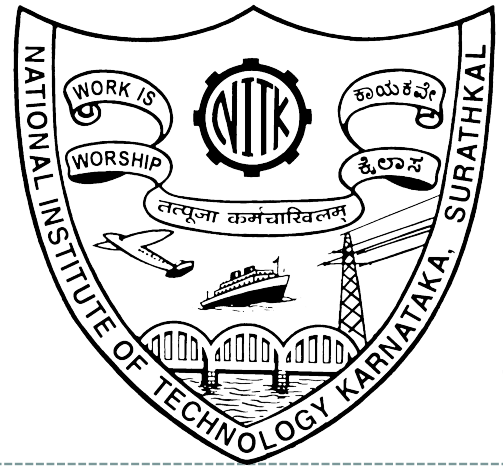



## National Institute of Technology Karnataka

## Automata and Compiler Design Project Presentation

# AUTOMATA FOR CHAIN REACTION GAME

**Under the guidance of**

**Ms Kajal Bangera**
**Assistant Lecturer**
**Department of Information Technology**

**Presented By:**

**Divija Nagaraju**
**14IT112**
**Mukta Kulkarni**
**14IT220**
**Pooja Soundalgekar**
**14IT230**

# Contents

- Introduction
- Problem Statement
- Objective
- Methodology
- Identification of states
- Influence of input on current grid
- Influence of input on neighbouring grids
- Non Deterministic Finite Automata
- Technology Stack used
- Future Work
- Individual Contribution
- Conclusion
- References

# Introduction

- What is Chain reaction ?
- Chain Reaction Game : Deterministic Combinatorial Game of Perfect Information
- Popular Computer Game developed by Buddy-Matt entertainment
- Unpredictable or Highly Volatile

# Problem Statement

- Analyze the Deterministic Details of the game
- Analyze Inherent States for the game
- Analyze State Transitions
- Develop a complete Finite State Machine for the Game

# Objective

- Explore the inherent property of the game.
- A new approach to define the heuristic of the game for 2 players
- Explore possibility for Developing AI

# Methodology

- Identification of states
- Influence of input on current grid
- Influence of input on neighbouring grids

# Identification of states

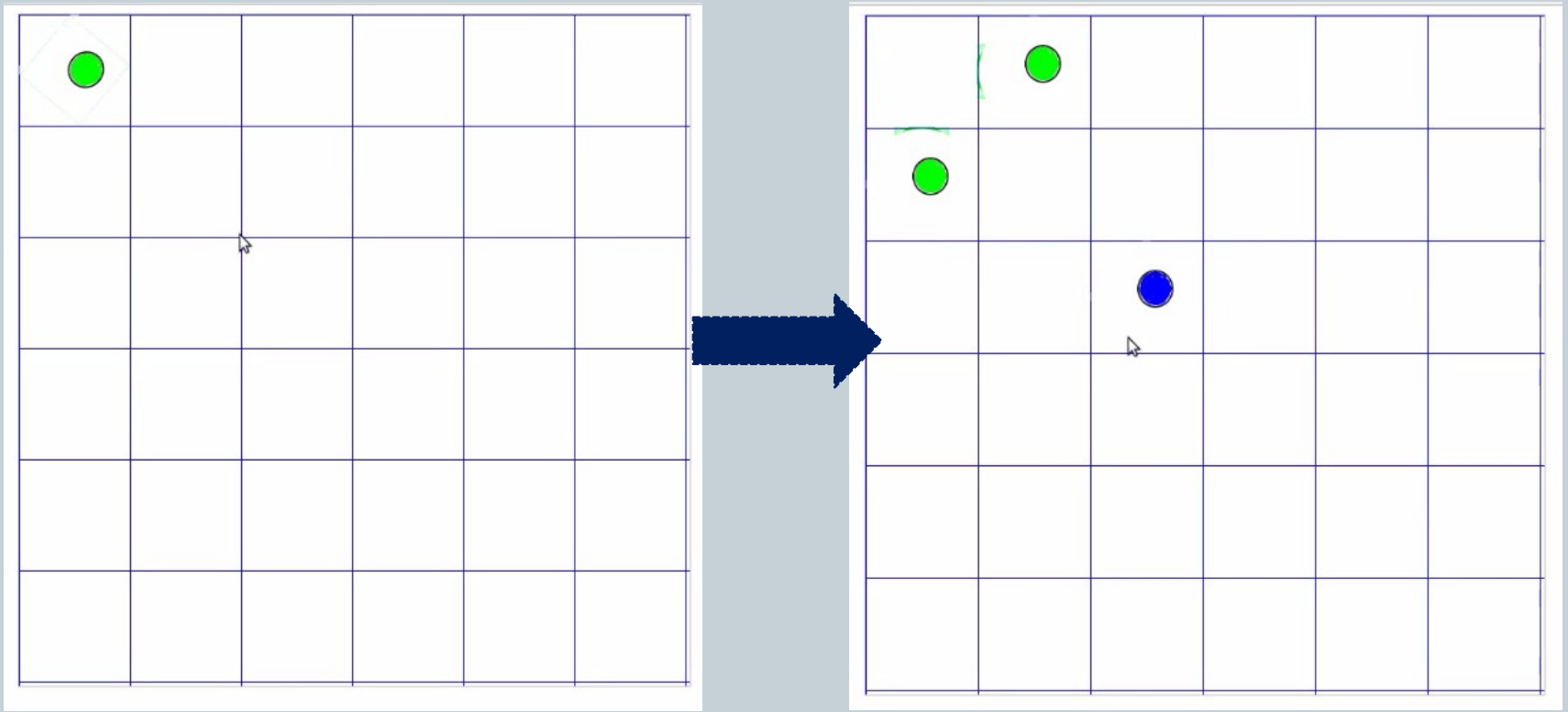
1. Start state : No Ball on the board
2. S10: Player A with 0 ball on grid with critical mass =2
3. S11: Player A with 1 ball on grid with critical mass =2
4. S20: Player A with 0 ball on grid with critical mass =3
5. S21: Player A with 1 ball on grid with critical mass =3
6. S22: Player A with 2 balls on grid with critical mass =3
7. S30: Player A with 0 ball on grid with critical mass =4
8. S31: Player A with 1 ball on grid with critical mass =4
9. S32: Player A with 2 balls on grid with critical mass =4
10. S33: Player A with 3 balls on grid with critical mass =4

# Identification of states (cont'd)

- 11. S'10: Player  B with 0 ball on grid with critical mass =2
- 12. S'11: Player B with 1 ball on grid with critical mass =2
- 13. S'20: Player B with 0 ball on grid with critical mass =3
- 14. S'21: Player B with 1 ball on grid with critical mass =3
- 15. S'22: Player B with 2 balls on grid with critical mass =3
- 16. S'30: Player B with 0 ball on grid with critical mass =4
- 17. S'31: Player B with 1 ball on grid with critical mass =4
- 18. S'32: Player B with 2 balls on grid with critical mass =4
- 19. S'33: Player B with 3 balls on grid with critical mass =4

