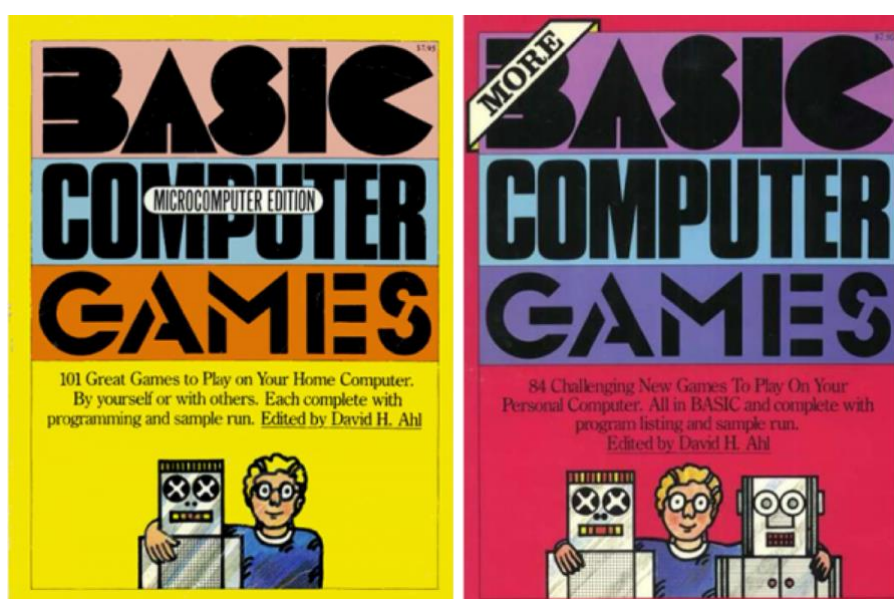


Manuale di Ispirazione per Progetti

Basic Computer Games

21+ giochi per il Computational Making con i ChatBot



*Basato sul classico Basic Computer Games di David H. Ahl
Pubblicato originariamente tra il 1973 e il 1978 da Creative Computing
Adattato per la scuola media e per i laboratori per insegnanti da Gary Stager e Robbo Outbackley*

Questi giochi sono basic, ma non più in BASIC

*Questa è un'opera in lavorazione: si prega di inviare correzioni, aggiunte o cancellazioni a
gary@stager.org*

Indice

Prefazione – Il Computational Making

Fase 1: Fondamenti

1. Hi-Lo — Indovina il numero | Principiante
2. Guess — Indovina il numero | Principiante
3. Stars — Indovina il numero | Principiante
4. Bagels — Logica / Decifrazione di codici | Principiante–Intermedio

Fase 2: Carte, dadi e fortuna

5. Acey Ducey — Gioco di carte | Principiante
6. Craps — Gioco di dadi | Principiante–Intermedio
7. Blackjack — Gioco di carte | Intermedio–Avanzato
8. War — Gioco di carte | Principiante

Fase 3: Strategia e pensiero

9. Nim — Strategia matematica | Intermedio
10. 23 Matches — Strategia matematica | Principiante
11. Reverse — Rompicapo | Intermedio
12. Tic-Tac-Toe — Gioco da tavolo | Intermedio

Fase 4: Stringhe e parole

13. Hangman — Gioco di parole | Intermedio
14. Buzzword — Parole / Umoreismo | Principiante

Fase 5: Simulazione e azione

15. Depth Charge — Ricerca / Militare | Intermedio
16. Gunner — Artiglieria / Fisica | Intermedio
17. Lunar LEM Rocket — Simulazione spaziale | Intermedio–Avanzato

Fase 6: Sistemi complessi

18. Hammurabi — Simulazione economica | Intermedio–Avanzato
19. Football — Simulazione sportiva | Avanzato

Fase 7: IA e apprendimento

20. Animal — IA / Apprendimento | Intermedio–Avanzato
21. Hexapawn — IA / Apprendimento | Avanzato

Prefazione – Il Computational Making

Ritorno al futuro

Questo eBook si fonda sugli ideali dell'educazione progressista (in particolare il costruzionismo), sulla convinzione che i progetti debbano essere la più piccola unità di attenzione di un insegnante, sul riconoscimento che il futuro è computazionale e sulla consapevolezza che l'intelligenza artificiale (IA) può essere usata per potenziare la gamma, l'ampiezza e la profondità dei progetti realizzabili dagli studenti.

Nel nostro libro, [*Invent To Learn – Making, Tinkering, and Engineering in the Classroom*](#), sosteniamo il valore dell'imparare facendo, un'idea senza tempo portata nella modernità da [Seymour Papert](#), [Cynthia Solomon](#) e dai loro colleghi.

L'attuazione efficace di un simile apprendimento basato sui progetti inizia dalla curiosità, generosità e inventiva dell'insegnante, dal rispetto per i bambini, dalla formulazione di prompt di alta qualità, dall'adesione al design generativo e dall'affetto per la serendipità.

Non è necessario avvicinarsi all'intelligenza artificiale con sogni estatici o terrori distopici. Fondamentalmente, l'IA generativa è solo software. Come Ken Kahn dimostra con tanta chiarezza nel suo libro, [*The Learner's Apprentice – AI and the Amplification of Human Creativity*](#), l'intelligenza artificiale può essere al tempo stesso mentore e apprendista, permettendo anche ai bambini di risolvere problemi che i loro insegnanti non avevano mai immaginato.

Il popolare "maker movement" moderno ci ha regalato una maggiore capacità di fabbricare cose con i bit e con gli atomi. Con le tecnologie di fabbricazione possiamo oggi realizzare ogni sorta di oggetti tangibili meravigliosi, con rapidità, economia e precisione. Con il codice possiamo fare lo stesso, AGGIUNGENDO interattività e intelligenza agli oggetti fisici. Purtroppo, le scuole hanno nuovamente lasciato indietro i bit a favore del ritaglio di cartone e di pool noodle. Ciò dipende in gran parte dalla paura e dall'ignoranza degli adulti riguardo al potere conferito dalla programmazione.

Il matematico, sviluppatore di software e MacArthur Genius Stephen Wolfram concorda sul fatto che il futuro è computazionale. "[*Per ogni disciplina X, esiste oggi o esisterà presto un ramo di quella disciplina chiamato Computational X.*](#)" Questo non rappresenta soltanto la frontiera più interessante (e spesso più ludica) di quella disciplina, ma anche il suo settore meglio retribuito. I nuovi e potenti linguaggi di programmazione a blocchi e i chatbot basati sull'IA abbassano le barriere alla democratizzazione dell'informatica, permettendo a chiunque di creare software — anche se è l'unico cliente per quel software.

Tutto ciò porta alla realizzazione odierna di un ideale che animava i primi anni della ricerca in intelligenza artificiale in luoghi come il MIT, dove tra la fine degli anni '60 e l'inizio degli anni '70 l'IA era profondamente interessata ai bambini, a Piaget e al computational making. Uno slogan non ufficiale dell'AI Lab del MIT era infatti "I computer sono per i bambini". Per saperne di più: <https://reggio.constructingmodernknowledge.com/roots>

Questo fenomeno a cerchio completo ci conduce a questo "progetto". Nel 1973, David Ahl scrisse una raccolta di giochi in linguaggio BASIC e ne distribuì copie ciclostilate agli appassionati di informatica ricreativa in tutto il mondo. I principali produttori di hardware pubblicarono versioni di questi giochi sui propri sistemi; nel 1978 uscì la prima edizione commerciale del libro Basic Computer Games. Fu il primo libro di informatica a vendere più di un milione di copie. Pensateci: quasi cinquant'anni fa, un libro di informatica per programmatori hobbisti vendette un milione di copie.

Condividiamo qui idee riassuntive tratte da quel libro come fonte d'ispirazione perché bambini e insegnanti collaborino con i chatbot basati sull'IA e "costruiscano" i propri giochi per computer. Avanti, fate un gioco! Convidetelo con i vostri amici! Potreste sorprendervi di ciò che riuscite a imparare e a realizzare con un piccolo aiuto dai vostri pari e dall'intelligenza artificiale.

Panoramica del libro

Questo manuale presenta 21 giochi selezionati da Basic Computer Games (edizione per microcomputer, 1978) per l'utilizzo in un laboratorio per insegnanti o in un corso di programmazione per la scuola media. I giochi sono disposti in ordine di complessità crescente, suddivisi in 7 fasi e 11 sessioni. Ogni voce fornisce una specifica completa: panoramica, obiettivo, regole, struttura dei turni, condizioni di fine gioco, concetti di programmazione trattati e un esempio di partita.

