

On the Designing a Text Protocol for the Game of Kalah

Kaludercic, Philip^{1 2}

Seminar Presentation, 28. July 2022

¹philip.kaludercic@fau.de

²<https://wwwcip.cs.fau.de/~oj14ozun/>

Subject Matter

How to organise a comparative competition between student?

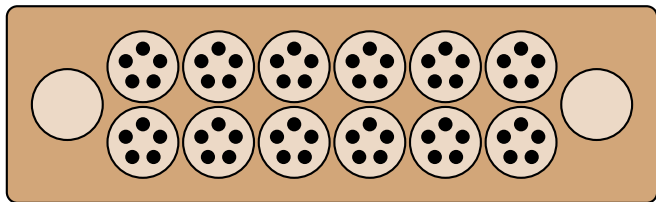
Case Study

The “Kalah” competition as part of the AI1 course.

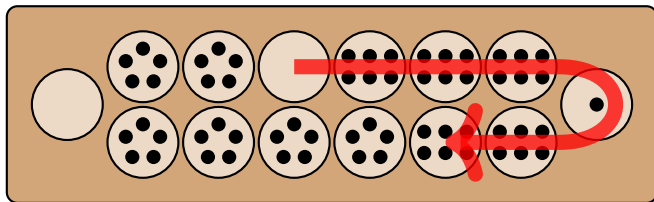
Intermission: What is “Kalah”?

An **abstract** board game between **two agents**.

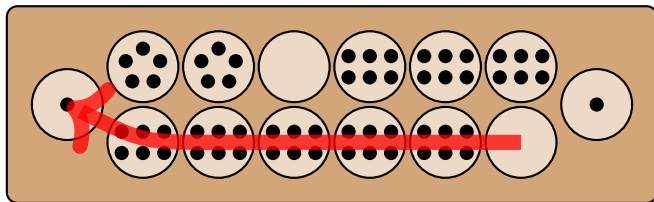
Intermission: What is “Kalah”?



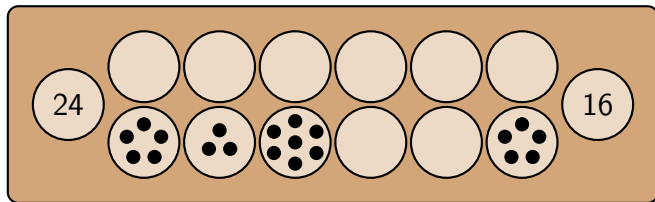
Intermission: What is "Kalah"?



Intermission: What is “Kalah”?



Intermission: What is “Kalah”?



Background

- ▶ There has already been a tournament for a few years

Background

- ▶ There has already been a tournament for a few years
- ▶ Clients were implemented in Java or Scala

Background

- ▶ There has already been a tournament for a few years
- ▶ Clients were implemented in Java or Scala
- ▶ The Framework was experiencing growth difficulties

Modest Proposal

Replace the old framework with an interactive protocol.

Design Choices

- ▶ Embed into an existing protocol or create something new?

Design Choices

- ▶ Embed into an existing protocol or create something new?
- ▶ Require a persistent connection or be stateless?

Design Choices

- ▶ Embed into an existing protocol or create something new?
- ▶ Require a persistent connection or be stateless?
- ▶ Simple or extensible?

... create something new?

Protobuf, MessagePack, ...

+ Ready-made libraries exist

Custom Plaintext Protocol

... create something new?

Protobuf, MessagePack, ...

- + Ready-made libraries exist
- Increased complexity

Custom Plaintext Protocol

... create something new?

Protobuf, MessagePack, ...

- + Ready-made libraries exist
- Increased complexity

Custom Plaintext Protocol

- + Can be kept simple

... create something new?

Protobuf, MessagePack, ...

- + Ready-made libraries exist
- Increased complexity

Custom Plaintext Protocol

- + Can be kept simple
- Requires some parsing

... a persistent connection?

Persistent

+ Conceptually simple

Short-lived

... a persistent connection?

Persistent

- + Conceptually simple
- Connection can break

Short-lived

... a persistent connection?

Persistent

- + Conceptually simple
- Connection can break

Short-lived

- + More resilient

... a persistent connection?

Persistent

- + Conceptually simple
- Connection can break

Short-lived

- + More resilient
- Conceptually less simple and slower

... extensible?

