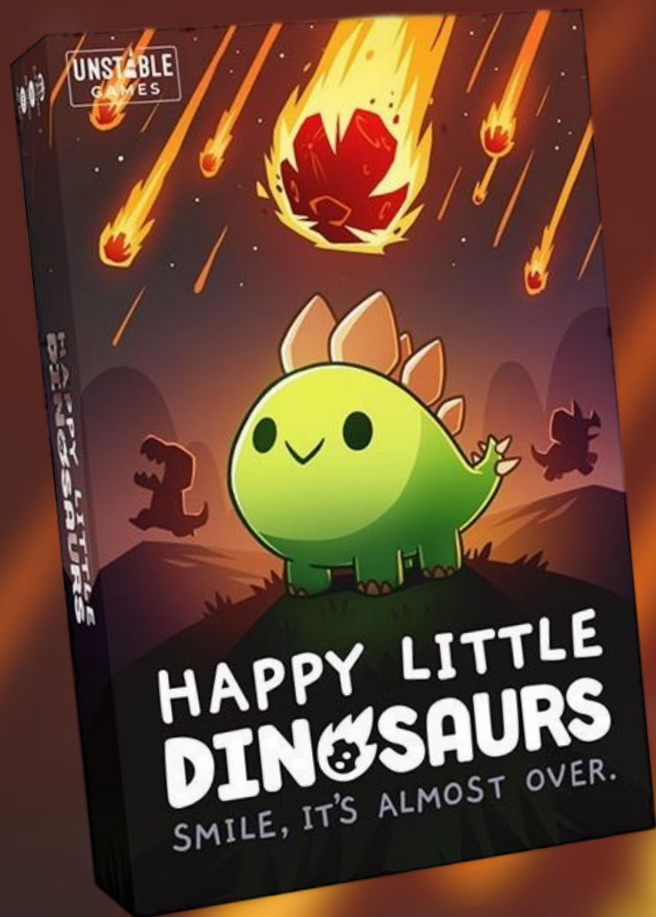


2024



Il progetto 2024 lo deciderete  
voi tra questi!





# HAPPY LITTLE DINOSAURS

Ottima scelta!



# CdL in Informatica - A.A. 2023 - 2024

## Laboratorio di Programmazione 1

### Progetto finale:



**Federico Meloni**

**Andrea Loddo**

**Luca Zedda**



# Organizzazione del progetto

---

- Il progetto è suddiviso in tre parti:
  - funzionalità BASE (20/30) consegnabile fino alle 23:59 del 23 maggio.
  - funzionalità MEDIE (30/30) consegnabile fino alle 23:59 del 31 luglio.
  - funzionalità AVANZATE (36/30) consegnabile fino alle 23:59 del 31 luglio.
- Il progetto può essere realizzato solo in modo **incrementale**: non possono essere implementate le funzionalità base e quelle avanzate tralasciando quelle medie.
- **Sufficienza**: voto  $\geq 15$
- Dopo la consegna del progetto, implementato nella difficoltà scelta, **si dovrà sostenere un colloquio orale**.
- Data e ora del colloquio verranno gestite a blocchi di consegna e/o concordati con lo studente.



# Organizzazione del progetto

---

- Il progetto è suddiviso in tre parti:
  - funzionalità BASE (20/30) consegnabile fino alle 23:59 del 23 maggio.
  - funzionalità MEDIE (30/30) consegnabile fino alle 23:59 del 31 luglio.
  - funzionalità AVANZATE (36/30) consegnabile fino alle 23:59 del 31 luglio.
- Deadline per non avere malus: 23:59 del 29 Febbraio 2024
- Dopo la 1<sup>a</sup> deadline: -0,4 per ogni settimana di ritardo sul voto:
  - ogni venerdì, a partire dal 1 marzo, verrà sottratto un malus di 0,4 dal voto del progetto.
- Tutto il progetto deve rispettare lo standard C99.
- È possibile utilizzare esclusivamente le librerie e i costrutti visti a lezione.





# Consegna

---

La consegna dovrà essere fatta tramite l'attività apposita su Teams corrispondente alla tipologia di progetto che avete scelto.

Le prime tre righe del file main dovranno rispettare il seguente formato:

//Nome: Austin Reaves ([a.reaves15@studenti.unica.it](mailto:a.reaves15@studenti.unica.it))

//Matricola: 60/61/66015

//Tipologia progetto: avanzato

Dopo la consegna attendete la convocazione per il colloquio orale.



# Deadline speciali

---

Abbiamo previsto deadline speciali per garantire correzione utile in concomitanza con date specifiche del vostro anno accademico:

- **31/05/2024:** per autocertificazione 20 CFU borsa di studio (30/06/2024)
- **30/06/2024:** per autocertificazione 20 CFU borsa di studio (10/08/2024)
- **07/05/2024:** per appello ASD (07/06/2024)
- **20/05/2024:** per appello ASD (20/06/2024)
- **08/06/2024:** per appello ASD (08/07/2024)
- **31/07/2024:** per appello ASD (11/09/2024)

Precisare nel form di consegna se avete tali necessità.



# Il gioco originale

Il gioco prevede 2-4 giocatori con:



Plance rappresentanti il giocatore



Carte giocabili



Carte disastro



# Il gioco originale

Il gioco prevede 2-4 giocatori con:



Plance rappresentanti il giocatore



Carte giocabili istantanee



Carte giocabili punto



Carte disastro





# Il gioco originale: la plancia



Ogni plancia rappresenta un dinosauro con:

- la sua personalità che darà malus o bonus in base al disastro estratto
- il contatore di punti
- un area dove accumulare disastri



# Il gioco originale: le carte giocabili

Carte punto



Le carte punto devono essere giocate ad ogni turno per evitare di prendere il disastro. Alcune di queste carte possono avere un effetto!