Fase	Sessioni	Focus	Giochi
Fase 1: Fondamenti	1–2	Variabili, cicli, condizionali, numeri casuali	Hi-Lo, Guess, Stars, Bagels
Fase 2: Carte, dadi e fortuna	3–4	Array, logica del mazzo, scommesse, stato di gioco	Acey Ducey, Craps, Blackjack, War
Fase 3: Strategia e pensiero	5–6	Teoria dei giochi, algoritmi, basi di IA	Nim, 23 Matches, Reverse, Tic-Tac-Toe
Fase 4: Stringhe e parole	7	Manipolazione di stringhe, liste di parole	Hangman, Buzzword
Fase 5: Simulazione e azione	8–9	Coordinate, fisica, decisioni in tempo reale	Depth Charge, Gunner, Lunar LEM
Fase 6: Sistemi complessi	10	Stato a più variabili, gestione delle risorse	Hammurabi, Football
Fase 7: IA e apprendimento	11	Apprendimento automatico, programmi adattivi	Animal, Hexapawn

Fase 1: Fondamenti

Gioco 1: Hi-Lo

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Indovina il numero	Principiante	Illimitata	Numero di tentativi illimitato per round.

Concetti di programmazione: Variabili, INPUT/PRINT, IF/THEN, numeri casuali

Panoramica

Il più semplice gioco di indovinelli possibile. Il computer sceglie un numero casuale tra 1 e 100 e tu devi indovinarlo. Troppo alto? Troppo basso? Il computer te lo dice. Un'aggiunta di stampo "gioco d'azzardo" introduce la posta in gioco: inizi con 1000 \$ e vinci o perdi denaro a ogni tentativo.

Obiettivo

Indovina il numero segreto scelto dal computer tra 1 e 100 gestendo al tempo stesso il tuo capitale.

Regole

1. Il computer sceglie un numero intero casuale da 1 a 100.
2. Il giocatore inserisce una proposta.
3. Il computer risponde: "TROPPO ALTO", "TROPPO BASSO" oppure "INDOVINATO!".
4. Un'ipotesi corretta fa guadagnare 100 \$. Un'ipotesi sbagliata costa l'importo della puntata.
5. Se il capitale scende a 0 \$, la partita finisce. Altrimenti inizia un nuovo round.
6. Il giocatore può ritirarsi in qualunque momento tra un round e l'altro.

Struttura dei turni e durata

Turni massimi: Numero di tentativi illimitato per round. I round proseguono finché il giocatore si ritira o resta senza soldi.

Struttura di gioco: Illimitata — il gioco continua di round in round fino a quando il giocatore decide di smettere o esaurisce il denaro.

Condizioni di fine gioco

Vittoria del round: indovinare il numero corretto (si vincono 100 \$). Sconfitta del round: ogni ipotesi sbagliata costa l'importo della puntata. Fine partita: il capitale raggiunge 0 \$ oppure il giocatore sceglie di smettere.

Esempio di partita

```
HI-LO
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY
```

```
HAI 1000 $.
ESTRARRÒ UN NUMERO TRA 1 E 100.
```

```
IL TUO TENTATIVO? 50
TROPPO ALTO.
```

```
IL TUO TENTATIVO? 25
TROPPO BASSO.
```

IL TUO TENTATIVO? 37
TROPPO BASSO.

IL TUO TENTATIVO? 42
INDOVINATO!!!! HAI VINTO 100 \$!
IL TUO TOTALE È ORA 1100 \$.
GIOCHI ANCORA (SI O NO)? SI

Gioco 2: Guess

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Indovina il numero	Principiante	Fino alla vittoria	Tentativi illimitati, ma il computer registra il numero ottimale e commenta la tua prestazione.

Concetti di programmazione: Cicli WHILE, variabili contatore, logaritmi (opzionale), strategia di ricerca binaria

Panoramica

Un passo avanti rispetto a Hi-Lo. Il giocatore sceglie il limite superiore e il computer conosce il numero ottimale di tentativi (logaritmo in base 2 del limite). Questo introduce naturalmente la ricerca binaria e permette agli studenti di scoprire da soli la strategia.

Obiettivo

Indovina il numero del computer nel minor numero possibile di tentativi, idealmente pareggiando o battendo l'ottimo teorico.

Regole

1. Il giocatore sceglie un limite superiore (ad esempio 100, 1000).
2. Il computer sceglie un numero intero casuale da 1 fino a quel limite.
3. Il giocatore prova; il computer risponde "TROPPO ALTO" o "TROPPO BASSO".
4. Quando il giocatore indovina, il computer comunica quanti tentativi sono stati necessari.
5. Il computer calcola il minimo teorico (parte intera superiore di \log_2 del limite) e commenta la prestazione del giocatore.
6. Il giocatore può scegliere di giocare di nuovo con lo stesso limite o con uno diverso.

Struttura dei turni e durata

Turni massimi: Tentativi illimitati, ma il computer registra il numero ottimale e commenta la tua prestazione.

Struttura di gioco: Fino alla vittoria — ogni round termina quando il giocatore indovina. Non c'è un limite massimo di turni.

Condizioni di fine gioco

Vittoria: il giocatore indovina il numero corretto. Il computer comunica il numero di tentativi e lo confronta con l'ottimo teorico. Il giocatore può allora scegliere se giocare di nuovo o smettere.

Esempio di partita

```
GUESS
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

QUALE LIMITE VUOI? 100
STO PENSANDO A UN NUMERO TRA 1 E 100.
DOVRESTI RIUSCIRCI IN AL MASSIMO 7 TENTATIVI.

IL TUO TENTATIVO? 50
TROPPO BASSO. PROVA UN NUMERO PIÙ GRANDE.
```

IL TUO TENTATIVO? 75
TROPPO ALTO. PROVA UN NUMERO PIÙ PICCOLO.

IL TUO TENTATIVO? 62
TROPPO ALTO.

IL TUO TENTATIVO? 56
ECCOLO! CI SEI RIUSCITO IN 4 TENTATIVI.
MOLTO BENE

Gioco 3: Stars

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Indovina il numero	Principiante	Lunghezza fissa	Al massimo 7 tentativi per round.

Concetti di programmazione: Funzione ABS(), mappatura di numeri in intervalli, ripetizione di stringhe, limiti fissi di tentativi

Panoramica

Come Guess, ma invece di "troppo alto/troppo basso" il computer mostra delle stelline per indicare la vicinanza. Una stella significa molto lontano; sette stelle molto vicino. Richiede una strategia di ricerca completamente diversa e insegna agli studenti a tradurre una distanza numerica in un feedback visivo.

Obiettivo

Indovina il numero segreto del computer (1–100) entro 7 tentativi, guidato solo dagli indizi di vicinanza sotto forma di stelle.

Regole

1. Il computer sceglie un numero casuale da 1 a 100.
2. Il giocatore dispone di esattamente 7 tentativi.
3. Dopo ogni tentativo, il computer stampa da 1 a 7 stelle: più stelle = più vicino al bersaglio.
4. Corrispondenza delle stelle: 1 stella (distanza ≥ 64), 2 stelle (32–63), 3 stelle (16–31), 4 stelle (8–15), 5 stelle (4–7), 6 stelle (2–3), 7 stelle (distanza 1).
5. Una corrispondenza esatta termina immediatamente il round con un messaggio di congratulazioni.
6. Dopo 7 tentativi errati, il numero viene rivelato.

Struttura dei turni e durata

Turni massimi: Al massimo 7 tentativi per round.

Struttura di gioco: Lunghezza fissa — sono consentiti esattamente 7 tentativi. Il round termina con una risposta corretta oppure dopo il settimo tentativo.

Condizioni di fine gioco

Vittoria: indovina il numero entro 7 tentativi. Sconfitta: non lo indovini in 7 tentativi; il computer rivela il numero. Fine partita: il giocatore può scegliere di giocare di nuovo o smettere.

Esempio di partita

```
STARS
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

STO PENSANDO A UN NUMERO. COMINCIA A INDOVINARE.

IL TUO TENTATIVO? 50
**

IL TUO TENTATIVO? 75
*
```

IL TUO TENTATIVO? 25

IL TUO TENTATIVO? 30

IL TUO TENTATIVO? 31
CI SEI RIUSCITO IN 5 TENTATIVI!!!

Gioco 4: Bagels

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Logica / Decifrazione di codici	Principiante–Intermedio	Lunghezza fissa	Al massimo 20 tentativi per round.

Concetti di programmazione: Confronto di stringhe, estrazione di cifre, cicli FOR, logica a più condizioni

Panoramica

Il computer sceglie un numero di 3 cifre tutte diverse. Tu provi e il computer risponde con indizi in codice: PICO (cifra giusta, posizione sbagliata), FERMI (cifra giusta, posizione giusta), BAGELS (nulla di corretto). È sostanzialmente un Mastermind numerico e insegna la deduzione sistematica.

Obiettivo

Deduci il numero segreto di 3 cifre scelto dal computer entro 20 tentativi usando gli indizi PICO, FERMI e BAGELS.