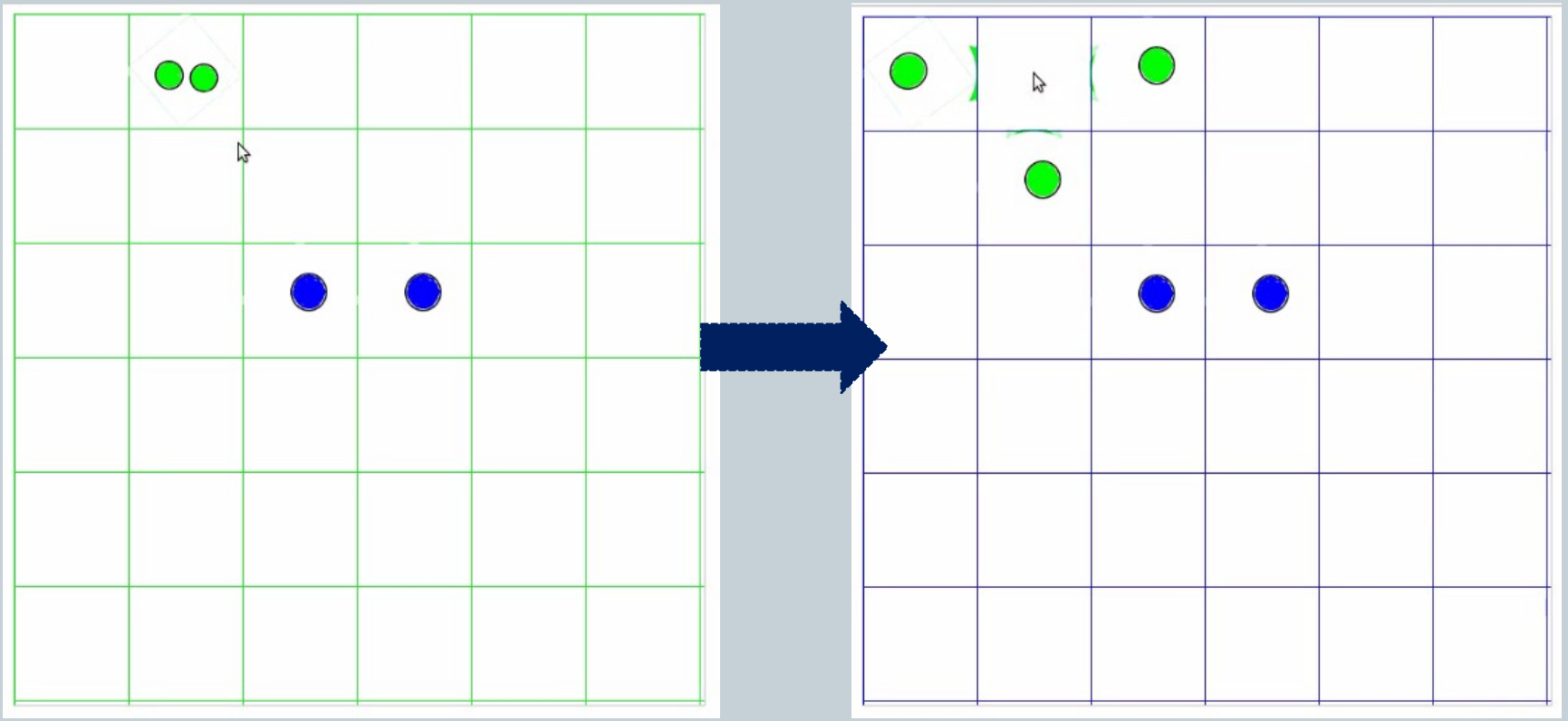
# Influence of input on current grid

- CASE 1:



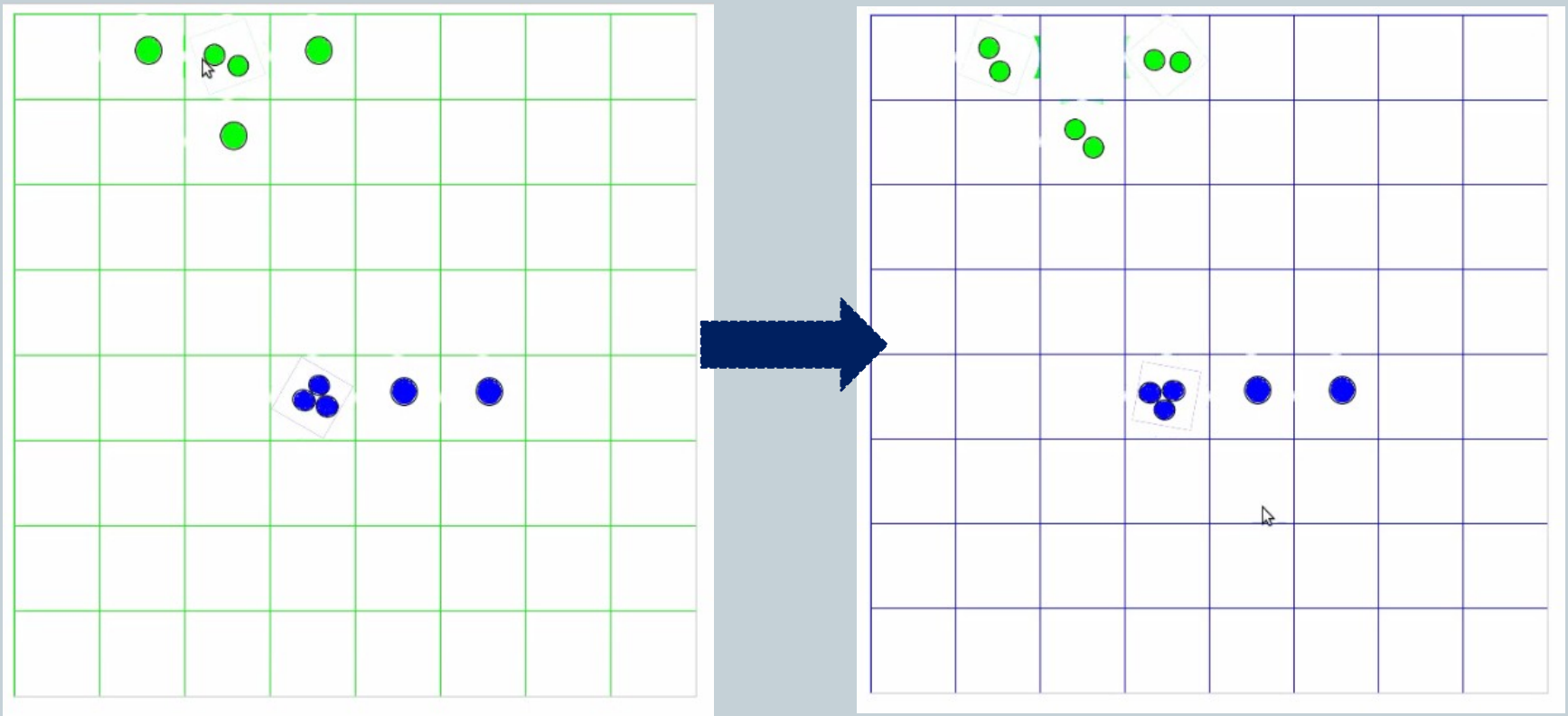
# Influence of input on current grid (cont'd)