Extensible

+ Futureproof

Fixed

... extensible?

Extensible

- + Futureproof
- Requires more overhead

Fixed

... extensible?

Extensible

- + Futureproof
- Requires more overhead

Fixed

- + More simple

... extensible?

Extensible

- + Futureproof
- Requires more overhead

Fixed

- + More simple
- Hard to adapt

Client-Server Protocol Example

kgp 1 0 0

Client-Server Protocol Example

```
kgp 1 0 0  
mode freeplay
```

Client-Server Protocol Example

```
kgp 1 0 0  
mode freeplay  
4 state <3,0,0,3,3,3,3,3,3>
```

Client-Server Protocol Example

```
kgp 1 0 0  
mode freeplay  
4 state <3,0,0,3,3,3,3,3,3>  
@4 move 1
```

Client-Server Protocol Example

```
kgp 1 0 0  
mode freeplay  
4 state <3,0,0,3,3,3,3,3,3>  
@4 move 1  
6@4 stop
```


Client-Server Protocol Example

```
kgp 1 0 0
mode freeplay
4 state <3,0,0,3,3,3,3,3,3>
@4 move 1
6@4 stop
8 state <3,1,3,0,4,4,4,3,3>
@8 move 3
```

Client-Server Protocol Example

```
kgp 1 0 0
mode freeplay
4 state <3,0,0,3,3,3,3,3,3>
@4 move 1
6@4 stop
8 state <3,1,3,0,4,4,4,3,3>
@8 move 3
@8 move 5
```

Client-Server Protocol Example

```
kgp 1 0 0
mode freeplay
4 state <3,0,0,3,3,3,3,3,3>
@4 move 1
6@4 stop
8 state <3,1,3,0,4,4,4,3,3>
@8 move 3
@8 move 5
```

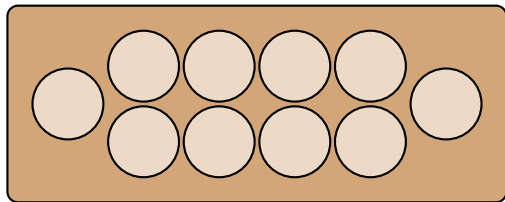
... and so on ...

