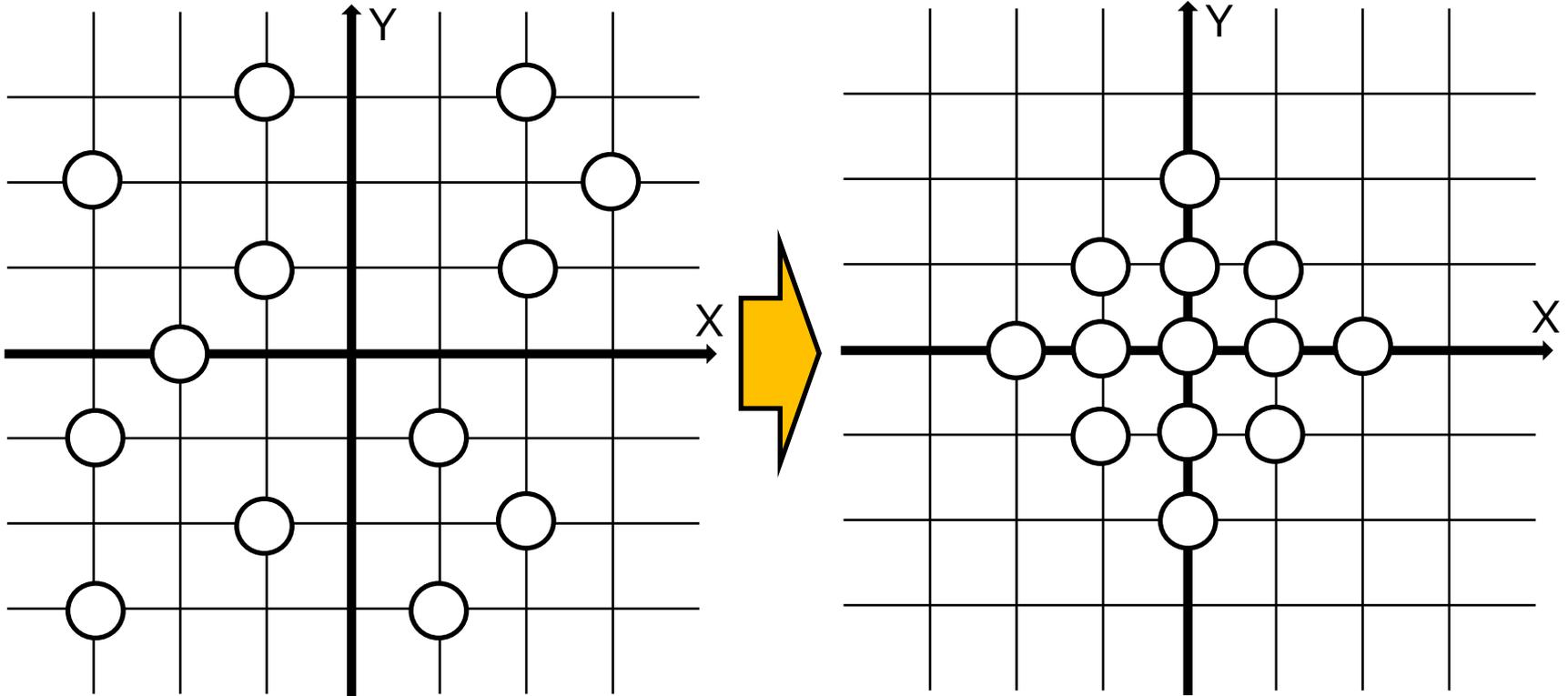




# Java課題発表

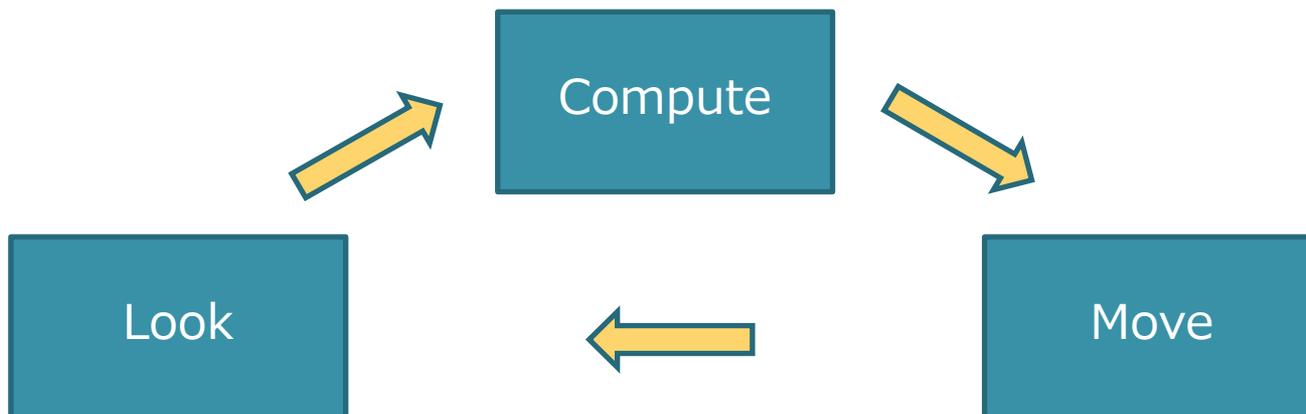
篠永智宏

- 直行格子平面上の原点を中心とした集合問題
  - 2020年度 高橋先輩修論

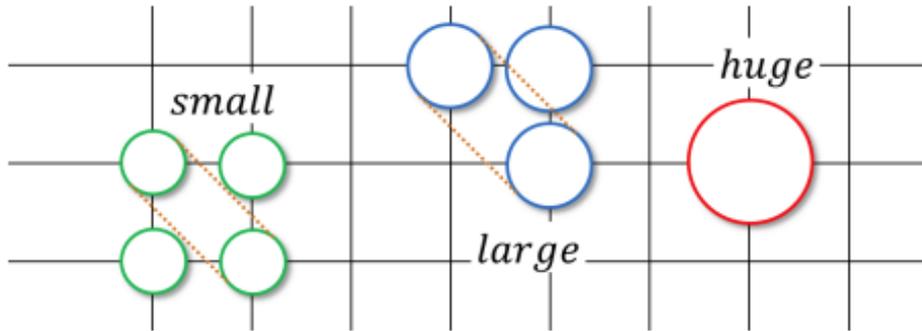


- ロボットモデル
  - 自律分散ロボット
  - Largeサイズのファットモデル
