



Programmazione 1 – Modulo II

CdL in Informatica – A.A. 2024 – 2025

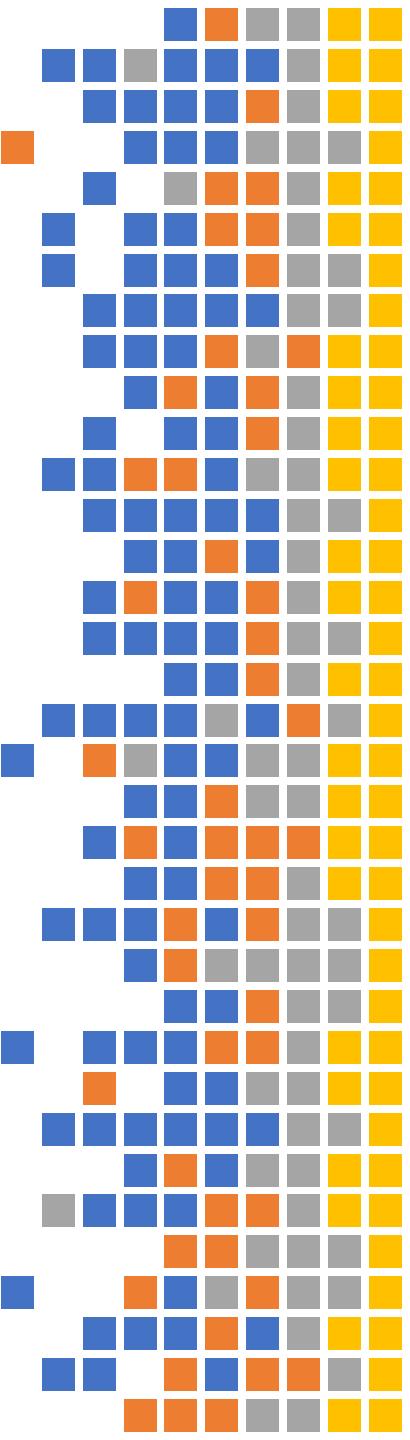
Progetto finale
Specifiche

Federico Meloni

Andrea Loddo

Alessandra Perniciano

Fabio Pili



Progetto negli anni

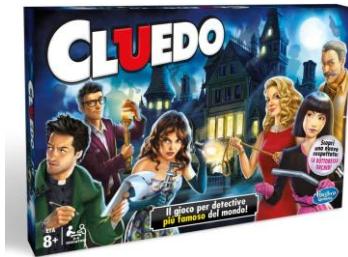
Storicamente il progetto di PR1 prevede un gioco di carte.
Quest'anno non fa eccezione!



2016



2017



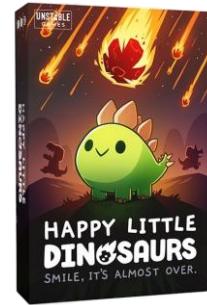
2018



2019



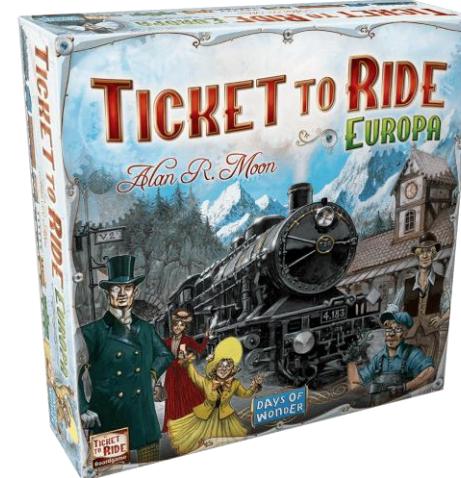
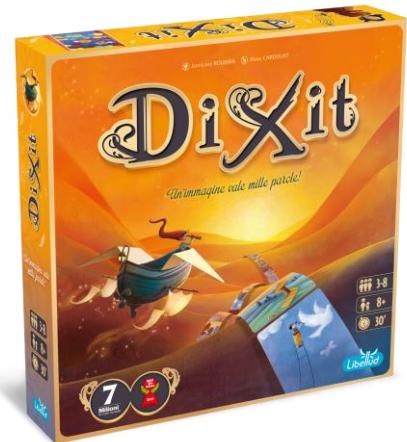
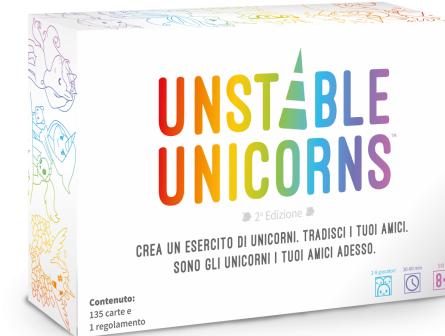
2023



2024

Progetto negli anni

Vogliamo darvi la possibilità di scelta



UNSTABLE UNICORNS™

2^a Edizione

CREA UN ESERCITO DI UNICORNI. TRADISCI I TUOI AMICI.
SONO GLI UNICORNI I TUOI AMICI ADESSO.

2-8 giocatori 30-60 min Età 8+

Contenuto:
135 carte e
1 regolamento





Organizzazione del progetto

- Il progetto è INDIVIDUALE.
- Il progetto è suddiviso in due parti:
 - funzionalità **BASE (30/30)**
 - funzionalità **AVANZATE (36/30)**
- Il progetto può essere realizzato solo in modo **incrementale**: non possono essere implementate le funzionalità avanzate tralasciando quelle base.
- **Sufficienza:** voto ≥ 15
- Dopo la consegna del progetto, implementato nella difficoltà scelta, **si dovrà sostenere un colloquio orale**.
- Data e ora del colloquio saranno organizzate in prossimità degli appelli ufficiali.
- **Tutto il progetto deve rispettare lo standard C99.**
- **È possibile utilizzare esclusivamente le librerie e i costrutti visti a lezione.**



Tempistiche

- Sono presenti due deadline:
 - SOFT: 23:59 del 28 febbraio 2025
 - HARD: 23:59 del 31 luglio 2025
- La consegna dopo la SOFT deadline prevede un malus sul voto finale di -0.4 alle 0:00 di ogni Sabato a partire dal 1 Marzo.
- Non è consentita la consegna DOPO l'HARD deadline.
- Il progetto può essere realizzato solo in modo **incrementale**: non possono essere implementate le funzionalità avanzate tralasciando quelle base.



Consegna del progetto

- Il progetto deve essere sviluppato e aggiornato tramite GitHub (*vedi slide 69-76: GitHub*).
- Le prime tre righe del file main.c, e del README, dovranno rispettare il seguente formato:

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

//Matricola: 60/61/66015

//Tipologia progetto: avanzato

- Quando siete pronti alla consegna del progetto dovrete consegnare, nell'apposita attività su Teams, un file strutturato come esemplificato nelle slide 10 e 11.
- Tale file fungerà da avviso di consegna. Da quel momento in poi, non saranno più ammesse modifiche al vostro repository GitHub.



Consegna del progetto

Dopo la consegna, attendete la convocazione per il colloquio orale.

Il colloquio si terrà nelle date degli appelli ufficiali o, in caso di un numero elevato di colloqui o cause di forza maggiore, nelle date vicine.

NOTA BENE: Aggiornare il vostro progetto su GitHub (con commit e push regolari) non è a pagamento, vi invitiamo a farlo il più possibile ogni qual volta il vostro progetto è in uno stato funzionante. Progetti con repository con pochi commit potrebbero non essere corrette.

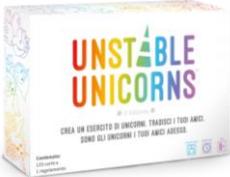


Consegna del progetto: deadline speciali

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

- **30/05/2025**: per autocertificazione 20 CFU borsa di studio (30/06/2025)
- **10/07/2025**: per autocertificazione 20 CFU borsa di studio (10/08/2025)
- **06/05/2025**: per appello ASD (06/06/2025)
- **19/05/2025**: per appello ASD (19/06/2025)
- **07/06/2025**: per appello ASD (07/07/2025)
- **10/08/2025**: per appello ASD (10/09/2025)

Precisare nella consegna su Teams se avete tali necessità.



Consegna del progetto: esempio

Lo studente **Austin Reaves** (matricola 60/61/12345) ha deciso di consegnare un progetto AVANZATO in data 29/05/2025 e ha bisogno dell'autocertificazione dei CFU di PR1 in tempo per la scadenza della prima rata della borsa di studio, prevista il 30/06/2025. Conseggerà un file txt così strutturato:

Studente: Austin Reaves

Matricola: 60/61/12345

Tipologia progetto: AVANZATO

Username GitHub: austinreaves12

Link GitHub: <https://github.com/unica-pr1/progetto-finale-austin-reaves-12>

10 Necessità: AUTOCERTIFICAZIONE CFU 30/06/2025



Consegna del progetto: esempio

Lo studente **Anthony Davis** (matricola 60/61/54321) ha deciso di consegnare un progetto BASE in data 15/02/2025 e non ha particolari necessità. Conseggerà un file txt così strutturato:

Studente: Anthony Davis

Matricola: 60/61/54321

Tipologia progetto: BASE

Username GitHub: antdavis23

Link GitHub: <https://github.com/unica-pr1/progetto-finale-ant-davis-23>

Necessità: NESSUNA

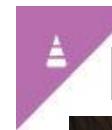


Cambio di contesto

UNSTABLE STUDENTS



BABY NARWHAL



Studente Multitasking



Card Type: Unicorn (Baby)

If this card would be sacrificed, destroyed, or returned to your hand, return it to the Nursery instead.

