Epari Aneesh	(IWM2014502)
Tara Prasad Tripathy	(IHM2014003)
Saurabh Singh	(IWM2014004)

OOM Project:
Game Of Life (Project No. 8)

Overview

- Motivation
- Project Goals
- Project Tasks
- About the Project
- Planning Process
- Patterns in LIFE
- Implementation

MOTIVATION

MOTIVATION

- Game Of Life offered a great chance to explore various Java Technologies and play around with different graphical tools.
- From a theoretical point of view, Game Of Life is fascinating because it has the power of a universal Turing machine: that is, anything that can be computed algorithmically can be computed within Conway's Game of Life.

Project Tasks

- Designing and Modelling Using OOM Principles.
- Prepare Class Diagrams, Use-Case Analysis, CRC Diagrams of the Project.
- Implementing an Graphical User Interface and Simulate the Game using JAVA’s Applet, Swing, AWT and 2D Graphics Libraries.

Project Goals

The player interacting with the game should be able to:

- Set an initial configuration of the grid before the game starts.
- Pause or restart the game, add, redraw or clear configurations to/from the grid.
- Exercise control over the rate of change of patterns and switch among the available rates.

About LIFE

Game of Life has analogies with the rise, fall and alterations of a society of living organisms. Thus, it belongs to the group of 'simulation games' (games that resemble real life processes).

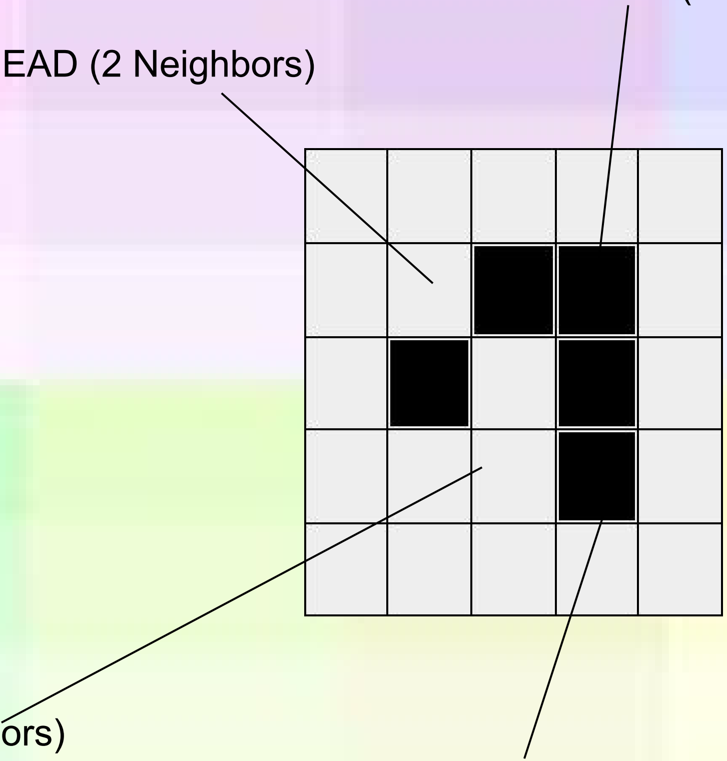
Conway's Game of Life

- The "**GAME**" is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input. Nonetheless, one can interact with the game while it runs by redrawing configurations on the grid.
- The game of life is played on an finite 2-Dimensional rectangular **GRID** of cells.
- Each **CELL** can be alive or dead.

Conway's Game of Life

- The status of a cell changes with each turn of the game, also called a **GENERATION**, depending on the statuses of the cell's neighbors.
- **NEIGHBORS** of a cell are cells that touch that cell from any direction - horizontal, vertical, and even diagonally.
- The initial **CONFIGURATION** is the first generation.
- The next generation is evolved from applying the rules of the game simultaneously to each **CELL.**

Rules Of LIFE



Remains ALIVE (2 Neighbors)

Remains DEAD (2 Neighbors)

Acquires LIFE (3 Neighbors)

DIES (1 Neighbors)

Physical Rules Governing the Simulation

1. Any live cell with fewer than two live neighbours dies, as if caused by **UNDER-POPULATION.**
2. Any live cell with two or three live neighbours lives on to the next generation by **PROCREATION.**.
3. Any live cell with more than three live neighbours dies, as if by **OVERCROWDING.**
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by **REPRODUCTION.**

Object Oriented Planning

Object Oriented Planning

STEP 1: Writing the use case description from the given project requirements.

STEP 2: Making a conceptual model of the application by finding out the main objects and drawing associations between the related objects of the application. These objects are the **nouns** found in the use case descriptions.

Ex:- Grid, Configuration, Generation, Cell, Pattern, Simulation, Rules, Neighbors etc.

