Assignment ID2010

GAME OF TAG FOR AGENTS
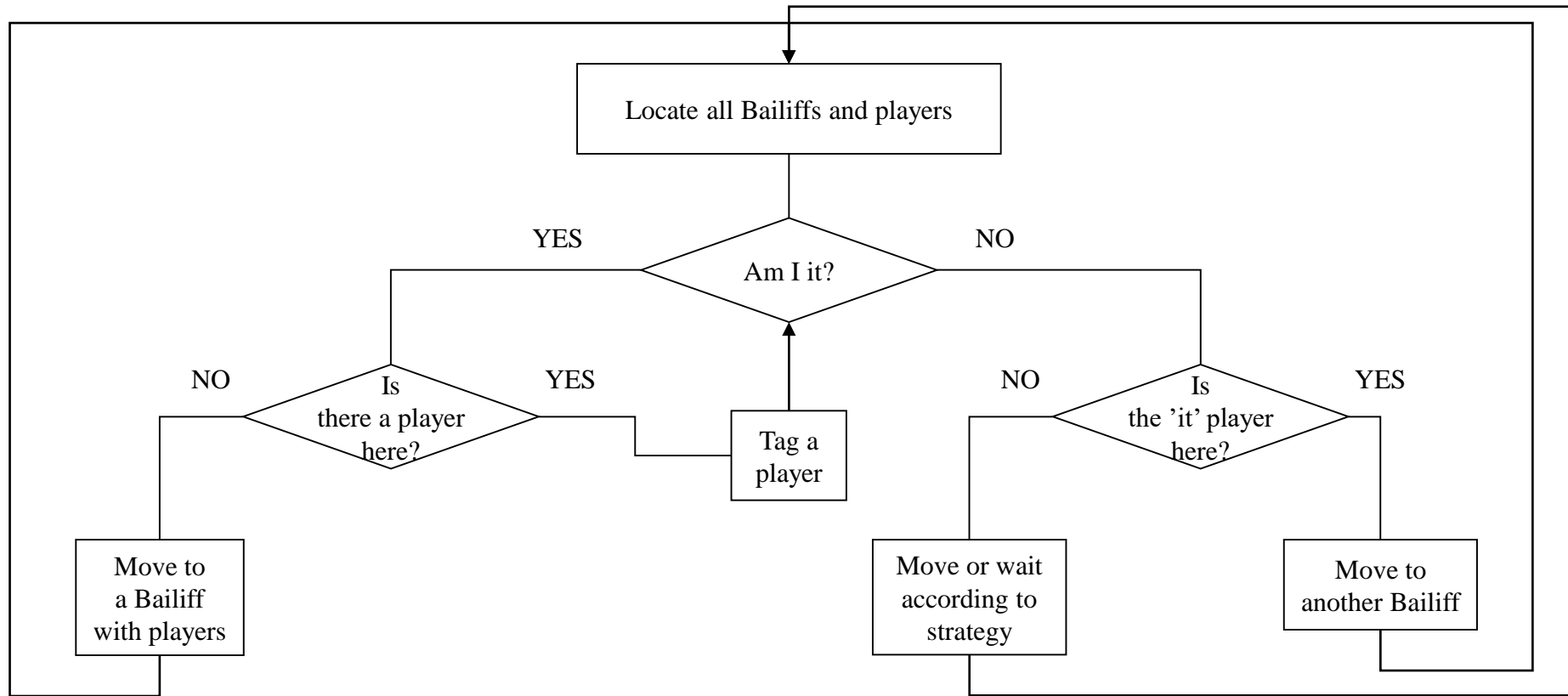
TAG player outline

```
        ┌─────────────────────────────────┐
        ↓                                 │
┌───────────────────────────┐            │
│   Acquire state of game    │            │
└───────────────────────────┘            │
        │                                 │
        ↓                                 │
┌───────────────────────────┐            │
│     Decide what to do      │            │
