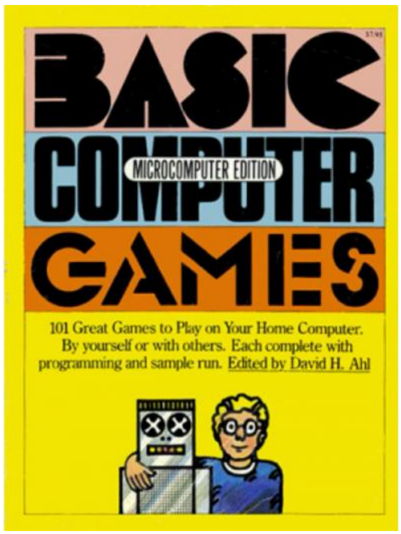
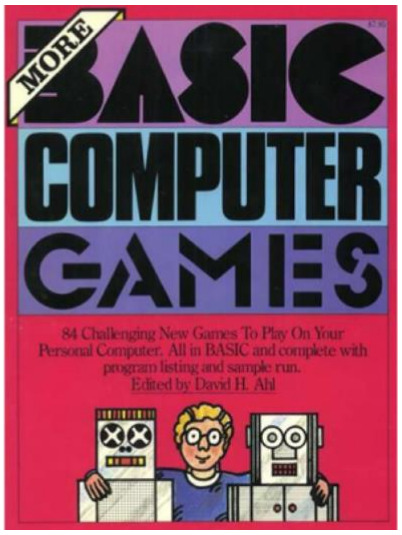
Basic Computer Games

Project Inspiration Handbook

21+ Games for Computational Making with ChatBots

Based on the classic book, Basic Computer Games, by David H. Ahl
Originally published 1973-1978 by Creative Computing
Adapted for middle school & teacher workshops by Gary Stager & Robbo Outbackley


BASIC
COMPUTER
MICROCOMPUTER EDITION
GAMES
101 Great Games to Play on Your Home Computer. By yourself or with others. Each complete with programming and sample run. Edited by David H. Ahl

MORE
BASIC
COMPUTER
GAMES
84 Challenging New Games To Play On Your Personal Computer. All in BASIC and complete with program listing and sample run. Edited by David H. Ahl

`These games are basic, but no longer in BASIC`

This is a work-in-progress, please share corrections, additions, or deletions with gary@stager.org

Table of Contents

Preface – Computational Making

Back to the Future

This eBook is predicated on a foundation of progressive education ideals (especially constructionism), a belief that projects should be the smallest unit of a teacher's concern, a recognition that the future is computational, and awareness that artificial intelligence (AI) may be used to supercharge the range, breadth, and depth of possible student projects.

In our book, *Invent To Learn – Making, Tinkering, and Engineering in the Classroom*, we make the case for the value of learning-by-making, a timeless idea ushered into modernity by Seymour Papert, Cynthia Solomon, and their colleagues.

The effective implementation of such project-based-learning begins with teacher curiosity, generosity, ingenuity, and respect for children, starting with high-quality prompt setting, an embrace of generative design, and affection for serendipity.

One need not approach artificial intelligence with ecstatic dreams or dystopian terror. Fundamentally, generative AI is just software. As Ken Kahn demonstrates so clearly in his book, *The Learner's Apprentice – AI and the Amplification of Human Creativity*, artificial intelligence can be a simultaneous mentor and apprentice that allows even young children to solve problems their teachers have never anticipated.

The popular modern "maker movement" blessed all of us with greater ability to make things with bits and atoms. With fabrication technology, we can now make all sorts of wondrous tangible objects with speed, thrift, and precision. With code we can make things too AND add interactivity and intelligence to physical artifacts. Sadly, schools have once again left bits behind in favor of cutting up cardboard and pool noodles. That is largely the result of adults' fear and ignorance of the empowerment associated with programming.

Mathematician, software developer, and MacArthur Genius Dr. Stephen Wolfram agrees that the future is computational. "For any discipline X, there is now or soon will be a branch of that discipline called *Computational X*." This represents not only the more interesting (and often playful) frontier of that discipline, but the better paying segment of it as well. Powerful new block-based programming languages and AI chatbots lower the barriers to democratizing computing allowing anyone to create software – even if they are the only customer for it.

All of this adds up to the modern realization of an ideal held during the early days of artificial intelligence research at places like MIT where in the late 1960s – early 1970s, AI research was deeply concerned with children, Piaget, and computational making. In fact, an unofficial slogan of the MIT AI Lab was, "Computers are for children." Read more here:: (https://reggio.constructingmodernknowledge.com/roots)

That full circle phenomena brings to this "project." In 1973, David Ahl wrote a collection of games in the BASIC programming language and distributed mimeographed copies to recreational computing enthusiasts far and wide. Major hardware companies featured versions of these games on their systems, in 1978, the first commercially published version of the book, *BASIC Computer Games* was published. It was the first computer book to ever sell over 1 million copies. Think about that, nearly 50 years ago, a computing book for hobbyist programmer sold a million copies.

We share summarized ideas from that book as inspiration for children and teachers alike to collaborate with AI chatbots and "make" their own computer games. Go ahead and make a game! Share it with your friends! You may be surprised by what you learn and can do with a little help from your peers and artificial intelligence.

Book Overview

This handbook presents 21 games selected from Basic Computer Games (Microcomputer Edition, 1978) for use in a teacher workshop or middle school programming course. The games are sequenced from simple to complex across 7 phases and 11 sessions. Each game entry provides a complete specification: overview, objective, rules, turn structure, termination conditions, programming concepts taught, and a sample run.

Phase	Sessions	Focus	Games
Phase 1: Foundations	1–2	Variables, loops, conditionals, random numbers	Hi-Lo, Guess, Stars, Bagels
Phase 2: Cards, Dice & Chance	3–4	Arrays, deck logic, betting, game state	Acey Ducey, Craps, Blackjack, War
Phase 3: Strategy & Thinking	5–6	Game theory, algorithms, AI basics	Nim, 23 Matches, Reverse, Tic-Tac-Toe
Phase 4: Strings & Words	7	String manipulation, word lists	Hangman, Buzzword
Phase 5: Simulation & Action	8–9	Coordinates, physics, real-time decisions	Depth Charge, Gunner, Lunar LEM
Phase 6: Complex Systems	10	Multi-variable state, resource management	Hammurabi, Football
Phase 7: AI & Learning	11	Machine learning, adaptive programs	Animal, Hexapawn

Phase 1: Foundations

Game 1: Hi-Lo

Category	Difficulty	Play Structure	Max Turns
Number Guessing	Beginner	Unlimited	Unlimited guesses per round.

Programming Concepts: Variables, INPUT/PRINT, IF/THEN, random numbers

Overview

The simplest possible guessing game. The computer picks a random number between 1 and 100, and you guess it. Too high? Too low? The computer tells you. A gambling wrapper adds stakes: you start with $1000 and win or lose money with each guess.