The Board Representation

$\langle 4, 2, 1, 2, 4, 3, 3, 1, 3, 0, 4 \rangle$

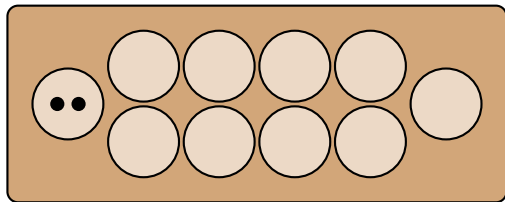
The Board Representation

$\langle 4, 2, 1, 2, 4, 3, 3, 1, 3, 0, 4 \rangle$



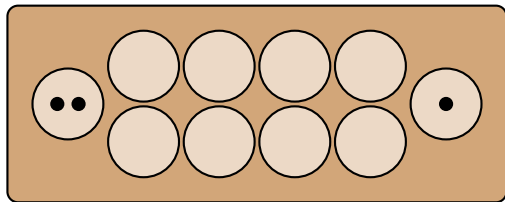
The Board Representation

$\langle 4, 2, 1, 2, 4, 3, 3, 1, 3, 0, 4 \rangle$



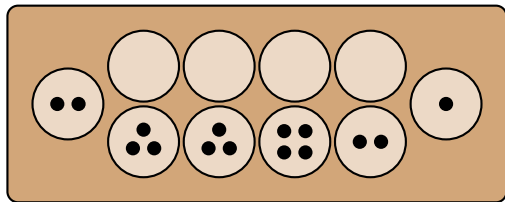
The Board Representation

$\langle 4, 2, 1, 2, 4, 3, 3, 1, 3, 0, 4 \rangle$



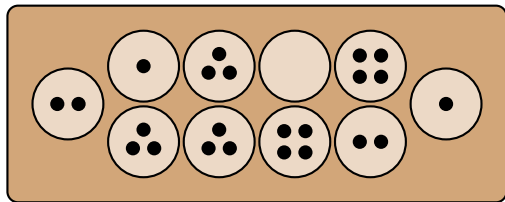
The Board Representation

$\langle 4, 2, 1, 2, 4, 3, 3, 1, 3, 0, 4 \rangle$



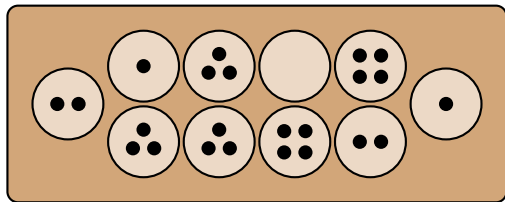
The Board Representation

$\langle 4, 2, 1, 2, 4, 3, 3, 1, 3, 0, 4 \rangle$



The Board Representation

$\langle 4, 2, 1, 2, 4, 3, 3, 1, 3, 0, 4 \rangle$



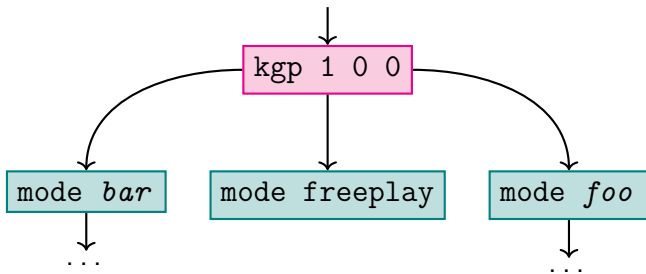
→ Compromise between a “statelessness” and
“persistence”