Unicorni

Fa gli esercizi, risponde alle domande e nel mentre si guarda pure House of Cards

Studenti



Carte: studenti

Esistono N tipologie di carte.

Le carte STUDENTE si suddividono in 3 sottocategorie:



Fa gli esercizi, risponde alle domande e nel mentre si guarda pure House of Cards

MATRICOLA



Chiacchiera sempre con il collega di fianco, ma sicuramente non parlano di puntatori...

STUDENTE SEMPLICE



Questo student fa questo, questo
e questo effetto.....
Blablablablablablablabla

LAUREANDO



Carte: bonus, malus, magia, mai..



CTM in ritardo



Descrizione molto molto
bella della carta malus



Eplode il Lab T



Descrizione molto molto
bella della carta magia

Le carte BONUS e MALUS si utilizzano per (s)vantaggiare un giocatore

Le carte MAGIA applicano degli effetti immediati su se stessi o sugli altri

Le carte MAI permettono di difendersi da MALUS o MAGIE che stanno per essere assegnati



Carte: effetto

Alcune carte possiedono degli effetti.

Ogni effetto è caratterizzato da N azioni che vanno eseguite sequenzialmente (se possibili):

- **GIOCARE** una carta
- **PESCAR**E una carta dal mazzo
- **SCARTARE** una carta
- **RUBA** una carta
- **BLOCCARE** gli effetti
- **MOSTRARE** la propria mano

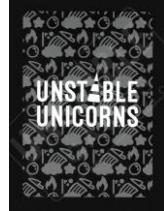
Etc...

Vedi slide 48 per approfondire...



Inizio del gioco

Non avevamo tempo per fare le grafiche di ogni carta, quindi questo esempio sarà con gli unicorni ☺



Pesca



Scarti



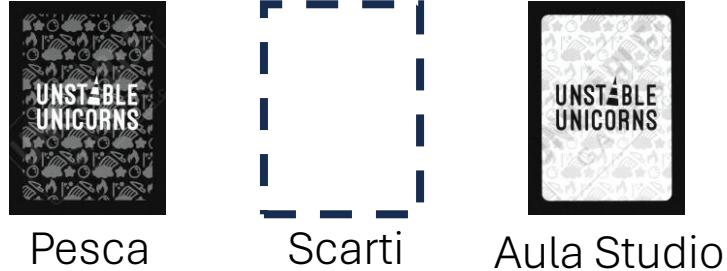
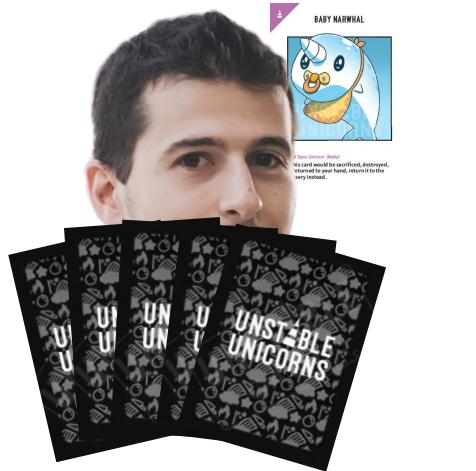
Aula Studio





Distribuzione 5 carte + 1 matricola

Ogni giocatore riceve 5 carte in mano
e 1 matricola nella propria aula.





Fase Pesca

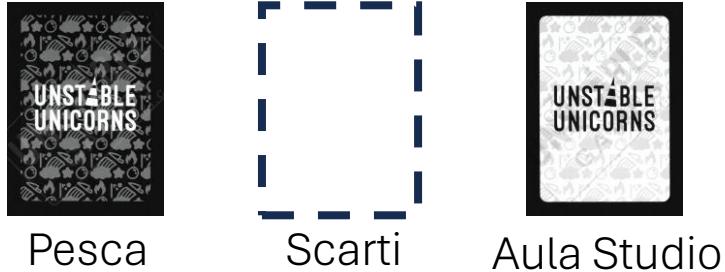
Prima di giocare una carta ognì giocatore deve pescare dal mazzo di pesca.





Fase Fase Azione – Giocare uno Studente

In questo caso viene giocata una carta nella propria aula



In fase azione si può fare UNA azione:

- **Giocare una carta**
- Pescare un'altra carta





Fase Fase Azione – Pescare ancora



In questo caso viene pescata un ulteriore carta



In fase azione si può fare UNA azione:

- Giocare una carta
- Pescare un'altra carta





Fase Azione – Giocare una Magia



In fase azione si può fare UNA azione:

- Pescare un'altra carta
- Giocare una carta



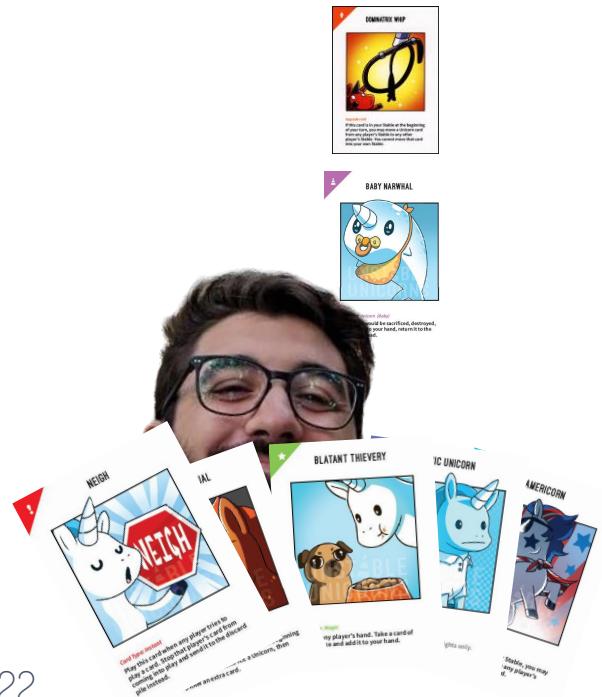
In questo caso viene giocata una carta magia (che poi finisce negli scarti dopo che vengono eseguiti gli effetti)





Fase Azione – Giocare un Bonus

In questo caso viene giocata una bonus nella propria sezione bonusmalus.



In fase azione si può fare UNA azione:

- Pescare un'altra carta
- Giocare una carta

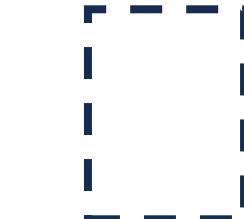




Fase Azione – Giocare un Malus



Pesca



Scarti



Aula Studio



In fase azione si può fare UNA azione:

- Pescare un'altra carta
- Giocare una carta



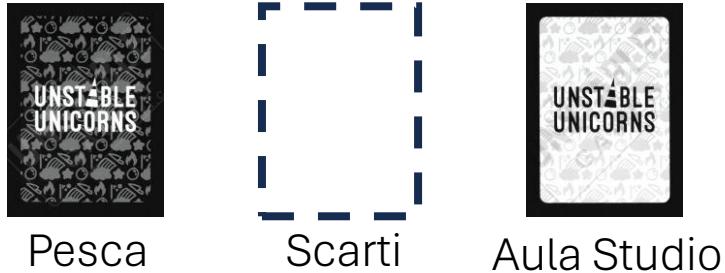
In questo caso viene giocata una carta malus nella sezione bonusmalus di una altro giocatore



Gli Effetti – Inizio, Subito, Fine...



Gli effetti si possono attivare in varie fasi di gioco...





Gli Effetti – Inizio, Subito, Fine...

Ci sono effetti che si applicano prima della fase di pesca e possono essere di:

- Laureandi
- Bonus
- Malus

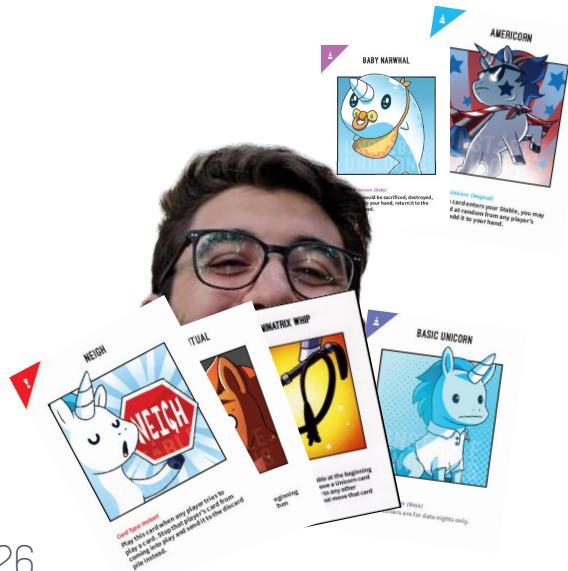




Gli Effetti – Inizio, Subito, Fine...

Ci sono effetti che si applicano appena si gioca la carta e possono essere:

- Laureandi
- Magia



Pesca



Scarti



Aula Studio





Gli Effetti – Inizio, Subito, Fine...