Objective

Guess the computer's secret number between 1 and 100 while managing your bankroll.

Rules

1. The computer selects a random integer from 1 to 100.
2. The player enters a guess.
3. The computer responds: 'TOO HIGH,' 'TOO LOW,' or 'GOT IT!'
4. A correct guess earns $100. An incorrect guess costs the player's bet.
5. If the bankroll hits $0, the game ends. Otherwise, a new round begins.
6. The player may quit at any time between rounds.

Turn Structure & Duration

Maximum Turns: Unlimited guesses per round. Multiple rounds continue until the player quits or goes broke.

Play Structure: Unlimited – play continues across rounds until the player decides to quit or runs out of money.

Termination Conditions

Win round: Guess the correct number (win $100). Lose round: Each incorrect guess costs the wager amount. Game over: Player's bankroll reaches $0, or player chooses to stop.

Sample Run

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

YOU HAVE $1000.
I WILL DRAW A NUMBER BETWEEN 1 AND 100.

YOUR GUESS? 50
TOO HIGH.

YOUR GUESS? 25
TOO LOW.

YOUR GUESS? 37
TOO LOW.

YOUR GUESS? 42
GOT IT!!!! YOU WIN $100!
YOUR TOTAL IS NOW $1100.
PLAY AGAIN (YES OR NO)? YES
```

Game 2: Guess

Category	Difficulty	Play Structure	Max Turns
Number Guessing	Beginner	Until win	Unlimited guesses, but the computer tracks the optimal count and comments on your performance.

Programming Concepts: WHILE loops, counter variables, logarithms (optional), binary search strategy

Overview

A step up from Hi-Lo. The player sets the upper limit, and the computer knows the optimal number of guesses (log base 2 of the limit). This naturally introduces binary search and lets students discover the strategy themselves.

Objective

Guess the computer's number in as few tries as possible, ideally meeting or beating the theoretical optimum.

Rules

1. The player chooses an upper limit (e.g. 100, 1000).
2. The computer picks a random integer from 1 to that limit.
3. The player guesses; the computer says 'TOO HIGH' or 'TOO LOW.'
4. When the player guesses correctly, the computer reports how many guesses were used.
5. The computer calculates the theoretical minimum (ceiling of log2 of the limit) and comments on the player's performance.
6. The player may choose to play again with the same or a different limit.

Turn Structure & Duration

Maximum Turns: Unlimited guesses, but the computer tracks the optimal count and comments on your performance.

Play Structure: Until win – each round ends when the player guesses correctly. No maximum turn limit.

Termination Conditions

Win: The player guesses the correct number. The computer reports the number of guesses taken and compares it to the theoretical optimum. The player may then choose to play again or quit.

Sample Run

```
GUESS
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

WHAT LIMIT DO YOU WANT? 100
I'M THINKING OF A NUMBER BETWEEN 1 AND 100.
YOU SHOULD NEED 7 GUESSES AT MOST.

YOUR GUESS? 50
TOO LOW. TRY A BIGGER NUMBER.

YOUR GUESS? 75
TOO HIGH. TRY A SMALLER NUMBER.

YOUR GUESS? 62
TOO HIGH.

YOUR GUESS? 56
THAT'S IT! YOU GOT IT IN 4 GUESSES.
VERY GOOD
```

Game 3: Stars

Category	Difficulty	Play Structure	Max Turns
Number Guessing	Beginner	Fixed-length	7 guesses maximum per round.

Programming Concepts: ABS() function, mapping numbers to ranges, string repetition, fixed-attempt limits

Overview

Like Guess, but instead of 'too high/too low,' the computer shows stars to indicate proximity. One star means far away; seven stars means very close. This requires a completely different guessing strategy and teaches students how to translate numeric distance into visual feedback.

Objective

Guess the computer's secret number (1–100) within 7 guesses, guided only by star-based proximity clues.

Rules

1. The computer picks a random number from 1 to 100.
2. The player has exactly 7 guesses.
3. After each guess, the computer prints 1 to 7 stars: more stars = closer to the target.
4. Star mapping: 1 star (64+ away), 2 stars (32–63), 3 stars (16–31), 4 stars (8–15), 5 stars (4–7), 6 stars (2–3), 7 stars (1 away).
5. An exact match ends the round immediately with a congratulations message.
6. After 7 wrong guesses, the number is revealed.

Turn Structure & Duration

Maximum Turns: 7 guesses maximum per round.

Play Structure: Fixed-length – exactly 7 guesses allowed. The round ends on a correct guess or after the 7th attempt.

Termination Conditions

Win: Guess the number within 7 tries. Lose: Fail to guess after 7 tries; the computer reveals the number. Game over: The player may choose to play again or quit.

Sample Run

```
STARS
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

I AM THINKING OF A NUMBER. START GUESSING.

YOUR GUESS? 50
**

YOUR GUESS? 75
*

YOUR GUESS? 25
*****

YOUR GUESS? 30
*******

YOUR GUESS? 31
YOU GOT IT IN 5 GUESSES!!!
```

Game 4: Bagels

Category	Difficulty	Play Structure	Max Turns
Logic / Code-Breaking	Beginner–Intermediate	Fixed-length	20 guesses maximum per round.

Programming Concepts: String comparison, digit extraction, FOR loops, multi-condition logic

Overview

The computer picks a 3-digit number with no repeated digits. You guess, and the computer gives coded clues: PICO (right digit, wrong position), FERMI (right digit, right position), BAGELS (nothing correct). This is essentially Mastermind with numbers and teaches systematic deduction.

Objective

Deduce the computer's 3-digit secret number within 20 guesses using PICO, FERMI, and BAGELS clues.

Rules

1. The computer picks a 3-digit number using digits 0–9 with no repeats.
2. The player enters a 3-digit guess.
3. For each digit: if it matches the same position, the computer says FERMI. If it matches a different position, PICO. If no digits match at all, BAGELS.
4. Multiple PICO/FERMI clues are given in a single response (e.g. 'PICO FERMI').
5. A response of three FERMIs means the player wins.
6. After 20 failed guesses, the number is revealed and the round ends.

Turn Structure & Duration

Maximum Turns: 20 guesses maximum per round.

Play Structure: Fixed-length – up to 20 guesses per round.

Termination Conditions

Win: Guess the exact 3-digit number (all three digits correct in correct positions). Lose: Exhaust all 20 guesses without a correct answer; the computer reveals the number. Game over: The player may choose to play again or quit.

Sample Run

```
BAGELS
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

O.K. I HAVE A NUMBER IN MIND.