- CASE 2:



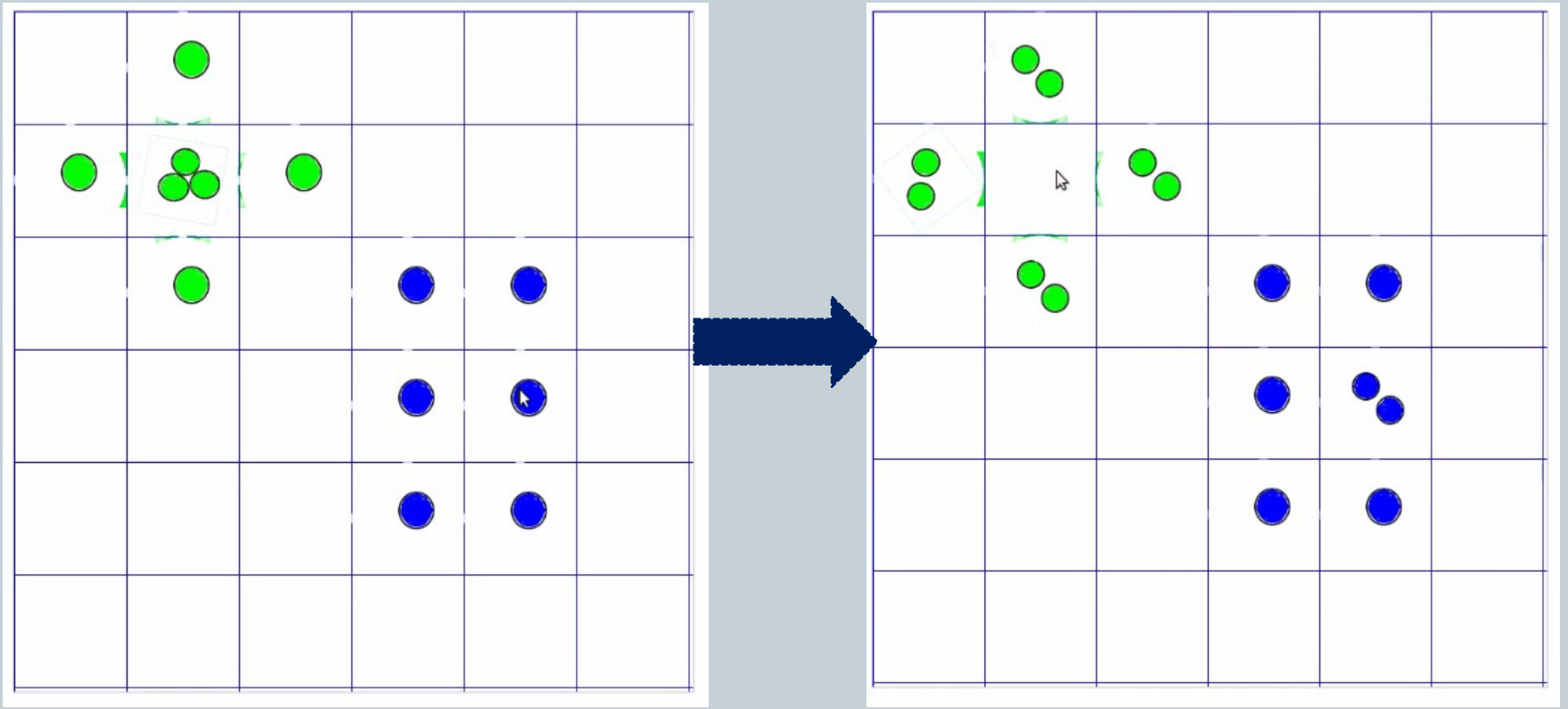
# Influence of input on current grid (cont'd)

- CASE 3:



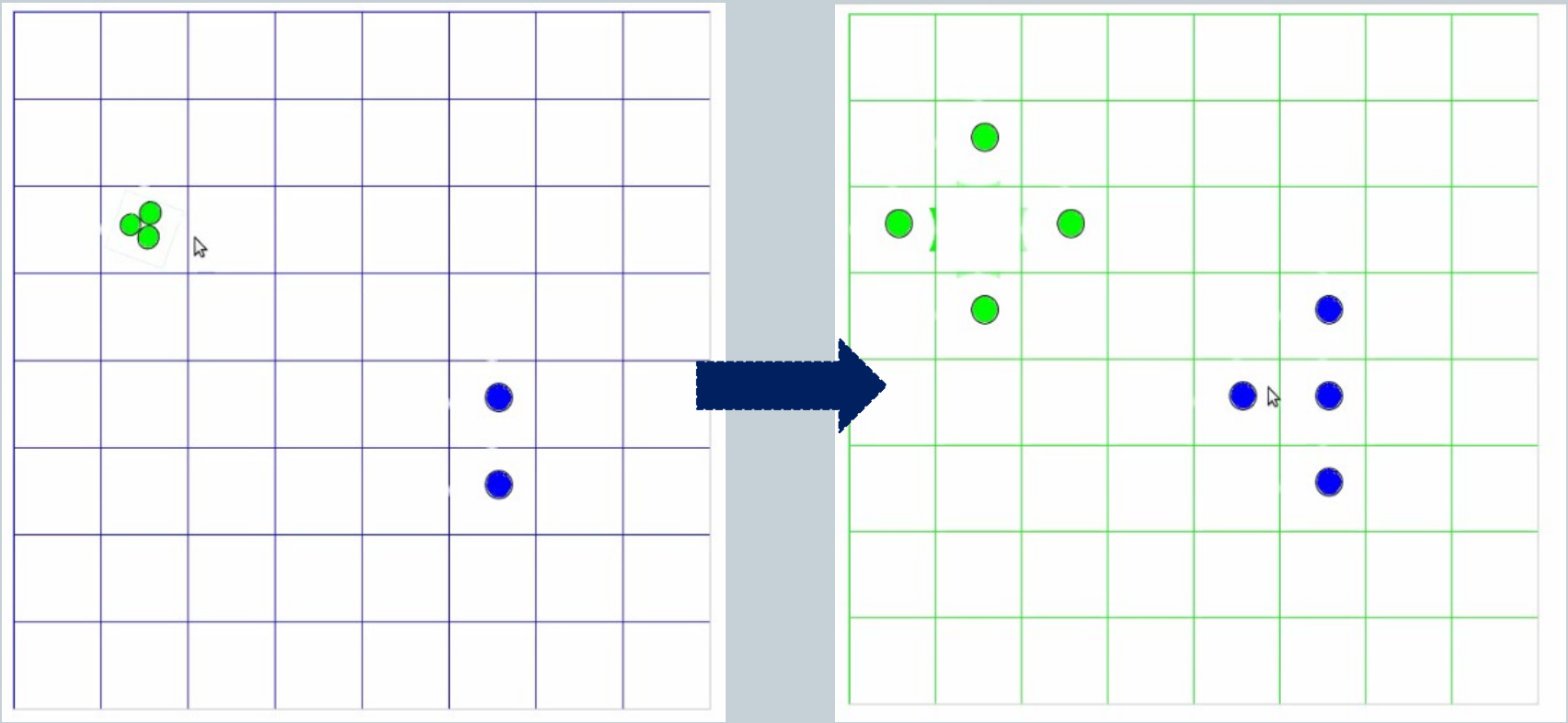
# Influence of input on current grid (cont'd)