  - 半同期
  - 距離3の視野
  - 原点, 一軸合意
  - ロボットの総台数
- 初期条件
  - X軸上に最低1台のロボットが存在

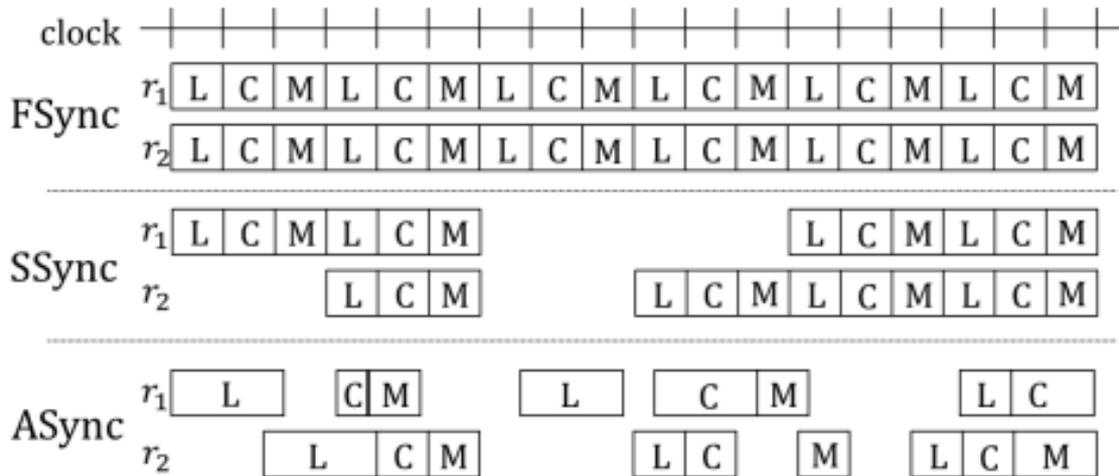
- 自律分散ロボット
  - 外見上区別不可
  - メモリ機能なし
  - LCMサイクルを繰り返す
    - Look:周囲の状況を観測
    - Compute:Lookの結果を用いて移動先を決定
    - Move:移動する