GUESS # 1
? 123
PICO FERMI

GUESS # 2
? 456
BAGELS

GUESS # 3
? 178
FERMI FERMI

GUESS # 4
? 172
YOU GOT IT!!!
```

Phase 2: Cards, Dice & Chance

Game 5: Acey Ducey

Category	Difficulty	Play Structure	Max Turns
Card Game	Beginner	Unlimited	Unlimited rounds.

Programming Concepts: Random number generation, card representation, betting logic, running totals

Overview

The dealer shows two cards. You bet on whether the next card will fall between them. Wide spread? Safe bet. Narrow spread? Risky. Bet zero to chicken out. It's simple, fast, and teaches probability thinking.

Objective

Maximize your bankroll by betting wisely on whether the next card falls between two dealt cards.

Rules

1. The player starts with $100.
2. Each round, the computer deals two cards face-up (values 2–14, where 11=Jack, 12=Queen, 13=King, 14=Ace).
3. The player places a bet ($0 to skip, or any amount up to their bankroll).
4. A third card is dealt. If its value is strictly between the first two cards, the player wins the bet amount. Otherwise, the player loses the bet amount.
5. If the player's bankroll reaches $0, the game ends.
6. Betting $0 results in the computer calling you 'CHICKEN!!' but no money changes hands.

Turn Structure & Duration

Maximum Turns: Unlimited rounds. Play continues until the player is broke or quits.

Play Structure: Unlimited – one betting round after another with no fixed endpoint.

Termination Conditions

Lose: Bankroll reaches $0 (the computer announces you're broke and offers to restart). Quit: The player interrupts the program or chooses not to continue. There is no 'win' condition; the goal is to accumulate as much money as possible.

Sample Run

```
ACEY DUCEY CARD GAME
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

YOU NOW HAVE 100 DOLLARS

HERE ARE YOUR NEXT TWO CARDS:
 3
 JACK

WHAT IS YOUR BET? 25
 8
YOU WIN!!!
YOU NOW HAVE 125 DOLLARS

HERE ARE YOUR NEXT TWO CARDS:
 9
 10

WHAT IS YOUR BET? 0
CHICKEN!!
```

Game 6: **Craps**

Category	Difficulty	Play Structure	Max Turns
Dice Game	Beginner–Intermediate	Until win/loss per round	Each round takes 1 roll (natural/craps) or unlimited rolls (point phase).

Programming Concepts: Multi-phase game state, DO/WHILE loops, compound conditions, dice simulation

Overview

The real casino dice game, faithfully simulated. The come-out roll, the point phase, naturals, craps – students love dice and the two-phase game state is a great teaching tool for program flow control.

Objective

Win money by rolling winning dice combinations according to standard craps rules.

Rules

1. The player places a bet.
2. Come-out roll: Two dice are rolled. 7 or 11 = instant win (natural). 2, 3, or 12 = instant loss (craps). Any other total (4, 5, 6, 8, 9, 10) becomes the 'point.'
3. Point phase: The player keeps rolling. Rolling the point number = win. Rolling a 7 = loss ('seven out'). Any other number = roll again.
4. Winnings equal the bet amount. Losses deduct the bet amount.
5. The player may quit between rounds.

Turn Structure & Duration

Maximum Turns: Each round takes 1 roll (natural/craps) or unlimited rolls (point phase). Multiple rounds continue until the player quits.

Play Structure: Until win/loss per round – each round ends on a decisive roll. The overall game is unlimited across rounds.

Termination Conditions

Win round: Roll 7 or 11 on the come-out roll (natural), OR roll your point number before rolling a 7. Lose round: Roll 2, 3, or 12 on the come-out roll (craps), OR roll a 7 before making your point. Game over: Player chooses to stop betting, or bankroll reaches $0.

Sample Run

```
CRAPS
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

ENTER YOUR BET? 10
5 PLUS 6 --- TOTAL = 11
NATURAL....A WINNER!!!
YOU WIN 10 DOLLARS.

ENTER YOUR BET? 20
3 PLUS 1 --- TOTAL = 4
YOUR POINT IS 4. I WILL ROLL AGAIN
6 PLUS 3 --- TOTAL = 9
1 PLUS 3 --- TOTAL = 4
YOU MADE YOUR POINT!
YOU WIN 20 DOLLARS.
```

Game 7: **Blackjack**

Category	Difficulty	Play Structure	Max Turns
Card Game	Intermediate–Advanced	Until win/loss per hand	Each hand allows unlimited hits.

Programming Concepts: Arrays, deck shuffling, multi-player logic, complex conditionals, ace valuation

Overview

The most iconic card game in the world. Even a simplified single-player version teaches deck management, hand evaluation, and multi-branch decision making (hit, stand, double down). Every student knows the rules already.

Objective

Beat the dealer by getting a hand value closer to 21 without going over.

Rules

1. The player places a bet. The dealer gives 2 cards to the player (face up) and 2 to themselves (one face up, one face down).
2. Card values: 2–10 face value; Jack/Queen/King = 10; Ace = 1 or 11 (whichever benefits the hand).
3. Player options: HIT (take another card), STAND (keep current hand), DOUBLE DOWN (double bet, take exactly one more card).
4. If the player's total exceeds 21, they bust and lose immediately.
5. Dealer reveals hidden card and must hit on 16 or less, stand on 17 or more.
6. Closest to 21 without busting wins. Ties push.

Turn Structure & Duration

Maximum Turns: Each hand allows unlimited hits. Rounds continue until the player quits or goes broke.

Play Structure: Until win/loss per hand – each hand resolves when both player and dealer stand or bust. The session is unlimited across hands.

Termination Conditions

Win hand: Player's hand total is closer to 21 than the dealer's without exceeding 21, OR dealer busts. Lose hand: Player's hand exceeds 21 (bust), OR dealer's hand is closer to 21. Push: Both hands have the same value (no money changes hands). Blackjack: A two-card hand totaling exactly 21 (pays 3:2). Game over: Player chooses to quit or bankroll is exhausted.

Sample Run

```
BLACKJACK
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

BET? 10

YOUR HAND:
  4
  JACK
DEALER SHOWS: 6

HIT? YES
  7 -- TOTAL IS 21

DEALER: 6  9  3 = 18

YOU WIN $10!
```

Game 8: War

Category	Difficulty	Play Structure	Max Turns
Card Game	Beginner	Fixed-length	Exactly 26 rounds (52 cards dealt 2 at a time), unless the player quits early.

Programming Concepts: Array shuffling, card deck representation, comparison operators, FOR loops, scorekeeping

Overview

The simplest card game possible – flip and compare. Despite its simplicity, it teaches the full lifecycle of a deck: creation, shuffling, dealing, and comparison. Students can enhance it with the 'war' tiebreaker mechanic.

Objective

Win more rounds than the computer after playing through the entire deck.

Rules

1. A standard 52-card deck is shuffled.
2. Each round, one card is dealt to the player and one to the computer.
3. The higher card value wins the round. Ties result in no score change.
4. Cards are displayed by suit and number (e.g. 'S-7' = 7 of Spades).
5. After 26 rounds (or when the player quits), the final score is displayed.
6. The player with more wins is the overall winner.

Turn Structure & Duration

Maximum Turns: Exactly 26 rounds (52 cards dealt 2 at a time), unless the player quits early.

Play Structure: Fixed-length – the game plays through the entire 52-card deck in 26 comparisons.

Termination Conditions

Win: Player has more round wins than the computer after 26 rounds. Lose: Computer has more wins. Tie: Equal number of wins. Early quit: Player may choose to stop between any two rounds; final score is reported at that point.

Sample Run

```
WAR
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

YOU: S-7   ME: H-3
YOU WIN!