- CASE 4:



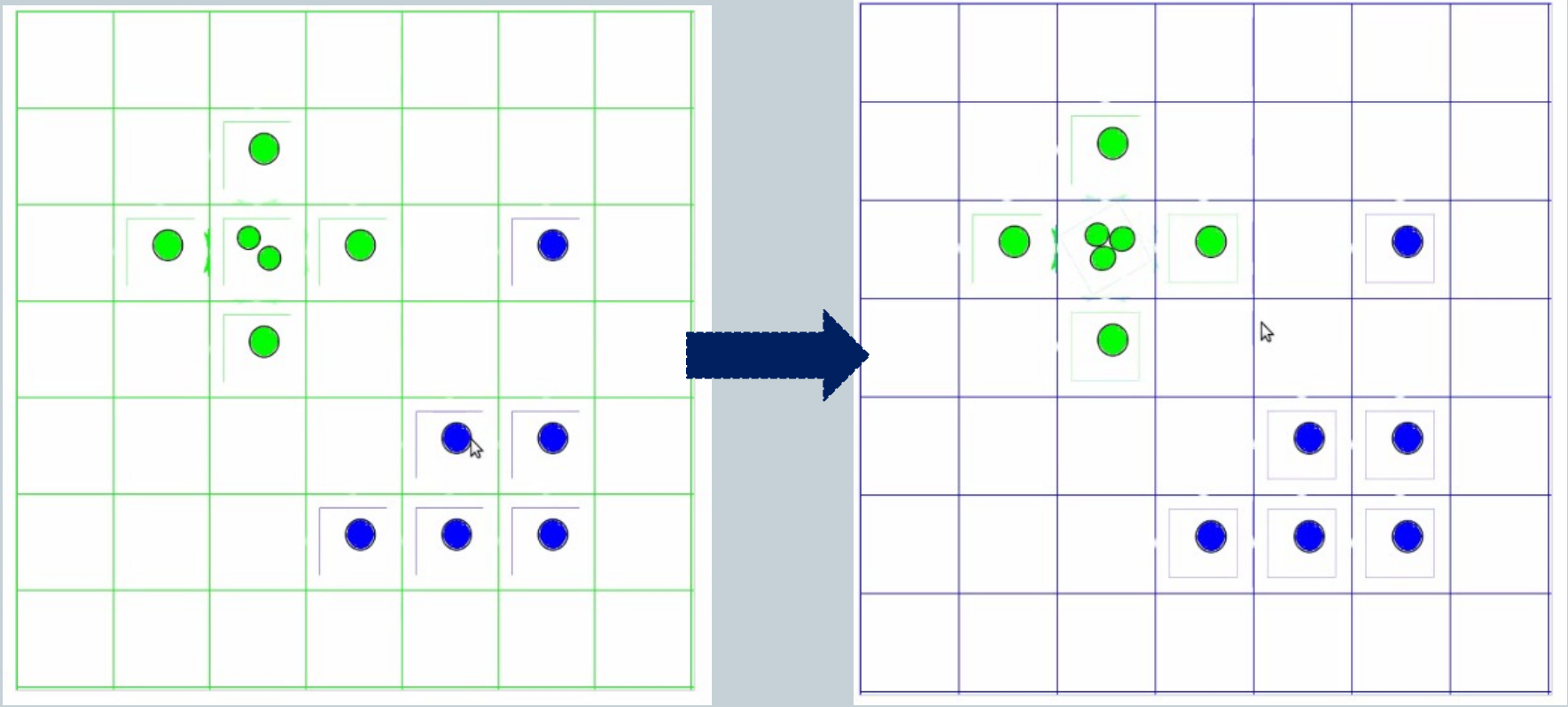
# Influence of input on current grid (cont'd)

- CASE 5:



