Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative

SUPSI

Minesweeper Project

Studente/i	Relatore	Correlatore
Galdiolo – Soldani - Vavassori	Giancarlo **Corti**	Tommaso **Agnola**
Corso di laurea	Modulo / Codice Progetto	Anno
Ingegneria Informatica	Software Engineering I	2024/25

Data

02/06/2025

Table of contents

1. Context & Motivation
2. State of Art
3. Problem & Approach
4. Diagrams
5. Results & Conclusions
6. Demo

Context

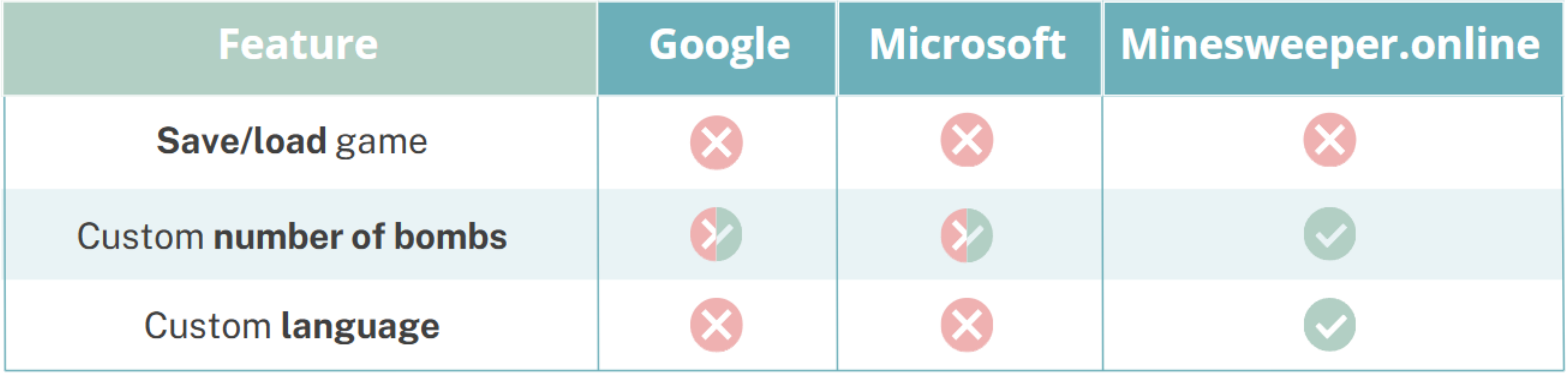
- **Requirements** and **versioning**
- **Software Production**
- **O**bject-**O**riented **P**rogramming

Motivation

- Application of **Software Engineering Principles**
- Use of
  - **MVC pattern** (Frontend)
  - **3-Layer architecture** (Backend)
  - **JavaFX** (Graphics)
- Build of a **standalone application** (Linux – Windows – MacOS)