YOU: D-10  ME: C-JACK
I WIN!

YOU: H-4   ME: H-9
I WIN!

YOU: S-ACE  ME: D-5
YOU WIN!

SCORE: YOU--2  COMPUTER--2
```

Phase 3: Strategy & Thinking

Game 9: Nim

Category	Difficulty	Play Structure	Max Turns
Mathematical Strategy	Intermediate	Until win/loss	Variable – depends on pile sizes.

Programming Concepts: Binary/XOR operations, game theory, optimal strategy, WHILE loops

Overview

One of the oldest strategy games in existence. Players take objects from piles, and the last player to take wins. The deep lesson: this game has a perfect mathematical solution (the nim-sum), and the computer knows it. Students learn what it means for a computer to play optimally.

Objective

Be the player who takes the last object (or force your opponent to take the last object, depending on the variant).

Rules

1. The player chooses the number of piles and how many objects in each pile.
2. Players alternate turns. On each turn, a player removes any number of objects (at least 1) from a single pile.
3. The player who takes the last object wins (normal play).
4. The computer uses the binary nim-sum strategy to play optimally.
5. If the nim-sum of all piles is 0 at the start of the computer's turn, the computer is in a losing position (if the player plays perfectly).

Turn Structure & Duration

Maximum Turns: Variable – depends on pile sizes. Each turn removes at least 1 object, so the game always terminates.

Play Structure: Until win/loss – play continues until all piles are empty.

Termination Conditions

Win: You take the last object(s) from the final pile (normal play), OR your opponent is forced to take the last object (misère variant). Lose: The opposite of the above. The game always terminates because the total number of objects decreases every turn.

Sample Run

```
NIM
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

NUMBER OF PILES? 3
PILE 1? 3
PILE 2? 5
PILE 3? 7

YOUR TURN. WHICH PILE? 2
HOW MANY? 4

PILE 1: 3   PILE 2: 1   PILE 3: 7

I REMOVE 6 FROM PILE 3.
PILE 1: 3   PILE 2: 1   PILE 3: 1
```

Game 10: 23 Matches

Category	Difficulty	Play Structure	Max Turns
Mathematical Strategy	Beginner	Until win/loss	8–23 turns total (depends on how many matches each player takes).

Programming Concepts: Modular arithmetic, strategy discovery, simplified Nim

Overview

A constrained version of Nim with exactly 23 matches and a take-1-2-or-3 rule. Simple enough that students can discover the winning strategy (always leave a multiple of 4) by playing a few rounds. A perfect gateway to game theory.

Objective

Force your opponent to take the last match.

Rules

1. 23 matches are placed on the table.
2. Players alternate turns. On each turn, a player removes 1, 2, or 3 matches.
3. The player who is forced to take the last match loses.
4. The computer knows the optimal strategy: always leave a number of matches that is 1 more than a multiple of 4 (i.e. 1, 5, 9, 13, 17, 21).

Turn Structure & Duration

Maximum Turns: 8–23 turns total (depends on how many matches each player takes). Always terminates.

Play Structure: Until win/loss – play alternates until exactly 1 match remains.

Termination Conditions

Win: Your opponent is forced to take the last (23rd) match. Lose: You are forced to take the last match. The game always terminates because the match count decreases by 1–3 every turn.

Sample Run

```
23 MATCHES
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

23 MATCHES ON THE TABLE.

YOUR TURN. HOW MANY? 3
20 MATCHES LEFT.

I TAKE 3.
17 MATCHES LEFT.

YOUR TURN. HOW MANY? 2
15 MATCHES LEFT.

I TAKE 2.
13 MATCHES LEFT.
```

Game 11: Reverse

Category	Difficulty	Play Structure	Max Turns
Puzzle	Intermediate	Until win	Unlimited moves.

Programming Concepts: Array reversal, algorithm design, heuristic vs. algorithmic thinking

Overview

A unique sorting puzzle: arrange numbers 1–9 by repeatedly reversing the first N elements. Students must think about algorithms (guaranteed solution in 2N-3 moves) versus heuristics (opportunistic moves that might do better). This is one of the most thought-provoking games in the book.

Objective

Arrange the scrambled list of numbers 1 through 9 into ascending order using only reversal operations.

Rules

1. The computer generates a random permutation of the numbers 1 through 9.
2. Each turn, the player enters a number N (from 1 to 9).
3. The first N numbers in the list are reversed in place.
4. Entering 0 quits the game.
5. When the list is sorted in ascending order, the player wins and the total move count is displayed.
6. Example: List is [2,4,5,1,9,6,3,7,8]. Reverse 4 → [1,5,4,2,9,6,3,7,8]. Reverse 5 → [9,2,4,5,1,6,3,7,8].

Turn Structure & Duration

Maximum Turns: Unlimited moves. There is no penalty for taking many moves, but fewer is better.

Play Structure: Until win – play continues until the list is sorted or the player gives up (reverses 0).

Termination Conditions

Win: The list reads 1 2 3 4 5 6 7 8 9 from left to right. The computer reports how many moves it took. Quit: The player enters 0, ending the game without solving. The game always has a solution.

Sample Run

```
REVERSE
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

THE LIST IS:
2 4 5 1 9 6 3 7 8

HOW MANY SHALL I REVERSE? 4
1 5 4 2 9 6 3 7 8

HOW MANY SHALL I REVERSE? 9
8 7 3 6 9 2 4 5 1

HOW MANY SHALL I REVERSE? 8
5 4 2 9 6 3 7 8 1

HOW MANY SHALL I REVERSE? 9
1 8 7 3 6 9 2 4 5
```

Game 12: Tic-Tac-Toe

Category	Difficulty	Play Structure	Max Turns
Board Game	Intermediate	Fixed-length	9 moves maximum (5 for the first player, 4 for the second).

Programming Concepts: 2D arrays, win-condition checking, simple AI, game trees

Overview

Every student knows it. Coding it transforms a childhood game into a programming milestone. The win-check logic alone (8 possible lines) teaches methodical thinking, and adding a computer opponent introduces basic AI.

Objective

Get three of your marks (X) in a row on a 3×3 grid before the computer (O) does.

Rules

1. The board is a 3×3 grid, positions numbered 1–9.
2. The player is X; the computer is O. The player moves first.
3. On each turn, the player enters the number of an empty square.
4. The computer then makes its move.
5. After each move, the program checks all 8 possible winning lines (3 rows, 3 columns, 2 diagonals).
6. First player to complete a line wins. If the board fills with no winner, it's a draw.

Turn Structure & Duration

Maximum Turns: 9 moves maximum (5 for the first player, 4 for the second). Always terminates.

Play Structure: Fixed-length – the game ends when someone gets three in a row or all 9 squares are filled.

Termination Conditions

Win: A player places three marks in a horizontal, vertical, or diagonal line. Draw: All 9 squares are filled with no three-in-a-row. The computer plays optimally, so the best a perfect player can achieve is a draw.

Sample Run

```
TIC-TAC-TOE
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

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

YOUR MOVE? 5
```

```
 . | . | .
-----------
 . | X | .
-----------
 . | . | .