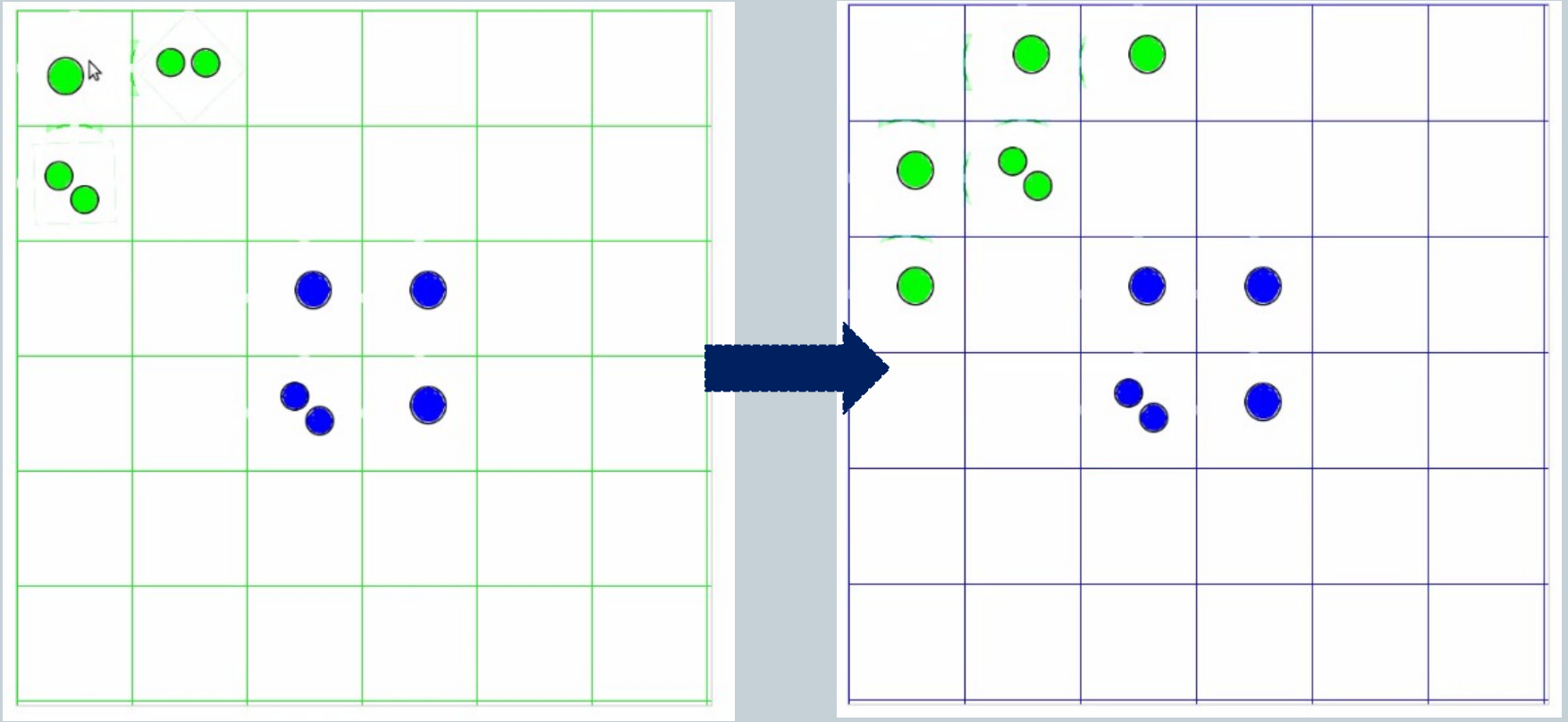
# Influence of input on current grid (cont'd)

- CASE 6:



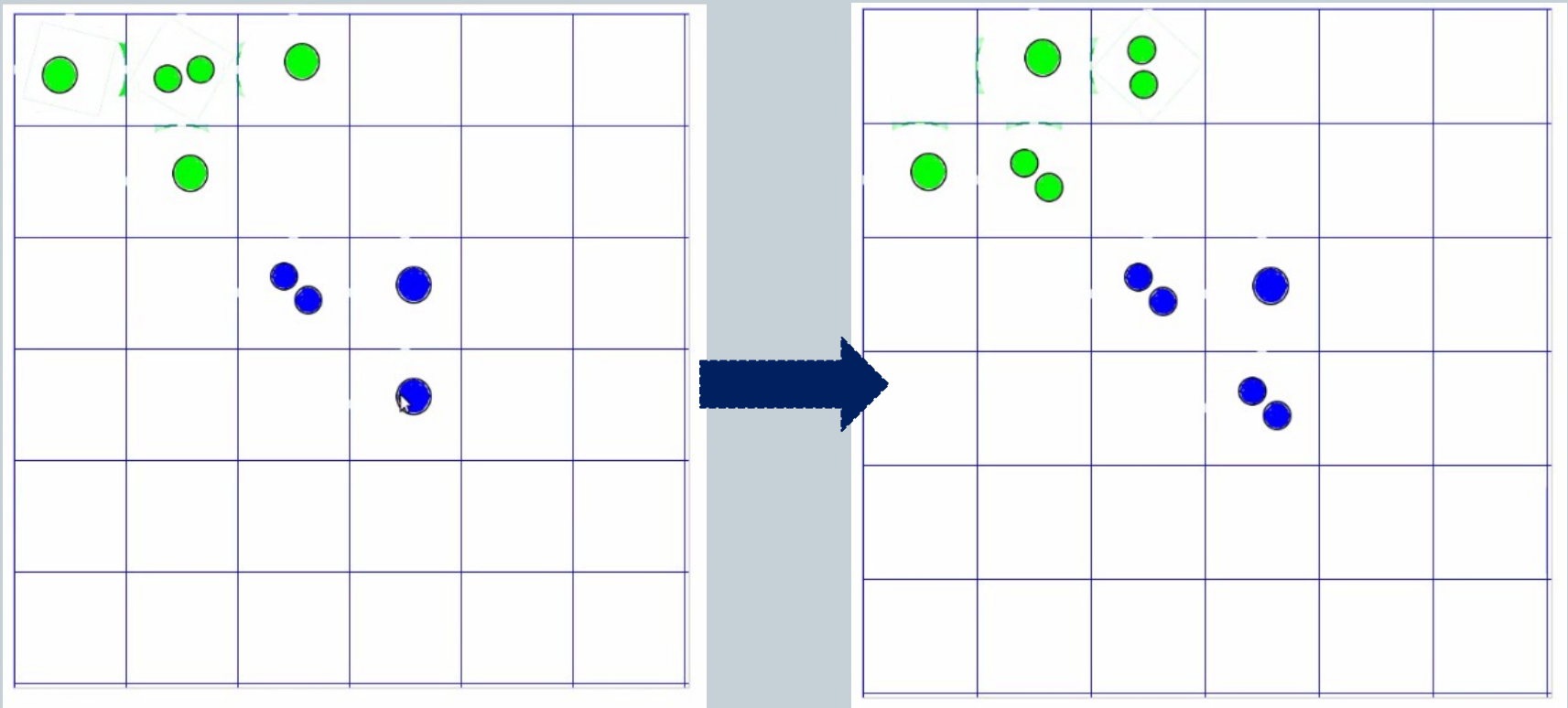
# Influence of input on neighbouring grids