└───────────────────────────┘            │
        │                                 │
        ↓                                 │
┌───────────────────────────┐            │
│      Execute choice        │            │
└───────────────────────────┘            │
        │                                 │
        └─────────────────────────────────┘
```
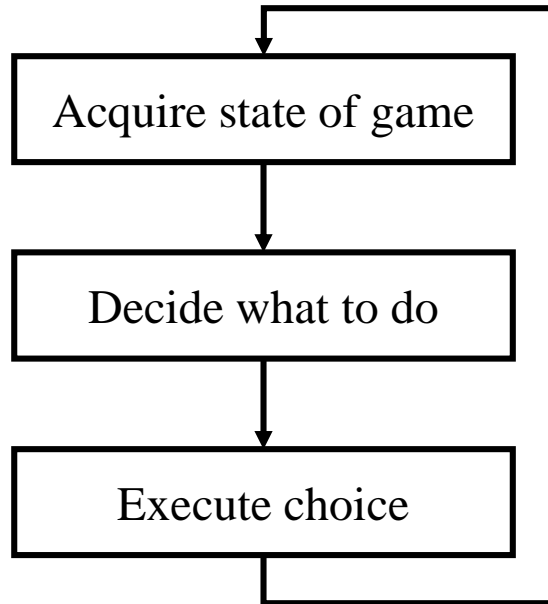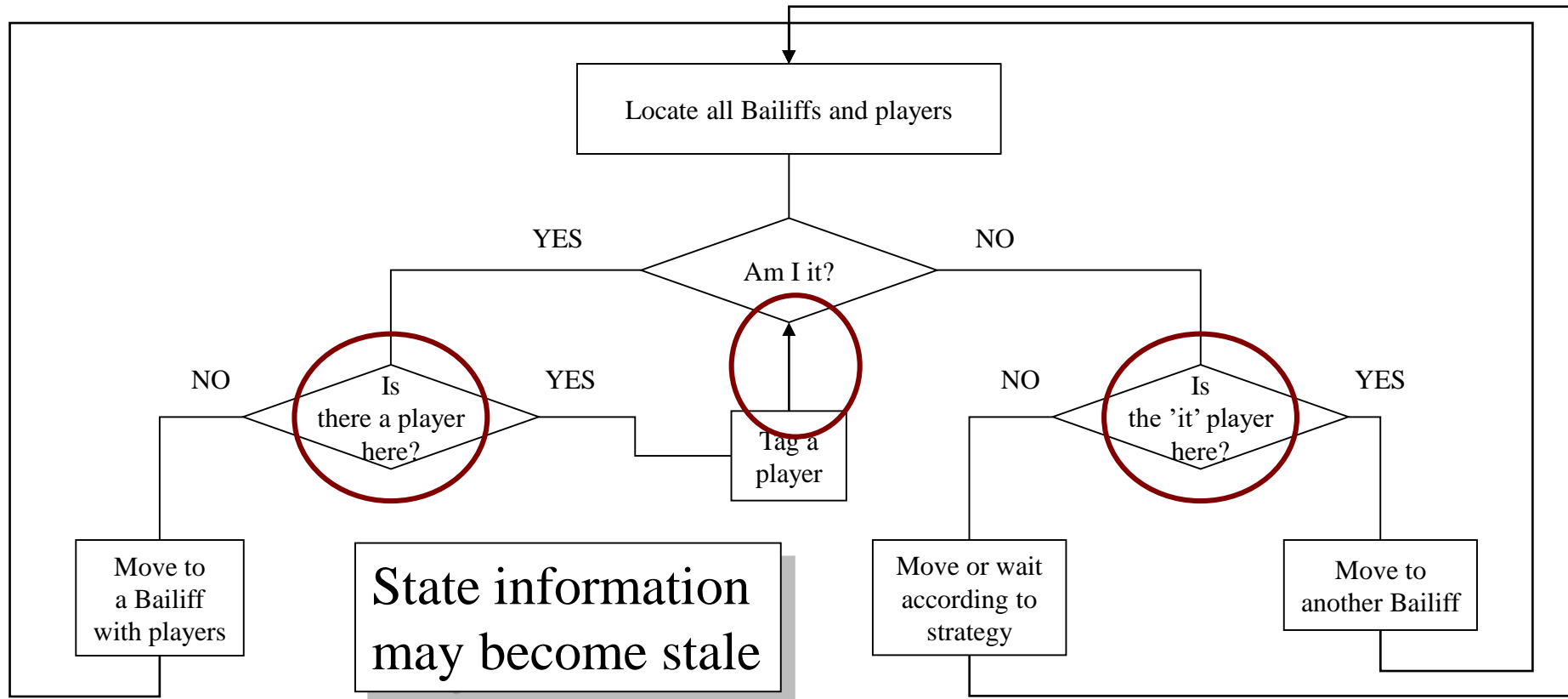
Game of Tag for agents