Regole

1. Il computer sceglie un numero di 3 cifre usando cifre da 0 a 9, senza ripetizioni.
2. Il giocatore inserisce un tentativo di 3 cifre.
3. Per ogni cifra: se coincide nella stessa posizione il computer dice FERMI; se coincide ma in posizione diversa, PICO; se nessuna cifra coincide, BAGELS.
4. Nella stessa risposta possono comparire più indizi PICO/FERMI (ad esempio "PICO FERMI").
5. Una risposta di tre FERMI significa che il giocatore ha vinto.
6. Dopo 20 tentativi falliti il numero viene rivelato e il round termina.

Struttura dei turni e durata

Turni massimi: Al massimo 20 tentativi per round.

Struttura di gioco: Lunghezza fissa — fino a 20 tentativi per round.

Condizioni di fine gioco

Vittoria: indovina l'intero numero di 3 cifre (tutte e tre le cifre corrette nelle posizioni corrette).

Sconfitta: esaurisci tutti i 20 tentativi senza la risposta giusta; il computer rivela il numero. Fine

partita: il giocatore può scegliere di giocare di nuovo o smettere.

Esempio di partita

BAGELS
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

OK, HO PENSATO A UN NUMERO.

TENTATIVO N. 1
? 123
PICO FERMI

TENTATIVO N. 2
? 456
BAGELS

TENTATIVO N. 3

? 178
FERMI FERMI

TENTATIVO N. 4
? 172
CI SEI RIUSCITO!!!

Fase 2: Carte, dadi e fortuna

Gioco 5: Acey Ducey

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Gioco di carte	Principiante	Illimitata	Round illimitati.

Concetti di programmazione: Generazione di numeri casuali, rappresentazione delle carte, logica delle scommesse, totali progressivi

Panoramica

Il mazziniere scopre due carte. Tu scommetti se la carta successiva sarà compresa tra le due. Ampio divario tra le carte? Scommessa sicura. Divario stretto? Rischiosa. Scommetti zero per passare. È semplice, veloce e insegna il ragionamento sulla probabilità.

Obiettivo

Massimizza il tuo capitale scommettendo saggiamente sul fatto che la carta successiva cada tra le due carte già scoperte.

Regole

1. Il giocatore parte con 100 \$.
2. Ogni round, il computer distribuisce due carte scoperte (valori 2–14, dove 11 = J, 12 = Q, 13 = K, 14 = Asso).
3. Il giocatore piazza la puntata (0 \$ per saltare, oppure qualunque importo fino al suo capitale).
4. Viene distribuita una terza carta. Se il suo valore è strettamente compreso tra le prime due, il giocatore vince l'importo puntato. Altrimenti lo perde.
5. Se il capitale del giocatore arriva a 0 \$, la partita termina.
6. Puntare 0 \$ fa sì che il computer ti chiami "CHICKEN!!" (pollo!), ma non c'è scambio di denaro.

Struttura dei turni e durata

Turni massimi: Round illimitati. Si continua finché il giocatore è in rovina o si ritira.

Struttura di gioco: Illimitata — un round di scommessa dopo l'altro, senza un punto di arrivo prefissato.

Condizioni di fine gioco

Sconfitta: il capitale raggiunge 0 \$ (il computer ti annuncia che sei in rovina e ti propone di ricominciare). Ritiro: il giocatore interrompe il programma o sceglie di non proseguire. Non esiste una condizione di "vittoria"; l'obiettivo è accumulare quanto più denaro possibile.

Esempio di partita

ACEY DUCEY CARD GAME
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

ORA HAI 100 DOLLARI

ECCO LE TUE DUE CARTE:

3

J

QUAL È LA TUA PUNTATA? 25

8

HAI VINTO!!!

ORA HAI 125 DOLLARI

ECCO LE TUE DUE CARTE:

9

10

QUAL È LA TUA PUNTATA? 0

POLLO!!

Gioco 6: Craps

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Gioco di dadi	Principiante– Intermedio	Fino a vittoria/sconfitta per round	Ogni round dura 1 lancio (natural/craps) o lanci illimitati (fase del point).

Concetti di programmazione: Stato di gioco a più fasi, cicli DO/WHILE, condizioni composte, simulazione di dadi

Panoramica

Il vero gioco di dadi da casinò, simulato fedelmente. Il come-out, la fase del point, i natural, i craps: gli studenti adorano i dadi e lo stato di gioco a due fasi è un ottimo strumento per insegnare il controllo del flusso del programma.

Obiettivo

Vinci denaro ottenendo combinazioni di dadi vincenti secondo le regole standard del craps.

Regole

1. Il giocatore piazza una puntata.
2. Lancio di come-out: si lanciano due dadi. 7 o 11 = vittoria immediata (natural). 2, 3 o 12 = sconfitta immediata (craps). Qualsiasi altro totale (4, 5, 6, 8, 9, 10) diventa il "point".
3. Fase del point: il giocatore continua a lanciare. Ottenere il point = vittoria. Ottenere un 7 = sconfitta ("seven out"). Qualunque altro numero = si lancia di nuovo.
4. Le vincite sono pari all'importo puntato. Le perdite sottraggono l'importo puntato.
5. Il giocatore può ritirarsi tra un round e l'altro.

Struttura dei turni e durata

Turni massimi: Ogni round dura 1 lancio (natural/craps) o lanci illimitati (fase del point). I round proseguono finché il giocatore non si ritira.

Struttura di gioco: Fino a vittoria/sconfitta per round — ogni round termina con un lancio decisivo. Il gioco nel suo complesso è illimitato di round in round.

Condizioni di fine gioco

Vittoria del round: ottenere 7 o 11 al lancio di come-out (natural), oppure ottenere il proprio point prima di un 7. Sconfitta del round: ottenere 2, 3 o 12 al lancio di come-out (craps), oppure ottenere un 7 prima del proprio point. Fine partita: il giocatore decide di smettere o il capitale arriva a 0 \$.

Esempio di partita

```
CRAPS
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY
```

```
PUNTATA? 10
5 PIÙ 6 --- TOTALE = 11
NATURAL....VITTORIA!!!
HAI VINTO 10 DOLLARI.
```

```
PUNTATA? 20
3 PIÙ 1 --- TOTALE = 4
IL TUO POINT È 4. LANCIO DI NUOVO
```

6 PIÙ 3 --- TOTALE = 9
1 PIÙ 3 --- TOTALE = 4
HAI OTTENUTO IL POINT!
HAI VINTO 20 DOLLARI.

Gioco 7: Blackjack

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Gioco di carte	Intermedio–Avanzato	Fino a vittoria/sconfitta per mano	Ogni mano consente un numero illimitato di hit.

Concetti di programmazione: Array, mescolamento del mazzo, logica multi-giocatore, condizionali complessi, valutazione dell'asso

Panoramica

Il gioco di carte più iconico al mondo. Anche una versione semplificata per un solo giocatore insegna la gestione del mazzo, la valutazione della mano e le decisioni a più rami (hit, stand, double down). Tutti gli studenti ne conoscono già le regole.

Obiettivo

Batti il banco ottenendo una mano con un valore più vicino a 21 senza sfiorare.

Regole

1. Il giocatore piazza una puntata. Il banco dà 2 carte al giocatore (scoperte) e 2 a se stesso (una scoperta, una coperta).
2. Valori delle carte: 2–10 valore nominale; J/Q/K = 10; Asso = 1 o 11 (a seconda di ciò che conviene alla mano).
3. Opzioni del giocatore: HIT (prendi un'altra carta), STAND (tieni la mano attuale), DOUBLE DOWN (raddoppia la puntata e prendi esattamente un'altra carta).
4. Se il totale del giocatore supera 21, sballa e perde immediatamente.
5. Il banco scopre la carta coperta e deve chiamare carta con 16 o meno e fermarsi con 17 o più.
6. Vince chi si avvicina di più a 21 senza sfiorare. Pareggio = push.

Struttura dei turni e durata

Turni massimi: Ogni mano consente un numero illimitato di hit. I round proseguono finché il giocatore si ritira o resta senza soldi.

Struttura di gioco: Fino a vittoria/sconfitta per mano — ogni mano si risolve quando sia il giocatore sia il banco stanno o sballano. La sessione complessiva è illimitata di mano in mano.

Condizioni di fine gioco

Vittoria della mano: il totale del giocatore è più vicino a 21 di quello del banco senza sfiorare, oppure il banco sballa. Sconfitta della mano: il giocatore supera 21 (bust), oppure il banco ha un totale più vicino a 21. Push: entrambe le mani hanno lo stesso valore (non cambia nulla nel denaro). Blackjack: mano da due carte dal valore esatto di 21 (paga 3:2). Fine partita: il giocatore decide di ritirarsi oppure il capitale è esaurito.

Esempio di partita

BLACKJACK
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

PUNTATA? 10

LA TUA MANO:

4

J

IL BANCO MOSTRA: 6

CARTA? SI

7 -- TOTALE 21

BANCO: 6 9 3 = 18

HAI VINTO 10 \$!

