OHMYGIT.COM. 30/09/2025.

Gamified git commands.

Common commands

git addd . ; git commit.

git init

git checkout

git checkout HEAD -go to prev.

git checkout HEAD~2 -go 2 back.

Got to, Branches make parallel commit..

Creating branches is next.

Next. A grip on git.

agripongit-vinothunru.com

1 initialise repository > git init

2. git add thanks.txt ← staging thanks.txt

3. git commit HEAD main Good rule of thumb to commit seperate files at once. e.g you can go back to specific edits

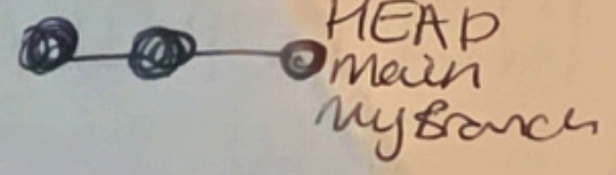
git add outline.txt

4. thanks HEAD outline. Lets say change more on thanks.txt then do:

git add thanks.txt → git commit "..."

5. main ← you can name main its a branch.
HEAD

6. Make new branch. git branch myBranch



7. As you continue on myBranch stays at that commit

8. Switch to persons branch via git switch myBranch

UP TO  New commit will follow branch HEAD is pointing to.

Developer Roadmap.sh

For my goal: To specialise between Audio DSP engineer, instrument developer, immersive interactive audio sound design.

Core Pathways.

Game developer/Server-side Game dev roadmap. ✓

+

C++ Roadmap. - JUCE, Steinberg VST3 Unreal C++. ✓
teaches memory management.

Systems Design:

+

Data Structures & Algos. for scalable audio engines plugins DSPs.

Rust - gaining traction for Safety + Performance.

Supporting.

Compsci. - for foundations of DSP math memory models, concurrency

Python - good for prototyping. DSP, analysis & machine-learning-driven sound design.

AI engineer - good but not the foundations. useful late on.

NOT CENTRAL: Frontend, Ux, technical, product manager. SQL, BI Analyst, Data analyst.

RoadMap ~~1st steps~~ Maths:

Game Dev & Server Side GD.
Linear algebra
Vector Matrix
Linear Transformations
Geometry Affine space
Affine Transformations
projection Orientation
Perspective Quaternion
Orthogonal Euler Angle

These are important, But much more, Print off maps.

C++
None? X
Systems design X
Data Structures Algos.
BigO BigΘ BigΩ
Rust.

Excel spreadsheet.

Main 1st Categories.

Game Mathematics, Physics & engines.
Basic operations, control flow & statements.
TCP, UDP, IP
Performance vs scalability, latency throughput
availability consistency consistency patterns
availability patterns
Basic data structures
Language basics error handling.
Pretrained AI models
Data structures again. Asynchronous relation with GD
Data structures again.

Common denominators.

Data Structures Algos.

Basic languages.

Big focus. C++ Python Data Structures & Algos.

OSSU, Courses. 1hour per day. CSVISTOOL.
helps visualise Algoalgorithmsbigos.
PSETO before lecl, simple questions on motivations.
In programming languages the primitives are
numbers, strings, simple operators
Sequences of characters
addn + - x ÷
< > etc.

Syntactically valid
e.g cat dog boy - not syntax valid
cat hugs boy - syntax valid
Python "hi" 5 - not syntax valid
"hi" * 5 - valid (means Hi Hi Hi Hi Hi)

Mon - CS50 30min Coursera OSSU 1hr Python ~~C++~~ System design
The - CS50 30min, MIT, OSSU 1hr C++ Game Dev
Wed - CS50 30min Coursera, OSSU 1hr Data struc. Algos Rust
Thur - CS50 30min MIT, OSSU 1hr Python System design
Fri - CS50 30min Coursera, OSSU 1hr C++ Game dev/serve side.
Sat - CS50 30min MIT, OSSU 1hr Data struc Algos. Compsci
Sun - CS50 Coursera OSSU Python AI engineer
M - CS50 MIT OSSU C++ Game Dev
T - CS50 Coursera OSSU Data structures Algos Rust
W - CS50 MIT OSSU Python system design
T - CS50 Coursera OSSU C++ Game Dev
F - CS50 MIT OSSU Data Structures Algos Comp Sci
S - CS50 Coursera OSSU Python AI engineer
S - CS50 MIT OSSU Left over.