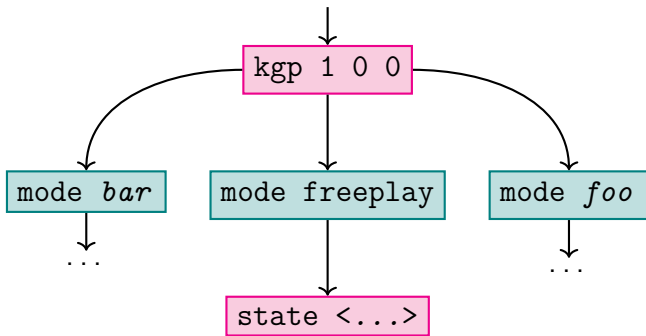


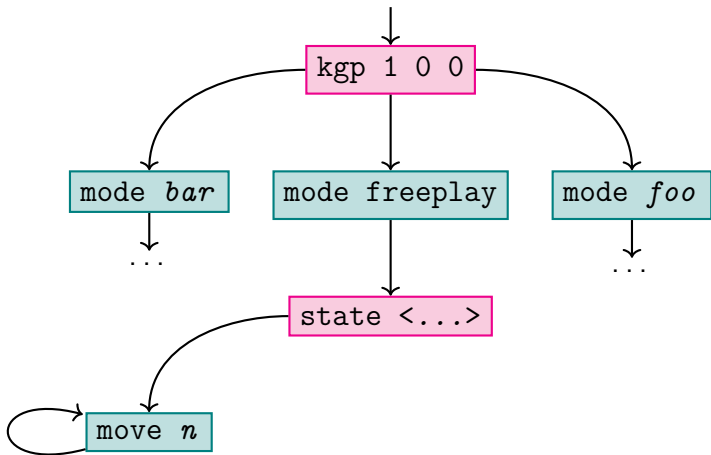
kgp 1 0 0

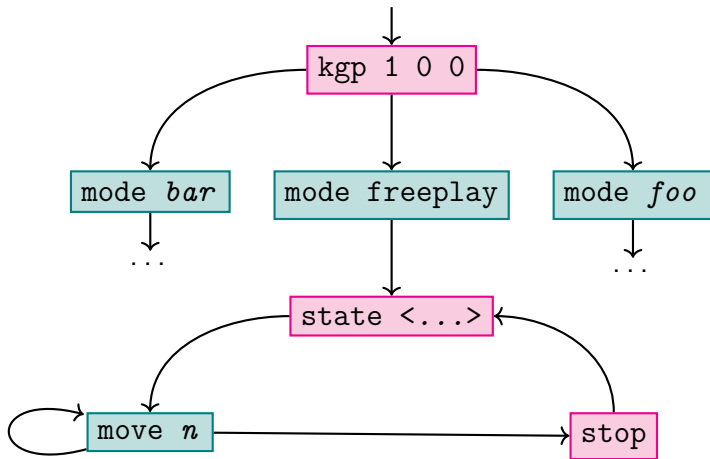
↓
kgp 1 0 0

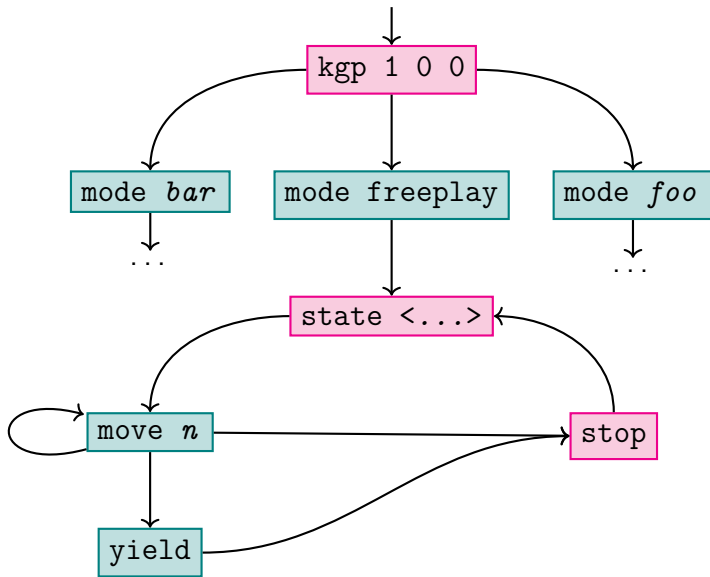
↓
mode freeplay

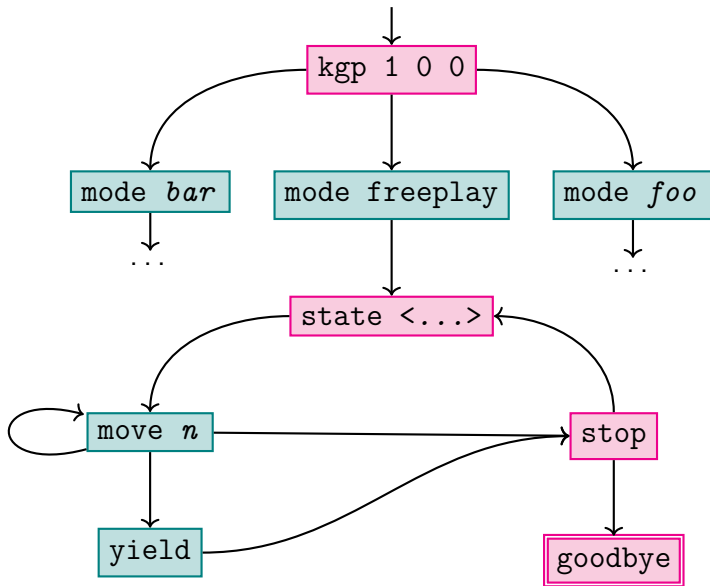


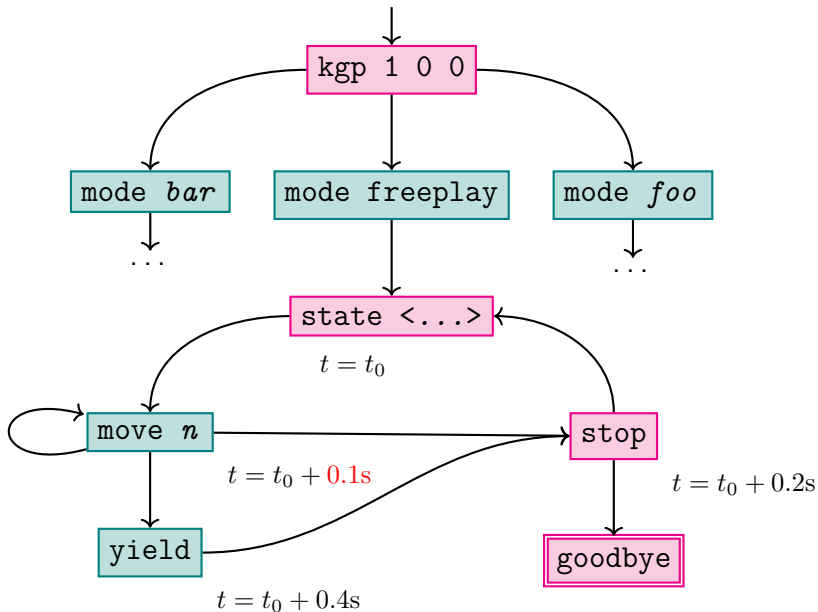


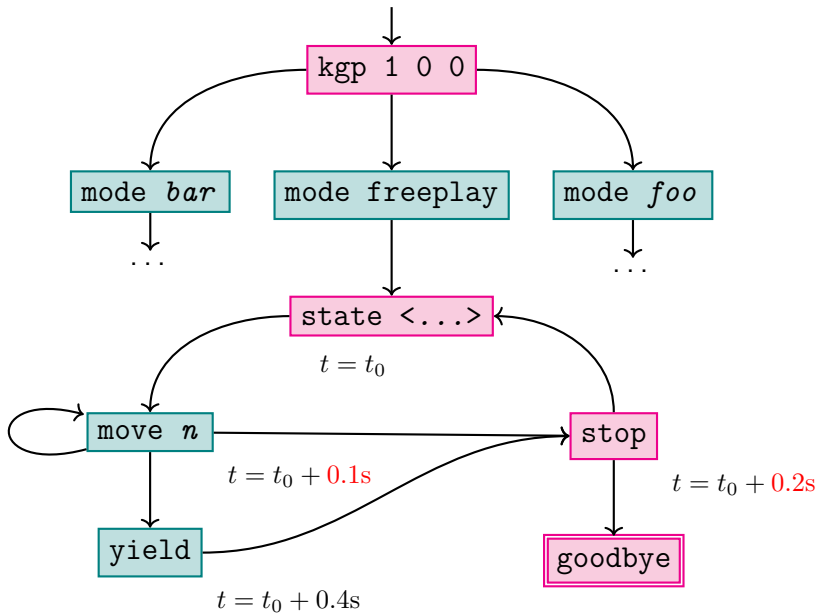


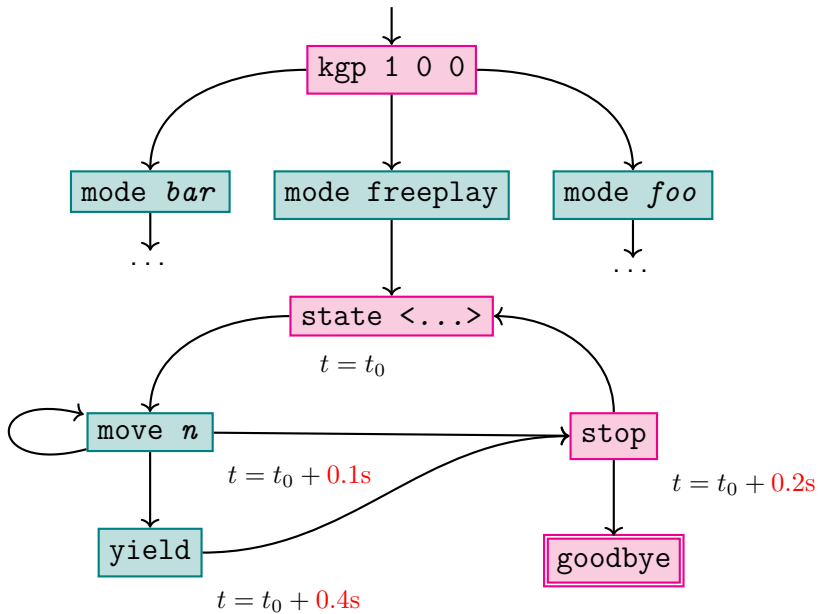


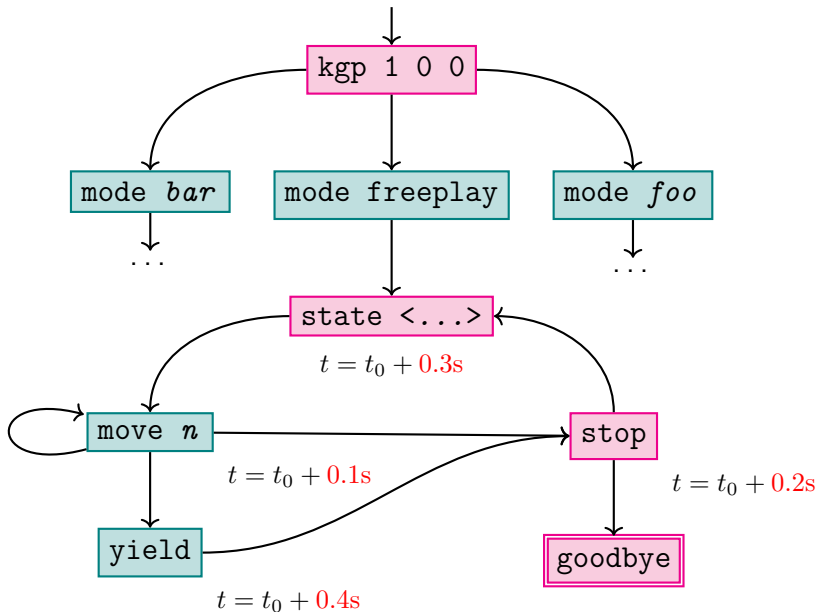












IDs and Reference numbers

Server

state <3,2,1,1,0,3,2,5,1>

stop

state <3,2,0,0,1,3,2,5,1>

stop

Client

move 1

move 1

move 1

move 2

move 3

IDs and Reference numbers

Server

4 state <3,2,1,1,0,3,2,5,1>

6 stop

8 state <3,2,0,0,1,3,2,5,1>

10 stop

Client

move 1

move 1

move 1

move 2

move 3

IDs and Reference numbers

Server

4 state <3,2,1,1,0,3,2,5,1>

6@4 stop

8 state <3,2,0,0,1,3,2,5,1>

10@6 stop

Client

@4 move 1

@4 move 1

@4 move 1

@8 move 2

@8 move 3

IDs and Reference numbers

Server

4 state <3,2,1,1,0,3,2,5,1>

6@4 stop