I MOVE TO 1.

 O | . | .
-----------
 . | X | .
-----------
 . | . | .
```

Phase 4: Strings & Words

Game 13: Hangman

Category	Difficulty	Play Structure	Max Turns
Word Game	Intermediate	Until win/loss	10 incorrect guesses allowed (one for each body part: head, body, left arm, right arm, left leg, right leg, left hand, right hand, left foot, right foot).

Programming Concepts: String searching, character arrays, ASCII display building, word lists

Overview

The universally known word game. Coding it teaches string searching (is the guessed letter in the word?), building a display string that updates, and tracking used letters. The visual hangman figure adds progressive feedback.

Objective

Guess all the letters in the hidden word before the hangman figure is fully drawn.

Rules

1. The computer selects a word from a built-in list.
2. The word is displayed as a series of dashes (one per letter).
3. The player guesses one letter at a time.
4. If the letter is in the word, all instances are revealed in their correct positions.
5. If the letter is not in the word, a body part is added to the hangman figure and the miss count increments.
6. Previously guessed letters should not be guessed again.
7. After 10 misses, the hangman is complete and the game is lost.

Turn Structure & Duration

Maximum Turns: 10 incorrect guesses allowed (one for each body part: head, body, left arm, right arm, left leg, right leg, left hand, right hand, left foot, right foot). Correct guesses do not count as turns.

Play Structure: Until win/loss – play continues until the word is fully revealed or 10 wrong guesses are made.

Termination Conditions

Win: All letters in the word have been correctly guessed. Lose: The player makes 10 incorrect guesses (the hangman is complete) and the word is revealed. The player may then choose to play again with a new word.

Sample Run

```
HANGMAN
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY


THE WORD IS: ------


GUESS A LETTER? E
  -E----


GUESS A LETTER? A
SORRY, NOT IN THE WORD. MISS #1
```

```
GUESS A LETTER? O
  -E--O-


GUESS A LETTER? R
  -E--OR


GUESS A LETTER? N
  -EN-OR


GUESS A LETTER? M
  MENTOR
YOU GOT IT!
```

Game 14: Buzzword

Category	Difficulty	Play Structure	Max Turns
Word / Humor	Beginner	Unlimited	Unlimited.

Programming Concepts: Arrays of strings, random selection, string concatenation, DATA statements

Overview

The lightest game in the workshop – but students love it. Three arrays of jargon words are randomly combined to produce impressive-sounding nonsense. Fast to code, instantly funny, and teaches array indexing and string building.

Objective

Generate random impressive-sounding three-word buzzword phrases.

Rules

1. Three arrays of adjectives/nouns are defined (typically 12 words each).
2. When the player says 'YES' (or presses enter), the computer randomly picks one word from each of the three columns.
3. The three words are printed as a single phrase.
4. The player is asked if they want another. 'NO' exits the program.

Turn Structure & Duration

Maximum Turns: Unlimited. The player requests phrases one at a time.

Play Structure: Unlimited – the program generates one phrase per request until the player says 'NO.'

Termination Conditions

Quit: The player responds 'NO' when asked if they want another phrase. There is no win or loss condition; this is a generator, not a competition.

Sample Run

```
BUZZWORD
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

READY? YES
  TOTAL MANAGEMENT OPTIONS

READY? YES
  INTEGRATED RECIPROCAL FLEXIBILITY

READY? YES
  COMPATIBLE TRANSITIONAL PROJECTION

READY? NO
COME BACK WHEN YOU NEED MORE BUZZWORDS...
```

Phase 5: Simulation & Action

Game 15: Depth Charge

Category	Difficulty	Play Structure	Max Turns
Search / Military	Intermediate	Fixed-length	The number of allowed trials equals the ceiling of log2(N+1), where N is the dimension of the search area.

Programming Concepts: 3D coordinates, search algorithms, spatial reasoning, logarithmic turn limits

Overview

Hunt a submarine in 3D space. Each guess gives directional feedback on three axes (north/south, east/west, deep/shallow). It's the 3D version of a binary search and teaches students to think in three dimensions simultaneously.

Objective

Locate and destroy the hidden submarine by specifying X, Y, and depth coordinates.

Rules

1. The player chooses the dimension N of the search area (creating an N+1 × N+1 × N+1 grid).
2. The computer hides a submarine at random coordinates (0 to N for each axis).
3. The player gets a limited number of guesses (based on grid size).
4. Each guess specifies three coordinates: horizontal, vertical, and depth.
5. The computer reports whether each coordinate is too high or too low (e.g. 'TOO FAR NORTH,' 'TOO SHALLOW').
6. An exact match on all three coordinates destroys the submarine.

Turn Structure & Duration

Maximum Turns: The number of allowed trials equals the ceiling of log2(N+1), where N is the dimension of the search area. For a 9×9×9 grid, that's approximately 3–4 trials.

Play Structure: Fixed-length – the player has a calculated number of trials based on grid size.

Termination Conditions

Win: Guess the submarine's exact coordinates. Lose: Exhaust all allowed trials without a direct hit; the submarine's location is revealed. The player may then choose to play again.

Sample Run

```
DEPTH CHARGE
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

DIMENSION OF SEARCH AREA? 9
YOU HAVE 3 TRIES.

TRIAL #1: WHAT IS YOUR GUESS? 5,5,5
TOO FAR NORTH.
TOO FAR WEST.
TOO SHALLOW.

TRIAL #2: WHAT IS YOUR GUESS? 8,8,8
TOO FAR SOUTH.
TOO FAR EAST.
TOO DEEP.