Python

Related Roadmaps

- Backend Roadmap
- DevOps Roadmap
- AI & Data Scientist

Find the detailed version of this roadmap and other similar roadmaps

roadmap.sh

- Basic Syntax
- Variables and Data Types
- Conditionals
- Loops
- Learn the Basics
- Type Casting
- Exceptions
- Functions, Builtin Functions
- Lists
- Tuples
- Sets
- Dictionaries
- PyPI
- Pip
- Conda
- uv
- Poetry
- Package Managers
- Common Packages
- pyproject.toml
- Configuration
- List Comprehensions
- Generator Expressions
- Paradigms
- Context Manager
- Data Structures & Algorithms
- Arrays and Linked Lists
- Hash Tables
- Heaps, Stacks and Queues
- Binary Search Tree
- Recursion
- Sorting Algorithms
- Builtin
- Custom
- Modules
- Lambdas
- Decorators
- Iterators
- Regular Expressions
- Object Oriented Programming
- Classes
- Inheritance
- Methods, Dunder
- typing
- mypy
- pyright
- pyre
- Pydantic
- Static Typing
- Pipenv
- virtualenv
- pyenv
- Environments
- Learn a Framework
- Plotly Dash
- Pyramid
- Synchronous
- gevent
- aiohttp
- Tornado
- Sanic
- Asynchronous
- Synchronous + Asynchronous
- Fast API
- Django
- Flask
- Concurrency
- Multiprocessing
- Asynchrony
- GIL
- Threading
- yapf
- black
- ruff
- Code Formatting
- Sphinx
- Documentation
- Testing
- tox
- nose
- unittest / pyUnit
- doctest
- pytest

Have a look at the following related roadmaps

- Backend
- DevOps
- AI / Data Science

Related Roadmaps

- Data Structures and Algorithms
- System Design Roadmap
- Software Design & Architecture

Other Resources

- Coding Interview University

Computer Science

Find the detailed version of this roadmap along with other similar roadmaps

roadmap.sh

Pick a Language

Python
Go
C#
C++
C
Java
Rust

Data Structures

Array
Linked List
Stack
Queue
Tree
Hash Table

Graph
Heap

Binary Tree
Binary Search Tree
Full Binary Tree
Complete Binary Tree
Balanced Tree
Unbalanced Tree

Directed Graph
Undirected Graph
Spanning Tree

Representation

Adjacency List
Adjacency Matrix

Common Runtimes

Constant
Logarithmic
Linear
Polynomial
Exponential
Factorial

Big O
Big-Theta
Big Omega

Asymptotic Notation

Common Algorithms

Graphs

Breadth First Search
Depth First Search
Bellman Ford's Algorithm
Dijkstra's Algorithm
A* Algorithm

Tree

Pre-Order Traversal
In-Order Traversal
Post Order Traversal
Breadth First Search
Depth First Search

Recursion

Tail Recursion
Non-Tail Recursion

Searching

Binary Search
Linear Search

Sorting

Bubble Sort
Selection Sort
Insertion Sort
Heap Sort
Quick Sort
Merge Sort

Greedy Algorithms

Dijkstra's Algorithm
Huffman Coding
Kruskal's Algorithm
Ford-Fulkerson Algorithm
Prim's Algorithm

Back Tracking

Finding Hamiltonian Paths
Solving N Queen Problem
Maze Solving Problem
The Knight's Tour Problem
Robin-Karp Algorithm