- CASE 1: Effect of One on Two



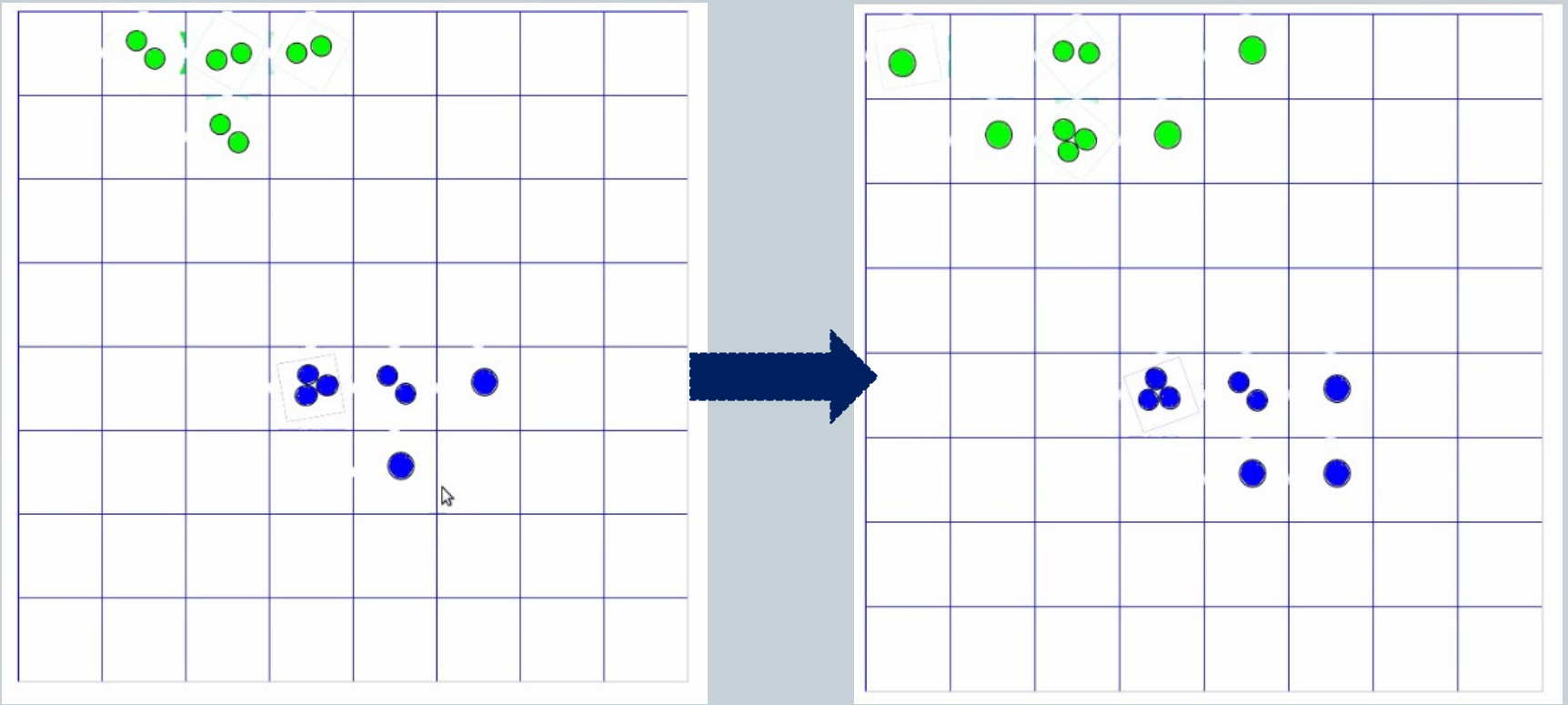
# Influence of input on neighbouring grids (cont'd)

- CASE 2: Effect of Two on One, Two and Three



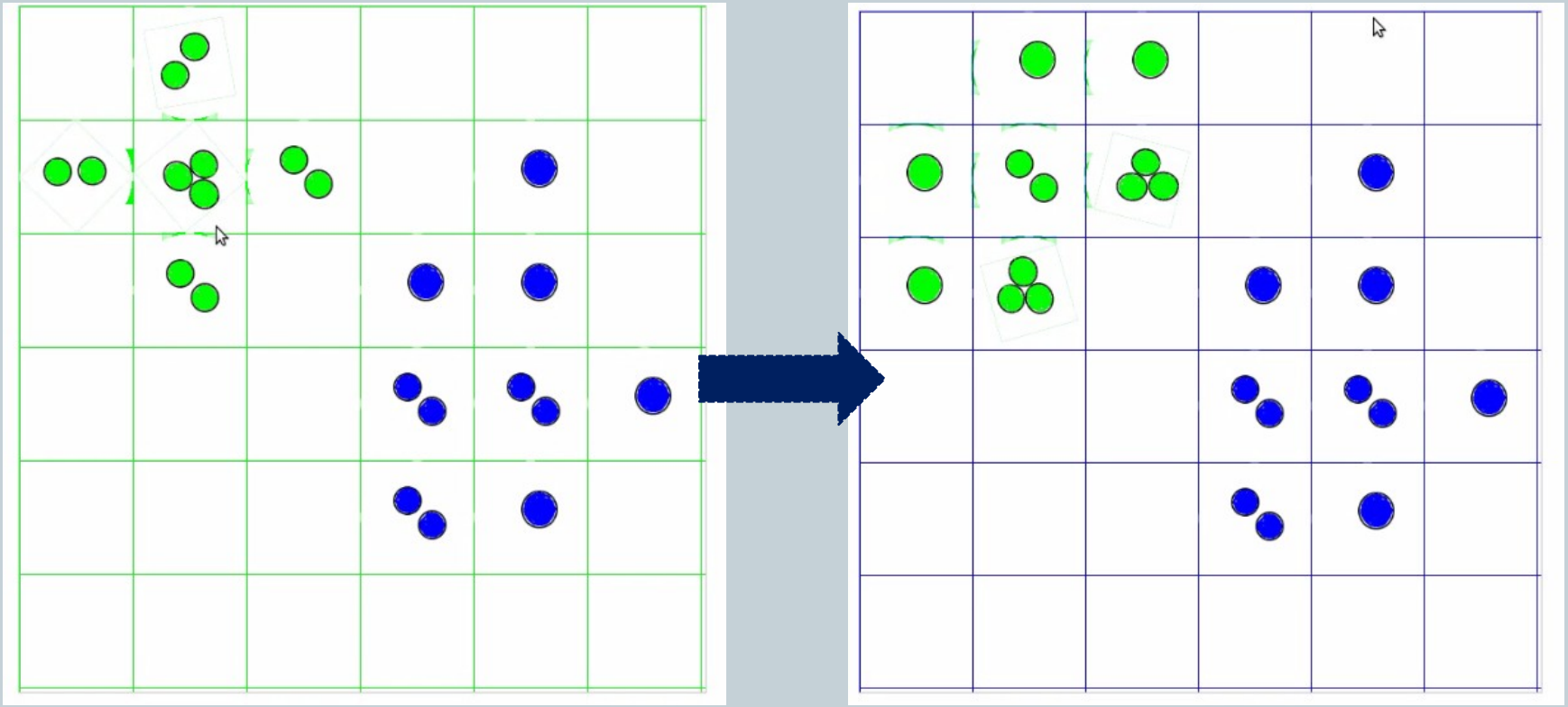
# Influence of input on neighbouring grids (cont'd)