TRIAL #3: WHAT IS YOUR GUESS? 7,7,6
BOOM!! YOU FOUND IT IN 3 TRIES!
```

Game 16: Gunner

Category	Difficulty	Play Structure	Max Turns
Artillery / Physics	Intermediate	Fixed-length per target (5 shots), multi-round (5 targets). The overall game ends after 5 targets or when the player is destroyed.	5 shots per target.

Programming Concepts: Trigonometry (sine function), physics simulation, iterative refinement, tolerance checking

Overview

Fire a cannon at a target by choosing the elevation angle. The range follows real ballistic physics (sine function). Students learn that 45° gives maximum range and must iteratively zero in on the target. Direct connection to math class.

Objective

Destroy 5 successive targets by adjusting your gun's elevation angle, with no more than 5 shots per target.

Rules

1. The gun's maximum range is randomly set between 20,000 and 60,000 yards.
2. A target appears at a random distance within the gun's range.
3. The player enters an elevation angle (1–89 degrees).
4. The shot range is calculated as: Range = MaxRange × sin(2 × angle).
5. If the shot lands within 100 yards of the target, it's a hit.
6. Otherwise, the computer reports how many yards over or short the shot was.
7. After 5 misses on one target, the enemy fires back and destroys you.
8. After 5 targets, total ammunition is tallied: under 18 = 'NICE SHOOTING!!'

Turn Structure & Duration

Maximum Turns: 5 shots per target. 5 targets per game. Maximum 25 shots total.

Play Structure: Fixed-length per target (5 shots), multi-round (5 targets). The overall game ends after 5 targets or when the player is destroyed.

Termination Conditions

Win target: Land a shot within 100 yards of the target (burst radius). Lose target: Miss 5 consecutive times; the enemy destroys you, and the game advances to the score. Win game: Destroy all 5 targets. The computer reports total ammunition expended and comments on accuracy. The player may choose to play again.

Sample Run

```
GUNNER
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY


MAXIMUM RANGE OF YOUR GUN IS 55684 YARDS.
DISTANCE TO THE TARGET IS 15755 YARDS.


ELEVATION? 9
OVER TARGET BY 1452 YARDS.

ELEVATION? 8
SHORT OF TARGET BY 407 YARDS.


ELEVATION? 8.2
*** TARGET DESTROYED ***
3 ROUNDS OF AMMUNITION EXPENDED

THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY
ACTIVITY...
```

Game 17: Lunar LEM Rocket

Category	Difficulty	Play Structure	Max Turns
Space Simulation	Intermediate–Advanced	Until win/loss	Variable – one decision per time step (every 10 seconds of simulation time).

Programming Concepts: Physics simulation, iterative calculation, fuel management, real-time decision-making

Overview

The crown jewel of the simulation games. You pilot a lunar lander, choosing your burn rate each second to control descent. Gravity pulls you down, fuel is finite, and the tension builds as the surface approaches. Students are riveted – and the physics is real.

Objective

Land the Lunar Excursion Module softly on the moon's surface by managing your retro rocket burn rate.

Rules

1. The LEM starts at 500 feet altitude, descending at 50 ft/sec, with 120 units of fuel.
2. Each time step, the player enters a burn rate K (0 to 200).
3. Fuel is consumed at rate K. If K exceeds remaining fuel, actual burn equals remaining fuel.
4. Velocity changes based on: new_velocity = old_velocity + gravity - (K × thrust_factor).
5. Altitude decreases based on average velocity during the time step.
6. Telemetry is displayed each step: time, altitude, velocity, and remaining fuel.
7. When altitude ≤ 0, the program interpolates to find exact touchdown velocity and reports the landing quality.

Turn Structure & Duration

Maximum Turns: Variable – one decision per time step (every 10 seconds of simulation time). The game ends when altitude reaches 0.

Play Structure: Until win/loss – the simulation runs continuously until the craft reaches the surface.

Termination Conditions

Perfect landing: Touchdown velocity is under 1 ft/sec. Good landing: Touchdown velocity is under 10 ft/sec. Hard landing: Touchdown velocity is 10–50 ft/sec (damage reported). Crash: Touchdown velocity exceeds 50 ft/sec (complete destruction). Out of fuel: If fuel runs out, the craft is in free-fall until impact. The game always terminates because gravity ensures the craft eventually reaches the surface.

Sample Run

```
LUNAR LEM ROCKET
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY


BEGINNING LANDING PROCEDURE...


    TIME  ALT      VEL     FUEL
     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


PERFECT LANDING!
CONGRATULATIONS!
```

Phase 6: Complex Systems

Game 18: Hammurabi

Category	Difficulty	Play Structure	Max Turns
Economic Simulation	Intermediate–Advanced	Fixed-length	Exactly 10 turns (one per year).

Programming Concepts: Multi-variable state management, resource allocation, random events, simulation scoring

Overview

The ancestor of every Civilization-style game. Govern ancient Sumeria for 10 years: buy land, feed people, plant crops. Plagues strike, rats eat grain, harvests vary. The math is straightforward but the interconnected systems teach real programming complexity.

Objective

Govern Sumeria for 10 years, keeping starvation low and population growing.

Rules

1. Starting conditions: 100 people, 1000 acres, 2800 bushels of grain.
2. Each year, land trades at a random price (17–26 bushels/acre).
3. The player decides: acres to buy/sell, bushels to feed the people, and acres to plant.
4. Constraints: Can't spend more grain than you have. Each person can farm at most 10 acres. Each acre requires 0.5 bushels of seed.
5. Each person needs 20 bushels of food per year. Undfed people starve.
6. Harvest yield is random (1–6 bushels/acre). Rats may eat 10–40% of grain stores.
7. Random events: plague (kills half the population, ~15% chance).
8. After 10 years, the player's governance is rated based on average starvation rate and final acres per person.

Turn Structure & Duration

Maximum Turns: Exactly 10 turns (one per year).

Play Structure: Fixed-length – exactly 10 rounds representing 10 years of governance.

Termination Conditions

Impeached (loss): More than 33% of the population starves in a single year. Survived but poor: Complete 10 years but with a bad performance rating (high cumulative starvation, low land per person). Good ruler: Complete 10 years with moderate starvation and reasonable growth. Legendary: Complete 10 years with minimal starvation, strong population growth, and significant land holdings. The game always ends after year 10 (if you survive).

Sample Run

```
HAMMURABI
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

HAMMURABI: I BEG TO REPORT TO YOU,
IN YEAR 1, 0 PEOPLE STARVED, 5 CAME TO THE
CITY.
POPULATION IS NOW 100.
THE CITY OWNS 1000 ACRES.
YOU HARVESTED 3 BUSHELS PER ACRE.
RATSON ATE 200 BUSHELS.
YOU NOW HAVE 2800 BUSHELS IN STORE.
LAND IS TRADING AT 17 BUSHELS PER ACRE.