Caches

MFU Cache
LRU Cache
LFU Cache

Search Pattern in Text
Suffix Arrays

Endianess
Floating Point Math
String Search & Manipulations

Unicode
ASCII

Big Endian
Little Endian

Character Encodings
Bitwise Operators
Common UML Diagrams

Design Patterns

GoF Design Patterns
Architectural Patterns
Dependency Injection
Null Object Pattern
Type Object Pattern

Substring Search

Brute Force Search
Robin-Karp
Knuth-Morris Pratt
Boyer-Moore

Class Diagrams
Usecase Diagrams
Activity Diagrams
Statemachine Diagrams
Sequence Diagrams

Basic Math Skills

Probability
Combinatorics

Complexity Classes

P
NP
P = NP
Co-NP
NP Hard
NP Complete
Travelling Salesman Problem
Knapsack Problem
Longest Path Problem

Tries

OSI Model
TCP/IP Model
DNS
HTTP
TLS & HTTPS
Sockets

SQL vs NoSQL Databases
Normalization / Denormalization
Entity-Relationship Model
DDL
DML
DQL
DCL

Databases

System Design

Balanced Search Trees

Networking

Locking
ACID Model
BASE
CAP Theorem
PACELC
Indexes
Views
Transactions
Stored Procedures
Database Federation
Replication
Sharding

Horizontal vs Vertical Scaling
Load Balancing
Clustering
Caching
CDN
Proxy
CAP Theorem
Queues
Architectural Styles
REST
GraphQL
gRPC
Cloud Design Patterns
Long Polling
Short Polling
Web Sockets
SSE

AVL Trees
Red / Black Trees
2 3 Search Trees
2 3 4 Trees
K-ary / M-ary Tree
B-Tree

K-D Trees
Skip Lists

Security

Hashing / Encryption / Encoding
Public Key Cryptography
Hashing Algorithms
OWASP Top 10

Process Forking
Memory Management
Lock / Mutex / Semaphore

How Computers Work

How CPU Executes Programs
Registers and RAM
Instructions and Programs
CPU Cache
How Computers Calculate

Processes and Threads

Concurrency in Multiple Cores
Scheduling Algorithms
CPU Interrupts
Processes vs Threads

Have a look at the following related roadmaps

Data Structures & Algorithms
API Design

Pre-requisites (One of these)

Frontend | Backend | Full-Stack

Scrimba is offering 20% off to roadmap users on their AI Engineer course that covers this roadmap in depth. Check them out!

Scrimba - AI Engineer Path

Related Roadmaps

- AI and Data Scientist Roadmap
- Prompt Engineering
- Data Analyst Roadmap

Find the detailed version of this roadmap along with other similar roadmaps

roadmap.sh

AI Engineer

Introduction

- What is an AI Engineer?
- AI Engineer vs ML Engineer
- Impact on Product Development
- Roles and Responsibilities

Using Pre-trained Models

- AI vs AGI
- LLMs
- Inference
- Training
- Embeddings
- Vector Databases
- AI Agents
- RAG
- Prompt Engineering

Common Terminology

Pre-trained Models

- Benefits of Pre-trained Models
- Limitations and Considerations
- OpenAI Models

Open AI Models
Capabilities / Context Length
Cut-off Dates / Knowledge

Popular AI Models

- Anthropic's Claude
- Google's Gemini
- Azure AI | AWS Sagemaker
- Hugging Face Models
- Mistral AI | Cohere
- Replicate

Open AI Platform

OpenAI API

- Chat Completions API
- Writing Prompts
- Maximum Tokens
- Token Counting
- Pricing Considerations

Managing Tokens

- Open AI Playground
- Fine-tuning
- Prompt Engineering Roadmap

AI Safety and Ethics

- Prompt Injection Attacks
- Security and Privacy Concerns
- Bias and Fairness

Understanding AI Safety Issues

- OpenAI Moderation API
- Adding end-user IDs in prompts
- Conducting adversarial testing
- Robust prompt engineering
- Know your Customers / Usecases
- Constraining outputs and inputs