- Largeサイズのファットモデル(衝突禁止, 視界制限)
  - 斜め移動を禁止することで実現



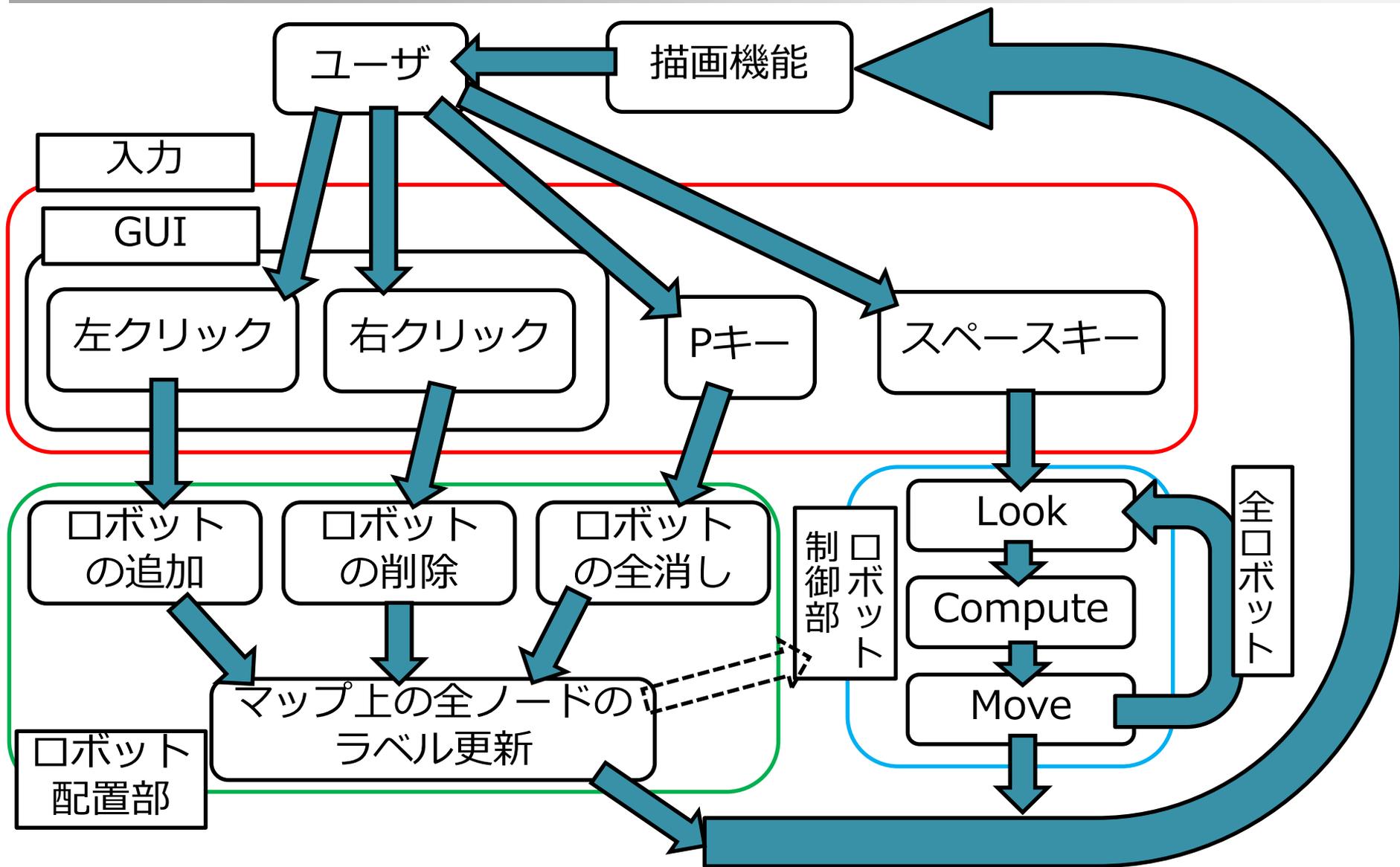
- 完全同期 (FSync)
  - 全てのロボットが同じタイミングでLCMサイクルを行う
- 半同期 (SSync)
  - LCMサイクルの各動作は同じタイミングだがサイクルを行わないロボットの存在を許す



- 完全同期モデルを想定
- 任意のタイミング、任意の場所のロボット追加, 削除
  - 左クリック: ロボット追加
  - 右クリック: ロボット削除
- 任意のタイミングによる全ロボットの削除
  - P key: 全ロボット削除
- 全ロボットの1サイクル動作
  - Space key: 全ロボット1サイクル動作
  - Space key押しっぱなしで複数サイクル動作