- Discover execution environments (Bailiffs)
- Query properties of the exec. env.
- Locate other agents
- Negotiate for status (being 'it' or not)
- Transaction (reliably transferring 'it' between players)
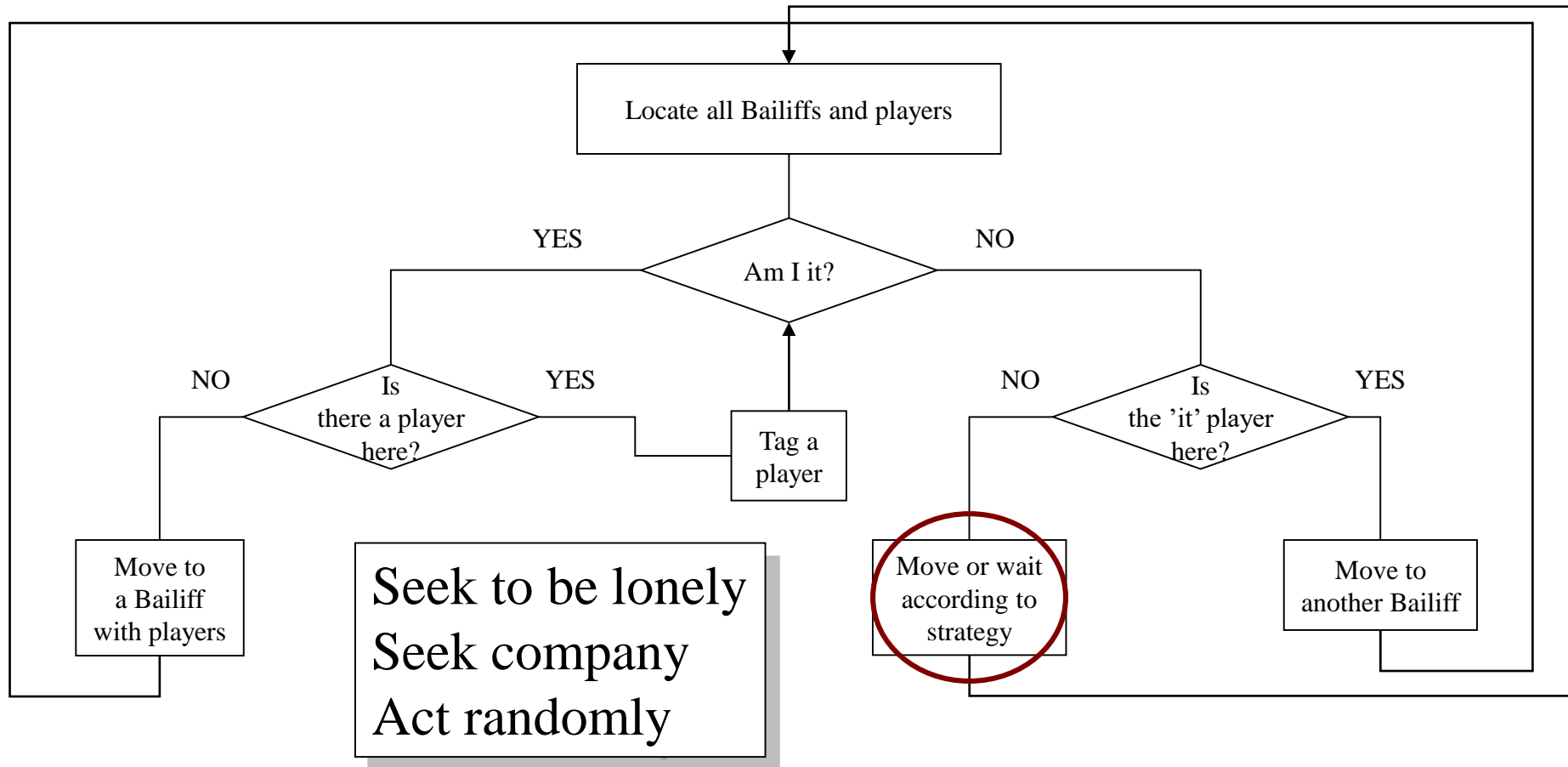- Multiple roles and tasks dep. on being 'it'.
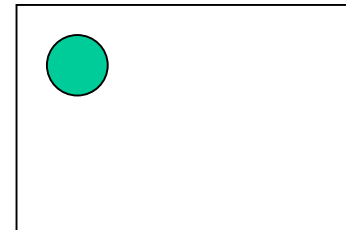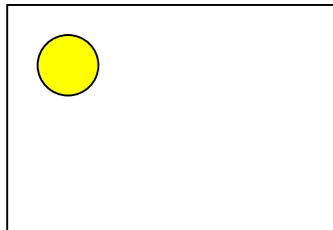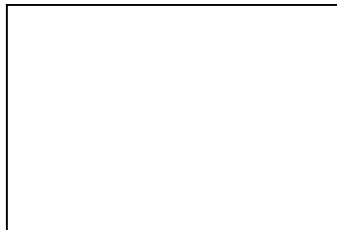
TAG player outline