❗ **Mandatory:** part of the **evaluation** of Software Engineering I course

State of Art



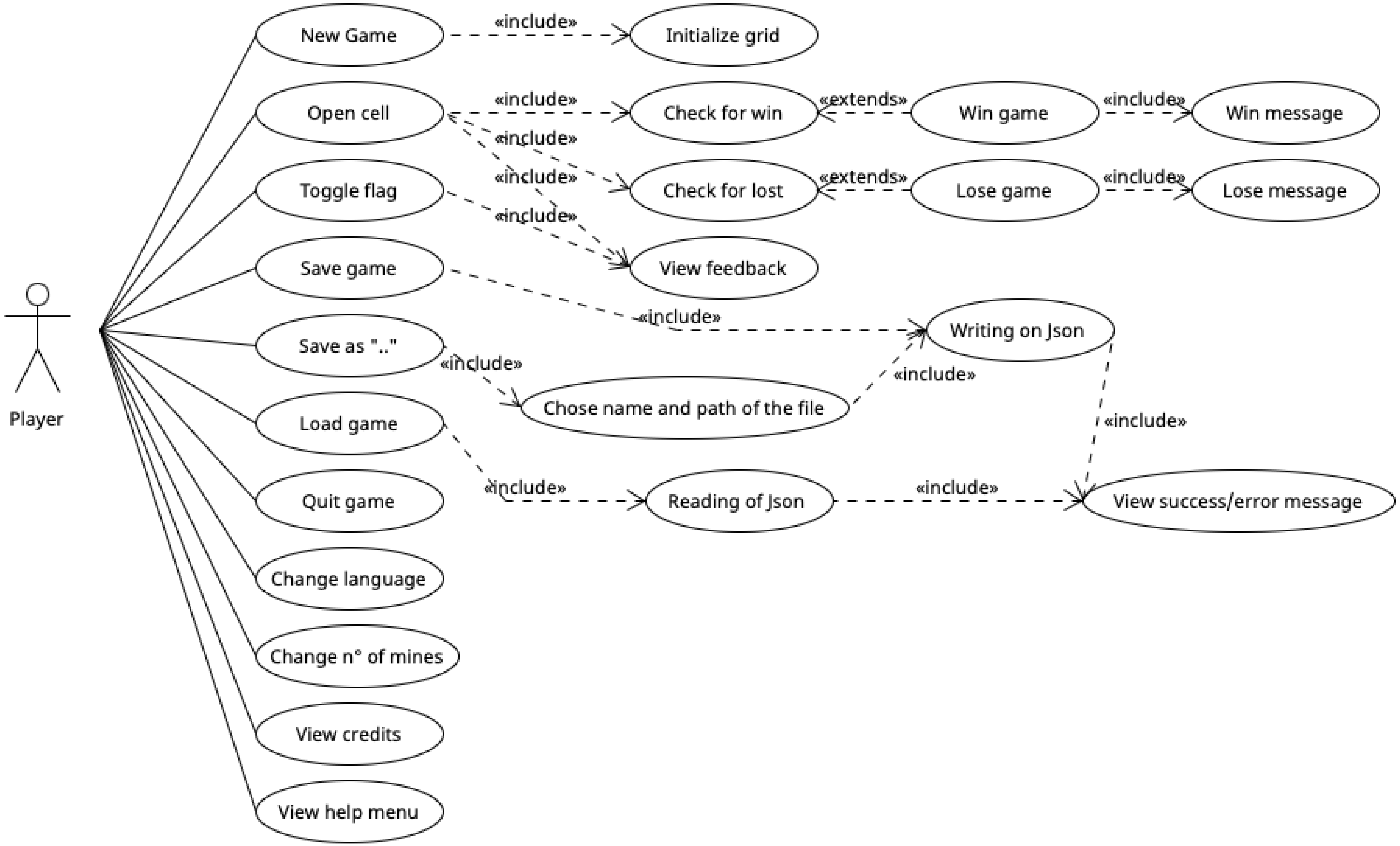
Feature	Google	Microsoft	Minesweeper.online
Save/load game	✗	✗	✗
Custom **number of bombs**	✗/✓	✗/✓	✓
Custom **language**	✗	✗	✓

STUDENTS

Problem

- **Preference file**: number of mines and language (*IT – EN*)
- **Cell interaction**: contextual feedback
- **Save/load**: possibility to save and load already existing games
- **App info**: version of number, build date and developers' credits
- **Build pipeline**: automatic build and artifact publishing

Problem – Use Case Diagram



Approach – Design Solutions

- **Requirements Elicitation and Formalization**
  - Asynchronous meetings and shared brainstorming
  - Discord (chat + call + screen sharing)