Ci sono effetti che si applicano quando lo student viene bocciato (scartato) e possono essere:

- Laureandi



Pesca

Scarti

Aula Studio

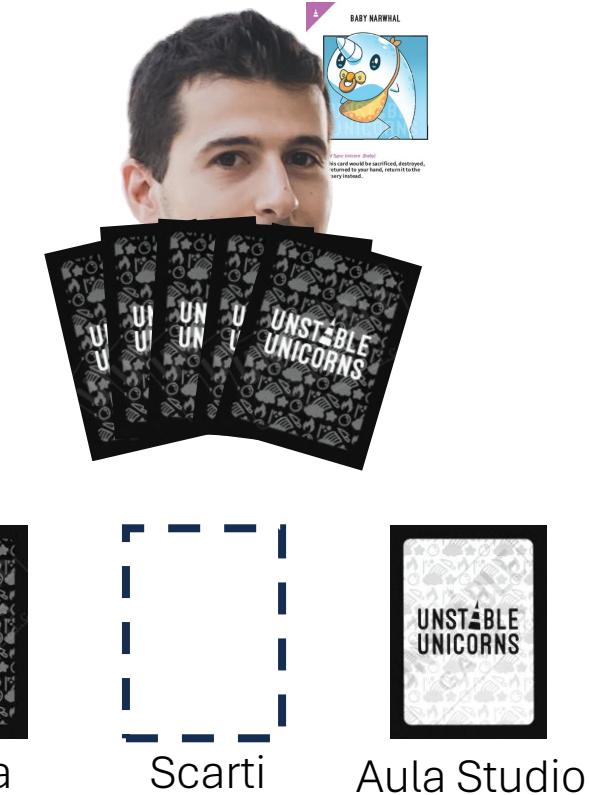




Gli Effetti – Far scartare carte

Negli effetti ce ne saranno alcuni che faranno scartare carte...

Tutte le carte sacratte vanno nel mazzo degli scarti.

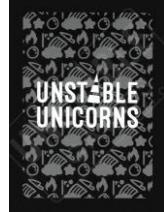




Gli Effetti – Far scartare carte

Negli effetti ce ne saranno alcuni che faranno scartare carte...

Tutte le carte sacratte vanno nel mazzo degli scarti. Che sia dalla mano, dall'aula oppure dai bonusmalus.



Pesca



Scarti



Aula Studio





Gli Effetti – Far scartare carte

Negli effetti ce ne saranno alcuni che faranno scartare carte...

Tutte le carte sacrate vanno nel mazzo degli scarti. Che sia dalla mano, dall'aula oppure dai bonusmalus.

Tranne le matricole che tornano in aula studio.



Pesca



Scarti



Aula Studio

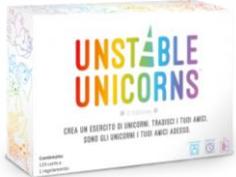




Difendersi dagli effetti

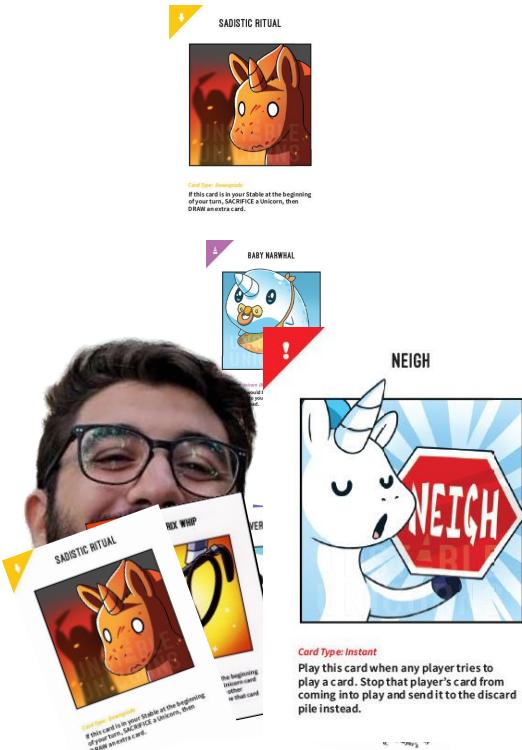
Se un giocatore viene attaccato da un qualsiasi effetto di un altro giocatore e ha una carta MAI può usarla per non subire questi effetti





Difendersi dagli effetti

Se un giocatore viene attaccato da un qualsiasi effetto di un altro giocatore e ha una carta MAI può usarla per non subire questi effetti





Difendersi dagli effetti

Se un giocatore viene attaccato da un qualsiasi effetto di un altro giocatore e ha una carta MAI può usarla per non subire questi effetti

Gli effetti dei laureandi semplicemente annullati, mentre bonus e malus finiscono negli scarti se ci si difende con un MAI.



Pesca



Scarti



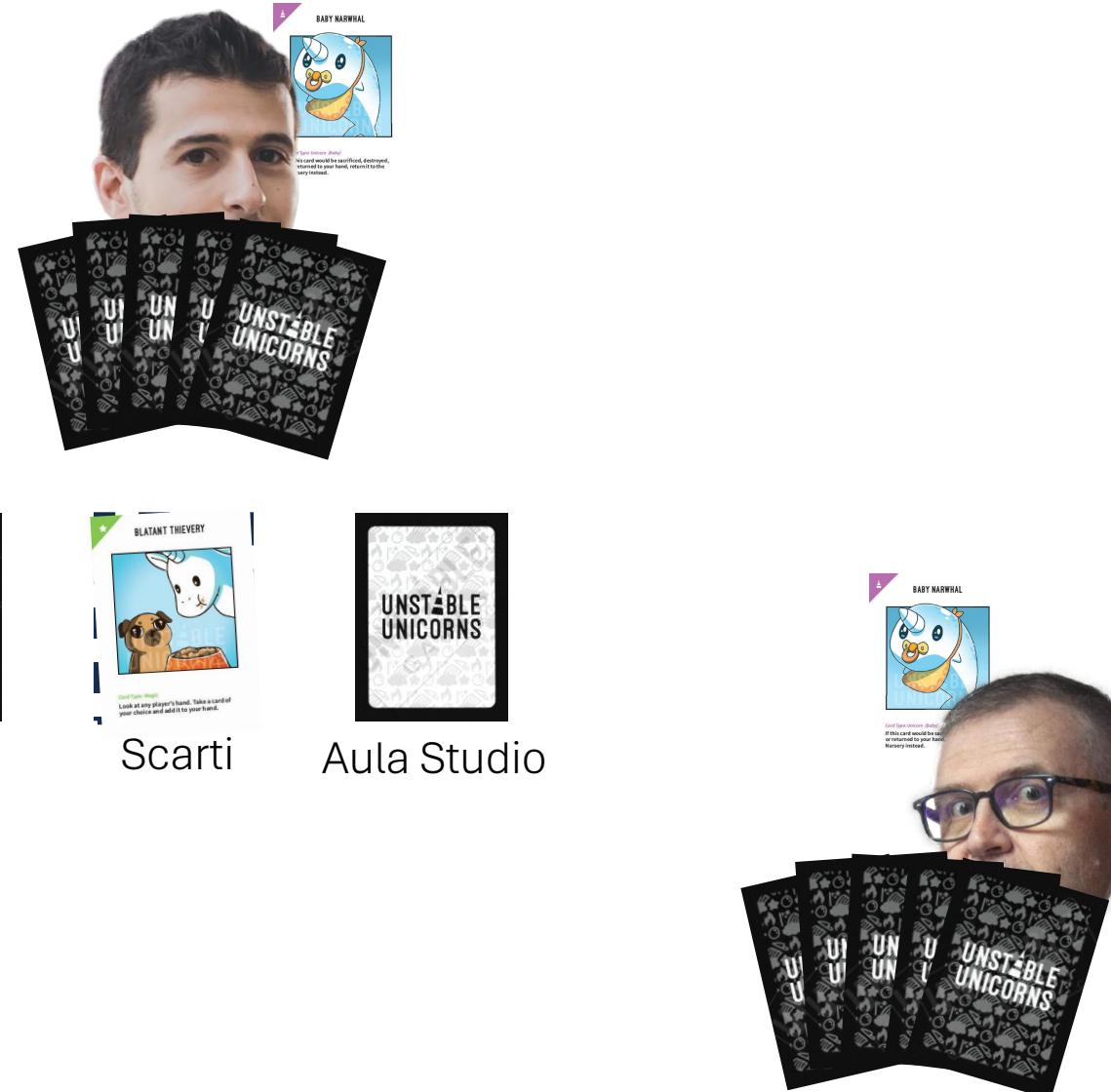
Aula Studio





Fine del turno

Alla fine del turno ogni giocatore può avere massimo 5 carte, quindi deve scartare quelle in più





Fine del gioco

Vince il primo giocatore che accumula 6 studenti.

Questa condizione può variare con dei malus.