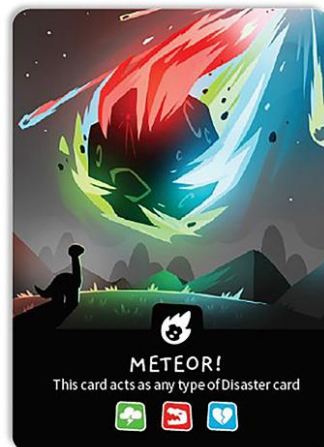
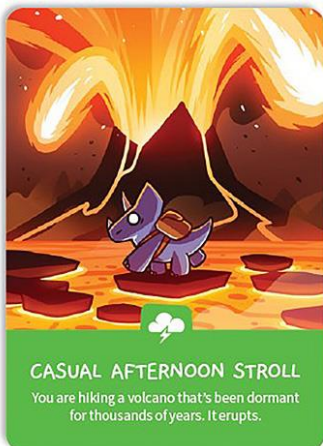
Carte istantanee



Le carte istantanee possono essere giocate durante o dopo (in base alla carta) il calcolo del punteggio per alterarlo o salvarsi da un disastro.



# Il gioco originale: le carte disastro



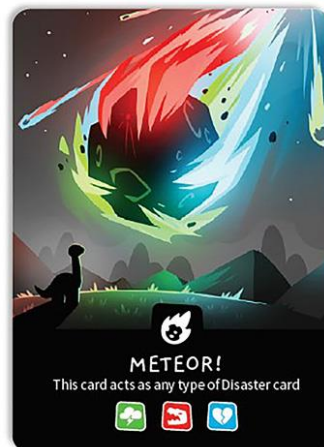
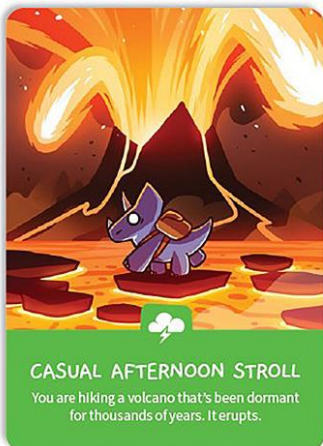
Le carte ostacolo possono essere di 3 tipi + meteorite che vale come jolly

Ad ogni turno si competerà per non prendere la carte disastro perché in base a quante se ne accumulano si potrebbe essere eliminati.





# Il gioco originale: le carte disastro



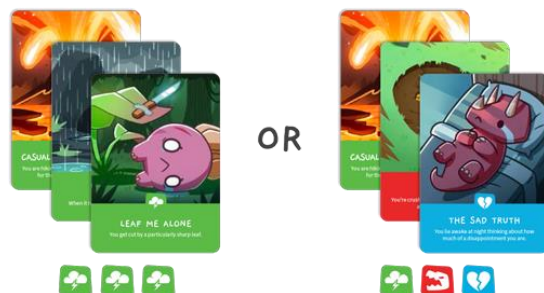
Le carte ostacolo possono essere di 3 tipi + meteorite che vale come jolly

Ad ogni turno si competerà per non prendere la carte disastro perché in base a quante se ne accumulano si potrebbe essere eliminati.

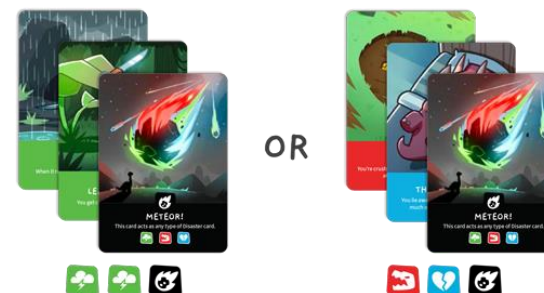
Si viene eliminati se accade una di queste due cose:

- si hanno tutti i colori
- si accumulano 3 carte dello stesso colore

Le carte Meteorite valgono come jolly, quindi un colore qualsiasi.



Esempio senza meteoriti



Esempio con meteoriti





# Il gioco originale: punti e scopo del gioco

---

Lo scopo del gioco è arrivare a 50 punti o rimanere l'ultimo dinosauro in vita.

Tutti i giocatori che risultano vincitori (vedi di più nella slide riguardante i pareggi) avanzano del tanto di punti che gli ha permesso di vincere il turno.

A fine turno, tutti avanzano del tanto di punti pari a quante carte disastro hanno accumulato nei turni precedenti (compreso quello appena finito).

In caso di più giocatori che arrivano a 50 punti nello stesso turno ci sarà una vittoria ex aequo.



# Il gioco originale: il turno

Tutti i giocatori prendono una  
plancia e si mescolano i mazzi





# Il gioco originale: il turno

Ad ogni giocatore vengono date 5 carte





# Il gioco originale: il turno

Viene estratta una carta disastro







# Il gioco originale: il turno

Tutti i giocatori giocano una carta tenendola coperta





# Il gioco originale: il turno

Contemporaneamente tutti  
girano la carta che hanno giocato





# Il gioco originale: il turno

Si fa un primo conto dei punti, tenendo conto di bonus/malus personaggio



CFU: 4



CFU: 7



CFU: 8-1



CFU: 2+1







# Il gioco originale: il turno

Si eseguono le azioni delle carte punto



CFU: 4



CFU: 7



CFU: 8-1



CFU: 2+1







# Il gioco originale: il turno

In questo momento si possono giocare le carte istantanee



CFU: 4



CFU: 7



CFU: 8-1+2



CFU: 2+1





# Il gioco originale: il turno

Si individua un vincitore e un perdente, quindi si assegnano punti e carta disastro. Gli altri rimangono fermi.