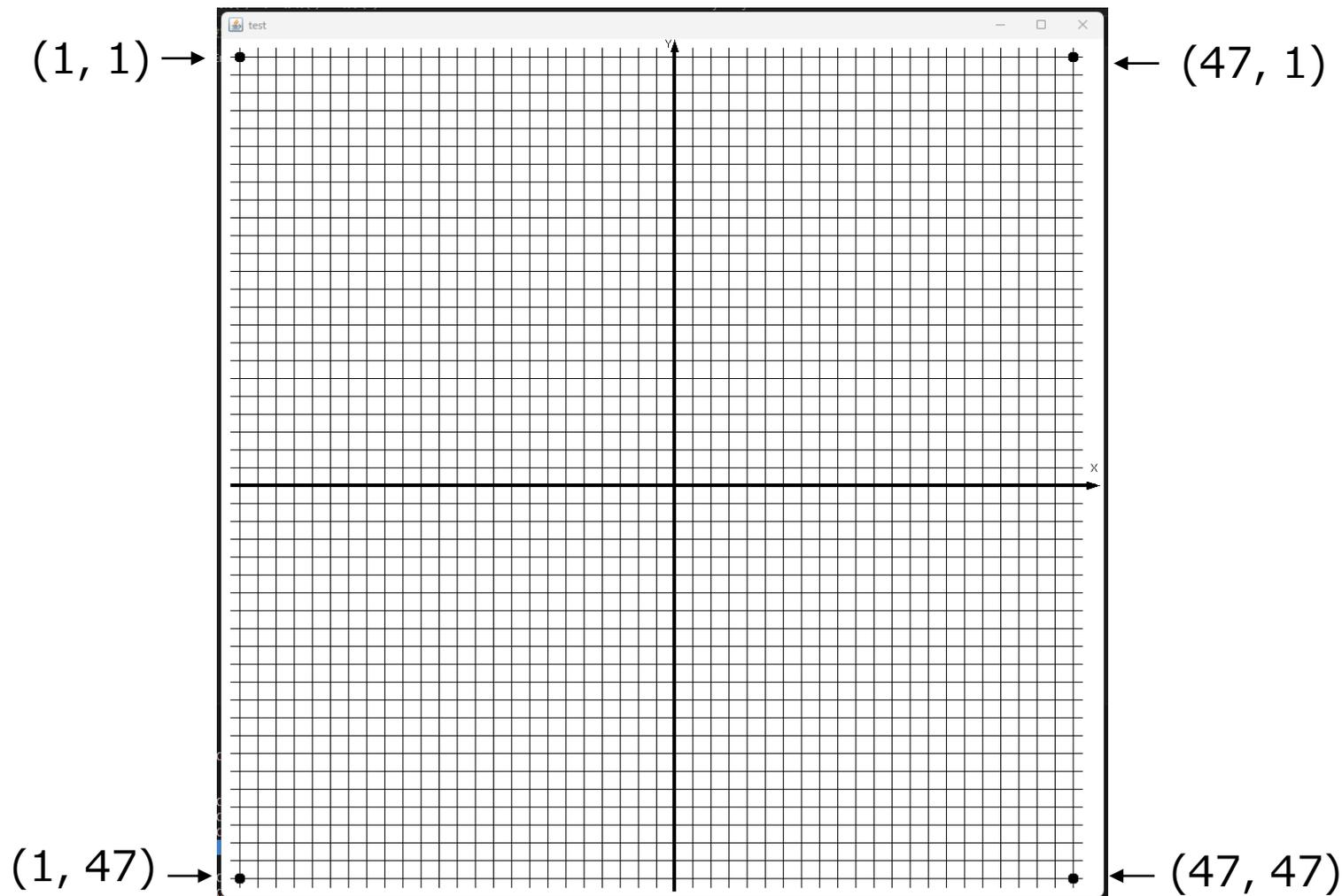
# K<sup>2</sup> LAB システム構成図

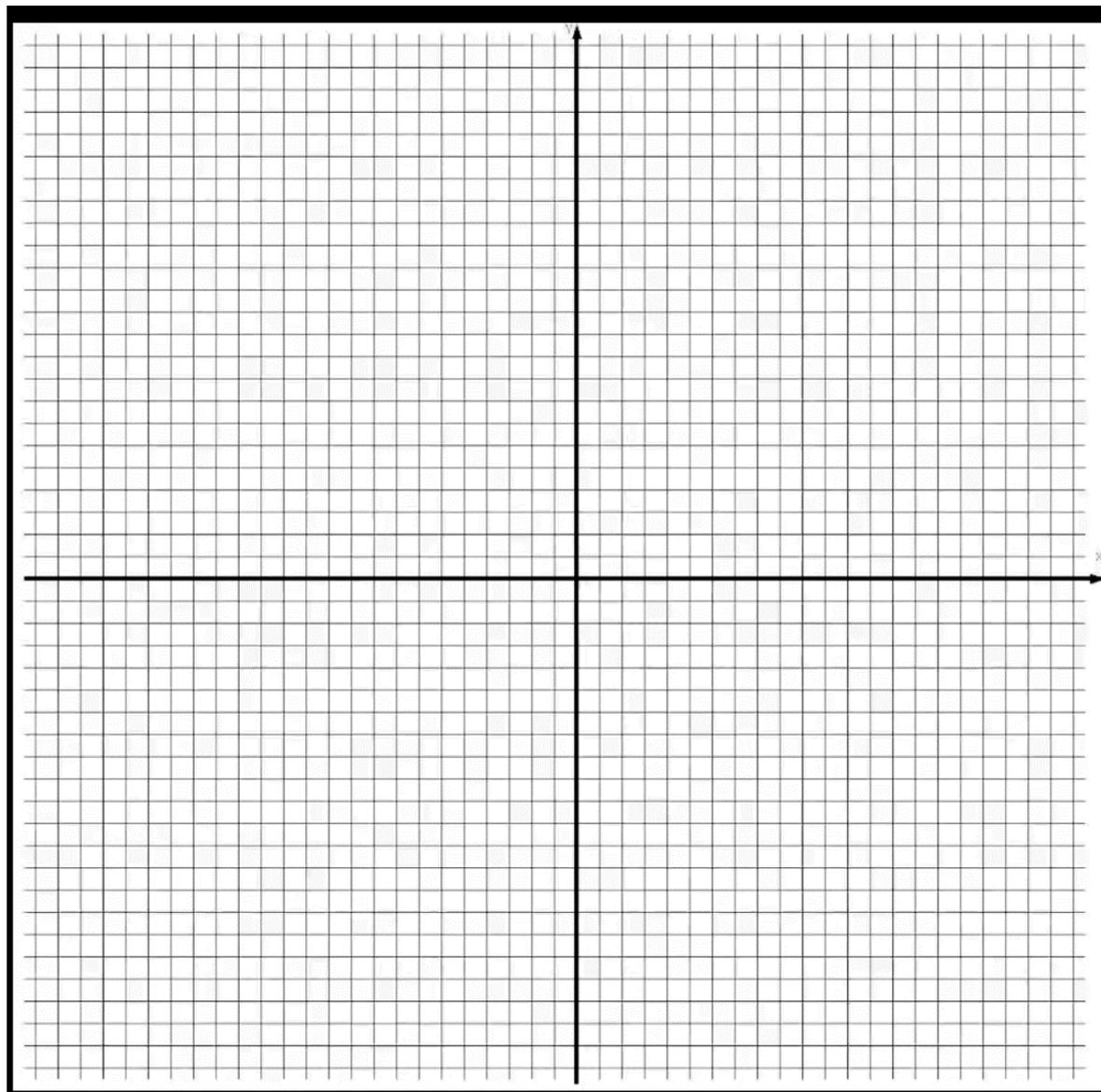
9



# K<sup>2</sup> LAB 実装内容

- X座標, Y座標の範囲は(0~48, 0~48), 原点は(25, 25)





## □ 難しかった点

- 原点が(25, 25)であるため、座標が把握しづらい点

## □ 工夫した点

- 実際の座標は(0~48, 0~48)のところ、  
入力できる範囲を(1~47, 1~47)に限定した点
  - 端に配置されてもlook時にエラーが発生しないようにした
- アルゴリズムを動作させている途中でもロボットの追加・  
削除が可能な点