Pesca



Scarti



Aula Studio





Carte Matricola

quantità	nome	descrizione	tipoCarta	numEffetti	effetto	quando	opzionale
1	Matricola Energy Drink	Impossibile seguire una lezione senza la Red Bull. Non si può.	MATRICOLA	0		MAI	False
1	Matricola Scout	Che bello programmare all'aria aperta.	MATRICOLA	0		MAI	False
1	Matricola Smanettone	Avrà una versione di ArchLinux su cui non girerà nulla.	MATRICOLA	0		MAI	False
1	Matricola Gamer	Frequenta perlopiù lezioni di Brawl Stars, Geometry Dash e TCG Pokemon.	MATRICOLA	0		MAI	False
1	Matricola Multitasking	Svolge gli esercizi, guarda House Of Cards e risponde alle domande del docente.	MATRICOLA	0		MAI	False
1	Matricola Scacchista	Arriverà in ritardo alla lezione perché sta finendo la partita all'ingresso del Palazzo.	MATRICOLA	0		MAI	False



Carte Studente Semplice

quantità	nome	descrizione	tipoCarta	numEffetti	effetto	quando	opzionale
3	Studente Freddoloso	Possono esserci 40 gradi, ma lui avrà comunque freddo.	STUDENTE_SEMPLICE	0		MAI	False
3	Studente Boscaiolo	Barba folta e passione per le camicie a quadri.	STUDENTE_SEMPLICE	0		MAI	False
3	Studente Trassato	Confonde il Palazzo delle Scienze con le Terrazze di Calamosca.	STUDENTE_SEMPLICE	0		MAI	False
3	Studente Erasmus	Profesor, es posible realizar el examen en inglés?	STUDENTE_SEMPLICE	0		MAI	False
3	Studente Influencer	La soluzione al tuo esercizio fa tendenza!	STUDENTE_SEMPLICE	0		MAI	False
3	Studente Palestrato	Ha le spalle così larghe che occupa due posti.	STUDENTE_SEMPLICE	0		MAI	False
3	Studente Anticipatore	Se la lezione parla degli array, lui sta già pensando alle liste.	STUDENTE_SEMPLICE	0		MAI	False
3	Studente Chiacchierone	Parlerà di tutto con il collega di fianco ma non dei puntatori.	STUDENTE_SEMPLICE	0		MAI	False



Carte Laureando

quantità	nome	descrizione	tipoCarta	numEffetti	effetto	quando	opzionale
4	Laureando Nerd	Quando questa carta esce dalla tua aula, puoi eliminare una carta Studente dall'aula di un altro giocatore.	LAUREANDO	1	[ELIMINA, TU, ALL]	FINE	True
4	Laureando Lagna	Quando questa carta entra nella tua aula, ogni giocatore incluso te stesso deve scartare una carta.	LAUREANDO	1	[SCARTA, TUTTI, ALL]	SUBITO	False
4	Laureando Seduttore	Quando questa carta entra nella tua aula, puoi scartare una carta e poi ruba una carta studente.	LAUREANDO	2	[SCARTA, IO, ALL], [RUBA, TU, STUDENTE]	SUBITO	True
4	Laureando Avido	Quando questa carta entra nella tua aula, pesca una carta.	LAUREANDO	1	[PESCA, IO, ALL]	SUBITO	False
4	Laureando Hipster	Quando questa carta entra nella tua aula, pesca due carte e scarta una carta.	LAUREANDO	3	[PESCA, IO, ALL], [PESCA, IO, ALL], [SCARTA, IO, ALL]	SUBITO	False
4	Laureando Casinista	Quando questa carta entra nella tua aula, tutti i giocatori devono eliminare una carta studente dalla propria aula.	LAUREANDO	1	[ELIMINA, TUTTI, STUDENTE]	SUBITO	False
4	Laureando Snitch	Quando questa carta entra nella tua aula, puoi eliminare una carta vantaggio dall'aula di un altro giocatore.	LAUREANDO	1	[ELIMINA, TU, BONUS]	SUBITO	True
4	Laureando Ladro	Quando questa carta entra nella tua aula, puoi rubare una carta vantaggio dall'aula di un altro giocatore.	LAUREANDO	1	[RUBA, TU, BONUS]	SUBITO	True



Carte Bonus

quantità	nome	descrizione	tipoCarta	numEffetti	effetto	quando	opzionale
3	CTM in orario	Se questa carta è nella tua aula all'inizio del tuo turno, puoi giocare due carte.	BONUS	1	[GIOCA, IO, ALL]	INIZIO	True
3	Salta fila in mensa	Se questa carta è nella tua aula all'inizio del tuo turno, puoi scartare 3 carte e poi rubare una carta studente.	BONUS	4	[SCARTA, IO, ALL], [SCARTA, IO, ALL], [SCARTA, IO, ALL], [RUBA, TU, STUDENTE]	INIZIO	True
3	Aula dei vecchi progetti	Se questa carta è nella tua aula all'inizio del tuo turno, puoi scartare 2 carte poi eliminare una carta studente dall'aula di un altro giocatore.	BONUS	3	[SCARTA, IO, ALL], [SCARTA, IO, ALL], [ELIMINA, TU, STUDENTE]	INIZIO	True
3	Macchinetta inceppata	Se questa carta è nella tua aula all'inizio del tuo turno, puoi scartare una carta e poi pesca una carta.	BONUS	2	[SCARTA, IO, ALL], [PESCA, IO, ALL]	INIZIO	True
3	Sovraccarico di caffeina	Se questa carta è nella tua aula all'inizio del tuo turno, puoi eliminare una carta dalla tua aula e poi pesca due carte.	BONUS	3	[ELIMINA, IO, ALL], [PESCA, IO, ALL], [PESCA, IO, ALL]	INIZIO	True
3	Sparacoriandoli	Se questa carta è nella tua aula all'inizio del tuo turno, puoi eliminare una carta dalla tua aula e poi elimina una carta dall'aula di un altro giocatore.	BONUS	2	[ELIMINA, IO, ALL], [ELIMINA, TU, ALL]	INIZIO	True



Carte Malus

quantità	nome	descrizione	tipoCarta	numEffetti	effetto	quando	opzionale
2	Aula minuscola	A inizio turno, elimina una carta studente dalla tua aula.	MALUS	1	[ELIMINA, IO, STUDENTE]	INIZIO	False
1	Portone chiuso	A inizio turno, scarta una carta.	MALUS	1	[SCARTA, IO, ALL]	INIZIO	False
1	Allerta meteo	A inizio turno, elimina una carta studente dalla tua aula e poi pesca una carta.	MALUS	2	[ELIMINA, IO, STUDENTE], [PESCA, IO, ALL]	INIZIO	False
1	Esame in Lab T	La tua mano deve essere visibile a tutti i giocatori.	MALUS	1	[MOSTRA,IO,ALL]	SEMPRE	False
1	Teams fuori servizio	Non puoi giocare carte BONUS.	MALUS	1	[IMPEDIRE,IO,BONUS]	SEMPRE	False
1	CTM in ritardo	Non puoi giocare carte MAI.	MALUS	1	[IMPEDIRE,IO,ISTANTANEA]	SEMPRE	False
1	Ingegnerizzazione	Tutti i tuoi studenti sono considerati ingegneri.	MALUS	1	[INGEGNERE,IO,STUDENTE]	SEMPRE	False



Carte Magia

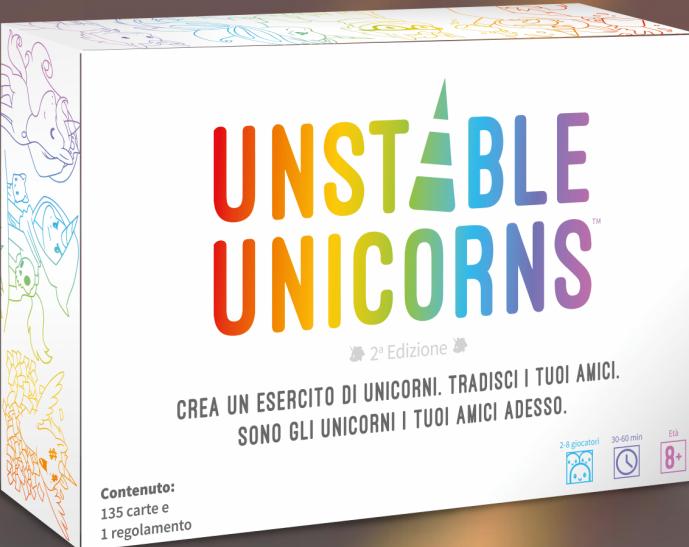
quantità	nome	descrizione	tipoCarta	numEffetti	effetto	quando	opzionale
3	Due per Uno	Elimina 1 carta dalla tua aula, poi elimina 2 carte dall'aula di un altro giocatore.	MAGIA	3	[ELIMINA, IO, ALL], [ELIMINA, TU, ALL], [ELIMINA, TU, ALL]	SUBITO	False
3	Scambio di esame	Scambia la tua mano con quella di un altro giocatore.	MAGIA	1	[SCAMBIA, TU, ALL]	SUBITO	False
3	Passaggio di appunti	Ruba 1 carta da una mano di un altro giocatore.	MAGIA	1	[PRENDI, TU, ALL]	SUBITO	False
3	Copia dal turno precedente	Scarta 1 carta dalla mano e ruba 2 carte dalla mano di un altro giocatore.	MAGIA	3	[SCARTA, IO, ALL], [PRENDI, TU, ALL], [PRENDI, TU, ALL]	SUBITO	False
3	Aperitivo da Tiffany	Elimina 1 carta studente dall'aula di un altro giocatore.	MAGIA	1	[ELIMINA, TU, STUDENTE]	SUBITO	False
2	Colpo di Fortuna	Pesca 2 carte, scarta 3 carte, poi gioca una carta.	MAGIA	6	[PESCA, IO, ALL], [PESCA, IO, ALL], [SCARTA, IO, ALL], [SCARTA, IO, ALL], [SCARTA, IO, ALL], [GIOCA, IO, ALL]	SUBITO	False
2	Ottimo Affare	Pesca 3 carte e scarta 1 carta.	MAGIA	4	[PESCA, IO, ALL], [PESCA, IO, ALL], [PESCA, IO, ALL], [SCARTA, IO, ALL]	SUBITO	False
2	Attacco Hacker	Elimina 1 carta vantaggio dall'aula di un altro giocatore.	MAGIA	1	[ELIMINA, TU, BONUS]	SUBITO	False
2	free(svantaggio);	Elimina 1 carta svantaggio dalla tua aula.	MAGIA	1	[ELIMINA, IO, MALUS]	SUBITO	False
2	Ruba Studente	Prendi 1 carta dall'aula di un altro giocatore.	MAGIA	1	[RUBA, TU, STUDENTE]	SUBITO	False
3	Sessione di Laurea	Ogni giocatore (incluso te stesso) elimina 1 carta studente dalla propria aula.	MAGIA	1	[ELIMINA, TUTTI, STUDENTE]	SUBITO	False
3	Tasto reset	Ogni giocatore (incluso te stesso) elimina 1 carta bonus e 1 carta malus dalla propria aula.	MAGIA	2	[ELIMINA, TUTTI, BONUS], [ELIMINA, TUTTI, MALUS]	SUBITO	False
3	Bagno intasato	Ogni giocatore (escluso te stesso) deve scartare 1 carta.	MAGIA	1	[SCARTA, VOI, ALL]	SUBITO	False



Carte Istantanee

quantità	nome	descrizione	tipoCarta	numEffetti	effetto	quando	opzionale
15	MAI	Blocca l'azione che un giocatore ti sta imponendo.	INSTANTANEA	1	[BLOCCA, IO, ALL]	SUBITO	False

IMPLEMENTAZIONE



Regole obbligatorie:

- Standard C99
- Suddivisione in più file .c/.h
- Lettura e scrittura file
- Utilizzo delle liste

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

Modificare le strutture obbligatorie **compromette la compatibilità dei salvataggi** da noi richiesti.