CFU: 4

CFU: 7



CFU: 8-1+2

CFU: 2+1



+1 CFU  
per avere un ostacolo

+9 CFU  
per vittoria turno



# Il gioco originale: il turno

Si pesca il tanto di carte per tornare a 5 carte in mano



CFU: 4

CFU: 7



CFU: 8-1+2

CFU: 2+1







# Il gioco originale: risoluzione dei pareggi

---

Cosa succede se due (o più) giocatori sono in **pareggio**?

Se il pareggio è **per la vittoria** tutti i “pareggianti” si considerano come vincitori ed avanzano.

Se il pareggio è **tra perdenti** si fa un turno supplementare, **senza bonus/malus ne effetti di alcun tipo**, finché non si decreta un perdente. **Non sono previsti pescaggi durante i turni supplementari.**

I turni supplementari vanno **ad oltranza** finché non ci sarà un perdente unico.

Se ci si ritrova nella situazione di non poter giocare carte si **perde in automatico** il turno supplementare.

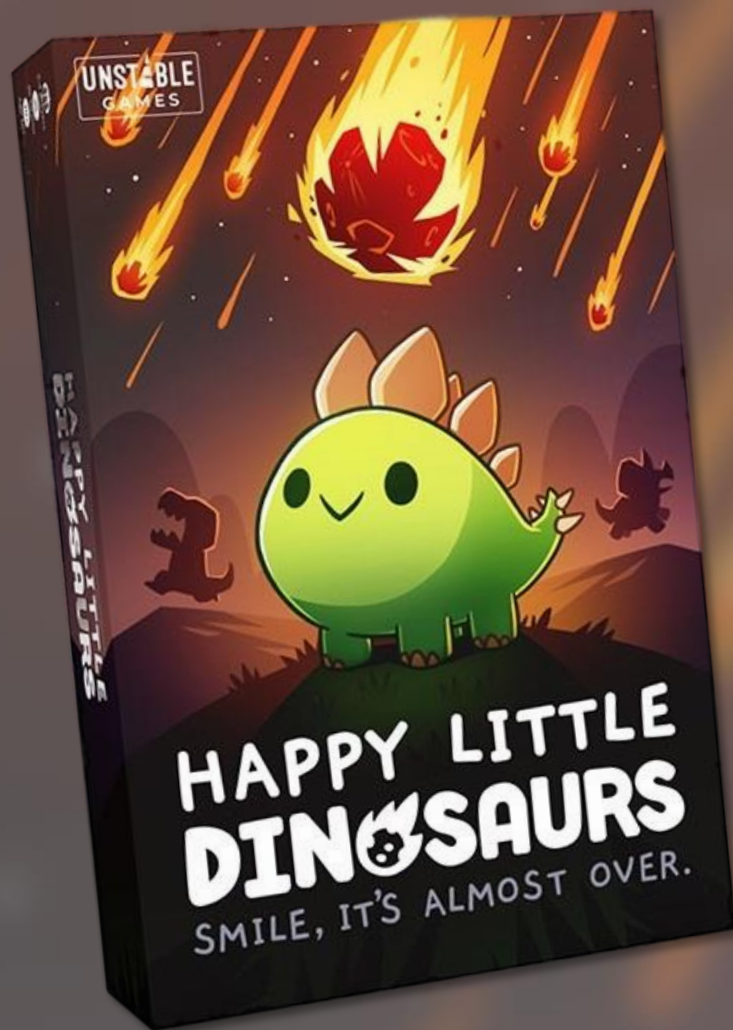
Se entrambi i giocatori non possono giocare carte **non ci sono perdenti** ed il disastro torna nel mazzo.

Se **tutti i giocatori sono in pareggio** non ci sono né vincitori né vinti, quindi il disastro torna nel mazzo.



# Il vostro progetto: Happy Little Students

---



## HAPPY LITTLE STUDENTS





# Il vostro progetto: Happy Little Students

---

Voi non avrete Happy Little Dinosaurs, ma Happy Little Students.  
Le regole sono le stesse, ma i punti verranno sostituiti dai CFU.

Quindi carte punto e istantanee verranno chiamate carte CFU punto e CFU istantanee.  
L'obiettivo diventa ottenere 60 CFU per finire il primo anno accademico.

La morte viene sostituita dalla rinuncia agli studi.  
Altro modo per vincere sarà rimanere l'ultimo studente a non aver fatto rinuncia agli studi.

Le carte diventano a tema vita universitaria, con riferimenti al vostro anno da matricola.  
Le carte disastro diventeranno carte di tipo ostacolo STUDIO, SOPRAVVIVENZA e SOCIALE,  
con i Meteoriti che diventano gli ESAMI.