Safety Best Practices

OpenSource AI

- Open vs Closed Source Models
- Popular Open Source Models
- Hugging Face

Finding Open Source Models

- Hugging Face Tasks
- Hugging Face Hub

Using Open Source Models

- Inference SDK
- Transformers.js
- Ollama

Ollama Models
Ollama SDK

Embeddings & Vector Databases

What are Embeddings

Semantic Search
Data Classification
Recommendation Systems
Anomaly Detection

Use Cases for Embeddings

Open AI Embedding Models
Pricing Considerations

Open AI Embeddings API

Sentence Transformers
Models on Hugging Face

Open-Source Embeddings

Vector Databases

Purpose and Functionality

Popular Vector DBs (pick one)

- Chroma | Pinecone
- Weaviate | FAISS
- LanceDB | Qdrant
- Supabase
- MongoDB Atlas

Implementing Vector Search

- Indexing Embeddings
- Performing Similarity Search

RAG & Implementation

- RAG Usecases
- RAG vs Fine-tuning

Implementing RAG

- Chunking
- Embedding
- Vector Database
- Retrieval Process
- Generation

Ways of Implementing RAG

- Using SDKs Directly
- Langchain | Llama Index

RAG Alternative

Open AI Assistant API

Agents Usecases
Prompt Engineering
ReAct Prompting

AI Agents

Building AI Agents

- Manual Implementation
- OpenAI Functions / Tools
- OpenAI Assistant API

Multimodal AI

Multimodal AI Usecases

- Image Understanding
- Image Generation
- Video Understanding
- Audio Processing
- Text-to-Speech
- Speech-to-Text

Multimodal AI Tasks

- OpenAI Vision API
- DALL-E API
- Whisper API
- Hugging Face Models
- LangChain for Multimodal Apps
- LlamaIndex for Multimodal Apps

Implementing Multimodal AI

AI Code Editors
Code Completion Tools

Development Tools

Continue learning with following relevant tracks

AI & Data Scientist | Prompt Engineering

Related Roadmaps

- Backend Roadmap
- DevOps Roadmap
- Go Roadmap

Find the interactive version of this roadmap and more roadmaps at

roadmap.sh

Rust

- Introduction
  - What is Rust?
  - Why use Rust?
  - Environment Setup
    - Installing Rust and Cargo
    - IDEs and Rust Toolchains
    - Rust REPL (Rust Playground)
- Language Basics
  - Syntax and Semantics
    - Variables, DataTypes and Constants
    - Control Flow and Constructs
    - Functions and Method Syntax
    - Pattern Matching & Destructuring
  - Constructs
    - Enums
    - Traits
    - Structs
    - Impl Blocks
  - Data Structures
    - Integers
    - Floats
    - Boolean
    - Character
    - String
    - Tuple
    - Array
    - Vector
    - Hashmap
    - Hashset
    - LinkedList
    - Stack
    - Queue
    - Binary Heap
  - Ownership System
    - Ownership Rules & Memory Safety
    - Borrowing, References and Slices
    - Deep Dive: Stack vs Heap
  - Advanced Topics
    - BTreeMap
    - BTreeSet
    - RC
    - Arc
    - Mutex
    - RwLock
    - Channels
- Error Handling
  - Option and Result Enumerations
  - Propagating Errors and '?' Operator
  - Custom Error Types and Traits
- Modules & Crates
  - Code Organization & Namespacing
  - Dependency Management with Cargo
  - Publishing on Crates.io
- Testing
  - Unit & Integration Testing
  - Mocking & Property Based Testing
- Concurrency & Parallelism
  - Threads, Channels and Message Passing
  - Atomic Operations & Memory Barriers
  - Futures and Async/Await Paradigm
- Traits & Generics
  - Trait Definitions & Implementations
  - Trait Bounds and Associated Types
  - Generics & Type-Level Programming
- Lifetimes & Borrow Checker
  - Explicit Lifetime Annotations
  - Lifetime Elision Rules
  - Covariant & Contravariant Lifetimes
- Macros & Metaprogramming
  - Declarative Macros with macro_rules!
  - Procedural Macros & Custom Derive
  - Domain Specific Languages (DSLs)
- Ecosystem and Libraries
  - Web Development
    - Axum
    - Actix
    - Leptos
    - Loco
    - Rocket