Rispettare anche l'ordine dei campi.



Strutture: Giocatore

Giocatore è una struttura composta dai seguenti campi:

- Il nome del giocatore (32 caratteri utili)
- Una lista di carte che rappresenta le carte del giocatore
- Una lista di carte che rappresenta l'aula del giocatore
- Una lista di carte che rappresenta le carte bonus e malus del giocatore
- Un puntatore al giocatore successivo



Strutture: Carta

Carta è una struttura composta dai seguenti campi:

- Il nome della carta (32 caratteri compreso il terminatore di stringhe)
- La descrizione della carta (256 caratteri compreso il terminatore di stringhe)
- La tipologia della carta (enumerazione TipologiaCarta)
- Un numero che rappresenta il numero degli effetti della carta
- Un puntatore di tipo Effetto che rappresenta l'insieme ordinato degli effetti della carta
- Quando gli effetti della carta devono essere eseguiti (enumerazione Quando)
- Un bool che indica se gli effetti delle carte possono essere giocati
- Un puntatore di tipo Carta che rappresenta la carta successiva



Strutture: Effetto

Effetto è una struttura composta dai seguenti campi:

- Un'azione (enumerazione **Azione**) che rappresenta l'azione dell'effetto
- Il targetGiocatori (enumerazione **TargetGiocatori**) a cui è rivolto l'effetto
- La tipologia della carta (enumerazione **TipologiaCarta**) target dell'effetto



Enumerazioni: TipoCarta

TipoCarta sarà un enumerazione che comprende i seguenti valori:

- ALL → enum generica per tutte le carte
- STUDENTE → enum generica per quando ci si riferisce ad uno dei 3 studenti
- MATRICOLA → quando una matricola viene eliminata dall'aula finisce in aula studio
- STUDENTE_SEMPlice
- LAUREANDO
- BONUS
- MALUS
- MAGIA
- ISTANTANEA



Enumerazioni: Azione

Azione sarà un enumerazione che comprende i seguenti valori:

- **GIOCA** → giocare una carta dalla mano
- **SCARTA** → scartare una carta dalla mano
- **ELIMINA** → scartare una carta da aula o bonusmalus (vedi targetCarta)
- **RUBA** → prendere una carta da aula o bonusmalus (vedi targetCarta) e metterlo nella propria aula
- **PESCA** → pescare una carta dal mazzo di pesca
- **PRENDI** → rubare una carta dalla mano di un giocatore e metterla nella propria
- **BLOCCA** → impedisce di subire un effetto
- **SCAMBIA** → scambia la propria mano con quella di un altro giocatore
- **MOSTRA** → fa vedere a tutti in chiaro la propria mano quando si visualizza lo stato degli altri giocatori
- **IMPEDIRE** → non permette al proprietario di utilizzare determinate tipologie di carte
- **INGEGNERE** → trasforma i propri studenti di informatica in studenti di ingegneria (impedisce la vittoria)



Enumerazioni: Quando

Questa enumerazione rappresenta quando l'effetto andrà ad essere eseguito.

Quando sarà un enumerazione che comprende i seguenti valori:

- **SUBITO** → non appena la carta viene giocata
- **INIZIO** → ad inizio di ogni turno
- **FINE** → quando la carta esce dall'aula
- **MAI** → mai
- **SEMPRE** → l'effetto è sempre in essere (*possono averlo solo bonus/malus*)



Enumerazioni: TargetGiocatori

TargetGiocatori rappresenta un enumerazione che indica su che giocatori agisce l'effetto della carta giocata. Una carta può avere più effetti, ogni effetto un target diverso.

TargetGiocatori sarà un enumerazione che comprende i seguenti valori:

- **IO** → Il giocatore che ha giocato la carta
- **TU** → Un giocatore scelto dal giocatore che ha scelto la carta
- **VOI** → Tutti i giocatori tranne il giocatore che ha giocato la carta
- **TUTTI** → Tutti i giocatori, compreso il giocatore che ha giocato la carta

NB: sono presenti effetti che, sebbene abbiano come TargetGiocatore **TU** richiedono anche la complicità del giocatore che ha giocato la carta (vedi azione **SCAMBIA**).



Fasi di gioco: fase 0 – preparazione

Il programma deve poter:

- Iniziare una nuova partita
- Caricare una partita da file salvataggio (SE passato come argomento del main)



Fasi di gioco: fase 0 – nuova partita

Iniziare una nuova partita:

- Chiedere quanti giocatori giocheranno (2-4) ed il nome per ogni giocatore
- L'ordine di gioco poi rispetterà l'ordine di inserimento
- Leggere le carte dal file mazzo.txt ed inizializzare il mazzo di pesca
 - È concesso dividere da subito le carte MATRICOLA dalle altre e metterle in Aula Studio
- Mischiare le carte nel mazzo di pesca (ed eventualmente in Aula Studio)
- Distribuire casualmente, a ogni giocatore, 5 carte per la mano e 1 matricola per l'aula



Fasi di gioco: fase 0 – caricamento

Caricare una partita da salvataggio:

- Caricare giocatori e mazzi dal file binario di salvataggio
- Nel caricamento dei giocatori l'ordine del file è preservato all'interno della lista. Il primo turno al riprendere della partita verrà effettuato dal primo giocatore letto dal file binario.
- Nel caricamento delle carte (mani, aule, mazzi pesca, mazzi scarti etc..) viene preservato l'ordine del file (prima carta in testa, ultima carta in coda).



Fasi di gioco: fase 1 – inizio turno

Prima dell'inizio di ogni turno va salvata la partita nell'apposito file binario di salvataggio (vedi slide 63 e 64: *File di salvataggio*).

All'inizio di ogni turno si dovrà stampare il numero del turno e il giocatore di quel turno.

Dopodiché si dovrà controllare se nel campo di gioco del giocatore sono presenti carte bonusmalus che hanno come QUANDO dell'Effetto "INIZIO" e, in caso affermativo, svolgere gli effetti di queste carte.

Successivamente fare la stessa cosa con le carte presenti in aula.

NB: tenere conto dell'eventuale opzionalità nell'esecuzione degli effetti.



Fasi di gioco: fase 2 – pesca

Finita la fase iniziale del turno, il giocatore deve pescare una carta dal mazzo di pesca.

Nel caso il mazzo di pesca non abbia più carte:

- Mescolare il mazzo degli scarti
- Scambiare il mazzo di pesca e quello degli scarti

NB: questa operazione di scambio dei mazzi va fatta in tutti i casi in cui sia richiesto di pescare una carta dal mazzo di pesca e questo è vuoto (vedi sezione Effetti).



Fasi di gioco: fase 3 – azione

Durante la fase di azione, i giocatori potranno eseguire le seguenti azioni:

- [TASTO 1] giocare una carta dalla propria mano
- [TASTO 2] pescare un’ulteriore carta
- [TASTO 3] visualizzare le carte nella propria mano, aula e bonusmalus
- [TASTO 4] controllare lo stato degli altri giocatori (numero carte mano, aula, bonusmalus)
- [TASTO 0] uscire dalla partita

Le azioni "giocare una carta" e "pescare un’ulteriore carta" fanno finire la fase di azione.

Non è possibile inserire due carte identiche nella propria aula (o bonusmalus) o quella di un altro giocatore.

NB: stampare le carte in mano quando si deve giocare una carta. Non è strettamente necessario far vedere tutti gli effetti delle carte in quel momento. Con [TASTO 3] stampare tassativamente tutti gli effetti di ogni carta. Scegliere il formato più opportuno.



Fasi di gioco: fase 3 – azione (difendersi)

Quando si applica un effetto ad un altro giocatore, questo può (se ne è in possesso) utilizzare una carta MAI che finirà negli scarti e che gli permetterà di interrompere l'esecuzione degli effetti della carta giocata dall'avversario.

Un giocatore può utilizzare il MAI anche per impedire che gli venga attribuito un bonus o malus da un altro giocatore.

Quando si utilizza un MAI per difendersi da:

- Un laureando → si interrompono gli effetti ma il laureando rimane nell'aula del giocatore
- Una magia → si interrompono gli effetti e la carta finisce (come normale) negli scarti
- Una bonusmalus → la carta finisce negli scarti



Fasi di gioco: fase 4 – fine del turno

Ogni giocatore alla fine del proprio turno può avere un massimo di 5 carte.