# Happy Little Students: le carte GIOCABILI

Ci saranno delle carte CFU punto che non hanno effetti:

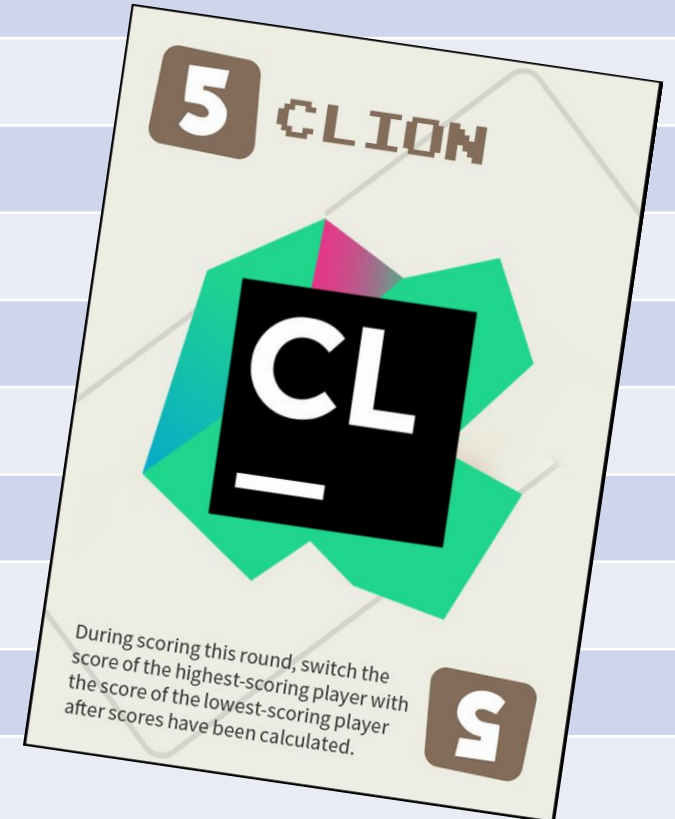
Nº Carte	Effetto [Enumerazione]	Valore	Nome
4	0 [NESSUNO]	1	Mouse
4	0 [NESSUNO]	2	Cavo Ethernet
4	0 [NESSUNO]	3	Esse3
4	0 [NESSUNO]	4	Unicamente
4	0 [NESSUNO]	5	DevC++
4	0 [NESSUNO]	6	Cappellino Natale
4	0 [NESSUNO]	7	Dispense Teoria
4	0 [NESSUNO]	8	Laboratorio T
4	0 [NESSUNO]	9	InfoLife



# Happy Little Students: le carte GIOCABILI

Per ogni carta CFU senza effetto ce ne sarà una corrispondente con un effetto:

Nº Carte	Effetto [Enumerazione]	Valore	Nome
4	1 [SCARTAP]	0	Monitor
2	2 [RUBA]	1	Tastiera
2	3 [SCAMBIADS]	2	Cavo USB
2	4 [SCARTAE]	3	MS Teams
2	5 [SCARTAC]	4	Eduroam
2	6 [SCAMBIAP]	5	CLion
2	7 [DOPPIOE]	6	Maglione Natale
2	8 [SBIRCIA]	7	Slide Lab
2	9 [SCAMBIAC]	8	Aula Magna Fisica
2	10 [ANNULLA]	9	BatCaverna







# Happy Little Students: le carte GIOCABILI

Le ultime carte giocabili sono quelle istantanee.

Queste si possono giocare durante il calcolo del punteggio:

Nº Carte	Effetto [Enumerazione]	Valore	Nome
5	11 [AUMENTA]	0	Usa ChatGPT
5	12 [DIMINUISCI]	0	Copia dal collega
2	13 [INVERTI]	0	Scambio di esame

Queste carte istantanee si possono giocare quando si perde un turno:

Nº Carte	Effetto [Enumerazione]	Valore	Nome
3	14 [SALVA]	0	Esame di recupero
1	15 [DIROTTA]	0	Snitcha il collega



# Happy Little Students: GLI EFFETTI [1/2]

Per gli effetti si fa riferimento alla colonna EFFETTO delle precedenti tabelle, che riporta il valore che dovrà avere l'enumerazione e nome dell'enumerazione.

Effetto [Enumerazione]	Descrizione effetto
1 [SCARTAP]	Scarta una carta CFU punto e aggiungi il suo punteggio a quello del turno
2 [RUBA]	Guarda la mano di un collega e ruba una carta a scelta.
3 [SCAMBIADS]	Scambia questa carta con quella di un altro giocatore, purché senza effetto
4 [SCARTAE]	Scarta una carta CFU punto con effetto e aggiungi il suo punteggio a quello del turno
5 [SCARTAC]	Scarta da uno a tre carte dalla tua mano
6 [SCAMBIAP]	Scambia il punteggio del turno maggiore e minore dopo il calcolo del punteggio di base
7 [DOPPIOE]	Raddoppia gli effetti delle carte che aumentano o diminuiscono il punteggio (per tutti)
8 [SBIRCIA]	Guarda due carte in cima al mazzo, prendine una e scarta l'altra
9 [SCAMBIAC]	Scambia la carta punto giocata nel turno da un giocatore G1 con quella di un giocatore G2, con G1 e G2 scelti dal giocatore che ha giocato la carta SCAMBIAC
10 [ANNULLA]	Annulla gli effetti di tutte le carte punto durante il turno



# Happy Little Students: GLI EFFETTI [2/2]

Per gli effetti si fa riferimento alla colonna EFFETTO delle precedenti tabelle, che riporta valore che dovrà avere l'enumerazione e nome dell'enumerazione.