    Personal recommendation: Axum
  - Asynchronous Programming
    - Tokio
    - async-std
    - smol
  - Networking
    - reqwest
    - hyper
    - quinn
  - Serialization / Deserialization
    - Serde
    - json-rust
    - toml-rust
  - Cryptography
    - ring
    - rust-crypto
    - sodiumoxide
  - Database and ORM
    - Diesel
    - sqlx
    - rusqlite
  - CLI Utilities
    - termion
    - structopt
    - clap
  - GUI Development
    - tauri
    - gtk-rs
    - relm
  - Embedded and Systems
    - embedded-hal
    - rppal
    - nrf-hal
  - Game Development
    - bevy
    - fyrox
    - ggez
    - macroquad
    - wgpu-rs
  - WebAssembly (WASM)
    - wasm-pack
    - wasmer
    - wasm-bindgen
  - Performance and Profiling
    - Criterion.rs
  - Documenting with `rustdoc`
  - Debugging
    - rust-gdb
    - rust-lldb

Continue learning with following roadmaps

- Backend
- DevOps

Data Structures & Algorithms

Related Roadmaps

- Computer Science
- Programming Languages
- System Design
- Software Design & Architecture

Find the interactive version of this roadmap and more at

roadmap.sh

C#
JavaScript
C++
Java
Python
Go
Rust
Ruby

Pick a Language

Programming Fundamentals

- Language Syntax
- Control Structures
- Pseudo Code
- Functions
- OOP Basics

What are Data Structures?

Why are Data Structures Important?

Basic Data Structures

- Array
- Linked Lists
- Queues
- Stacks
- Hash Tables

Algorithmic Complexity

- Time vs Space Complexity
- How to Calculate Complexity?
- Asymptotic Notation
  - Big-O Notation
  - Big-Θ Notation
  - Big-Ω Notation

Common Runtimes

- Constant
- Logarithmic
- Linear
- Polynomial
- Exponential
- Factorial

Sorting Algorithms

- Bubble Sort
- Merge Sort
- Insertion Sort
- Quick Sort
- Selection Sort
- Heap Sort

Search Algorithms

- Binary Search
- Linear Search

Graph Data Structures

- Directed Graph
- Undirected Graph

Search Algorithms

- Breadth First Search
- Depth First Search

Shortest Path Algorithms

- Dijkstra's Algorithm
- Bellman-Ford Algoritm
- A* Algorithm

Minimum Spanning Tree

- Prim's Algorithm
- Kruskal's Algorithm

Tree Data Structures

- Binary Trees
- Binary Search Trees
- AVL Trees
- B-Trees
- Heap

Tree Traversal

- In-Order Traversal
- Pre-Order Traversal
- Post-Order Traversal

Search Algorithms

- Breadth First Search
- Depth First Search

Advanced Data Structures

- Trie
- Segment Trees
- Fenwick Trees
- Disjoint Set (Union-Find)
- Suffix Trees and Arrays

Complex Data Structures

- 2-3 Trees
- B/B+ Trees
- Skip List
- ISAM

Indexing

- Linear
- Tree-Based

Problem Solving Techniques

- Brute Force
- Backtracking
- Greedy Algorithms
- Island traversal
- Randomised Algorithms
- Multi-threaded
- DMide and Conquer
- Two Heaps
- Kth Element
- Recursion
- Merge Intervals
- Dynamic Programming
- Cyclic Sort
- Two Pointer Technique
- Fast and Slow Pointers
- Sliding Window Technique