HOW MANY ACRES DO YOU WISH TO BUY? 0
HOW MANY ACRES DO YOU WISH TO SELL? 0
HOW MANY BUSHELS TO FEED YOUR PEOPLE? 2000
HOW MANY ACRES TO PLANT WITH SEED? 1000
```

Game 19: NFL Football

Category	Difficulty	Play Structure	Max Turns
Sports Simulation	Advanced	Fixed-length	Approximately 50–80 plays per game (varies by clock usage).

Programming Concepts: Complex state machines, multi-phase game logic, play selection, scoring systems, random outcomes with weighted probabilities

Overview

A full American football simulation: four quarters, downs, yardage, turnovers, scoring. Students call offensive plays from a menu and the outcome depends on weighted random selection. This is the most ambitious game in the workshop and ties together every concept learned previously.

Objective

Score more points than the computer by selecting effective offensive plays over four quarters of football.

Rules

1. A coin toss determines who receives the opening kickoff.
2. The receiving team starts at their own 20-yard line. First down and 10 yards to go.
3. The offensive player selects from plays: left end around, right end around, quarterback sneak, long bomb, and others depending on the version.
4. Each play gains or loses yardage based on weighted random outcomes.
5. Failing to gain 10 yards in 4 downs results in a turnover on downs. The player may punt on 4th down.
6. Scoring: Touchdown = 6 points + extra point attempt (1 point). Field goal = 3 points. Safety = 2 points.
7. Turnovers (fumbles, interceptions) can occur randomly on any play.
8. The clock advances with each play. Quarters are time-limited.
9. At the end of the 4th quarter, the team with the most points wins.

Turn Structure & Duration

Maximum Turns: Approximately 50–80 plays per game (varies by clock usage). The game is clock-governed.

Play Structure: Fixed-length – four quarters of game time, each consuming clock on every play.

Termination Conditions

Win: Player's score exceeds the computer's score at the end of the 4th quarter. Lose: Computer's score exceeds the player's. Tie: Scores are equal at the end. The game always terminates after the 4th quarter clock expires. No overtime.

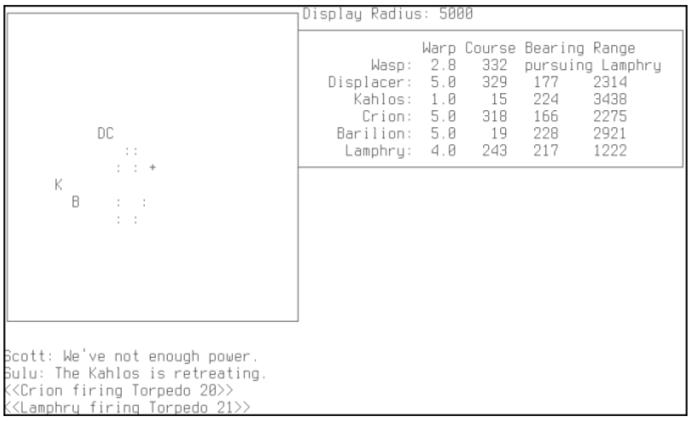
Sample Run

```
FOOTBALL
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY


TOSS OF THE COIN...
YOU WIN THE TOSS!
DO YOU WANT TO RECEIVE? YES


BALL ON YOUR 20 YARD LINE.
FIRST DOWN AND 10.


SELECT PLAY:
  1 = LEFT END AROUND
  2 = RIGHT END AROUND
  3 = QUARTERBACK SNEAK
  4 = LONG BOMB
```

```
NEXT PLAY? 2
GAIN OF 7 YARDS.
2ND AND 3 ON YOUR 27.

NEXT PLAY? 4
COMPLETE! 35 YARD GAIN.
1ST AND 10 ON THEIR 38.
```

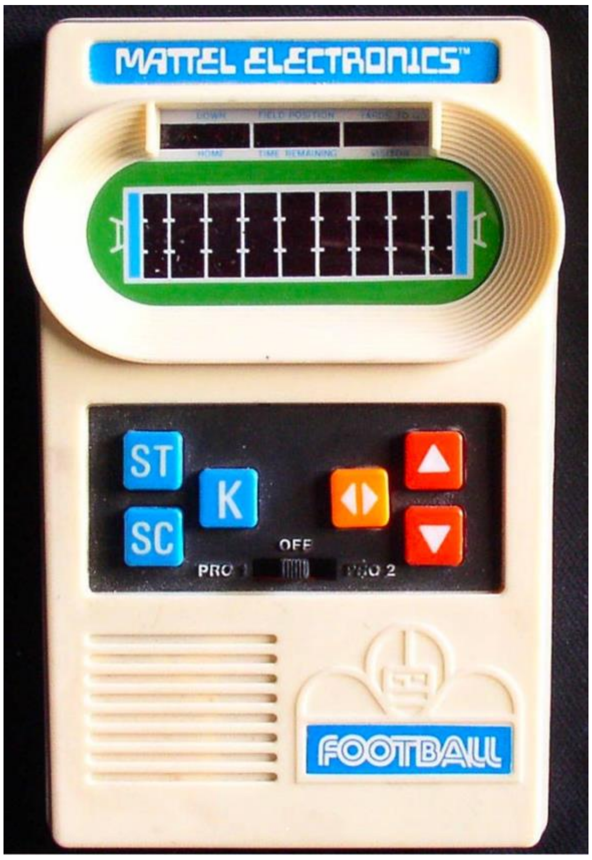
Further Explorations

- Create a computer-based version of soccer, tennis, boxing, baseball, cricket, etc.. in a similar text-based style. It's possible. Such programs were ubiquitous on 1970s timeshare systems.
- Add graphics and/or animation.
    - The Mattel Electronic Football (depicted) suggests a simple interface to build upon.
- Star Trek (specifically Trek 73) games were also popular in the 1970s and could use an update. (Video at 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

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

- How about putting your own spin on Pong, Space Invaders, or other early arcade games?



Phase 7: AI & Learning

Game 20: Animal

Category	Difficulty	Play Structure	Max Turns
AI / Learning	Intermediate–Advanced	Unlimited	Variable per round – the number of questions depends on the depth of the current knowledge tree.

Programming Concepts: Binary trees, dynamic data structures, yes/no decision logic, machine learning fundamentals

Overview

The computer tries to guess your animal by asking yes/no questions. When it fails, it asks YOU for a distinguishing question and adds it to its knowledge. The computer literally learns and gets smarter with each game. This is the most conceptually powerful program in the entire book.

Objective

Think of an animal. The computer asks yes/no questions to guess it. If the computer fails, you teach it.

Rules

1. The computer starts with a small knowledge tree (e.g. one question: 'Does it swim?' with FISH and BIRD as leaves).
2. The computer asks a series of yes/no questions, traversing the tree.
3. At a leaf node, it guesses an animal.
4. If correct, the computer celebrates.
5. If wrong, the player types the correct animal AND a yes/no question that distinguishes it from the computer's guess.
6. The player specifies which answer (yes or no) applies to the new animal.
7. The new question and animal are inserted into the tree.
8. Over many rounds, the tree grows and the computer becomes increasingly knowledgeable.

Turn Structure & Duration

Maximum Turns: Variable per round – the number of questions depends on the depth of the current knowledge tree. Unlimited rounds.

Play Structure: Unlimited – rounds continue as long as the player keeps thinking of animals.

Termination Conditions

Computer wins round: It guesses the player's animal correctly. Computer loses round: It guesses wrong, and the player provides a new animal and distinguishing question (the tree grows). Game over: The player answers 'NO' to 'Are you thinking of an animal?' The computer displays all animals it knows.

Sample Run