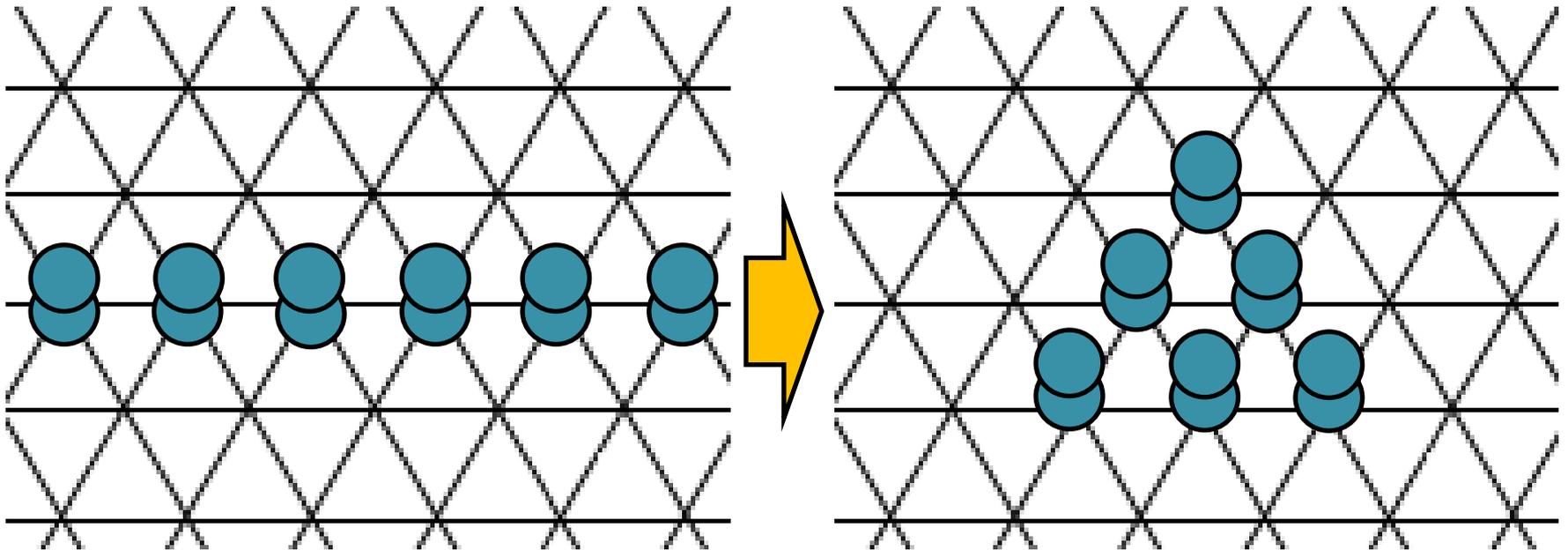
## □ まとめ

- look, compute, moveの流れが理解できた

## □ 改善点

- 半同期の実装
- サイクル実行前に戻す機能の実装

- 三角格子平面上の直線から三角形への形状形成問題
  - 2020年度 田口先輩卒論



- ロボットモデル
  - ペアロボット
  - 二軸合意
  - 距離1の視野
  - 半同期
- 初期条件
  - X軸に沿って直線を形成するように配置

## □ ペアボットの説明

### □ Short状態とLong状態を交互に繰り返す

- Short状態のままになる移動やLong状態のままになる移動は禁止

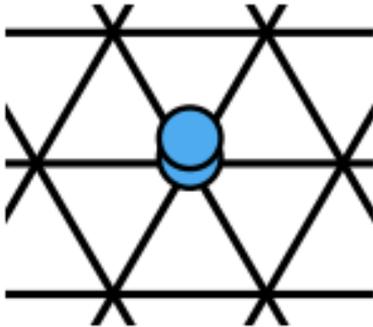


図 2.2: Short 状態

