

CoType - Cooperative typing game

Distributed systems 25/26

Simone Ceredi

Alma Mater Studiorum - Università di Bologna

September 1, 2026

Index

- 1 Introduction
 - Idea
- 2 Architecture & design
 - Architecture
 - Design
- 3 Implementation & Quality Assurance
 - Technologies & methodologies
 - Terminal user interface

Idea

- Cooperative MonkeyType^a
- Shared code snippet
- Distributed keyboard
- Error handling
- Scalability
- No persistent storage

^a<https://monkeytype.com/>

Requirements

Functional

- Lobby management
- Character selection
- Gameplay
- Real time updates
- End-game statistics

Non-functional

- Scalability
- Fault tolerance
- Availability

Index

- 1 Introduction
 - Idea
- 2 Architecture & design
 - Architecture
 - Design
- 3 Implementation & Quality Assurance
 - Technologies & methodologies
 - Terminal user interface

Architecture

- Multiple clients
- Multiple servers
- One broker
- Service discovery
- Load balancing
- Error recovery
- AP over C

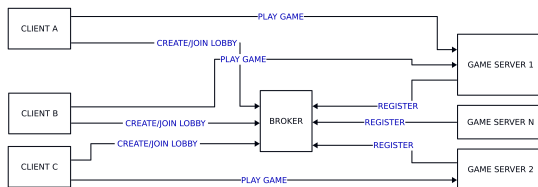


Figure: System architecture

Interactions pt. 1

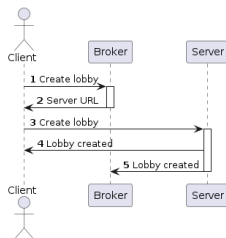


Figure: Lobby creation

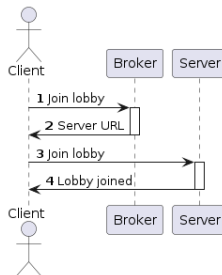


Figure: Lobby join

Interactions pt. 2

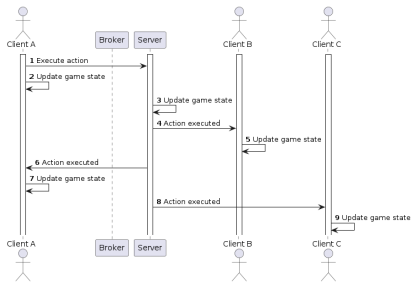


Figure: Normal gameplay

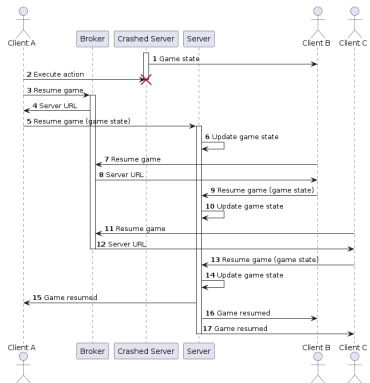


Figure: Server crash recovery

Index

- 1 Introduction
 - Idea
- 2 Architecture & design
 - Architecture
 - Design
- 3 Implementation & Quality Assurance
 - Technologies & methodologies
 - Terminal user interface

Implementation & Quality Assurance

- GoLang
- gRPC
- Kubernetes
- Test driven development
- Unit tests
- Integration tests
- Manual error recovery tests
- Manual end to end tests
- Code style
- Code linting
- Test check
- Automatic release

TUI pt.1



Figure: Homepage screen

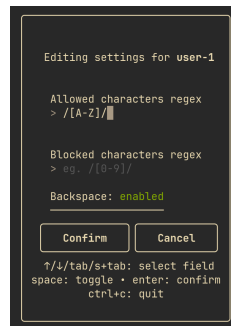


Figure: Statistics screen

TUI pt.2

```
server "github.com/sceredi/co-
type/server"
}

// Random reads a random .go source file
(excluding test files) from the
// embedded server module sources and returns
a random window of 10-30 lines.
func Random() string {
    var goFiles []string

    // WalkDir error is intentionally
    ignored; an empty slice falls back
    gracefully.
    fs.WalkDir(server.SourceFiles, ".",
    func(path string, d fs.DirEntry, err error)
    error {
        if err != nil {
            return nil
        }

        if !d.IsDir() &&
        strings.HasSuffix(path, ".go") &&
        !strings.HasSuffix(path, "_test.go") {
```

Players:
user-1
user-2

Game statistics:
Duration -> 03:54 (mm:ss)
Characters per second -> 2.32

Use "ctrl+c" to close co-type

Figure: Statistics screen

Figure: Gameplay screen