Effetto [Enumerazione]	Nome
11 [AUMENTA]	Aumenta di 2 CFU il punteggio del turno di un giocatore a scelta
12 [DIMINUISCI]	Diminuisce di 2 CFU il punteggio del turno di un giocatore a scelta
13 [INVERTI]	Inverti punteggio minimo e massimo del turno
14 [SALVA]	Metti la carta Ostacolo che stai per prendere in fondo al mazzo
15 [DIROTTA]	Dai la carta che stai per prendere ad un altro giocatore a tua scelta
0 [NESSUNO]	Nessun effetto



# Happy Little Students: GLI EFFETTI FAQ

---

Alcune domande che sicuramente ci saranno:

- **In che ordine vanno eseguiti gli effetti delle carte punto?**  
Vanno eseguiti in ordine decrescente del valore della carta
- **Si possono giocare carte CFU istantanee durante gli effetti delle carte CFU punto?**  
No, sono momenti separati del turno
- **Posso usare la carta 'Esame di recupero' quando mi giocano contro 'Snitcha il collega'?**  
Sì
- **C'è un ordine in cui i giocatori devono giocare le carte CFU istantanee?**  
No, dovrete chiedere se qualcuno vuole giocare (e chi) una carta istantanea
- **Un giocatore può giocare più carte istantanee nello stesso turno? Anche non consecutive?**  
Sì e sì





# Happy Little Students: le carte

Le carte disastro diventano carte ostacolo: tipologia STUDIO

Nome	Descrizione
Perdi tutto	Prof. Loddo chiede se i PC vanno e il tuo esplode perdendo il parziale
Sbiccato copiare	Federico passa tra i banchi e ti becca con Discord aperto
Maledetto Lab T	I PC del laboratorio non partono e il parziale inizia con 45 minuti di ritardo
Debugging infinito	Dopo ore di debugging, scopri che il problema era una parentesi mancante
Troppo veloce	Ti distrai mezzo secondo per guardare TikTok e la Prof. Carrisi ha spiegato gli integrali tripli
Va tutto bene?	Prof. Puglisi ti chiede se va tutto bene, ma non va tutto bene
Printf scurrili	Consegna il progetto di PR1 e scopri che hai lasciato un printf con una parolaccia
Tutti fermi	Durante PR1 parte la musica di Pokemon Smeraldo, Luca si ferma e tutti si girano a guardarti



# Happy Little Students: le carte

Le carte disastro diventano carte ostacolo: tipologia SOPRAVVIVENZA

Nome	Descrizione
Una fila infinita	Arrivi in mensa in via Trentino e c'è una fila chilometrica
Macchinetta ladra	La macchinetta del piano terra ti prende i soldi ma non ti dà nulla
Sciopero del CTM	Il CTM è in sciopero e non puoi tornare a casa
Rotta della caffeina	La macchinetta del caffè è fuori servizio e devi affrontare la giornata senza caffeina
Mauro infuriato	Durante la pausa pranzo sposti i puff e Mauro ti urla contro
Intossicazione alimentare	In mensa prendi le polpette e torni a casa con un colorito strano
Batteria scarica	Torni a casa e ti accorgi che ti sei dimenticato il caricabatterie in aula magna
Metti il caschetto	Stai seguendo la lezione e crolla un calcinaccio dal soffitto



# Happy Little Students: le carte

Le carte disastro diventano carte ostacolo: tipologia SOCIALE

Nome	Descrizione
Festa in casa	I tuoi coinquilini organizzano una festa in casa e non riesci a studiare per il parziale di PR1
Doppio appuntamento	Inviti il tuo collega a studiare, ma lui porta anche il suo gruppo di amici (ingegneri)
Ateneika fallimentare	Ti iscrivi al torneo di biliardino di Ateneika e arrivi ultimo (e neanche di poco)
Le priorità della vita	Vorresti studiare per il parziale di PR1 ma esce Fortnite OG
Ignorato come sempre	Arrivi in biblioteca di ingegneria e i tuoi colleghi non ti hanno tenuto il posto
Parcheggio not found	Non trovi parcheggio e ti tocca mettere le macchina a Buon Cammino
Nessuno mi vuole	Compri il gioco di società Happy Little Dinosaurs ma nessuno vuole giocare con te
Molletta per il naso	Vai in bagno durante la pausa di PR1 e trovi fila con il bagno che puzza moltissimo



# Happy Little Students: le carte

---

Le carte disastro diventano carte ostacolo: tipologia ESAME

Nome	Descrizione
PR1 Teoria	Prof. Scateni scatena la sua furia e ti bocchia al parziale di PR1
PR1 Lab	Il progetto di PR1 non compila e non riesci a consegnarlo
Analisi	Sbagli e vai all'esame di Analisi di quelli di Matematica
Fondamenti	Questa volta l'intero esame è scritto in binario





# IMPLEMENTAZIONE

Regole obbligatorie:

- Standard C99
- Suddivisione in più file .c/.h, in base alle funzionalità implementate in quel file

Sono ammesse strutture ausiliarie aggiuntive, purché non modifichino le strutture richieste e i file di salvataggio.



# Le strutture: il Giocatore

---

Per ogni giocatore sarà necessario salvare le seguenti informazioni:

- Nome utente (32 caratteri, compreso terminatore)
- Struttura personaggio (vedi slide 46)
- Il numero di CFU accumulati
- Una lista di carte CFU
- Una lista di carte ostacolo
- Puntatore al prossimo giocatore

I CAMPI DEL GIOCATORE VANNO DICHIARATI IN QUEST'ORDINE PER COMPATIBILITÀ CON I NOSTRI SALVATAGGI



# Le strutture: Carte CFU

---

Per ogni carta sarà necessario salvare le seguenti informazioni:

- Nome della carta (32 caratteri, compreso terminatore)
- Il numero di CFU della carta
- Enumerazione effetto carta (vedi slide 34, 35, 45)
- Puntatore alla prossima carta