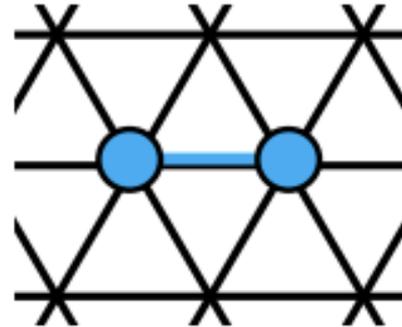


図 2.3: Long 状態

### □ Short状態から動作するときにはどちらか1台のみが動作する

- 2台が動作してしまうとLong状態にならないため

## □ ペアボットモデルの説明

### □ 弱重複検知を持っている

- 視野範囲内のノード上にロボットが何台存在しているか判別可能

### □ 禁止事項

- 状態が遷移しない移動 (Short→Short, Long→Long)
- 下の図のような状態

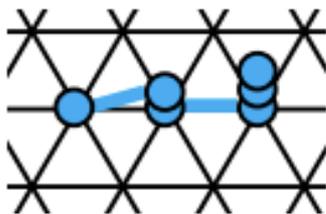


図 2.6: 同一点上に 3 台以上のロボットが存在する状態の例

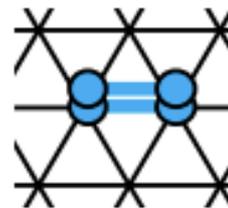
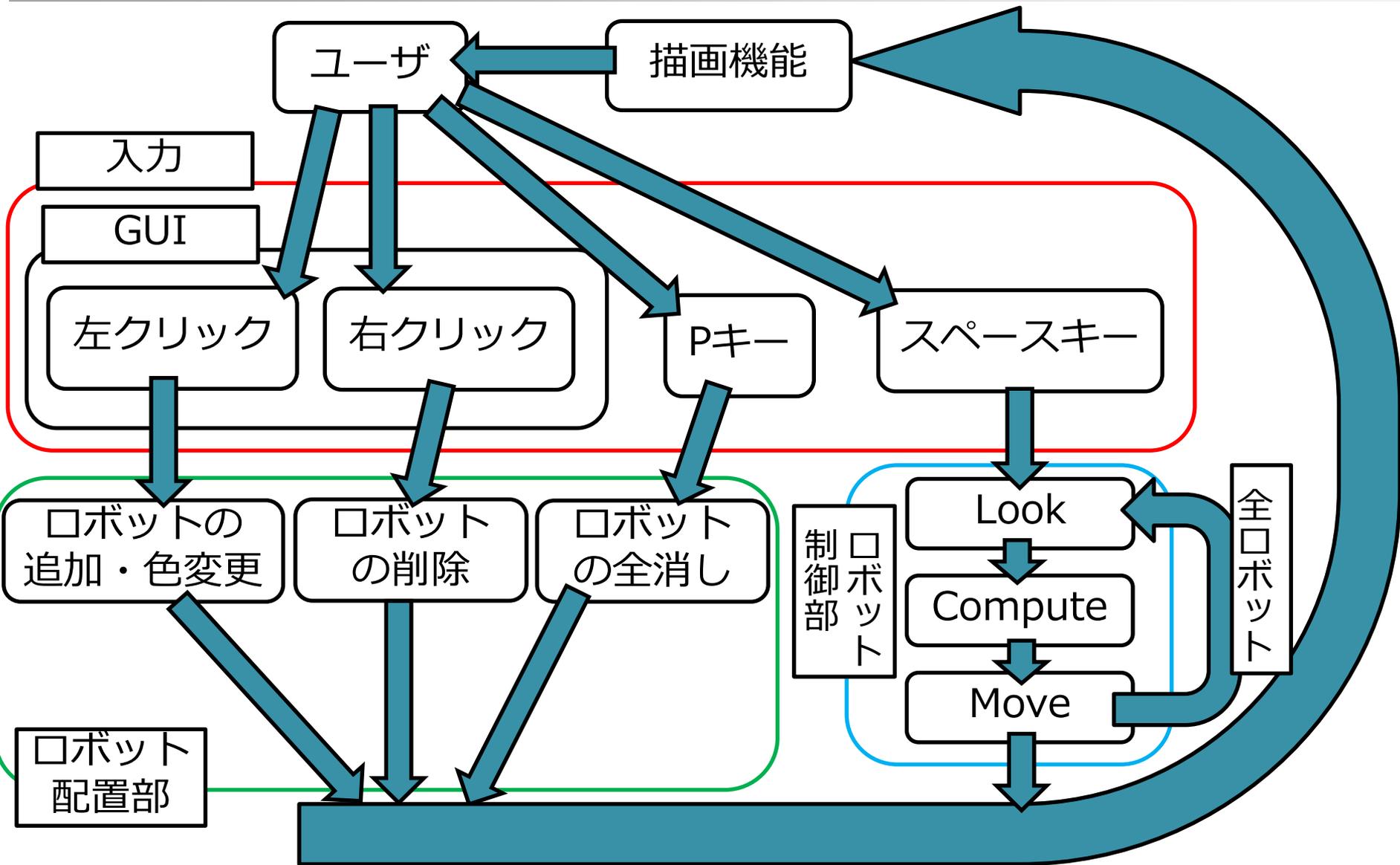