- CASE 3: Effect of Two on Two, Two and Three



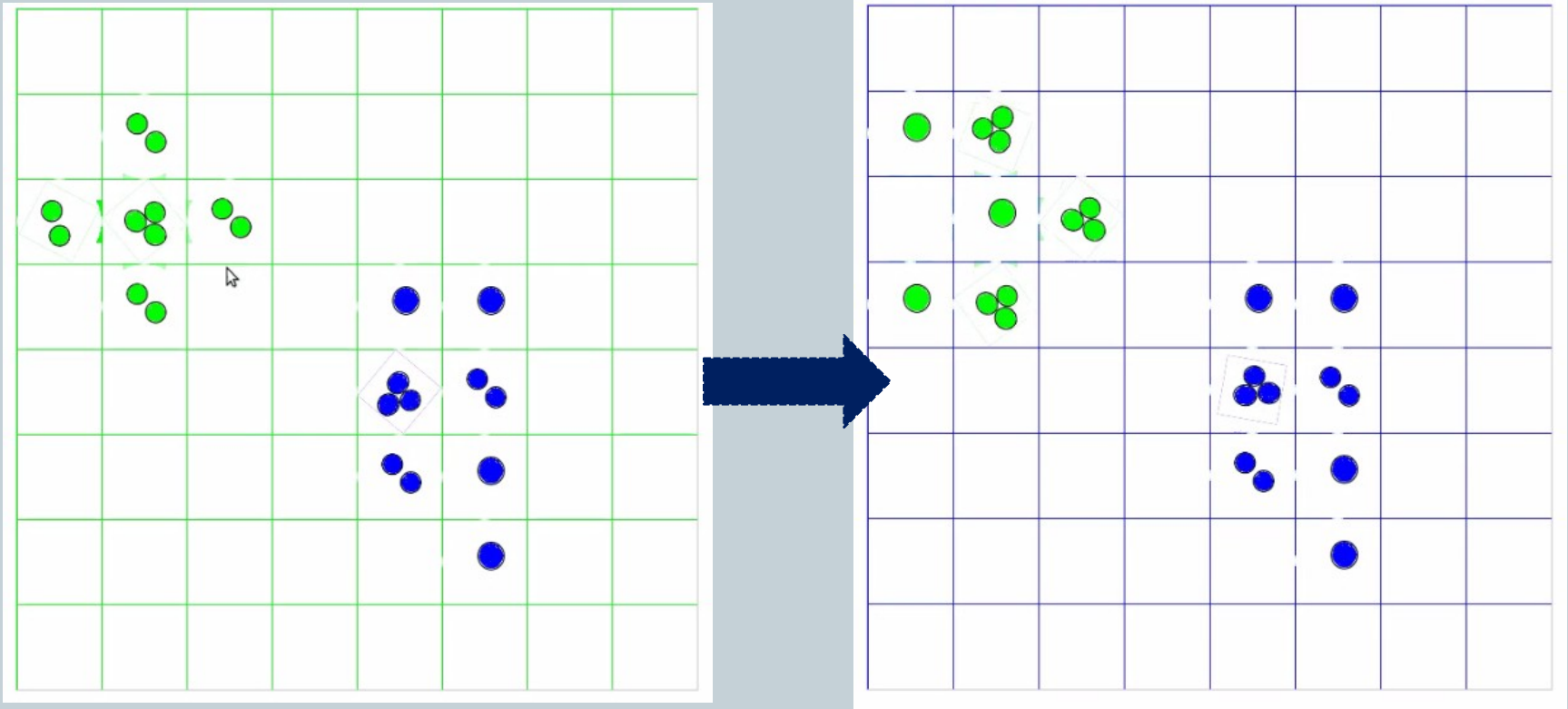
# Influence of input on neighbouring grids (cont'd)

- CASE 4: Effect of Three on Two, Two, Three and Three



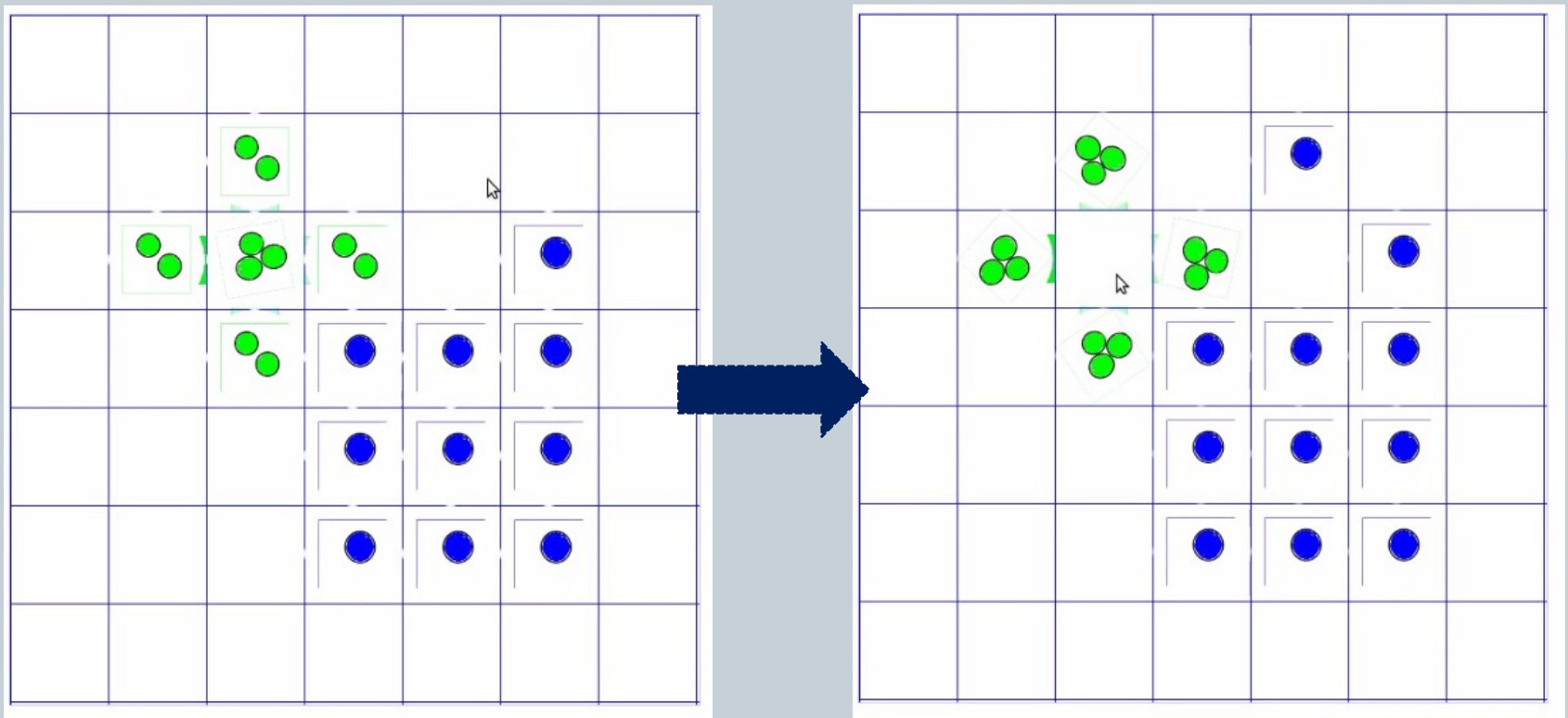
# Influence of input on neighbouring grids (cont'd)