Gioco 8: War

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Gioco di carte	Principiante	Lunghezza fissa	Esattamente 26 round (52 carte distribuite 2 alla volta), salvo abbandono anticipato.

Concetti di programmazione: Mescolamento di array, rappresentazione di un mazzo di carte, operatori di confronto, cicli FOR, tenuta del punteggio

Panoramica

Il gioco di carte più semplice possibile: scopri e confronta. Nonostante la semplicità, insegna l'intero ciclo di vita di un mazzo: creazione, mescolamento, distribuzione e confronto. Gli studenti possono arricchirlo con la meccanica dello spareggio "war".

Obiettivo

Vinci più round del computer dopo aver giocato l'intero mazzo.

Regole

1. Si mescola un mazzo standard di 52 carte.
2. A ogni round viene distribuita una carta al giocatore e una al computer.
3. La carta dal valore più alto vince il round. In caso di parità il punteggio non cambia.
4. Le carte vengono mostrate con seme e numero (ad esempio "S-7" = 7 di picche).
5. Dopo 26 round (o al ritiro del giocatore) viene mostrato il punteggio finale.
6. Vince la partita chi ha ottenuto più vittorie.

Struttura dei turni e durata

Turni massimi: Esattamente 26 round (52 carte distribuite 2 alla volta), salvo abbandono anticipato.

Struttura di gioco: Lunghezza fissa — si gioca l'intero mazzo di 52 carte in 26 confronti.

Condizioni di fine gioco

Vittoria: il giocatore ha più round vinti del computer dopo 26 round. Sconfitta: il computer ha più vittorie. Pareggio: stesso numero di vittorie. Abbandono anticipato: il giocatore può scegliere di fermarsi tra due round qualsiasi; il punteggio finale viene riportato in quel momento.

Esempio di partita

```
WAR
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY
```

```
TU: P-7   IO: C-3
VINCI TU!
```

```
TU: Q-10  IO: F-J
VINCO IO!
```

```
TU: C-4   IO: C-9
VINCO IO!
```

TU: P-A IO: Q-5
VINCI TU!

PUNTEGGIO: TU--2 COMPUTER--2

Fase 3: Strategia e pensiero

Gioco 9: Nim

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Strategia matematica	Intermedio	Fino a vittoria/sconfitta	Variabile — dipende dalla dimensione delle pile.

Concetti di programmazione: Operazioni binarie/XOR, teoria dei giochi, strategia ottimale, cicli WHILE

Panoramica

Uno dei giochi di strategia più antichi esistenti. I giocatori prelevano oggetti da delle pile e vince chi prende l'ultimo oggetto. La lezione profonda: questo gioco ha una soluzione matematica perfetta (la nim-sum), e il computer la conosce. Gli studenti comprendono cosa significhi che un computer giochi in modo ottimale.

Obiettivo

Essere il giocatore che prende l'ultimo oggetto (oppure costringere l'avversario a prenderlo, a seconda della variante).

Regole

1. Il giocatore sceglie il numero di pile e quanti oggetti ci siano in ciascuna.
2. I giocatori si alternano. A ogni turno un giocatore rimuove un numero qualsiasi di oggetti (almeno 1) da una sola pila.
3. Vince il giocatore che prende l'ultimo oggetto (gioco normale).
4. Il computer utilizza la strategia binaria della nim-sum per giocare in modo ottimale.
5. Se la nim-sum di tutte le pile è 0 all'inizio del turno del computer, il computer è in posizione perdente (se il giocatore gioca in modo ottimale).

Struttura dei turni e durata

Turni massimi: Variabile — dipende dalla dimensione delle pile. Ogni turno rimuove almeno 1 oggetto, quindi la partita termina sempre.

Struttura di gioco: Fino a vittoria/sconfitta — si gioca finché tutte le pile sono vuote.

Condizioni di fine gioco

Vittoria: prendi l'ultimo oggetto (o gli ultimi oggetti) dall'ultima pila (gioco normale), oppure il tuo avversario è costretto a prendere l'ultimo oggetto (variante misère). Sconfitta: l'opposto. La partita termina sempre perché il numero totale di oggetti diminuisce a ogni turno.

Esempio di partita

```
NIM
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY
```

```
NUMERO DI PILE? 3
PILA 1? 3
PILA 2? 5
PILA 3? 7
```

È IL TUO TURNO. QUALE PILA? 2
QUANTI? 4

PILA 1: 3 PILA 2: 1 PILA 3: 7

TOLGO 6 DALLA PILA 3.
PILA 1: 3 PILA 2: 1 PILA 3: 1

Gioco 10: 23 Matches (23 Fiammiferi)

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Strategia matematica	Principiante	Fino a vittoria/sconfitta	Da 8 a 23 turni in totale (dipende da quanti fiammiferi prende ogni giocatore).

Concetti di programmazione: Aritmetica modulare, scoperta della strategia, Nim semplificato

Panoramica

Una versione vincolata del Nim con esattamente 23 fiammiferi e la regola prendi-1-2-o-3. Abbastanza semplice da permettere agli studenti di scoprire la strategia vincente (lasciare sempre un multiplo di 4) dopo pochi round. Una porta d'accesso perfetta alla teoria dei giochi.

Obiettivo

Costringi il tuo avversario a prendere l'ultimo fiammifero.

Regole

1. Sul tavolo ci sono 23 fiammiferi.
2. I giocatori si alternano. A ogni turno un giocatore rimuove 1, 2 o 3 fiammiferi.
3. Perde il giocatore costretto a prendere l'ultimo fiammifero.
4. Il computer conosce la strategia ottimale: lasciare sempre un numero di fiammiferi pari a 1 più un multiplo di 4 (cioè 1, 5, 9, 13, 17, 21).

Struttura dei turni e durata

Turni massimi: Da 8 a 23 turni in totale (dipende da quanti fiammiferi prende ogni giocatore). Termina sempre.

Struttura di gioco: Fino a vittoria/sconfitta — i turni si alternano finché resta esattamente 1 fiammifero.

Condizioni di fine gioco

Vittoria: il tuo avversario è costretto a prendere l'ultimo (23°) fiammifero. Sconfitta: sei tu a essere costretto a prenderlo. Il gioco termina sempre perché il conteggio dei fiammiferi diminuisce di 1–3 a ogni turno.

Esempio di partita

```
23 MATCHES  
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY
```

```
23 FIAMMIFERI SUL TAVOLO.
```

```
È IL TUO TURNO. QUANTI? 3  
RESTANO 20 FIAMMIFERI.
```

```
NE PRENDO 3.  
RESTANO 17 FIAMMIFERI.
```

```
È IL TUO TURNO. QUANTI? 2  
RESTANO 15 FIAMMIFERI.
```

NE PRENDO 2.
RESTANO 13 FIAMMIFERI.

Gioco 11: Reverse

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Rompicapo	Intermedio	Fino alla vittoria	Mosse illimitate.

Concetti di programmazione: Inversione di array, progettazione di algoritmi, pensiero euristico vs. algoritmico

Panoramica

Un rompicapo di ordinamento unico: riordina i numeri 1–9 invertendo ripetutamente i primi N elementi. Gli studenti devono pensare in termini di algoritmi (soluzione garantita in $2N-3$ mosse) contro euristiche (mosse opportunistiche che potrebbero fare meglio). È uno dei giochi più stimolanti del libro.

Obiettivo

Ordina l'elenco mescolato dei numeri da 1 a 9 in ordine crescente utilizzando solo operazioni di inversione.

Regole

1. Il computer genera una permutazione casuale dei numeri da 1 a 9.
2. A ogni turno, il giocatore inserisce un numero N (da 1 a 9).
3. I primi N numeri della lista vengono invertiti sul posto.
4. Inserendo 0 si esce dal gioco.
5. Quando la lista è ordinata in modo crescente, il giocatore vince e viene mostrato il numero totale di mosse.
6. Esempio: lista [2,4,5,1,9,6,3,7,8]. Inverti 4 → [1,5,4,2,9,6,3,7,8]. Inverti 5 → [9,2,4,5,1,6,3,7,8].

Struttura dei turni e durata

Turni massimi: Mosse illimitate. Non c'è penalità per aver fatto molte mosse, ma meno è meglio.

Struttura di gioco: Fino alla vittoria — si gioca finché la lista non è ordinata o il giocatore si arrende (inverte 0).

Condizioni di fine gioco

Vittoria: la lista si legge 1 2 3 4 5 6 7 8 9 da sinistra a destra. Il computer comunica quante mosse sono state necessarie. Ritiro: il giocatore inserisce 0 e termina la partita senza risolverla. Una soluzione esiste sempre.