```
ANIMAL
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

ARE YOU THINKING OF AN ANIMAL? YES
DOES IT SWIM? YES
IS IT A FISH? NO
THE ANIMAL YOU WERE THINKING OF WAS A ?
DOLPHIN
PLEASE TYPE A QUESTION THAT DISTINGUISHES A
DOLPHIN FROM A FISH.
? IS IT A MAMMAL
FOR A DOLPHIN THE ANSWER WOULD BE? YES

ARE YOU THINKING OF AN ANIMAL? YES
DOES IT SWIM? YES
IS IT A MAMMAL? YES
IS IT A DOLPHIN? YES
WHY NOT TRY ANOTHER ANIMAL?
```

Game 21: Hexapawn

Category	Difficulty	Play Structure	Max Turns
AI / Learning	Advanced	Until win/loss per game	Each game lasts 2–5 moves total (the 3×3 board limits game length).

Programming Concepts: Machine learning by elimination, state tables, game memory, adaptive AI

Overview

A 3×3 board game with chess pawns. The computer starts as a terrible player but eliminates losing moves from memory after each defeat. Students literally watch an AI emerge from trial and error – the most dramatic teaching moment in the entire workshop.

Objective

Beat the computer at Hexapawn. But beware: the computer learns from each loss and will eventually become unbeatable.

Rules

1. The board is 3×3. Each player has 3 pawns on their home row.
2. Pawns can move forward one square (if empty) or capture diagonally forward.
3. A player wins by: (a) advancing a pawn to the opponent's home row, or (b) leaving the opponent with no legal moves.
4. The computer maintains a table of all possible board states and legal moves.
5. When the computer loses, it deletes the last move it made from the table for that board state.
6. If all moves for a state are eliminated, the computer deletes the move that led to that state (cascading learning).
7. Over 10–20 games, the computer converges on perfect play.

Turn Structure & Duration

Maximum Turns: Each game lasts 2–5 moves total (the 3×3 board limits game length). Unlimited rematches.

Play Structure: Until win/loss per game – each game ends when one side wins. Multiple games are played as the computer learns.

Termination Conditions

Player wins game: Get a pawn to the opposite row, OR leave the computer with no legal moves. Computer wins game: Get a pawn to the opposite row, OR leave the player with no legal moves. Learning: When the computer loses, it removes the move that led to the loss from its strategy table. Over many games, the computer's play improves until it is unbeatable. Session ends when the player chooses to stop.

Sample Run

```
HEXAPAWN
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

  X X X
  . . .
  O O O

YOUR MOVE? 23,22
  X X X
  . O .
  O . O
```

```
I MOVE 11,22
  X . X
  . X .
  O . O

YOUR MOVE? 31,22
YOU WIN.

--- GAME 2 ---
(I won't make that mistake again!)
```

Appendix: Quick Reference Table

A summary of all 21 games with their key specifications at a glance.

#	Game	Category	Difficulty	Structure	Termination
1	Hi-Lo	Number Guessing	Beginner	Unlimited	Win round: Guess the correct number (win $100). Lose round: Each incorrect gu...
2	Guess	Number Guessing	Beginner	Until win	Win: The player guesses the correct number. The computer reports the number o...
3	Stars	Number Guessing	Beginner	Fixed-length	Win: Guess the number within 7 tries. Lose: Fail to guess after 7 tries; the ...
4	Bagels	Logic / Code-Breaking	Beginner–Intermediate	Fixed-length	Win: Guess the exact 3-digit number (all three digits correct in correct posi...
5	Acey Ducey	Card Game	Beginner	Unlimited	Lose: Bankroll reaches $0 (the computer announces you're broke and offers to ...
6	Craps	Dice Game	Beginner–Intermediate	Until win/loss per round	Win round: Roll 7 or 11 on the come-out roll (natural), OR roll your point nu...
7	Blackjack	Card Game	Intermediate–Advanced	Until win/loss per hand	Win hand: Player's hand total is closer to 21 than the dealer's without excee...
8	War	Card Game	Beginner	Fixed-length	Win: Player has more round wins than the computer after 26 rounds. Lose: Comp...
9	Nim	Mathematical Strategy	Intermediate	Until win/loss	Win: You take the last object(s) from the final pile (normal play), OR your o...
10	23 Matches	Mathematical Strategy	Beginner	Until win/loss	Win: Your opponent is forced to take the last (23rd) match. Lose: You are for...
11	Reverse	Puzzle	Intermediate	Until win	Win: The list reads 1 2 3 4 5 6 7 8 9 from left to right. The computer report...
12	Tic-Tac-Toe	Board Game	Intermediate	Fixed-length	Win: A player places three marks in a horizontal, vertical, or diagonal line....
13	Hangman	Word Game	Intermediate	Until win/loss	Win: All letters in the word have been correctly guessed. Lose: The player ma...
14	Buzzword	Word / Humor	Beginner	Unlimited	Quit: The player responds 'NO' when asked if they want another phrase. There ...
15	Depth Charge	Search / Military	Intermediate	Fixed-length	Win: Guess the submarine's exact coordinates. Lose: Exhaust all allowed trial...

16	Gunner	Artillery / Physics	Intermediate	Fixed-length per target (5 shots), multi-round (5 targets). The overall game ends after 5 targets or when the player is destroyed.	Win target: Land a shot within 100 yards of the target (burst radius). Lose t...
17	Lunar LEM Rocket	Space Simulation	Intermediate–Advanced	Until win/loss	Perfect landing: Touchdown velocity is under 1 ft/sec. Good landing: Touchdow...
18	Hammurabi	Economic Simulation	Intermediate–Advanced	Fixed-length	Impeached (loss): More than 33% of the population starves in a single year. S...
19	Football	Sports Simulation	Advanced	Fixed-length	Win: Player's score exceeds the computer's score at the end of the 4th quarte...
20	Animal	AI / Learning	Intermediate–Advanced	Unlimited	Computer wins round: It guesses the player's animal correctly. Computer loses...
21	Hexapawn	AI / Learning	Advanced	Until win/loss per game	Player wins game: Get a pawn to the opposite row, OR leave the computer with ...

References

Hansen, S. (Host). (2024, May 29). *What's worth making?* [Audio podcast episode]. 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. (Ed.). (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. (2016, September 7). *How to teach computational thinking*. Stephen Wolfram Writings. https://writings.stephenwolfram.com/2016/09/how-to-teach-computational-thinking/