8 state <3,2,0,0,1,3,2,5,1>

10@6 stop

Client

@4 move 1

@4 move 1

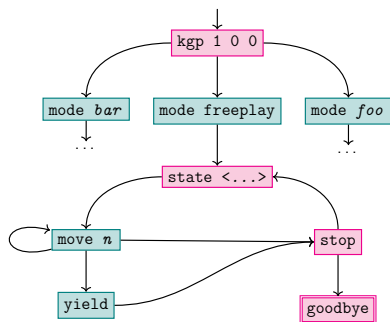
@4 move 1

@8 move 2

@8 move 3

Ensuring **transactional behaviour** to avoid **race conditions**

Weak Extensibility



- ▶ Small and modular “core language” keeps implementations simple

Weak Extensibility

```
kgp 1 0 0
set info:name "John Doe"
mode freeplay
set game:id 98031512
4 state <3,0,0,3,3,3,3,3,3>
@4 move 1
set info:comment "Guess"
```

- ▶ Small and modular “core language” keeps implementations simple
- ▶ Server and client can communicate *hints* using set-commands

Weak Extensibility

```
set auth:token "60b725f10c"  
set game:id 98031512
```

- ▶ Small and modular “core language” keeps implementations simple
- ▶ Server and client can communicate *hints* using set-commands
- ▶ Set-commands can model shared state or “pseudo-Variables”

Kalah Game Protocol

K
G
P

Kalah Game Protocol

Kaludercic, Philip

Völk, Tobias

Abstract

This document specifies a protocol for playing the game Kalah, a member of the Mancala family. It has been designed to be modularized, so that not all implementations have to implement all features. The main modules presented here are freeplay, evaluation and validation.

This document specified version 1.0.0 of the KGP protocol.

Contents

1	Prelude	1
1.1	Definitions	1
1.2	Formal Structure	1
1.3	Protocol Overview	2
2	Default Modes	2
2.1	Freeplay Mode	3
3	Freeplay commands	3
4	Simple Mode	3
4.1	Simple commands	3

1 Prelude

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “NOT RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.1 Definitions

A server organizes activities between one or more clients. The server waits for clients to request an activity, that the server may or may not organize. Activities cannot be changed, after they have been requested.

The server and the client communicate using a text-based, line-oriented protocol, over a reliable, ordered and error-checked transport layer (e.g. TCP).