Esempio di partita

```
REVERSE  
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY
```

```
LA LISTA È:  
2 4 5 1 9 6 3 7 8
```

```
QUANTI NE INVERTO? 4  
1 5 4 2 9 6 3 7 8
```

```
QUANTI NE INVERTO? 9  
8 7 3 6 9 2 4 5 1
```

QUANTI NE INVERTO? 8
5 4 2 9 6 3 7 8 1

QUANTI NE INVERTO? 9
1 8 7 3 6 9 2 4 5

Gioco 12: Tic-Tac-Toe (Tris)

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Gioco da tavolo	Intermedio	Lunghezza fissa	Al massimo 9 mosse (5 per il primo giocatore, 4 per il secondo).

Concetti di programmazione: Array 2D, verifica delle condizioni di vittoria, IA semplice, alberi di gioco

Panoramica

Tutti gli studenti lo conoscono. Programmarlo trasforma un gioco d'infanzia in una pietra miliare della programmazione. La sola logica di verifica della vittoria (8 linee possibili) insegna il pensiero metodico, e l'aggiunta di un avversario computerizzato introduce le basi dell'IA.

Obiettivo

Metti tre dei tuoi segni (X) in fila su una griglia 3×3 prima che lo faccia il computer (O).

Regole

1. La scacchiera è una griglia 3×3, con le posizioni numerate da 1 a 9.
2. Il giocatore è X; il computer è O. Il giocatore muove per primo.
3. A ogni turno, il giocatore inserisce il numero di una casella vuota.
4. Il computer effettua poi la propria mossa.
5. Dopo ogni mossa il programma controlla tutte le 8 possibili linee vincenti (3 righe, 3 colonne, 2 diagonali).
6. Il primo giocatore a completare una linea vince. Se la scacchiera si riempie senza vincitore, è patta.

Struttura dei turni e durata

Turni massimi: Al massimo 9 mosse (5 per il primo giocatore, 4 per il secondo). Termina sempre.

Struttura di gioco: Lunghezza fissa — la partita finisce quando qualcuno allinea tre segni o quando tutte le 9 caselle sono occupate.

Condizioni di fine gioco

Vittoria: un giocatore mette tre segni in linea orizzontale, verticale o diagonale. Patta: tutte le 9 caselle sono occupate senza tre in fila. Il computer gioca in modo ottimale, quindi al massimo un giocatore perfetto può ottenere una patta.

Esempio di partita

TIC-TAC-TOE
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

```
1 | 2 | 3
-----
4 | 5 | 6
-----
7 | 8 | 9
```

LA TUA MOSSA? 5

. | . | .

. | X | .

. | . | .

MUOVO IN 1.

O | . | .

. | X | .

. | . | .

Fase 4: Stringhe e parole

Gioco 13: Hangman (L'impiccato)

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Gioco di parole	Intermedio	Fino a vittoria/sconfitta	10 tentativi errati consentiti (uno per ogni parte del corpo: testa, busto, braccio sinistro, braccio destro, gamba sinistra, gamba destra, mano sinistra, mano destra, piede sinistro, piede destro).

Concetti di programmazione: Ricerca in stringhe, array di caratteri, costruzione di una visualizzazione ASCII, liste di parole

Panoramica

Il gioco di parole universalmente conosciuto. Programmarlo insegna la ricerca in stringhe (la lettera proposta è nella parola?), la costruzione di una stringa da aggiornare e il tracciamento delle lettere già usate. La figura visiva dell'impiccato aggiunge un feedback progressivo.

Obiettivo

Indovina tutte le lettere della parola nascosta prima che la figura dell'impiccato sia completata.

Regole

1. Il computer sceglie una parola da una lista predefinita.
2. La parola è mostrata come una serie di trattini (uno per ogni lettera).
3. Il giocatore propone una lettera alla volta.
4. Se la lettera è nella parola, tutte le occorrenze vengono rivelate nelle posizioni corrette.
5. Se la lettera non è nella parola, viene aggiunta una parte del corpo alla figura dell'impiccato e il contatore degli errori aumenta.
6. Non si dovrebbero riproporre lettere già tentate.
7. Dopo 10 errori, l'impiccato è completo e la partita è persa.

Struttura dei turni e durata

Turni massimi: 10 tentativi errati consentiti (uno per ogni parte del corpo: testa, busto, braccio sinistro, braccio destro, gamba sinistra, gamba destra, mano sinistra, mano destra, piede sinistro, piede destro). Le ipotesi corrette non contano come turni.

Struttura di gioco: Fino a vittoria/sconfitta — si gioca finché la parola non è interamente rivelata o non si raggiungono 10 errori.

Condizioni di fine gioco

Vittoria: tutte le lettere della parola sono state indovinate. Sconfitta: il giocatore commette 10 errori (l'impiccato è completato) e la parola viene rivelata. Il giocatore può quindi scegliere di giocare di nuovo con una nuova parola.

Esempio di partita

HANGMAN
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

LA PAROLA È: -----

INDOVINA UNA LETTERA? E
-E-----

INDOVINA UNA LETTERA? A
SPIACENTE, NON È NELLA PAROLA. ERRORE N. 1

INDOVINA UNA LETTERA? O
-E--O-

INDOVINA UNA LETTERA? R
-E--OR

INDOVINA UNA LETTERA? N
-EN-OR

INDOVINA UNA LETTERA? M
MENTOR
CI SEI RIUSCITO!

Gioco 14: Buzzword

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Parole / Umore	Principiante	Illimitata	Illimitato.

Concetti di programmazione: Array di stringhe, selezione casuale, concatenazione di stringhe, istruzioni DATA

Panoramica

Il gioco più leggero del laboratorio — ma gli studenti lo adorano. Tre array di parole gergali vengono combinati casualmente per produrre sciocchezze dal suono altisonante. Veloce da programmare, immediatamente divertente, insegna l'indicizzazione degli array e la costruzione di stringhe.

Obiettivo

Genera frasi di tre parole chiave casuali dal suono altisonante.

Regole

1. Si definiscono tre array di aggettivi/sostantivi (tipicamente 12 parole ciascuno).
2. Quando il giocatore risponde "SI" (o preme invio), il computer sceglie casualmente una parola da ciascuna delle tre colonne.
3. Le tre parole vengono stampate come un'unica frase.
4. Al giocatore viene chiesto se ne vuole un'altra. "NO" fa uscire dal programma.

Struttura dei turni e durata

Turni massimi: Illimitato. Il giocatore chiede frasi una alla volta.

Struttura di gioco: Illimitata — il programma genera una frase per richiesta finché il giocatore non risponde "NO".

Condizioni di fine gioco

Ritiro: il giocatore risponde "NO" quando gli viene chiesto se ne vuole un'altra. Non esiste una condizione di vittoria o sconfitta: questo è un generatore, non una competizione.

Esempio di partita

```
BUZZWORD  
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY
```

```
PRONTO? SI  
GESTIONE TOTALE DELLE OPZIONI
```

```
PRONTO? SI  
FLESSIBILITÀ RECIPROCA INTEGRATA
```

```
PRONTO? SI  
PROIEZIONE TRANSITORIA COMPATIBILE
```

```
PRONTO? NO  
TORNA QUANDO AVRAI BISOGNO DI ALTRI BUZZWORD...
```

Fase 5: Simulazione e azione

Gioco 15: Depth Charge (Carica di profondità)

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Ricerca / Militare	Intermedio	Lunghezza fissa	Il numero di tentativi consentiti è pari alla parte intera superiore di $\log_2(N+1)$, dove N è la dimensione dell'area di ricerca.

Concetti di programmazione: Coordinate 3D, algoritmi di ricerca, ragionamento spaziale, limiti di turni logaritmici

Panoramica

Dai la caccia a un sottomarino in uno spazio 3D. Ogni tentativo fornisce un feedback direzionale sui tre assi (nord/sud, est/ovest, profondo/superficiale). È la versione 3D della ricerca binaria e insegna agli studenti a pensare contemporaneamente in tre dimensioni.

Obiettivo

Individua e distruggi il sottomarino nascosto specificando le coordinate X, Y e di profondità.

Regole

1. Il giocatore sceglie la dimensione N dell'area di ricerca (creando una griglia $(N+1) \times (N+1) \times (N+1)$).
2. Il computer nasconde un sottomarino in coordinate casuali (da 0 a N per ciascun asse).
3. Il giocatore dispone di un numero limitato di tentativi (basato sulla dimensione della griglia).
4. Ogni tentativo specifica tre coordinate: orizzontale, verticale e di profondità.
5. Il computer segnala se ciascuna coordinata è troppo alta o troppo bassa (ad esempio "TROPPO A NORD", "TROPPO IN SUPERFICIE").
6. Una corrispondenza esatta su tutte e tre le coordinate distrugge il sottomarino.

Struttura dei turni e durata

Turni massimi: Il numero di tentativi consentiti è pari alla parte intera superiore di $\log_2(N+1)$, dove N è la dimensione dell'area di ricerca. Per una griglia $9 \times 9 \times 9$, si tratta di circa 3–4 tentativi.

Struttura di gioco: Lunghezza fissa — il giocatore dispone di un numero calcolato di tentativi in base alla dimensione della griglia.