I CAMPI DELLA CARTA VANNO DICHIARATI IN QUEST'ORDINE PER COMPATIBILITÀ CON I NOSTRI SALVATAGGI



# Le strutture: Carte Ostacolo

---

Per ogni carta sarà necessario salvare le seguenti informazioni:

- Nome della carta (32 caratteri, compreso terminatore)
- Descrizione della carta (128 caratteri, compreso terminatore)
- Enumerazione con il tipo di ostacolo:
  - 1} STUDIO
  - 2) SOPRAVVIVENZA
  - 3) SOCIALE
  - 4) ESAME
- Puntatore alla prossima carta

I CAMPI DELLA CARTA VANNO DICHIARATI IN QUEST'ORDINE PER COMPATIBILITÀ CON I NOSTRI SALVATAGGI





# Le strutture: Effetti delle carte

---

Sarà necessario avere un enumerazione rappresentante gli effetti delle carte:

- VEDERE SLIDE N.34/35 per nomi e valori enumerazione



# Le strutture: il personaggio

---

Per ogni personaggio sarà necessario salvare le seguenti informazioni:

- Nome del personaggio (32 caratteri, compreso terminatore)
- Un array di interi di dimensione pari al numero di tipologie di ostacoli
  - All'interno ci saranno salvati bonus/malus con corrispondenza biunivoca tra enumerazione e indice

I CAMPI DEL MAZZO VANNO DICHIARATI IN QUEST'ORDINE PER COMPATIBILITÀ CON I NOSTRI SALVATAGGI



# Fasi di gioco: fase 1 — preparazione

---

- Si caricano i personaggi dal file **giocatori.txt** e i mazzi da **carte.txt** e **ostacoli.txt**
- Preso in input il numero di giocatori ed i nomi di ognuno, si assegna loro un personaggio casuale.
  - Ad ogni giocatore deve essere assegnato un personaggio diverso.
- Vengono mescolati i mazzi contenuti carte CFU e carte Ostacolo.
- Ogni giocatore parte con:
  - 0 CFU
  - 0 carte ostacolo
  - 5 carte CFU (che vengono distribuite dopo la mescolatura)
- A questo punto il gioco può iniziare



## Fasi di gioco: fase 2 – turno

---

Prima dell'inizio di ogni turno va salvata la partita nell'apposito file binario di salvataggio (*vedi slide 58*).

**All'inizio di ogni turno** si dovrà stampare il numero del turno, dopodiché si dovrà estrarre una carta ostacolo e si dovrà stampare il suo contenuto.

Durante il turno i giocatori potranno, a turno, eseguire le seguenti azioni:

[TASTO 1] giocare una carta CFU dalla propria mano

[TASTO 2] controllare lo stato degli altri giocatori (CFU e carte ostacolo)

[TASTO 0] uscire dalla partita

Di default tutte le informazioni relative al giocatore che deve effettuare la scelta sono visualizzate a schermo (nome, info personaggio, carte etc...)





## Fasi di gioco: fase 2 – turno

---

Se al momento della scelta della carta da giocare un giocatore non ha carte CFU punto, scarta tutte le carte CFU in suo possesso finché non avrà almeno una carta CFU punto da giocare.

Se le carte dal mazzo di pesca non sono sufficienti a permettere il pescaggio, il mazzo degli scarti diventerà il nuovo mazzo di pesca, dopo apposito rimescolamento.

Dopo che ogni giocatore ha scelto una carta da giocare, può iniziare la fase di calcolo del punteggio.



# Fasi di gioco: calcolo del punteggio

---

Una volta che tutti i giocatori hanno scelto una carta inizia il calcolo del punteggio del turno.

Al punteggio di ogni giocatore vengono applicati i bonus/malus rispetto alla personalità del personaggio assegnato al giocatore.

Dopodiché vengono applicati gli effetti delle carte CFU in ordine decrescente rispetto al valore della carta.



## Fasi di gioco: calcolo del punteggio

---

Una volta esauriti gli effetti delle carte CFU i giocatori hanno la possibilità di giocare le loro carte CFU istantanee.

Dato che non c'è un ordine predefinito di gioco in questa fase si dovrà prima chiedere chi vuole giocare e poi far giocare una carta a quel giocatore. Se non ha carte CFU istantanee si nega l'azione.

Finita questa fase si passa all'eventuale risoluzione dei pareggi e all'assegnazione dei punti.



# Fasi di gioco: risoluzione dei pareggi ed assegnazione punti

---

Una volta calcolati i punteggi bisogna individuare un perdente e almeno un vincitore per il turno.

Se c'è un pareggio per il punteggio più alto tutti i giocatori in stato di pareggio risultano vincitori del turno.

Se c'è un pareggio per il punteggio più basso viene effettuato un turno aggiuntivo solo tra i giocatori che stanno pareggiando.

- si possono utilizzare solo carte CFU punto
- qualsiasi effetto e/o modificatore, compresa personalità, vengono ignorati
- si continua ad oltranza finché non c'è un vincitore
- se un giocatore non ha carte CFU punto perde il turno
- se entrambi i giocatori non hanno carte CFU punto non ci sono perdenti





# Fasi di gioco: risoluzione dei pareggi ed assegnazione punti

---

Nel caso in cui tutti i giocatori alla fine del calcolo del punteggio stiano pareggiando non ci sono né vincitori né perdenti.

I giocatori risultanti vincitori avanzano del numero di CFU pari ai punti con cui hanno vinto il turno.

Il giocatore perdente prende la carta ostacolo. Nel caso in cui non ci siano perdenti la carta ostacolo viene messa in fondo al mazzo.