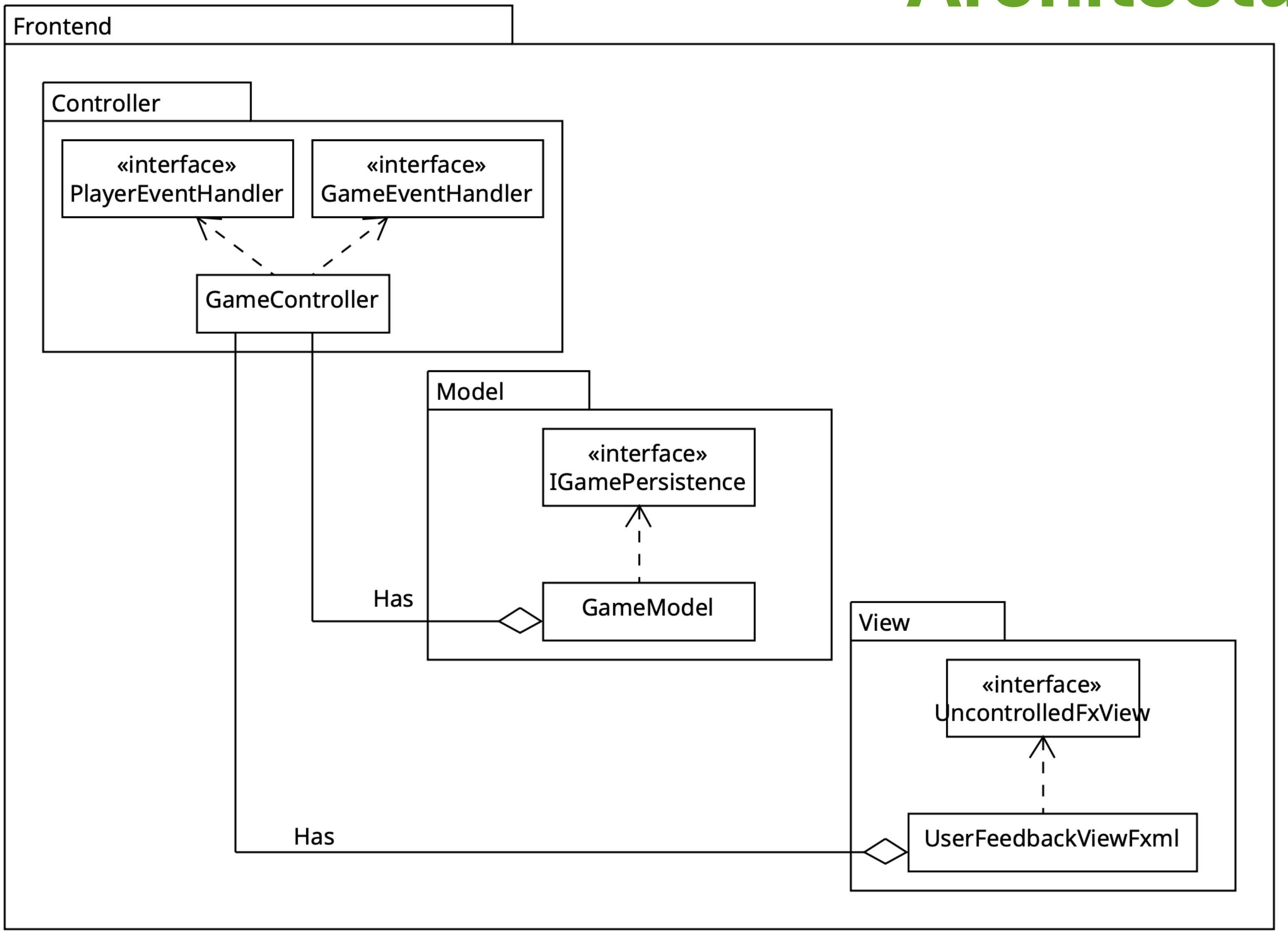
- **Product versioning**
  - Incremental versioning and branch work
  - Git - GitLab

- **Ensuring cross-platform compatibility**
  - Using the user directory as a basis for configuration files
  - System.getProperty("user.home") + Path, Files