図 2.7: 2つの Short 状態のペアが完全に重なっている状態

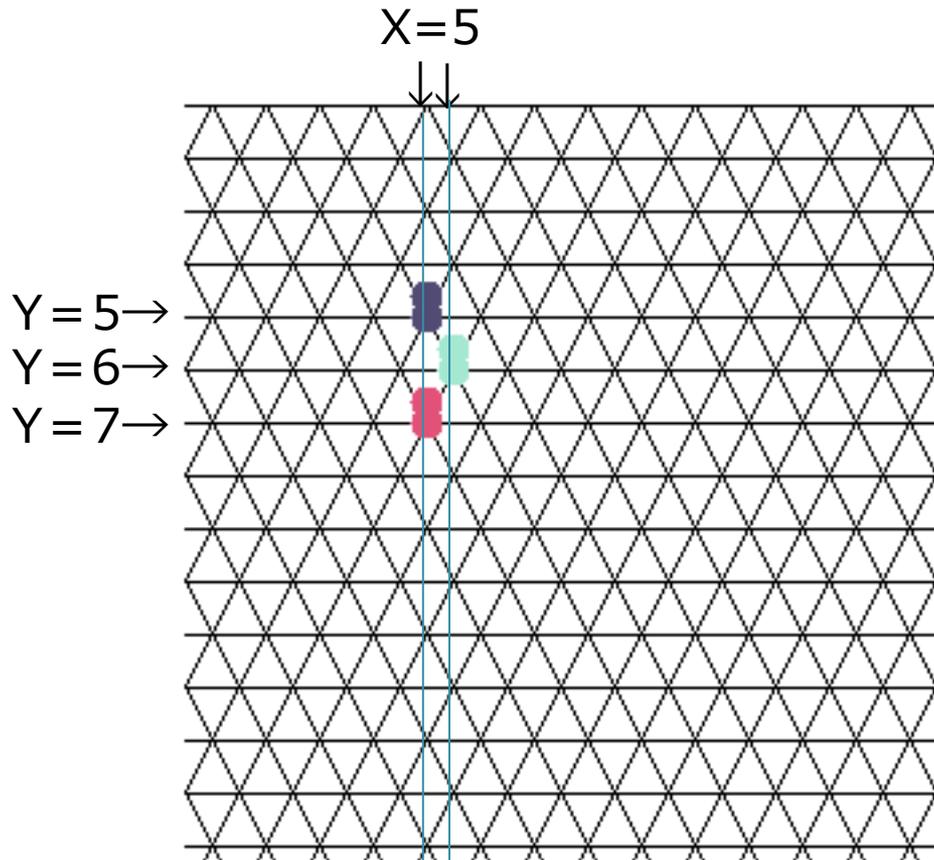
- 完全同期モデルを想定
- 任意のタイミング、任意の場所のロボット追加, 削除
  - 左クリック: Short状態のロボット追加, 色変更
  - 右クリック: Short状態のロボット削除
- 任意のタイミングによる全ロボットの削除
  - P key: 全ロボット削除
- 全ロボットの1サイクル動作
  - Space key: 全ロボット1サイクル動作
  - Space key押しっぱなしで複数サイクル動作