Alla fine del turno tutti i giocatori avanzano in CFU del numero di carte ostacolo accumulate e pescano il tanto di carte necessario per tornare a 5 nella propria mano.



## Fasi di gioco: fine del gioco

---

Vince la partita il primo studente che arriva a 60 CFU o l'ultimo che non ha fatto rinuncia agli studi.

Quando si ha almeno un vincitore della partita il gioco (ed il programma) finisce.

Se più giocatori arrivano a 60 CFU nello stesso turno ci sono più vincitori.

Per morire un giocatore deve soddisfare una delle seguenti condizioni:

- 3 carte ostacolo dello stesso tipo
- una carta ostacolo di ogni tipo

Quando un giocatore muore le sue carte vanno scartate e il giocatore viene deallocato. Le carte ostacolo in suo possesso vengono rimesse in fondo al mazzo.

Le carte ostacolo di tipo ESAME sono delle carte jolly, valgono contemporaneamente come tutti i tipi di ostacolo (vedi slide 16).



# Implementazione: progetto BASE (max 20pt)

---

Implementazioni da realizzare nelle differenti versioni:

- BASE:
  - Il **numero di giocatori** è deciso a **runtime** e viene chiesto un **username** ad ogni giocatore.
  - Bisogna **implementare tutta la logica del gioco**, senza tener conto degli eventuali effetti delle carte CFU e delle carte CFU istantanee.  
Le carte CFU con effetto saranno comunque presenti, ma l'effetto sarà ignorato.
  - Le informazioni vanno caricate dai file .txt forniti.
  - All'inizio di ogni turno si deve salvare la partita in un file binario 'savegame.sav' strutturato come mostrato in slide 58.
  - Questo salvataggio, se presente, deve poter essere caricato ad ogni avvio del programma.



# Implementazione: progetto MEDIO (max 30pt)

---

Implementazioni da realizzare nelle differenti versioni:

- MEDIO:

Specifiche BASE +

- Si devono implementare tutta la logica del gioco, tenendo conto degli effetti che le carte CFU possono avere durante il calcolo del punteggio e delle carte CFU istantanee.
- Si devono permettere salvataggi multipli permettendo all'utente di scegliere il nome del file di salvataggio (nel caso sia già presente verrà sovrascritto) e di conseguenza scegliere quale file caricare, gestendo gli eventuali casi di input inconsistenti da parte dell'utente.
- Generare un file log.txt in cui viene scritta la cronologia della partita

TURNO 1: Riccardo gioca un Clion

TURNO 1: Andrea gioca Tastiera

TURNO 1: Federico gioca Mouse

TURNO 2: Luca gioca Lab T

...





# Implementazione: progetto **AVANZATO** (max 36pt)

---

Implementazioni da realizzare nelle differenti versioni:

- **AVANZATO:**

Specifiche MEDIO +

- Dare la possibilità di eseguire la partita in modalità automatica, il tester dovrà solamente specificare il numero di giocatori e la partita avrà solo giocatori CPU, salvando ad ogni turno la partita.
- In questa modalità dovranno essere mostrati in chiaro tutti i dati della partita.
- Le scelte di ogni giocatore dovranno essere fatte in modo autonomo, seguendo comunque la logica del gioco. Ovvero: tutti i giocatori, comandati dal computer, dovranno cercare di vincere la partita.
- Maggiore sarà la fedeltà dell'IA rispetto al comportamento umano, più alta sarà la valutazione. Quindi: maggiore sarà la completa casualità nelle scelte, minore sarà il voto, maggiore sarà la logica (con un senso e una strategia dietro), maggiore sarà il voto.
- Dopo ogni partita aggiornare un file con le statistiche che permetta di stabilire il numero di vittorie di ogni personaggio, tenendo il conto separato tra modalità manuale e modalità automatica. Si possono aggiungere un numero a piacere di informazioni a questo file. Visualizzare tale file prima di ogni nuova partita. Gestire il caso in cui il file non venga trovato o non sia presente.
- È necessario consegnare un file di statistiche con dei dati già presenti insieme al progetto.



# File di salvataggio

---

Il file di salvataggio è un file binario "savegame.sav", che contiene:

- un intero N indicante il numero di giocatori partecipanti alla partita (da 2 a 4).
- Per ognuno degli N giocatori:
  - Un blocco rappresentante il giocatore.
  - Una lista rappresentante le carte in mano al giocatore
  - Un numero rappresentante il numero di carte ostacolo accumulate
  - Una lista rappresentante le carte ostacolo del giocatore
- Per ogni mazzo (in questo ordine CFU, CFU scarti, Ostacoli):
  - Un numero rappresentate il numero di carte nel mazzo
  - Una lista rappresentante le carte nel mazzo

È possibile memorizzare ulteriori informazioni a piacimento alla fine del file di salvataggio. Gestire il caso in cui queste non siano presenti.