Scartare il quantitativo di carte necessario affinché il numero di carte in mano sia minore o uguale a 5. Il giocatore sceglie che carte scartare nel mazzo degli scarti.



Fasi di gioco: fase 5 – fine del gioco

Alla fine di ogni turno si controlla la condizione di vittoria del giocatore corrente.

Se la condizione è soddisfatta il gioco finisce e si da un messaggio di congratulazioni al giocatore, altrimenti si passa il turno al giocatore successivo.

REMINDER: un giocatore vince quando accumula almeno 6 studenti nella propria aula.

NB: la condizione di vittoria può essere influenzata da dei malus
(vedi INGEGNERIZZAZIONE)



Implementazione: progetto BASE

- Il numero di giocatori è deciso a runtime e vengono gestiti con un lista concatenata lineare circolare.
- Implementare tutta la logica del gioco come descritto nelle *slide Fasi di gioco* (vedi slide da 51 a 59).
- Le carte vanno caricate dal file mazzo.txt fornito.
- All'inizio di ogni turno si deve salvare la partita in un file binario ‘savegame.sav’ strutturato come mostrato nelle *slide File di salvataggio* (vedi slide 63 e 64).
- Questo salvataggio, se presente, deve poter essere caricato ad ogni avvio del programma.
- Generare un file log.txt in cui viene scritta la cronologia della partita, esempio:
TURNO 1: Riccardo gioca un Laureando Ammaliante
TURNO 2: Andrea pesca Studente Trassato
TURNO 2: Andrea gioca Aula dei vecchi progetti



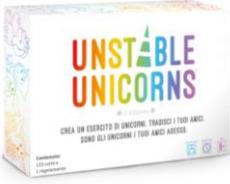
Implementazione: progetto AVANZATO

- 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.
 - maggiore sarà la completa casualità nelle scelte, minore sarà il voto della parte avanzata
 - maggiore sarà la logica (con un senso e una strategia dietro), maggiore sarà il voto della parte avanzata.



Implementazione: progetto AVANZATO

- 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.
- 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.
- Il file di salvataggio si deve poter fornire sia da riga di comando che durante l'esecuzione della partita; se, implementando la versione AVANZATA, viene fornito un file da terminale, non si dovrà chiedere a runtime il file di salvataggio.
- È concesso, se desiderato, utilizzare file .txt ausiliari per tenere traccia dei salvataggi presenti per la gestione dei salvataggi a runtime.



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.
 - Un numero rappresentante il numero di carte che ha in mano il giocatore
 - Una lista rappresentante le carte in mano al giocatore
 - Un numero rappresentante il numero di carte che ha nell'aula il giocatore
 - Una lista rappresentante le carte dell'aula del giocatore
 - Un numero rappresentante il numero di carte bonusmalus
 - Una lista rappresentante le carte bonusmalus del giocatore



File di salvataggio

- Per ogni mazzo (in questo ordine pesca, scarti, aula studio):
 - Un numero rappresentante 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 assicurandosi che tutto funzioni nel caso in cui esse non siano presenti.



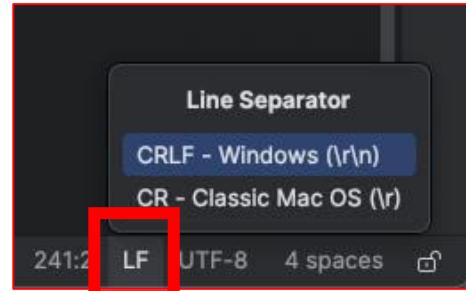
File da caricare

Avete a disposizione un file che si chiama "mazzo.txt" formattato in questa maniera:

- Numero di copie da creare della carta
- Il nome della carta
- La descrizione della carta
- La tipologia della carta
- Il numero degli effetti della carta
- Una riga per ogni effetto della carta dove ogni riga è composta dai campi della struttura Effetto.
- Un numero che rappresenta il valore dell'enumerazione Quando
- Un numero che rappresenta il flag che indica se gli effetti sono obbligatori o meno

NOTA BENE: una volta importato il file mazzo.txt all'interno del vostro progetto, apritelo con CLion e impostate LF come *line separator* del file:
Trovate l'opzione in basso a destra.

Non dovete avere CRLF o CR.





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



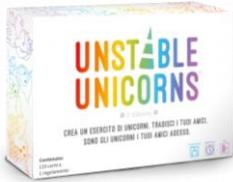
BEST PRACTICES ai fini della valutazione

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.

- Per turno si intende ogni volta che tocca ad un giocatore. NON si intende un giro completo dei giocatori.
- Nel momento in cui bisogna scartare una carta ma non si hanno carte in mano, la scartata è presa per buona.
- Nel momento in cui si devono rubare carte dalla mano, aula o bonusmalus di un altro giocatore, se il giocatore selezionato non ha carte in quella lista l'effetto è perso e non si deve scegliere un altro giocatore.
- Non è necessario salvare la partita dopo la vittoria di un giocatore. Il salvataggio rimane fermo a prima che il giocatore che vincerà la partita abbia giocato.
- Il mazzo aula studio viene utilizzato solo in fase di preparazione e quando una matricola viene eliminata dall'aula di un giocatore (le matricole anziché negli scarti vanno in aula studio)
- La carta "CTM in Orario" permette di giocare due carte durante il turno, la prima carta va giocata prima della fase di pesca, in quanto è un effetto con quando INIZIO.
- Nella versione del progetto BASE, il file di salvataggio si può fornire solo da riga di comando. Nella versione del progetto AVANZATA, si deve poter fornire sia da riga di comando che durante l'esecuzione della partita. Le specifiche dell'AVANZATO si aggiungono a quelle BASE, non le sostituiscono. Se, implementando la versione AVANZATA, viene fornito un file da terminale, non si dovrà chiedere a runtime il file di salvataggio. È concesso, se desiderato, utilizzare file .txt ausiliari per tenere traccia dei salvataggi presenti per la gestione dei salvataggi a runtime.



GitHub: iscrizione

Dopo esservi iscritti a GitHub, accedete al link della nostra GitHub classroom e accettate l'assignment:

<https://classroom.github.com/a/ldtyEsnH>

A screenshot of the GitHub Classroom header bar. It includes the GitHub Classroom logo, navigation links for 'GitHub Education', 'Messages', 'Help', and 'Profile', and a search icon.

unica-pr1-2425

Accept the assignment –
Progetto finale

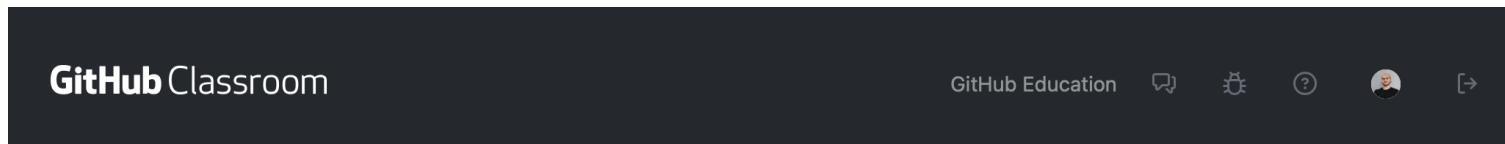
Once you accept this assignment, you will be granted access to the
[progetto-finale-bron3v](#) repository in the [unica-pr1](#) organization on
GitHub.

[Accept this assignment](#)



GitHub: iscrizione

Il link che otterrete dopo aver accettato è il link alla vostra repository, nonché quello che dovete indicarci in fase di consegna del progetto (*vedi slide da 7 a 11*).



You're ready to go!

You accepted the assignment, **Progetto finale**.

Your assignment repository has been created:

 <https://github.com/unica-pr1/progetto-finale-bron3v>

We've configured the repository associated with this assignment.