Locate all Bailiffs and players

Am I it?

YES   NO

Is there a player here?

NO   YES

Tag a player

Is the 'it' player here?

NO   YES

Move to a Bailiff with players

Move or wait according to strategy

Move to another Bailiff

TAG player outline

```
                    ┌─────────────────────────────────────────┐
                    │   Locate all Bailiffs and players        │
                    └─────────────────────────────────────────┘
                                      │
          YES                         ◇ Am I it?                        NO
        ┌────────────────────────────/      \──────────────────────────────┐
        │                                                                   │
 NO     ◇ Is              YES                    ┌──────────┐    NO     ◇ Is         YES
┌───── there a player ──────┐                    │  Tag a   │   ┌──── the 'it' player ────┐
│       here?               │                    │  player  │   │      here?              │
│                           └────────────────────┘          │   │                         │
│                                                            │   │                         │
┌──────────┐          ┌────────────────────────────┐   ┌──────────────┐        ┌──────────────┐
│ Move to  │          │ State information          │   │ Move or wait │        │ Move to      │
│ a Bailiff│          │ may become stale           │   │ according to │        │ another Bailiff│
│ with players│       └────────────────────────────┘   │ strategy     │        └──────────────┘
└──────────┘                                            └──────────────┘
```

TAG player outline

Locate all Bailiffs and players

Am I it?

YES

NO

Is there a player here?

NO

YES

Is the 'it' player here?

NO

YES

Tag a player

Move to a Bailiff with players

Seek to be lonely
Seek company
Act randomly
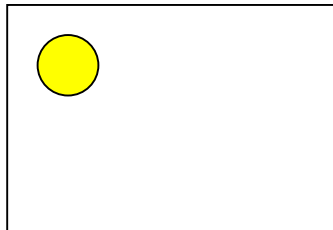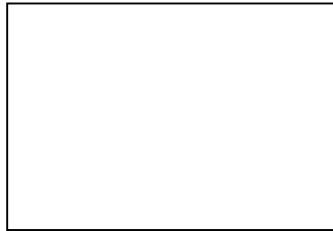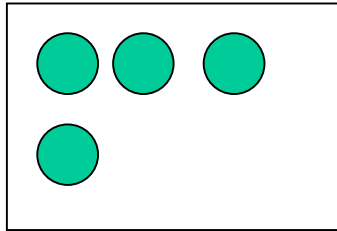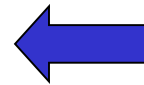
Move or wait according to strategy

Move to another Bailiff

TAG player outline

Seek to be lonely
Seek company
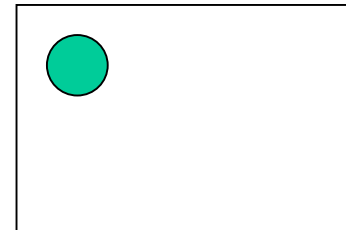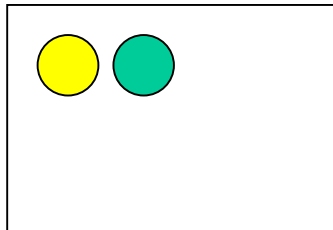Act randomly
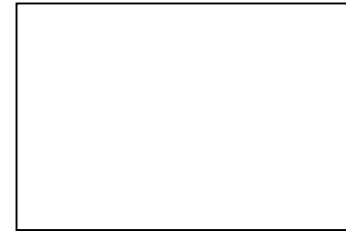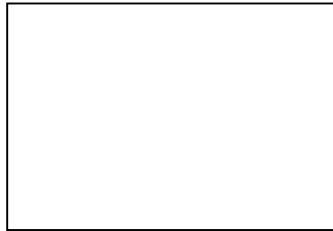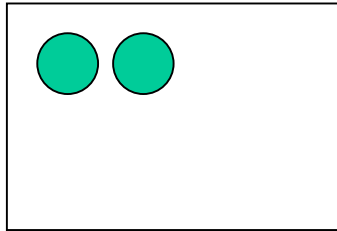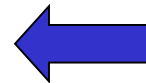
TAG player outline

Seek to be lonely
Seek company
Act randomly

TAG player outline

Seek to be lonely
Seek company
Act randomly

TAG player outline

The lonely strategy
- increases the chance that the it player goes to another Bailiff
- makes the player faster
- needs as many Bailiffs as there are players, but no more
- with fewer Bailiffs it approaches the company strategy

The company strategy
- increases the chance that the it player tags another player
- needs fewer Bailiffs
- makes the player slower

Bailiff service interface

- Avoid local calls from player to Bailiff
- Always use the Bailiff service interface
- For example, ask the "remote" Bailiff to tag another player
  - Requires player identification – UUID
- Decisions should be in the players
- Bailiffs provide information and message passing

Caveats - timings

- Agents typically use 1-2 seconds delay
  - Brings the game to human speed
- To avoid live-lock:
  - Let the it player be slightly faster
  - Or apply a random variation to each delay

Caveats - SDM

- Make sure to call
  `ServiceDiscoveryManager.terminate()`
  when migrating away from a Bailiff

- The SDM starts five threads

- If not terminated, the SDM remains orphaned and active

- Threads accumulate in the Bailiff JVM

Caveats – copy remote object

- If a Bailiff remote service method returns a player object it is a *copy*

- Tagging the copy will not stick (the tag will be lost)

- This holds *even when both agents happen to be in the same Bailiff*

End