# File da caricare

---

- I mazzi di carte e i personaggi si trovano nei file .txt forniti da noi, la struttura delle carte è data come segue.
- Carte CFU: la carta è salvata secondo lo schema, per esempio, **2 0 8 Laboratorio T**
  - in **rosso** troviamo le **occorrenze** della carta
  - in **blu** troviamo la **tipologia (vedere enumerazione)** della carta
  - in **viola** troviamo il **valore** della carta
  - in **verde** troviamo il **nome** della carta
- Carte Ostacolo: la carte sono salvate secondo uno schema a blocchi, ogni blocco corrisponde ad una categoria di carte, nell'ordine STUDIO, SOPRAVVIVENZA, SOCIALE, ESAME.  
In ogni blocco abbiamo:  
**numero\_carte**  
**nome\_carta1**  
**descrizione\_carta1**  
**nome\_carta2**  
...
  - Personaggi: i personaggi sono salvati secondo lo schema, per esempio, **0 -1 1 0 Federicosauro**
    - in **rosso** abbiamo **bonus e malus** per ogni categoria di ostacolo (stesso ordine dell'enumerazione)
    - in **verde** abbiamo il **nome** del personaggio



# BEST PRACTICES ai fini della valutazione

---

Buone norme di programmazione:

- Macro maiuscole
- Nomi delle variabili camelCase o snake\_case (scegliete uno stile e mantenetelo)
- Blocchi di codice non troppo lunghi (nel caso valutare la suddivisione in subroutine)
- Suddividere le varie funzionalità in apposite subroutine
- Suddividere le varie funzionalità in appositi file .c/.h
- No variabili globali
- Evitare break e continue
- Ricordare i controlli sui puntatori e di liberare la memoria in caso di allocazione dinamica (DI TUTTO QUELLO ALLOCATO)
- Ricordare il controlli sui file e di chiuderli sempre
- Seguire stile di scrittura e indentazioni consoni, come visto a lezione
- L'uso del goto sarà un goto a riseguire il corso nel 24/25



# BEST PRACTICES ai fini della valutazione

---

## Stile, indentazione e leggibilità:

- Rendere il proprio codice leggibile
- Indentare correttamente i vari blocchi di codice
- Utilizzare uno stile e seguirlo per tutto il progetto
- Utilizzare nomi per variabili, subroutine etc.. che abbiano un senso con quello che fanno
- Commentare tutti i passaggi non ovvi
- Prima dell'implementazione di ogni subroutine descriverla brevemente con un commento

Per ogni subroutine, utilizzare il seguente formato di commenti Doxygen (due asterischi in apertura di commento):

```
/** La funzione randRange restituisce un numero intero casuale
 * compreso tra min e max. Va chiamata la srand() nel main prima
 * della sua prima invocazione.
 *
 * @param min il numero minimo
 * @param max il numero massimo
 * @return un numero intero casuale compreso tra min e max
 */
int randRange(int min, int max) {
    return rand() % (max - min + 1) + min;
}
```





## F.A.Q.

---

D: Cosa rappresentano i 4 numeri prima del nome del personaggio nel file personaggi.txt?

R: Sono i bonus/malus da applicare al punteggio del turno, uno per ogni tipologia di ostacolo (STUDIO, SOPRAVVIVENZA, SOCIALE, ESAME). Sono nello stesso ordine dell'enumerazione che dovete dichiarare riportata nella slide riguardante le carte ostacolo.

D: Ottenere una ostacolo di tipo esame/nera è morte istantanea, considerando che le carte esame sono jolly? Oppure è necessario ottenere una ostacolo di ogni tipo normale + un jolly?

R: per morire servono comunque un minimo di 3 carte, le carte di tipo ESAME valgono come jolly, cioè come uno qualsiasi dei tre colori. Se avete 2 STUDIO + 1 ESAME si viene eliminati, se si hanno 1 STUDIO + 1 SOCIALE + 1 ESAME si viene eliminati, se si hanno 2 ESAME + qualsiasi altra carta si viene eliminati e così via.

D: Ogni giocatore ha un suo punteggio "temporaneo" per il turno, che verrà aggiunto a quello totale (quindi quello che viene effettivamente salvato nello struct e file) se è un vincitore, e quello dei non vincitori verrà scartato, e il loro pt totale rimarrà invariato?

R: il punteggio del turno viene sommato a quello della partita solo per i vincitori del turno, quindi chi non vince non prende punti se non quelli per aver accumulato carte disastro.



## F.A.Q.

---

D: Il punteggio "temporaneo" è sempre 0 all'inizio di un nuovo turno?

R: Sì, il punteggio del turno parte da 0. A questo va sommato il punteggio della carta, bonus/malus e poi eventuali effetti.

D: Si possono utilizzare codici ANSI per rendere l'estetica più gradevole?

R: Sì, ma valgono le regole relative ai magic number.

D: Possiamo modificare gli struct?

R: Sì, a vostro rischio e pericolo. Le struct devono essere salvate nei file di salvataggio con la strutturazione da noi richiesta.

D: Per controllare la fine del file, possiamo utilizzare la funzione di libreria "feof"?

R: No, usate il valore di ritorno della fscanf.

D: L'effetto [SCAMBIAC] scambia TUTTE le carte CFU punto di 2 giocatori o una?

R: Scambia la carta punto giocata nel turno da un giocatore G1 con quella di un giocatore G2, con G1 e G2 scelti dal giocatore che ha giocato la carta SCAMBIAC



## F.A.Q.

---

D: L'enumerazione del tipo ostacolo deve partire obbligatoriamente con il valore 1?

R: Non è "strettamente" obbligatorio, è obbligatorio che sia così nel file di salvataggio. Internamente, si può fare partire sia da 1 che da 0, l'importante è che siano scritti e letti con quei numeri nei file di salvataggio per preservare la compatibilità dei salvataggi

D: Si può utilizzare un while per pulire il buffer?

R: Sì, è possibile. Non usate fflush o simili

D: Nel file delle carte, il numero di occorrenze indica che quella carta si troverà in n copie nel mazzo e quindi si dovranno allocare n posizioni in memoria per quella carta?

R: Sì, per occorrenze si intende quante copie della stessa carta ci sono nel mazzo