Platforms to Practice

- Leetcode
- Edabit

Continue learning with following roadmaps

- Computer Science
- System Design

Other Roadmaps

- Backend Roadmap
- DevOps Roadmap
- Software Design and Architecture

Find the detailed version of this roadmap along with other similar roadmaps

System Design

- Introduction
  - What is System Design?
  - How to approach System Design?
- Performance vs Scalability
- Latency vs Throughput
- Availability vs Consistency
  - CAP Theorem
    - CP - Consistency + Partition Tolerance
    - AP - Availability + Partition Tolerance
- Consistency Patterns
  - Weak Consistency
  - Eventual Consistency
  - Strong Consistency
- Availability Patterns
  - Fail-Over
    - Active - Passive
    - Active - Active
  - Replication
    - Master - Slave
    - Master - Master
  - Availability in Numbers
    - Availability in Parallel vs Sequence
    - 99.99% Availability - four 9s
    - 99.9% Availability - three 9s
- Background Jobs
  - Scheduled Driven
  - Event-Driven
  - Returning Results
- Domain Name System
- Content Delivery Networks
  - Push CDNs
  - Pull CDNs
- Load Balancers
  - Horizontal Scaling
  - Layer 4 Load Balancing
  - Layer 7 Load Balancing
  - Load Balancing Algorithms
  - LB vs Reverse Proxy
- Application Layer
  - Service Discovery
  - Microservices
- Databases
  - RDBMS
    - SQL Tuning
    - Sharding
    - Federation
    - Denormalization
    - Replication
  - NoSQL
    - Graph Databases
    - Wide Column Store
    - Document Store
    - Key-Value Store
  - SQL vs NoSQL
- Caching
  - Strategies
    - Refresh Ahead
    - Write-behind
    - Write-through
    - Cache Aside
  - Types of Caching
    - Application Caching
    - Database Caching
    - Web Server Caching
    - CDN Caching
    - Client Caching
- Asynchronism
  - Message Queues
  - Task Queues
  - Back Pressure
- Idempotent Operations
- Communication
  - TCP
  - UDP
  - HTTP
  - RPC
  - gRPC
  - REST
  - GraphQL
- Performance Antipatterns
  - Busy Database
  - Busy Frontend
  - Chatty I/O
  - Retry Storm
  - No Caching
  - Improper Instantiation
  - Monolithic Persistance
  - Noisy Neighbor
  - Synchronous I/O
  - Extraneous Fetching
- Monitoring
  - Usage Monitoring
  - Security Monitoring
  - Performance Monitoring
  - Availability Monitoring
  - Health Monitoring
  - Instrumentation
  - Visualization & Alerts

The design patterns given in this section are of varying importance. You don't need to master all of them. Simply get an overview of each and this will give you some insight into designing scalable systems.

- Cloud Design Patterns
  - Messaging
    - Publisher/Subscriber
    - Priority Queue
    - Pipes and Filters
    - Competing Consumers
    - Choreography
    - Claim Check
    - Async Request Reply
    - Sequential Convoy
    - Scheduling Agent Supervisor
    - Queue-based Load Leveling
  - Data Management
    - Valet Key
    - Static Content Hosting
    - Sharding
    - Materialized View
    - Index Table
    - Event Sourcing
    - CQRS
    - Cache-Aside
- Design & Implementation
  - Backends for Frontend
  - Gateway Aggregation
  - Gateway Offloading
  - Gateway Routing
  - Ambassador
    - Pipes & Filters
  - CQRS
    - Leader Election
  - Sidecar
    - Strangler Fig
  - Anti-Corruption Layer
  - External Config Store
  - Compute Resource Consolidation
  - Static Content Hosting
- Reliability Patterns
  - Resiliency
    - Queue-Based Load Leveling
    - Scheduler Agent Supervisor
    - Retry
    - Leader Election
    - Health Endpoint Monitoring
    - Compensating Transaction
    - Circuit Breaker
    - Bulkhead
  - High Availability
    - Security
      - Valet Key
      - Gatekeeper
      - Federated Identity
    - Circuit Breaker
    - Health Endpoint Monitoring
    - Bulkhead
    - Geodes
    - Deployment Stamps
  - Availability
    - Queue-Based Load Leveling
    - Health Endpoint Monitoring
    - Throttling
    - Geodes
    - Deployment Stamps

Visit the following relevant tracks to learn more

- Backend
- Software Architect
- DevOps

Shout out to Chris Ohk who helped make the initial version of this roadmap.