- CASE 5: Effect of Three on Two, Three, Three and Three

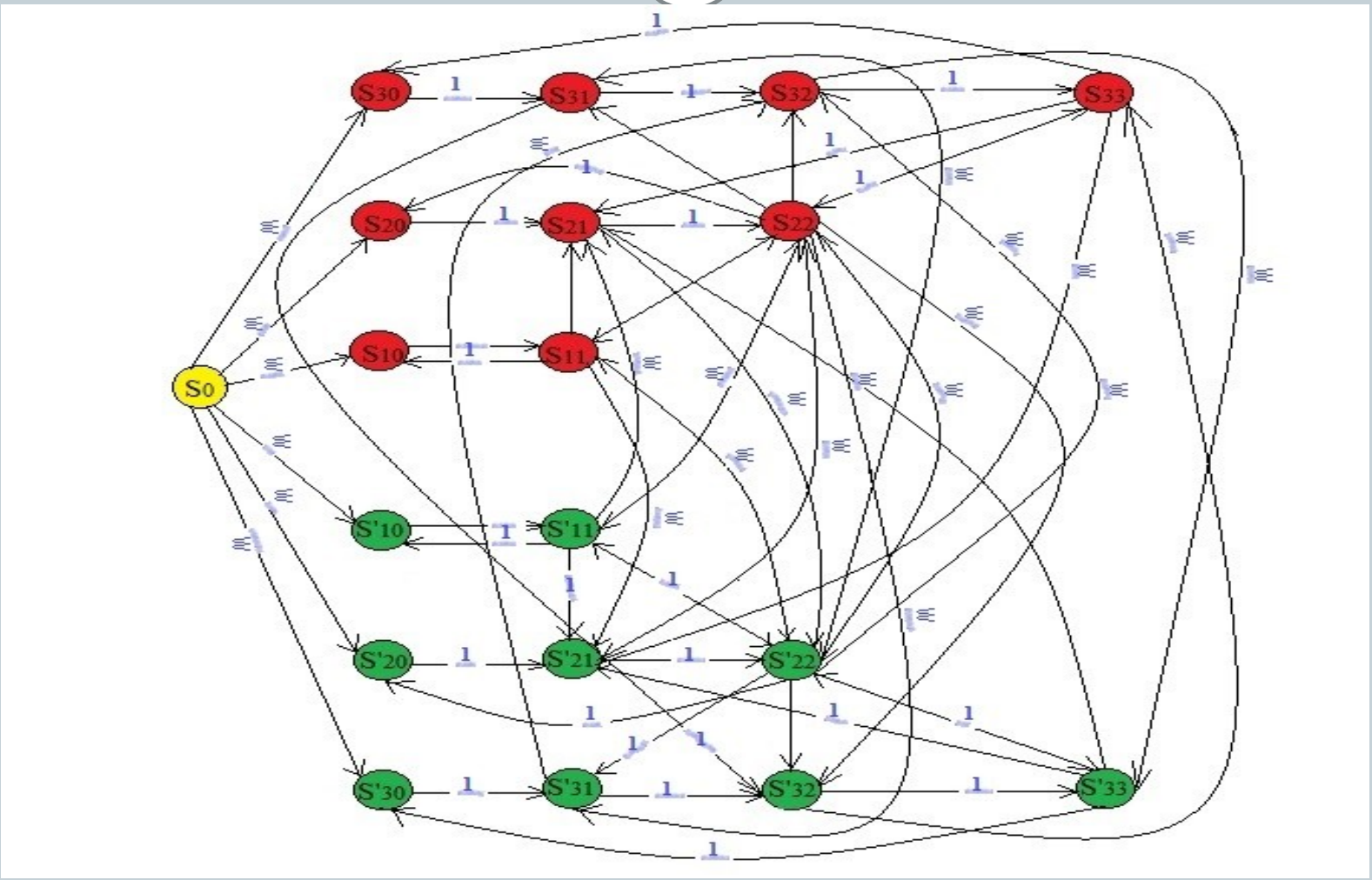


# Influence of input on neighbouring grids (cont'd)

- CASE 6: Effect of Three on Three, Three, Three and Three



# Non Deterministic Finite Automata



# Technology Stack used

- Front End:
  Graphical User Interface: HTML
  Styling: CSS
- Back End: JavaScript

# Future Work

- Current techniques – Minmax Algorithm and Tree based Heuristic and Monte Carlo Method
- Develop AI on the lines of the Finite State Machine
- Compare the methods to the existing ones for computation speed and memory usage

# Individual Contribution

- Divija: Graphical User Interface, Styling, reset and update functionality of game.
- Mukta: JavaScript skeleton, Animation functionality of the game.
- Pooja: Linking of JavaScript code with HTML, state creation functionality of the game.

# Conclusion

- The progression of the game itself occurs in a series of states.
- This inherent linking with the automaton concept makes it suitable for studying and modelling the state transitions involved.

# References


- [1] Noman Sohaib Qureshi , Hassan Mushtaq 2,Computing Game Design with Automata Theory ,May 2012.
- [2] T. L. B. v. S. Turocy, "Game Theory," Encyclopedia of Information Systems, Academic Press, 2002.
- [3] J. C. H. R. S. J. W. W. E. v. D. J. F. N. P. H. Harold W. Kuhn, "The Work of John Nash in Game Theory," in journal of economic theory, 1994.
- [4] Dafyd Jenkins and Colin Frank, “Highly Volatile Game Tree Search in Chain Reaction” http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4100131&tag=1 ,May 2006.
- [5] Agnishom Chattopadhyay, Pranshu Gaba, and SatyabrataDash, “Chain Reaction Game” https://brilliant.org/wiki/chain-reaction-game/ ,April 2014.

# THANK YOU!