 Your assignment is due by **Jul 31, 2025, 21:59 UTC**



Join the GitHub Student Developer Pack

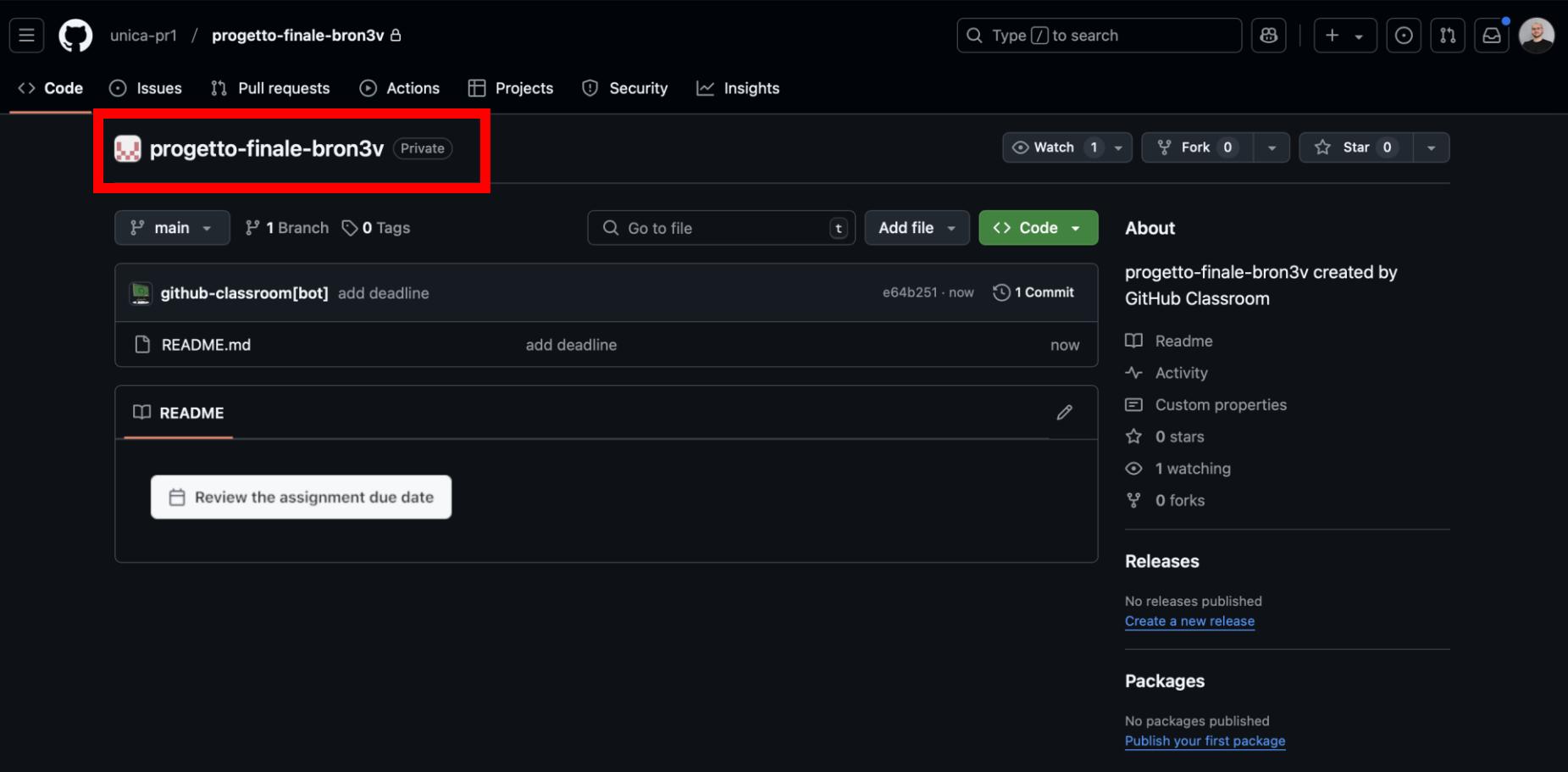
Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. For more information, visit [GitHub Student Developer Pack](#).

Apply



GitHub: iscrizione

Troverete nel vostro profilo una repository chiamata `unica-pr1/progetto-finale-username`



The screenshot shows a GitHub repository page for 'progetto-finale-bron3v'. A red box highlights the repository name 'progetto-finale-bron3v' in the top navigation bar. The repository is private. It contains one branch ('main') and one commit ('e64b251 · now'). The commit was made by 'github-classroom[bot]' and adds a deadline. There is also a 'README.md' file. The repository has 0 stars, 1 watching, and 0 forks. It includes sections for 'About', 'Releases', and 'Packages'.

Code Issues Pull requests Actions Projects Security Insights

progetto-finale-bron3v Private

main 1 Branch 0 Tags

github-classroom[bot] add deadline e64b251 · now 1 Commit

README.md add deadline now

README

Review the assignment due date

About

progetto-finale-bron3v created by GitHub Classroom

Readme Activity Custom properties 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package



GitHub: caricare i file

Per caricare le modifiche del vostro progetto, potete caricare direttamente la cartella dei file tramite il sito:

A screenshot of a GitHub repository page for 'progetto-finale-bron3v'. The interface shows a dark theme. At the top, there are navigation links: Code, Issues, Pull requests, Actions, Projects, Security, and Insights. Below this, the repository name 'progetto-finale-bron3v' is shown as Private. The main area displays a single branch 'main' with 1 branch and 0 tags. A commit from 'github-classroom[bot]' titled 'add deadline' is listed, along with files 'README.md' and 'README'. On the right, there's a 'Code' dropdown menu with options: '+ Create new file' and 'Upload files', with 'Upload files' highlighted by a red box. A message at the bottom says 'Review the assignment due date'.

A screenshot of the GitHub commit changes dialog box. It starts with a large text input field with the placeholder 'Drag files here to add them to your repository' and 'Or choose your files'. Below this is a section titled 'Commit changes' with a red border around it. It contains a text area with the placeholder 'Esempio: aggiunta struttura base' and a list of files: 'Caricati i file main.c, funzioni.c/h, strutture.c/h.'. It also includes a note: 'Nel file main.c è presente la struttura base del progetto. Nel file funzioni.c sono state implementate le funzioni mischiaMazzo e caricaPartita. Nel file strutture.c sono state implementate le strutture Carte e Giocatore.' At the bottom, there are two radio buttons: one for committing directly to the 'main' branch and another for creating a new branch. A green 'Commit changes' button is highlighted with a red box, and a 'Cancel' button is also visible.

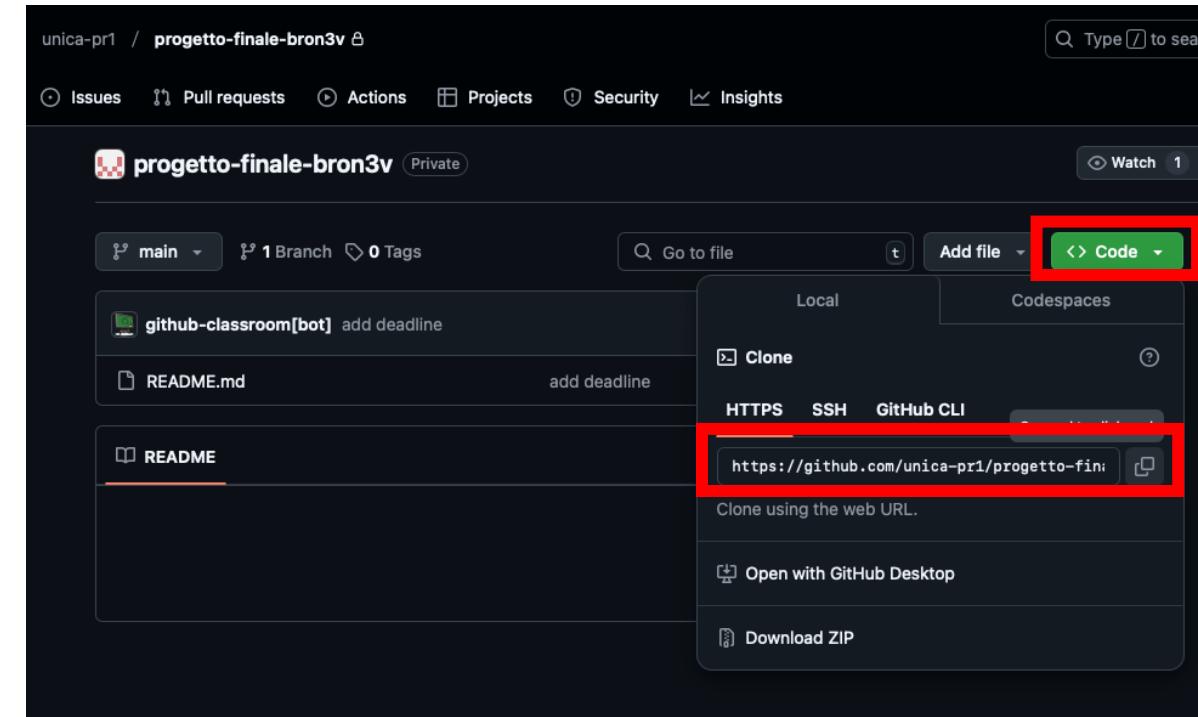
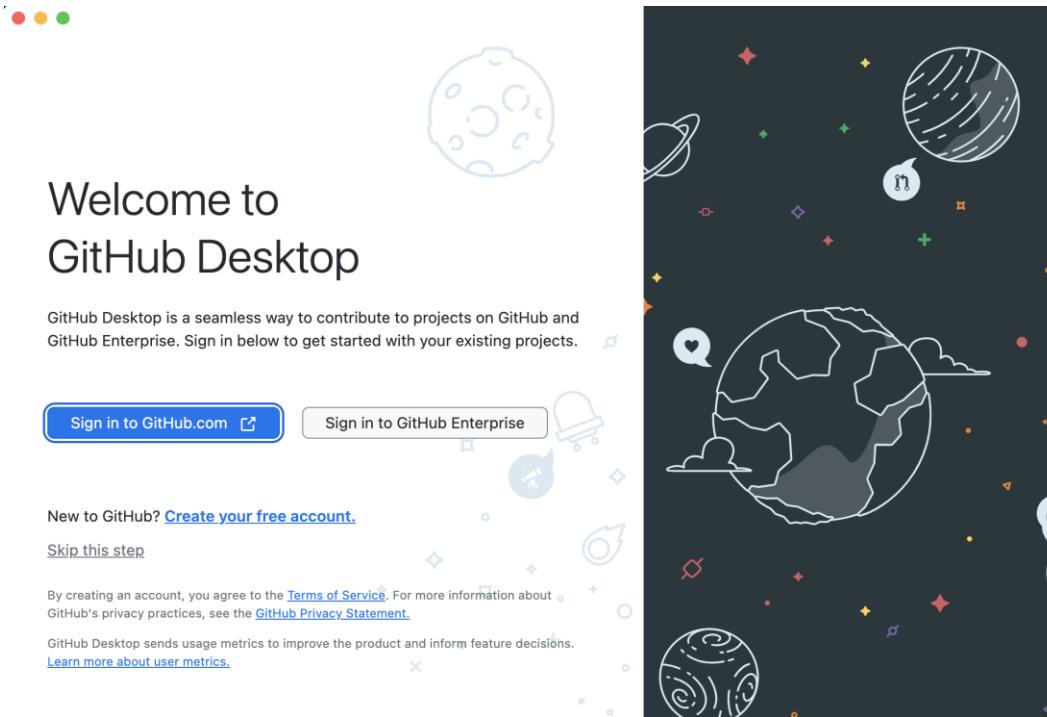


GitHub: caricare i file

In alternativa, potete usare il client di **GitHub Desktop** (in futuro userete prevalentemente questo).