Visit his GitHub

Game Developer

Find the detailed version of this roadmap along with other similar roadmaps

roadmap.sh

Client Side

Server-Side Development

Checksum

Segment Structure

Operations

Max Segment Size

Window Scaling

Connection

Resource Usage

Max Segment Scaling

Timestamp

Data Transfer

Out-of-Band Data

Selective Ack.

TCP

Vulnerability

Denial of Service

Connection Hijacking

Veto

Congestion Control

Reliable Transmission

Flow Control

Error Detection

UDP

Reliability

Datagram

Congestion Control

Checksum

Packet Structure

TCP vs UDP

Ordered vs Unordered

Heavy vs Lightweight

Packet vs Datagram

Reliable vs Unreliable

Streaming vs Broadcast

IP

Capability

Reliability

Link Capacity

Byte Manipulation

Address Conversion

Descriptor

Socket Programming

API

BSD Socket

Winsock

Programming Languages

Java

C#

C/C++

Erlang

JavaScript

Go

Serialization

JSON

TOML

XML

YAML

Protobuf

ARP

DNS

DHCP

IPv6

IPv4

TLS

Transactions

Category

Security

Function

Datagram Construction

IP Addressing

Routing

Sharding

Fiber

Multithreading

Thread Local Storage

Windows

pthread

Design Patterns

TDD

Dependency Injection

Dump Analysis

Functional Programming

Programming Techniques

Synchronization

Condition Variable

Spinlock

Barrier

Future & Promises

Mutex

Key-Value

Redis

Memcached

MS SQL

MySQL

PostgreSQL

RDBMS

DynamoDB

MongoDB

Cassandra

Couchbase

NoSQL

Databases

ORM

DAL

Coroutine

Channel

Semaphore

SSL

Cipher

REST

gRPC

RPC / REST

Apache Kafka

RabbitMQ

Message Queues

Thread Building Block (C++)

Concurrency (Java)

Async-await (C#)

goroutine (Go)

Task-Based

Actor Model

Akka.net (C#)

Akka (Java)

Update Process

Determinism

Synchrony

Reactive Model

Asynchronous

Reactor

WSA Poll

select

kqueue

epoll

OORP

FRP

Reactive Approach

Proactor

IOCP

io_uring

Registered IO

Maximise your skills

Multithreading

Serverless

Cloud

Azure

GCP

AWS

Containerization

Docker

Docker Compose

Kubernetes

Data Clustering

Apache Spark

AI

Cloud ML

Amazon ML

Azure ML

Deep Learning

TensorFlow

Pytorch

Visit the following relevant roadmaps

Client-side game Development

API Design

C++

Beginner Topics / Start here
Intermediate Topics / Pick Next
Advanced Topics / Pick Later
Optional / Learn if you want

References
Balmer's C++ Roadmap

Find the detailed version of this roadmap along with other similar roadmaps

roadmap.sh

Introduction to Language
What is C++?
Why use C++
C vs C++

Setting up your Environment
Installing C++
Code Editors / IDEs
Running your First Program

Basic Operations
Arithmetic Operators
Logical Operators
Bitwise Operators

If else / switch / goto
for / while / do while loops
Control Flow & Statements

Pointers and References
References
Memory Model
Lifetime of Objects
Smart Pointers
weak_ptr
shared_ptr
unique_ptr
Raw Pointers
New/Delete Operators
Memory Leakage

Data Types
Static Typing
Dynamic Typing
RTTI

Functions
Operator Overloading
Lambdas
Static Polymorphism
Function Overloading

Forward Declaration
Headers / CPP Files
Namespaces
Scope

Structuring Codebase

Virtual Methods
Virtual Tables
Dynamic Polymorphism
Object Oriented Programming

Structures and Classes
Multiple Inheritance
Diamond Inheritance
Rule of Zero, Five, Three