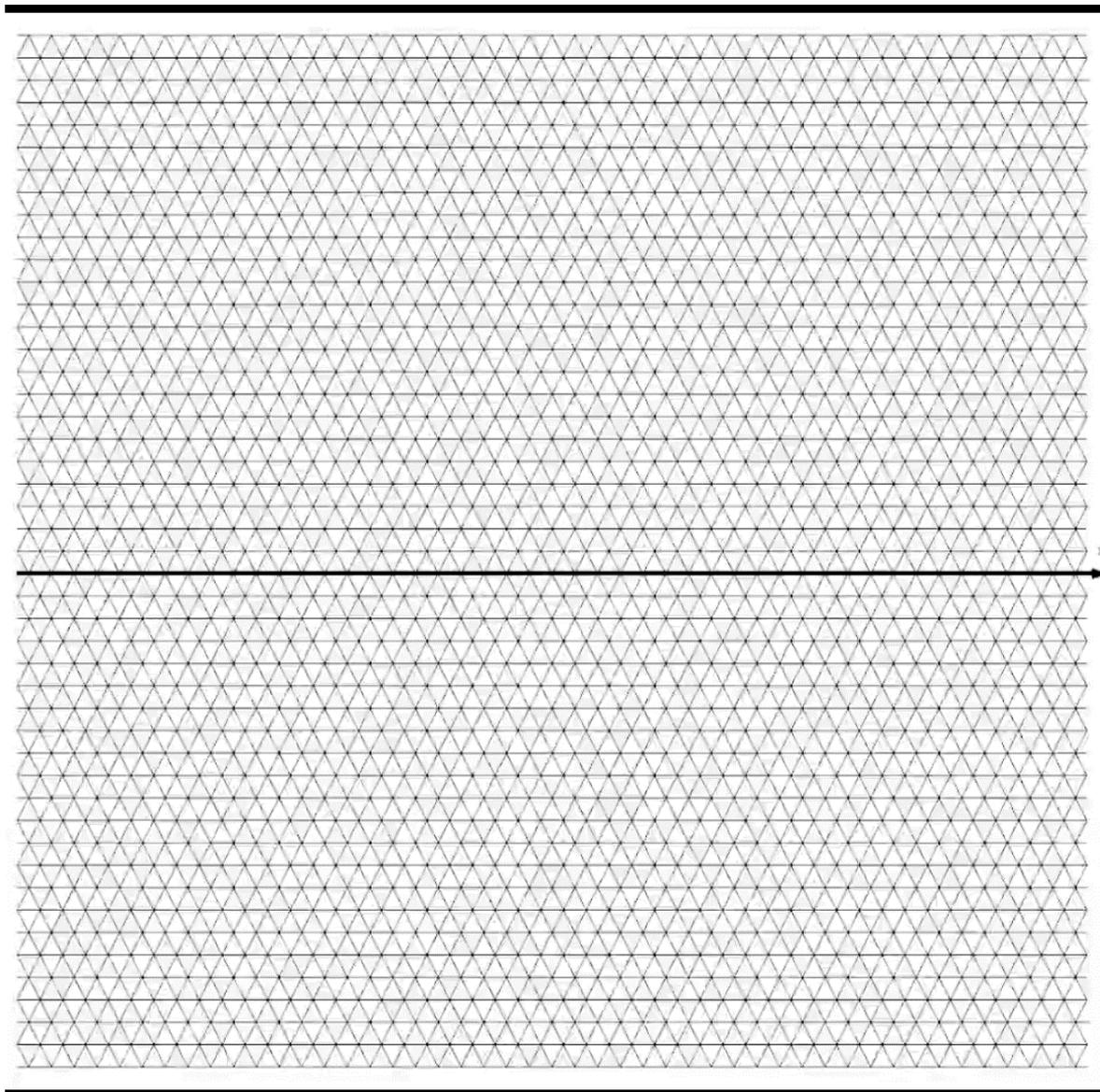
# K<sup>2</sup> LAB システム構成図

35



- 上から(X座標, Y座標) = (5, 5), (5, 6), (5, 7)
  - Y座標によって上下を見たいときに見る座標が異なる
    - (5, 6)の左上は(5, 5) (5,7)の左上は(4, 6)





- 難しかった点

- 上下のlook時やmove時のX座標の扱い
- 自身と同じペアのロボットの識別の処理
- 三角格子の描画

- 工夫した点

- ペアロボットのペアを色で視覚的に判別できるようにした点

- まとめ
  - ペアボットについての知識が深まった
  - プログラミングスキルが向上した
  
- 今後の課題
  - 半同期の実装
  - Long状態のペアボットの配置