Scaricate l'installer e accedete con le vostre credenziali.

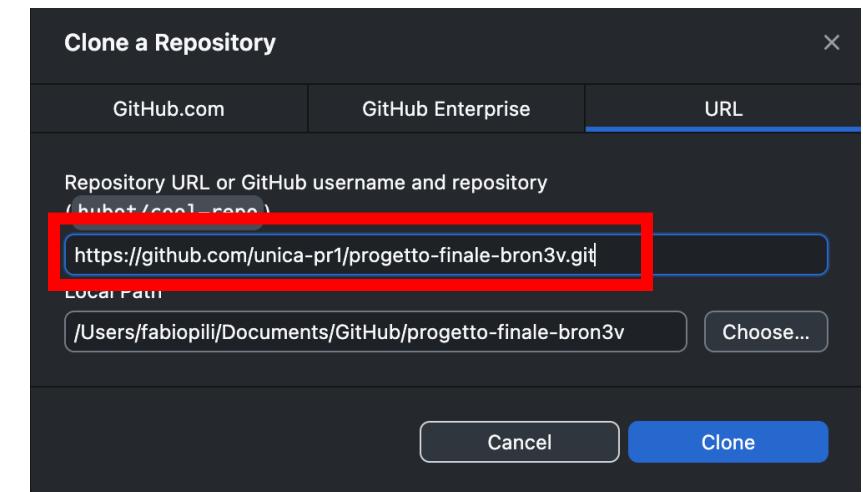
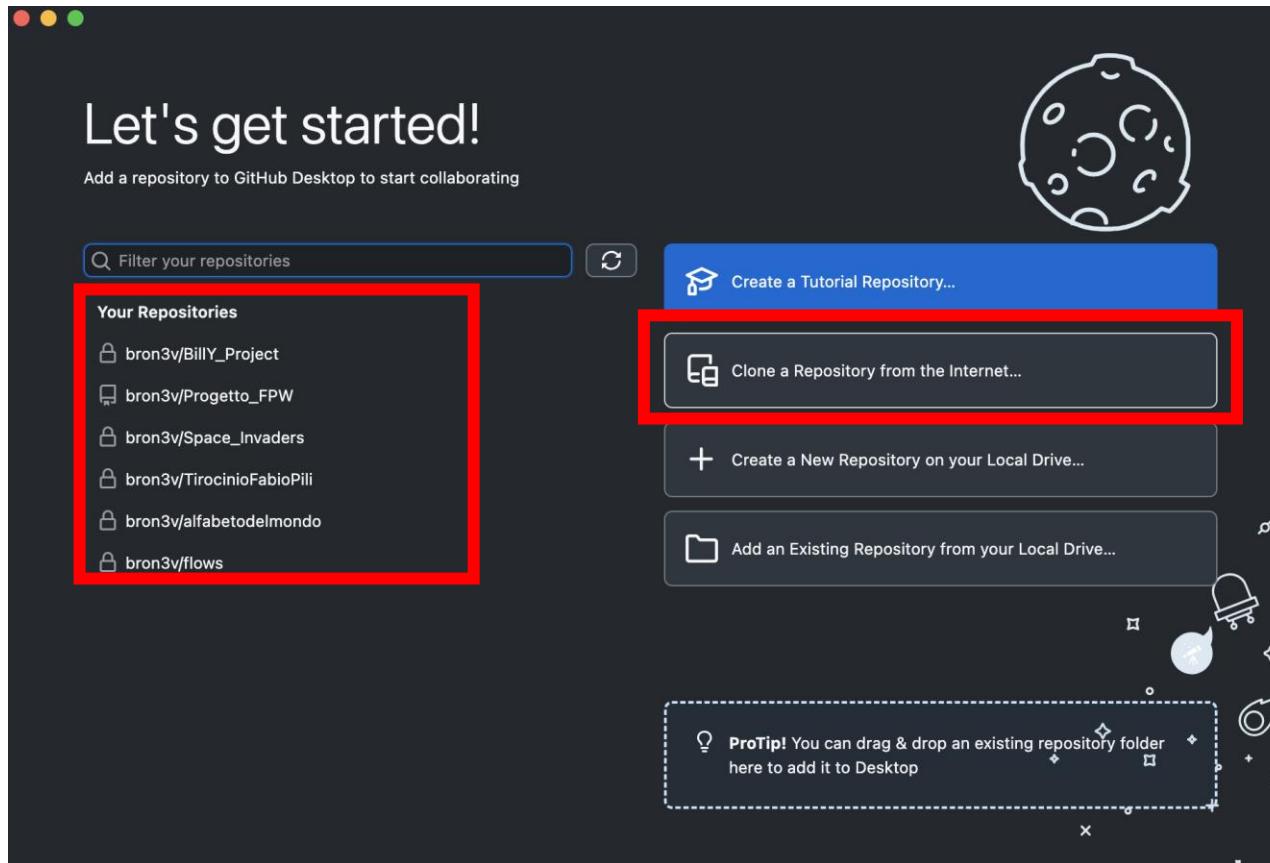
Nella pagina della vostra repository, premete su **<> Code**, e copiate il link della sezione **HTTPS**





GitHub: caricare i file

Una volta effettuato il login, vi troverete davanti alla schermata iniziale. Sulla sinistra troverete le vostre repository (se ne avete) mentre sulla destra il menù per aggiungere/cloneare una repository. Voi dovrete clonare una repository da internet: andate nel tab URL e incollate il link della vostra repository copiato precedentemente.





GitHub: caricare i file

Una volta clonata la repository, vi troverete davanti a questa finestra. Qui troverete tutti i file aggiunti/modificati che dovranno essere caricati. In basso a sinistra troverete le stesse voci che abbiamo visto nella pagina di GitHub, dove inserire titolo e descrizione del commit. Una volta *committato*, premete su **Push origin** per caricare i file.

This screenshot shows the GitHub Desktop application interface. On the left, there's a sidebar with 'Changes' and 'History' tabs. Under 'Changes', two files are selected: 'main.c' and 'main.c'. Below this, a commit message box contains 'Primo commit' and 'Aggiunto il file main.c'. At the bottom is a large blue button labeled 'Commit to main'. A red box highlights the 'Changes' tab and the commit message area. The main pane displays the code differences for 'main.c' between the local and remote branches.

```
@@ -0,0 +1,212 @@
+ #include <stdio.h>
+ #include <stdlib.h>
+
+ #define DIM_STR 32
+
+ typedef struct smartphone {
+     char nome[DIM_STR];
+     char cpu[DIM_STR];
+     int mp;
+     struct smartphone *next;
+ } Smartphone;
+
+ Smartphone *leggiSmartphone();
+ Smartphone *aggiungiInCoda(Smartphone *lista, Smartphone new);
+ Smartphone *deallocaLista(Smartphone *lista);
+ void stampaLista(Smartphone *lista);
+ Smartphone *estraiDaLista(Smartphone *lista, int pos);
+ Smartphone *inserisciInTesta(Smartphone *lista, Smartphone *newTesta);
+ void salvaSmartphone(Smartphone *lista);
+
+ int main(int argc, char **argv) {
+     int choice;
+     Smartphone *node = NULL, *lista = NULL;
+
+     // lettura e inizializzazione lista
+     lista = leggiSmartphone();
```

This screenshot shows the GitHub Desktop application after the commit has been pushed. The 'Changes' tab now shows '0 changed files'. The main pane displays a summary: 'No local changes' with the message 'There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.' It includes a 'Push commits to the origin remote' section with a 'Push origin' button, and three other sections: 'Open the repository in your external editor', 'View the files of your repository in Finder', and 'Open the repository page on GitHub in your browser'. A red box highlights the 'Push origin' button in the top right corner of the main pane.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote

You have 1 local commit waiting to be pushed to GitHub.

Always available in the toolbar when there are local commits waiting to be pushed or ⌘ P

Push origin

Open the repository in your external editor

Select your editor in Settings

Repository menu or ⌘ ⌘ A

Open in Visual Studio Code

Show in Finder

View the files of your repository in Finder

Repository menu or ⌘ ⌘ F

View on GitHub

Open the repository page on GitHub in your browser

Repository menu or ⌘ ⌘ G



GitHub: README

Il file README è un documento che descrive il progetto presente su GitHub. È un biglietto da visita e serve a spiegare lo scopo di esso, fornire istruzioni su come usarlo o in generale documentare il progetto con informazioni tecniche, struttura dei file etc.

Noi vi chiediamo di inserire le seguenti informazioni:

- Nome, cognome e matricola
- Descrizione della struttura del progetto e spiegazione di ogni file .h e .c
- Descrizione di ogni tipo e struttura aggiuntiva
- Descrizione del flusso di gioco
- **SOLO PER L'AVANZATO:** descrizione della logica dell'AI implementata spiegando:
 - Come è stata gestita la randomicità e il controllo degli avversari CPU
 - Se ci sono priorità nella scelta delle carte
 - Se cerca di ostacolare gli avversari e come etc...