Language Concepts
auto (Automatic Type Deduction)
Type Casting
static_cast
const_cast
dynamic_cast
reinterpret_cast
Undefined Behavior (UB)
Argument Dependent Lookup (ADL)
Name Mangling
Macros

Exception Handling
Exit Codes
Exceptions
Access Violations

Type Traits
SFINAE
Template Specialization
Full Template Specialization
Partial Template Specialization
Variadic Templates

Standard Library + STL
Iterators
Iostream
Algorithms
Date / Time
Multithreading
Containers

Templates

WinDBG
GDB
Debugging Symbols
Understanding Debugger Messages

C++ 11 / 14

Idioms
Non-Copyable / Non-Moveable
Erase-Remove
Copy and Swap
Copy on Write
RAII
Pimpl
CRTP

Debuggers
Language Tools
Standards
C++ 17
C++ 20
Newest
C++ 0x

Compilers
Compiler Stages
Compilers and Features
C/clang++ / LLVM
Intel C++
MSVS C++
GCC
MinGW

Licensing
Library Inclusion

Libraries
Boost
OpenCV
POCO
Tensorflow
protobuf
spdlog
gRPC
pybind11
fmt
opencl
ranges_v3

Working with Libraries

Build Systems
CMAKE
Makefile
Ninja

Package Managers
vcpkg
Spack
Conan
NuGet

gtest / gmock
Qt
Catch2
Orbit Profiler
PyTorch C++
Frameworks

Keep learning with following relevant tracks
Backend Roadmap

Shout out to Chris Ohk who helped make the initial version of this roadmap.

Visit his GitHub

Game Developer

Find the detailed version of this roadmap along with other similar roadmaps

roadmap.sh

Client Side Development

Server Side

Linear Algebra

Vector

Matrix

Linear Transformation

Geometry

Affine Space

Affine Transformation

Projection

Orientation

Perspective

Quaternion

Orthogonal

Euler Angle

Game Mathematics

Curve

Spline

Hermite

Bezier

Catmull-Rom

Center of Mass

Moment of Inertia

Acceleration

Joints

Restitution

Force

Angular Velocity

Buoyancy

Friction

Linear Velocity

Dynamics

Unity 3D

Unreal Engine

Native

Godot

Game Engine

Game Physics

SAT

GJK

EPA

Intersection

Convexity

Collision Detection

CCD

Narrow Phase

Broad Phase

C#

C/C++

Rust

Python

GDScript

Programming Languages

Convex Hull

Convex Decomposition

Convex

Concave

Getting deeper

OBB

AABB

Sort & Sweep

DBVT

BVH

Bounding Volume

Spatial Partitioning

Computer Animation

Color

Visual Perception

Tone Reproduction

Stencil Shadow

Ray Tracing

Rasterization

Graphics Pipeline

Sampling

Shader

Rendering Equation

Mapping

Reflection

Texture

Diffuse

Bump

Specular

Parallax

Horizon

Computer Graphics

Lighting and Shadow

Shadow Map

Light Source

2D

Cube

Directional

Spot

Cascaded

Infinite

Point

Visibility and Occlusion

Occluder

Culling

Clipping

Fog

Frustum

Polygon

Light

Polyhedron

Shadow

HLSL

GLSL

SPIR-V

DirectX

OpenGL

Vulkan

Graphics API

OpenGL ES

Metal

WebGL

Game AI

Decision Making

Movement

Naive Bayes Classifier

Decision Learning

Reinforcements Learning

Decision Tree Learning

Deep Learning

Artificial Neural Network

Maximise your skills

Learning

Game AI

Decision Tree

State Machine

Behavior Tree

Fuzzy Logic

Markov System

Goal Oriented Behavior

Board Game

Minimax

AB Pruning

MCTS

Physically-Based Rendering

Advanced Rendering

Real-time Ray Tracing

Microsurface Scattering

Translucency & Transparency

Conservation of Energy

Metallicity

Visit the following relevant roadmaps

Backend

API Design

DirectX Ray Tracing

Vulkan Ray Tracing

OptiX