1.2 Formal Structure

The protocol consists of commands sent between client and server. Server-to-client and client-to-server commands have the same form, consisting of:

Client Libraries

Concrete *KGP* Implementations

Client Libraries

jkgp (Java)

Concrete *KGP* Implementations

Client Libraries

jkpg (Java)

pykpg (Python)

Concrete *KGP* Implementations

Client Libraries

jkgp (Java)

pykgp (Python)

libakgp (C/C++), WIP

Concrete *KGP* Implementations

Client Libraries

jkgp (Java)

pykgp (Python)

libakgp (C/C++), WIP

kgpc (Generic Wrapper)

Concrete *KGP* Implementations

Client Libraries

jkgp (Java)

pykgp (Python)

libakgp (C/C++), WIP

kgpc (Generic Wrapper)

Concrete *KGP* Implementations

Client Libraries

jkgp (Java)

pykgp (Python)

libakgp (C/C++), WIP

kgpc (Generic Wrapper)

Server

go-kgp (Go)

Minimal client example using pykgp

```
import kgp
```

Minimal client example using pykgp

```
import kgp
```

```
kgp.connect(random_agent)
```

Minimal client example using pykgp

```
import kgp

def random_agent(state):

kgp.connect(random_agent)
```

Minimal client example using pykgp

```
import kgp

def random_agent(state):
    moves = state.legal_moves(kgp.SOUTH)

kgp.connect(random_agent)
```

Minimal client example using pykgp

```
from random import choice
import kgp

def random_agent(state):
    moves = state.legal_moves(kgp.SOUTH)
    yield choice(moves)

kgp.connect(random_agent)
```

Tournament Design

Tournament Design

- Stage 1 Open “training” tournament for student to get a feeling for how well their agents perform (using *ELO*-Ranking)

Tournament Design

- Stage 1 Open “training” tournament for student to get a feeling for how well their agents perform (using *ELO*-Ranking)
- Stage 2 A closed competition of submitted agents under fixed conditions for additional credit

This is the web interface of a KGP („Kalah Game Protocol“) Server. The server implementation is `go-kgp`.

If a token is set, the agent will be listed in the local Scoreboard (see below). Clients without a token will be regarded as anonymous.

Name	Rating (ELO)	Author
HappyNewYear	2182	<i>anonymous</i>
KalahIstSchrecklich	2181	This framework is awesome
KalahMarie4	2159	Marco, Lukas
KalahIstFuerchterlich2	2133	This framework is awesome
HopefullyNotSoShallowAgent	2130	Pumping Laemmer
KalahIstFuerchterlich	2122	This framework is awesome
2ndtest	2054	<i>anonymous</i>
AlphaBetaKalah	2046	Flo
TheAgent	1916	Julian Peters, Timothee Glörfeld
MaybeShallowerAgent	1828	Pumping Laemmer
MinMax 5	1734	Tobias Völk [Former Tutor]
Molly	1694	Tim, Vale

Expectations and Complications

1. Use a plain TCP connection for the public server

Expectations and Complications

1. Use a plain TCP connection for the public server

⚡ RRZE do not like opening TCP sockets

Expectations and Complications

1. Use a plain TCP connection for the public server

⚡ RRZE do not like opening TCP sockets

→ Tunnel through Websocket

Expectations and Complications

1. Use a plain TCP connection for the public server
 - ⚡ RRZE do not like opening TCP sockets
 - Tunnel through Websocket
 - + Encryption available “for free”

Expectations and Complications

1. Use a plain TCP connection for the public server
 - ⚡ RRZE do not like opening TCP sockets
 - Tunnel through Websocket
 - + Encryption available “for free”
2. Just submit “regular” programmes

Expectations and Complications

1. Use a plain TCP connection for the public server
 - ⚡ RRZE do not like opening TCP sockets
 - Tunnel through Websocket
 - + Encryption available “for free”
2. Just submit “regular” programmes
 - ⚡ *How to compile/interpret them?*

Expectations and Complications

1. Use a plain TCP connection for the public server
 - ⚡ RRZE do not like opening TCP sockets
 - Tunnel through Websocket
 - + Encryption available “for free”
2. Just submit “regular” programmes
 - ⚡ *How to compile/interpret them?*
 - Build and run using Docker

- ▶ Language Popularity

- ▶ Language Popularity

 - 16 Python

Data, Comments and Future Ideas

- ▶ Language Popularity

 - 16 Python

 - 13 Java

Data, Comments and Future Ideas

- ▶ Language Popularity
 - 16 Python
 - 13 Java
 - 1 C++ and Python

- ▶ Language Popularity
 - 16 Python
 - 13 Java
 - 1 C++ and Python
 - 1 C and Python