Condizioni di fine gioco

Vittoria: indovina le coordinate esatte del sottomarino. Sconfitta: esaurisci tutti i tentativi consentiti senza un colpo diretto; viene rivelata la posizione del sottomarino. Il giocatore può quindi scegliere di giocare di nuovo.

Esempio di partita

DEPTH CHARGE
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

DIMENSIONE DELL'AREA DI RICERCA? 9
HAI 3 TENTATIVI.

TENTATIVO N. 1: QUAL È LA TUA IPOTESI? 5,5,5
TROPPO A NORD.
TROPPO A OVEST.
TROPPO IN SUPERFICIE.

TENTATIVO N. 2: QUAL È LA TUA IPOTESI? 8,8,8
TROPPO A SUD.
TROPPO A EST.
TROPPO IN PROFONDITÀ.

TENTATIVO N. 3: QUAL È LA TUA IPOTESI? 7,7,6
BOOM!! L'HAI TROVATO IN 3 TENTATIVI!

Gioco 16: Gunner (Artigliere)

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Artiglieria / Fisica	Intermedio	Lunghezza fissa per bersaglio (5 colpi), multi-round (5 bersagli). Il gioco termina dopo 5 bersagli o quando il giocatore viene distrutto.	5 colpi per bersaglio.

Concetti di programmazione: Trigonometria (funzione seno), simulazione fisica, affinamento iterativo, controllo di tolleranza

Panoramica

Spara un colpo di cannone a un bersaglio scegliendo l'angolo di elevazione. La gittata segue la fisica balistica reale (funzione seno). Gli studenti scoprono che 45° dà la massima gittata e devono avvicinarsi al bersaglio in modo iterativo. Collegamento diretto con la matematica.

Obiettivo

Distruggi 5 bersagli successivi regolando l'angolo di elevazione del tuo cannone, con non più di 5 colpi per bersaglio.

Regole

1. La gittata massima del cannone è impostata casualmente tra 20.000 e 60.000 iarde.
2. Un bersaglio appare a una distanza casuale entro la gittata del cannone.
3. Il giocatore inserisce un angolo di elevazione (1–89 gradi).
4. La gittata del colpo è calcolata come: $Gittata = GittataMax \times \sin(2 \times \text{angolo})$.
5. Se il colpo cade entro 100 iarde dal bersaglio, è un colpo a segno.
6. Altrimenti, il computer riporta di quante iarde il colpo è andato lungo o corto.
7. Dopo 5 mancate sullo stesso bersaglio, il nemico risponde al fuoco e ti distrugge.
8. Dopo 5 bersagli, vengono contate le munizioni totali: meno di 18 = "BEL TIRO!!".

Struttura dei turni e durata

Turni massimi: 5 colpi per bersaglio. 5 bersagli per partita. Massimo 25 colpi totali.

Struttura di gioco: Lunghezza fissa per bersaglio (5 colpi), multi-round (5 bersagli). Il gioco termina dopo 5 bersagli o quando il giocatore viene distrutto.

Condizioni di fine gioco

Vittoria del bersaglio: un colpo entro 100 iarde dal bersaglio (raggio di scoppio). Sconfitta del bersaglio: 5 mancate consecutive; il nemico ti distrugge e la partita passa al punteggio. Vittoria della partita: distruggi tutti e 5 i bersagli. Il computer riporta le munizioni totali consumate e commenta la precisione. Il giocatore può scegliere di giocare di nuovo.

Esempio di partita

GUNNER
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

LA GITTATA MASSIMA DEL TUO CANNONE È DI 55684 IARDE.
DISTANZA DAL BERSAGLIO: 15755 IARDE.

ELEVAZIONE? 9
OLTRE IL BERSAGLIO DI 1452 IARDE.

ELEVAZIONE? 8
SOTTO IL BERSAGLIO DI 407 IARDE.

ELEVAZIONE? 8.2
*** BERSAGLIO DISTRUTTO ***
3 COLPI DI MUNIZIONE CONSUMATI

L'OSSERVATORE AVANZATO HA AVVISTATO ALTRA ATTIVITÀ NEMICA...

Gioco 17: Lunar LEM Rocket (Razzo LEM Lunare)

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Simulazione spaziale	Intermedio–Avanzato	Fino a vittoria/sconfitta	Variabile — una decisione per passo temporale (ogni 10 secondi di tempo di simulazione).

Concetti di programmazione: Simulazione fisica, calcolo iterativo, gestione del carburante, decisioni in tempo reale

Panoramica

Il gioiello della corona tra i giochi di simulazione. Piloti un modulo lunare e scegli la portata di combustione ogni secondo per controllare la discesa. La gravità ti attira verso il basso, il carburante è finito e la tensione cresce man mano che la superficie si avvicina. Gli studenti sono rapiti — e la fisica è reale.

Obiettivo

Atterra dolcemente il Modulo di Escursione Lunare sulla superficie della Luna gestendo la portata di combustione dei retrorazzi.

Regole

1. Il LEM parte da un'altitudine di 500 piedi, in discesa a 50 ft/s, con 120 unità di carburante.
2. A ogni passo temporale il giocatore inserisce una portata di combustione K (da 0 a 200).
3. Il carburante viene consumato al ritmo K. Se K supera il carburante residuo, la combustione effettiva è pari al carburante rimasto.
4. La velocità cambia secondo: $\text{nuova_velocità} = \text{vecchia_velocità} + \text{gravità} - (K \times \text{fattore_spinta})$.
5. L'altitudine diminuisce in base alla velocità media durante il passo temporale.
6. A ogni passo viene mostrata la telemetria: tempo, altitudine, velocità e carburante residuo.
7. Quando l'altitudine ≤ 0 , il programma interpola per trovare la velocità esatta al tocco e segnala la qualità dell'atterraggio.

Struttura dei turni e durata

Turni massimi: Variabile — una decisione per passo temporale (ogni 10 secondi di tempo di simulazione). La partita termina quando l'altitudine raggiunge 0.

Struttura di gioco: Fino a vittoria/sconfitta — la simulazione prosegue senza interruzioni finché la navicella non raggiunge la superficie.

Condizioni di fine gioco

Atterraggio perfetto: velocità al tocco inferiore a 1 ft/s. Buon atterraggio: velocità al tocco inferiore a 10 ft/s. Atterraggio duro: velocità al tocco tra 10 e 50 ft/s (danni segnalati). Schianto: velocità al tocco superiore a 50 ft/s (distruzione completa). Esaurimento carburante: se finisce il carburante, la navicella è in caduta libera fino all'impatto. La partita termina sempre perché la gravità fa sì che la navicella raggiunga prima o poi la superficie.

Esempio di partita

LUNAR LEM ROCKET
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

INIZIO DELLA PROCEDURA DI ATTERRAGGIO...

	TEMPO	ALT	VEL	CARB
	0	500	-50	120
K=?	10			
	10	413	-32	110
K=?	20			
	20	362	-12	90
K=?	15			
	30	337	-7	75
K=?	5			
	40	320	-9	70
K=?	30			
	50	270	2	40

ATTERRAGGIO PERFETTO!
CONGRATULAZIONI!

Fase 6: Sistemi complessi

Gioco 18: Hammurabi

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Simulazione economica	Intermedio–Avanzato	Lunghezza fissa	Esattamente 10 turni (uno per anno).

Concetti di programmazione: Gestione di uno stato a più variabili, allocazione delle risorse, eventi casuali, punteggio di simulazione

Panoramica

L'antenato di ogni gioco in stile Civilization. Governa l'antica Sumer per 10 anni: compra terra, nutri la popolazione, pianta i raccolti. Le pestilenze colpiscono, i topi mangiano il grano, i raccolti variano. La matematica è semplice, ma i sistemi interconnessi insegnano la vera complessità della programmazione.

Obiettivo

Governa la Sumer per 10 anni mantenendo bassa la fame e facendo crescere la popolazione.

Regole

1. Condizioni iniziali: 100 persone, 1000 acri, 2800 staia di grano.
2. Ogni anno, la terra viene scambiata a un prezzo casuale (17–26 staia/acro).
3. Il giocatore decide: acri da comprare/vendere, staia con cui nutrire la popolazione e acri da coltivare.
4. Vincoli: non puoi spendere più grano di quello che hai. Ogni persona può coltivare al massimo 10 acri. Ogni acro richiede 0,5 staia di semente.
5. Ogni persona ha bisogno di 20 staia di cibo all'anno. Chi non è nutrito muore di fame.
6. La resa del raccolto è casuale (1–6 staia/acro). I topi possono mangiare il 10–40% delle scorte di grano.
7. Eventi casuali: pestilenza (uccide metà della popolazione, ~15% di probabilità).
8. Dopo 10 anni, il governo del giocatore viene valutato in base al tasso medio di fame e agli acri finali per persona.

Struttura dei turni e durata

Turni massimi: Esattamente 10 turni (uno per anno).