… Continued

Object Oriented Planning (Continued)

STEP 3: Some of the objects identified in Step 2 gets converted into classes. Appropriate class relationships are forged wherever required.

Ex- Grid, Rules, Generation, Simulation etc.

STEP 4: Finding the responsibilities of each of the classes formed in Step 3. These responsibilities are the **verbs** found in the use case description.

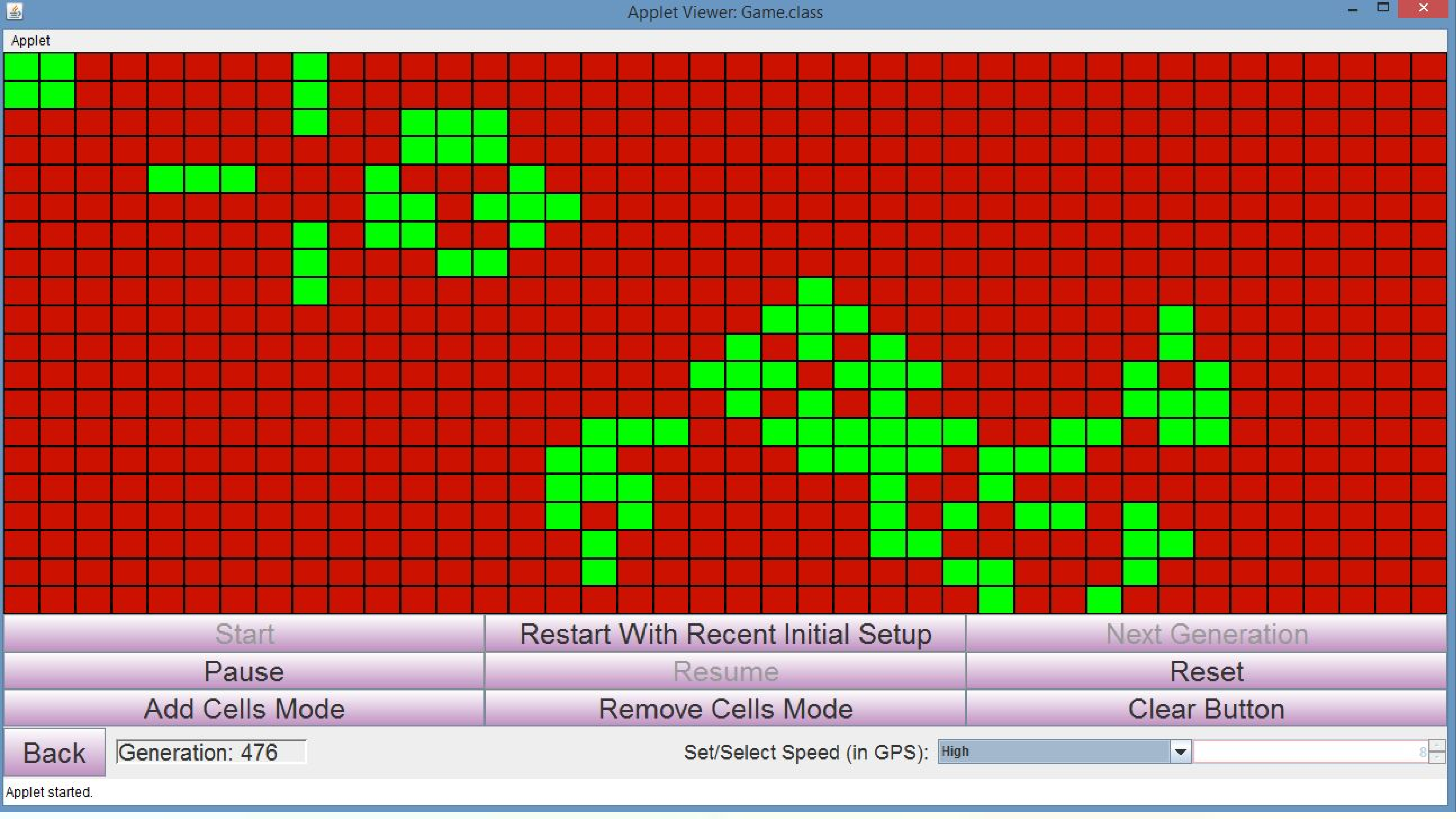
Ex- Generation class has to calculate neighbors and generate next generation.

Patterns in Life

Why is **LIFE** So Interesting?

- A popular implication is that a simple configuration on being subjected to changes over generation can lead to amusing **patterns** in life.
- Many different types of patterns occur in the Game of Life including static patterns ("**still lives**"), repeating patterns ("**oscillators**"), and patterns that translate themselves across the board ("**spaceships**").

Implementation



Generation: 476

Set/Select Speed (in GPS): High 8

Applet started.

Thank You