▶ Language Popularity

16 Python

13 Java

1 C++ and Python

1 C and Python

1 Kotlin

Data, Comments and Future Ideas

- ▶ Language Popularity
 - 16 Python
 - 13 Java
 - 1 C++ and Python
 - 1 C and Python
 - 1 Kotlin
- ▶ 5 “borked” submissions

Data, Comments and Future Ideas

- ▶ Language Popularity
 - 16 Python
 - 13 Java
 - 1 C++ and Python
 - 1 C and Python
 - 1 Kotlin
- ▶ 5 “borked” submissions
- ▶ 3 low-effort plagiarism attempts

Data, Comments and Future Ideas

- ▶ Language Popularity

 - 16 Python

 - 13 Java

 - 1 C++ and Python

 - 1 C and Python

 - 1 Kotlin

- ▶ 5 “borked” submissions

- ▶ 3 low-effort plagiarism attempts

- ▶ Rethink the “training” tournament

Data, Comments and Future Ideas

- ▶ Language Popularity

 - 16 Python

 - 13 Java

 - 1 C++ and Python

 - 1 C and Python

 - 1 Kotlin

- ▶ 5 “borked” submissions

- ▶ 3 low-effort plagiarism attempts

- ▶ Rethink the “training” tournament

- ▶ Provide *ready-to-use* Docker templates

Data, Comments and Future Ideas

- ▶ Language Popularity

 - 16 Python

 - 13 Java

 - 1 C++ and Python

 - 1 C and Python

 - 1 Kotlin

- ▶ 5 “borked” submissions

- ▶ 3 low-effort plagiarism attempts

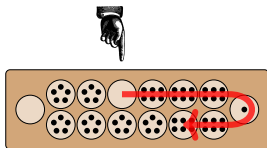
- ▶ Rethink the “training” tournament

- ▶ Provide *ready-to-use* Docker templates

- ▶ Implement more libraries

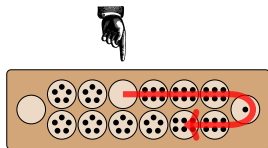
Summarié

Summarié

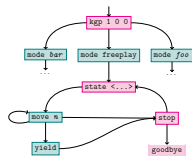


The Game of Kalah

Summarié

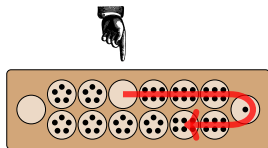


The Game of Kalah

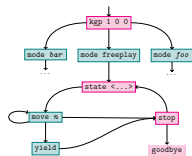


A Protocol

Summarié



The Game of Kalar



A Protocol

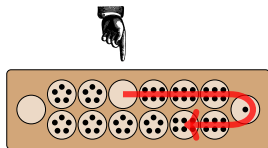
```
from random import choice
import kgp

def random_agent(state):
    moves = state.legal_moves(kgp.SOUTH)
    yield choice(moves)

kgp.connect(random_agent)
```

Libraries

Summarié



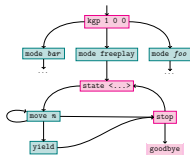
The Game of Kalah

```
from random import choice
import kgp
```

```
def random_agent(state):
    moves = state.legal_moves(kgp.SOUTH)
    yield choice(moves)
```

```
kgp.connect(random_agent)
```

Libraries



A Protocol

KGP Server | About

This is the web interface of a KGP („Kalah Game Protocol“) Server. The server implementation is `kgp`. If a token is set, the agent will be listed in the local Scoreboard (see below). Clients without a token will be regarded as anonymous.

Name	Rating (ELO)	Author
HappyNewYear	2182	anonymous
KalahSchrecklich	2181	This framework is awesome
KalahMaster	2159	Mirco, Lukas
KalahFurchtlich2	2133	This framework is awesome
HopeIJustGotSoShallowAgent	2130	Pumping Laetimer
KalahFurchtlich	2122	This framework is awesome
2ndbest	2054	anonymous
AlphaBetaKalah	2046	Flo
TheAgent	1916	Julian Peters, Timothee Girard
MaybeShallowAgent	1828	Pumping Laetimer
MiniMax 5	1734	Tobias Volk (Former Tutor)
Molly	1694	Tin, Vale

Tournament