Struttura di gioco: Lunghezza fissa — esattamente 10 round che rappresentano 10 anni di governo.

Condizioni di fine gioco

Deposto (sconfitta): più del 33% della popolazione muore di fame in un solo anno. Sopravvissuto ma povero: completa i 10 anni ma con una valutazione scadente (elevata fame cumulativa, poca terra per persona). Buon sovrano: completa i 10 anni con fame moderata e una ragionevole crescita. Leggendaro: completa i 10 anni con fame minima, forte crescita della popolazione e notevoli possedimenti terrieri. La partita termina sempre dopo l'anno 10 (se sopravvivi).

Esempio di partita

HAMMURABI
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

HAMMURABI: MI ONORO DI RIFERIRTI,
NELL'ANNO 1, 0 PERSONE SONO MORTE DI FAME, 5 SONO ARRIVATE IN CITTÀ.
LA POPOLAZIONE È ORA DI 100.
LA CITTÀ POSSIEDE 1000 ACRI.
HAI RACCOLTO 3 STAIA PER ACRO.
I TOPI HANNO MANGIATO 200 STAIA.
HAI ORA 2800 STAIA IN DEPOSITO.
LA TERRA VIENE SCAMBIATA A 17 STAIA PER ACRO.

QUANTI ACRI VUOI COMPRARE? 0
QUANTI ACRI VUOI VENDERE? 0
QUANTE STAIA PER NUTRIRE LA POPOLAZIONE? 2000
QUANTI ACRI VUOI COLTIVARE? 1000

Gioco 19: NFL Football

Categoria	Difficoltà	Struttura di gioco	Turni massimi
Simulazione sportiva	Avanzato	Lunghezza fissa	Circa 50–80 giocate per partita (varia in base all'uso dell'orologio).

Concetti di programmazione: Macchine a stati complesse, logica di gioco multi-fase, selezione delle giocate, sistemi di punteggio, esiti casuali con probabilità pesate

Panoramica

Una simulazione completa del football americano: quattro quarti, down, iarde, palle perse, punteggio. Gli studenti chiamano giocate d'attacco da un menu e l'esito dipende da una selezione casuale pesata. È il gioco più ambizioso del laboratorio e unisce tutti i concetti appresi in precedenza.

Obiettivo

Segna più punti del computer selezionando giocate d'attacco efficaci nel corso di quattro quarti di football.

Regole

1. Un lancio della moneta determina chi riceve il kickoff di apertura.
2. La squadra che riceve parte dalla propria linea delle 20 iarde. Primo down e 10 iarde da guadagnare.
3. L'attaccante seleziona tra le giocate: corsa all'ala sinistra, corsa all'ala destra, quarterback sneak, lancio lungo e altre a seconda della versione.
4. Ogni giocata fa guadagnare o perdere iarde in base a esiti casuali pesati.
5. Il mancato avanzamento di 10 iarde in 4 down comporta la perdita del possesso (turnover on downs). Il giocatore può puntare al 4^o down.
6. Punteggio: touchdown = 6 punti + tentativo di trasformazione (1 punto). Field goal = 3 punti. Safety = 2 punti.
7. I turnover (fumble, intercetti) possono avvenire casualmente in qualunque giocata.
8. L'orologio avanza a ogni giocata. I quarti sono a tempo limitato.
9. Alla fine del 4^o quarto, la squadra con più punti vince.

Struttura dei turni e durata

Turni massimi: Circa 50–80 giocate per partita (varia in base all'uso dell'orologio). La partita è regolata dall'orologio.

Struttura di gioco: Lunghezza fissa — quattro quarti di tempo di gioco, ognuno dei quali consuma il cronometro a ogni giocata.

Condizioni di fine gioco

Vittoria: il punteggio del giocatore supera quello del computer al termine del 4^o quarto. Sconfitta: il punteggio del computer supera quello del giocatore. Pareggio: punteggi uguali al termine. La partita termina sempre allo scadere del cronometro del 4^o quarto. Non sono previsti tempi supplementari.

Esempio di partita

FOOTBALL
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

LANCIO DELLA MONETA...
HAI VINTO IL LANCIO!
VUOI RICEVERE? SI

PALLONE SULLA TUA LINEA DELLE 20.
PRIMO DOWN E 10.

SELEZIONA GIOCATA:
1 = CORSA ALL'ALA SINISTRA
2 = CORSA ALL'ALA DESTRA
3 = QUARTERBACK SNEAK
4 = LANCIO LUNGO

PROSSIMA GIOCATA? 2
GUADAGNO DI 7 IARDE.
2° DOWN E 3 SULLE TUE 27.

PROSSIMA GIOCATA? 4
COMPLETO! GUADAGNO DI 35 IARDE.
1° DOWN E 10 SULLE LORO 38.

Ulteriori esplorazioni

- Crea una versione computerizzata di calcio, tennis, pugilato, baseball, cricket, ecc., in uno stile testuale simile. È possibile: programmi di questo tipo erano onnipresenti sui sistemi a tempo condiviso degli anni '70.
- Aggiungi grafica e/o animazioni.
- L'Electronic Football Mattel (nell'immagine) suggerisce una semplice interfaccia da cui partire.



- Anche i giochi di Star Trek (in particolare Trek 73) erano popolari negli anni '70 e potrebbero essere aggiornati. (Video disponibile su <https://bit.ly/4baXVvS>)

Display Radius: 5000			
	Warp	Course	Bearing Range
Wasp:	2.8	332	pursuing Lamphry
Displacer:	5.0	329	177 2314
Kahlos:	1.0	15	224 3438
Crion:	5.0	318	166 2275
Barilion:	5.0	19	228 2921
Lamphry:	4.0	243	217 1222


```

DC
  ::
  :: +
K
  B  ::
      ::

```

Scott: We've not enough power.
 Bulu: The Kahlos is retreating.
 <<Crion firing Torpedo 20>>
 <<Lamphry firing Torpedo 21>>

- Che ne dici di dare un tuo tocco personale a Pong, Space Invaders o altri classici giochi arcade?

Fase 7: IA e apprendimento

Gioco 20: Animal

Categoria	Difficoltà	Struttura di gioco	Turni massimi
IA / Apprendimento	Intermedio–Avanzato	Illimitata	Variabile per round — il numero di domande dipende dalla profondità dell'albero di conoscenza corrente.

Concetti di programmazione: Alberi binari, strutture dati dinamiche, logica decisionale sì/no, fondamenti del machine learning

Panoramica

Il computer cerca di indovinare l'animale a cui stai pensando ponendo domande sì/no. Quando sbaglia, chiede a TE una domanda distintiva e la aggiunge alla sua conoscenza. Il computer letteralmente impara e diventa più intelligente a ogni partita. È il programma concettualmente più potente di tutto il libro.

Obiettivo

Pensa a un animale. Il computer pone domande sì/no per indovinarlo. Se fallisce, glielo insegni.

Regole

1. Il computer parte con un piccolo albero di conoscenza (ad esempio una sola domanda: "Nuota?", con PESCE e UCCELLO come foglie).
2. Il computer pone una serie di domande sì/no, percorrendo l'albero.
3. Arrivato a un nodo foglia, azzarda un animale.
4. Se indovina, il computer festeggia.
5. Se sbaglia, il giocatore digita l'animale corretto E una domanda sì/no che lo distingue dall'ipotesi del computer.
6. Il giocatore specifica quale risposta (sì o no) corrisponde al nuovo animale.
7. La nuova domanda e il nuovo animale vengono inseriti nell'albero.
8. Dopo molti round, l'albero cresce e il computer diventa sempre più competente.

Struttura dei turni e durata

Turni massimi: Variabile per round — il numero di domande dipende dalla profondità dell'albero di conoscenza corrente. Round illimitati.

Struttura di gioco: Illimitata — i round proseguono fintanto che il giocatore continua a pensare ad animali.

Condizioni di fine gioco

Il computer vince il round: indovina l'animale del giocatore. Il computer perde il round: sbaglia e il giocatore fornisce un nuovo animale e una domanda distintiva (l'albero cresce). Fine partita: il giocatore risponde "NO" alla domanda "Stai pensando a un animale?". Il computer mostra tutti gli animali che conosce.

Esempio di partita

ANIMAL
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

STAI PENSANDO A UN ANIMALE? SI
NUOTA? SI
È UN PESCE? NO
L'ANIMALE A CUI STAVI PENSANDO ERA UN ? DELFINO
PER FAVORE DIGITA UNA DOMANDA CHE DISTINGUA UN
DELFINO DA UN PESCE.
? È UN MAMMIFERO
PER UN DELFINO LA RISPOSTA SAREBBE? SI

STAI PENSANDO A UN ANIMALE? SI
NUOTA? SI
È UN MAMMIFERO? SI
È UN DELFINO? SI
PERCHÉ NON PROVI CON UN ALTRO ANIMALE?

Gioco 21: Hexapawn

Categoria	Difficoltà	Struttura di gioco	Turni massimi
IA / Apprendimento	Avanzato	Fino a vittoria/sconfitta per partita	Ogni partita dura in totale 2–5 mosse (la scacchiera 3×3 limita la durata).

Concetti di programmazione: Apprendimento per eliminazione, tabelle di stato, memoria di gioco, IA adattiva

Panoramica

Un gioco su scacchiera 3×3 con pedoni degli scacchi. Il computer parte come un pessimo giocatore ma elimina dalla propria memoria le mosse perdenti dopo ogni sconfitta. Gli studenti vedono letteralmente emergere un'IA da prove ed errori — il momento didattico più emozionante dell'intero laboratorio.

Obiettivo

Batti il computer a Hexapawn. Ma attenzione: il computer impara da ogni sconfitta e finirà per diventare imbattibile.

Regole

1. La scacchiera è 3×3. Ogni giocatore ha 3 pedoni sulla propria riga di partenza.
2. I pedoni possono avanzare di una casella (se vuota) o catturare in diagonale in avanti.
3. Un giocatore vince: (a) portando un pedone sulla riga di partenza dell'avversario, oppure (b) lasciando l'avversario senza mosse legali.
4. Il computer mantiene una tabella di tutti i possibili stati della scacchiera e delle mosse legali.
5. Quando il computer perde, cancella dalla tabella l'ultima mossa effettuata per quello stato della scacchiera.
6. Se tutte le mosse per uno stato sono state eliminate, il computer cancella anche la mossa che ha condotto a quello stato (apprendimento a cascata).
7. In 10–20 partite, il computer converge al gioco perfetto.

Struttura dei turni e durata

Turni massimi: Ogni partita dura in totale 2–5 mosse (la scacchiera 3×3 limita la durata). Rematch illimitati.

Struttura di gioco: Fino a vittoria/sconfitta per partita — ogni partita finisce quando una delle due parti vince. Si giocano più partite mentre il computer impara.

Condizioni di fine gioco

Il giocatore vince la partita: porta un pedone alla riga opposta, oppure lascia il computer senza mosse legali. Il computer vince la partita: porta un pedone alla riga opposta, oppure lascia il giocatore senza mosse legali. Apprendimento: quando il computer perde, rimuove dalla sua tabella di strategia la mossa che ha portato alla sconfitta. In molte partite, il gioco del computer migliora finché diventa imbattibile. La sessione termina quando il giocatore sceglie di fermarsi.

Esempio di partita

HEXAPAWN
CREATIVE COMPUTING MORRISTOWN, NEW JERSEY

```
X X X
. . .
O O O
```

LA TUA MOSSA? 23,22

```
X X X
. O .
O . O
```

MUOVO 11,22

```
X . X
. X .
O . O
```

LA TUA MOSSA? 31,22
HAI VINTO.

--- PARTITA 2 ---
(Non commetterò più quell'errore!)

Appendice: Tabella di riferimento rapido

Un riepilogo di tutti i 21 giochi con le loro specifiche principali a colpo d'occhio.

#	Gioco	Categoria	Difficoltà	Struttura	Fine gioco
1	Hi-Lo	Indovina il numero	Principiante	Illimitata	Vittoria del round: indovinare il numero corretto (vinci 100 \$). Sconfitta del round: ogni errore costa la puntata...
2	Guess	Indovina il numero	Principiante	Fino alla vittoria	Vittoria: il giocatore indovina. Il computer comunica i tentativi...
3	Stars	Indovina il numero	Principiante	Lunghezza fissa	Vittoria: indovinare entro 7 tentativi. Sconfitta: fallire dopo 7 tentativi; il computer rivela il numero...
4	Bagels	Logica / Codici	Principiante–Intermedio	Lunghezza fissa	Vittoria: indovinare le tre cifre nella posizione corretta. Sconfitta: esaurire i 20 tentativi...
5	Acey Ducey	Gioco di carte	Principiante	Illimitata	Sconfitta: capitale a 0 \$ (il computer ti annuncia la rovina e propone di ricominciare)...
6	Craps	Gioco di dadi	Principiante–Intermedio	Fino a vittoria/sconfitta per round	Vittoria round: 7 o 11 al come-out (natural), OPPURE il point prima del 7...
7	Blackjack	Gioco di carte	Intermedio–Avanzato	Fino a vittoria/sconfitta per mano	Vittoria mano: totale più vicino a 21 senza sfiorare...
8	War	Gioco di carte	Principiante	Lunghezza fissa	Vittoria: più round vinti del computer dopo 26 round. Sconfitta: il computer vince più round...
9	Nim	Strategia matematica	Intermedio	Fino a vittoria/sconfitta	Vittoria: prendere gli ultimi oggetti dall'ultima pila (gioco normale), OPPURE costringere l'avversario...
10	23 Matches	Strategia matematica	Principiante	Fino a vittoria/sconfitta	Vittoria: l'avversario è costretto a prendere l'ultimo (23 ^o) fiammifero. Sconfitta: sei tu a essere costretto...
11	Reverse	Rompicapo	Intermedio	Fino alla vittoria	Vittoria: la lista è 1 2 3 4 5 6 7 8 9 da sinistra a

#	Gioco	Categoria	Difficoltà	Struttura	Fine gioco
					destra. Il computer comunica le mosse...
12	Tic-Tac-Toe	Gioco da tavolo	Intermedio	Lunghezza fissa	Vittoria: tre segni in linea orizzontale, verticale o diagonale...
13	Hangman	Gioco di parole	Intermedio	Fino a vittoria/sconfitta	Vittoria: tutte le lettere sono state indovinate. Sconfitta: il giocatore commette 10 errori...
14	Buzzword	Parole / Umore	Principiante	Illimitata	Ritiro: il giocatore risponde "NO" quando gli viene chiesto se ne vuole un'altra...
15	Depth Charge	Ricerca / Militare	Intermedio	Lunghezza fissa	Vittoria: indovinare le coordinate esatte del sottomarino. Sconfitta: esaurire i tentativi...
16	Gunner	Artiglieria / Fisica	Intermedio	Lunghezza fissa per bersaglio (5 colpi), multi-round (5 bersagli).	Vittoria bersaglio: colpo entro 100 iarde. Sconfitta bersaglio: 5 mancate...
17	Lunar LEM Rocket	Simulazione spaziale	Intermedio–Avanzato	Fino a vittoria/sconfitta	Atterraggio perfetto: velocità al tocco inferiore a 1 ft/s. Buon atterraggio: inferiore a 10 ft/s...
18	Hammurabi	Simulazione economica	Intermedio–Avanzato	Lunghezza fissa	Deposto (sconfitta): più del 33% della popolazione muore di fame in un solo anno. Sopravvissuto...
19	NFL Football	Simulazione sportiva	Avanzato	Lunghezza fissa	Vittoria: il punteggio del giocatore supera quello del computer al termine del 4° quarto...
20	Animal	IA / Apprendimento	Intermedio–Avanzato	Illimitata	Vittoria round del computer: indovina correttamente. Sconfitta round: sbaglia...
21	Hexapawn	IA / Apprendimento	Avanzato	Fino a vittoria/sconfitta per partita	Vittoria del giocatore: pedone alla riga opposta, OPPURE computer senza mosse legali...

Riferimenti

- Hansen, S. (conduttore). (29 maggio 2024). What's worth making? [Episodio di podcast audio]. In Chalk Radio. MIT OpenCourseWare. <https://chalk-radio.simplecast.com/episodes/whats-worth-making-with-prof-hal-abelson>
- Kahn, K. (2025). The learner's apprentice: AI and the amplification of human creativity. Constructing Modern Knowledge Press.
- Martinez, S. L., & Stager, G. S. (2013). Invent to learn: Making, tinkering, and engineering in the classroom. Constructing Modern Knowledge Press.
- Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books.
- Papert, S. (1993). The children's machine: Rethinking school in the age of the computer. Basic Books.
- Papert, S., & Solomon, C. (1971). Twenty things to do with a computer (Artificial Intelligence Memo No. 248). MIT Artificial Intelligence Laboratory.
<https://dspace.mit.edu/bitstream/handle/1721.1/5836/AIM-248.pdf?sequence=2>
- Solomon, C. (1986). Computer environments for children: A reflection on theories of learning and education. MIT Press.
- Stager, G. S. (2025). Our roots: Logo, Piaget, and AI.
<https://reggio.constructingmodernknowledge.com/roots>
- Stager, G. S. (A cura di). (2022). Twenty things to do with a computer forward 50: Future visions of education inspired by Seymour Papert and Cynthia Solomon's seminal work. Constructing Modern Knowledge Press.
- Wolfram, S. (7 settembre 2016). How to teach computational thinking. Stephen Wolfram Writings.
<https://writings.stephenwolfram.com/2016/09/how-to-teach-computational-thinking/>

© 2026 Gary Stager & Robbo Outbackley