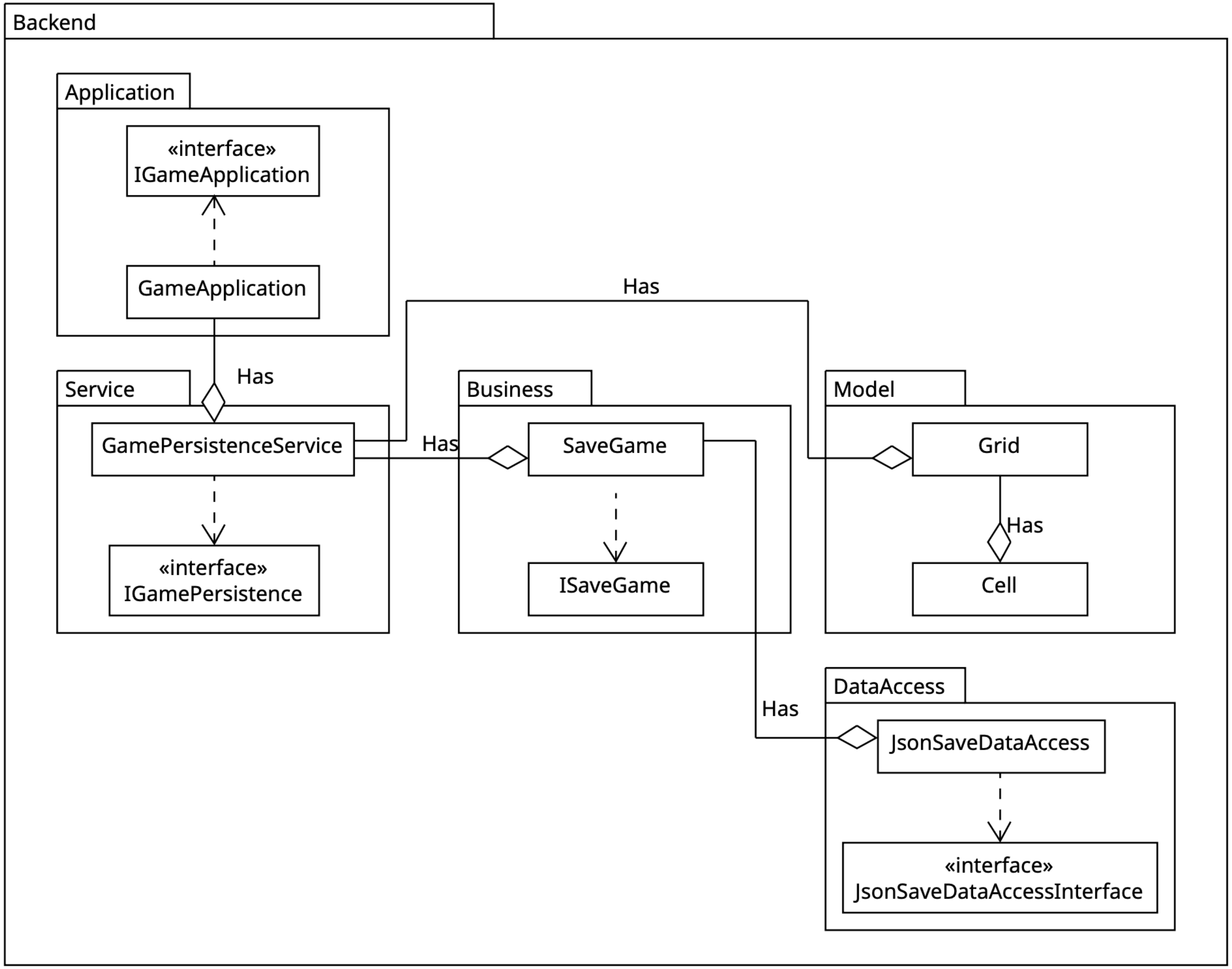
Approach – Design Solutions

- **Save/load functionality**
  - Serialising grid and cell status in a file
  - Jackson (ObjectMapper)
- **Produce clear and updatable diagrams**
  - Readable, lightweight diagrams for collaboration
  - UMLetino (.uxf file)
- **Separation of concerns to facilitate maintenance**
  - Layered and MVC Architecture
  - Use of packages and interfaces to exploit the IoC principle

Architecture



Architecture

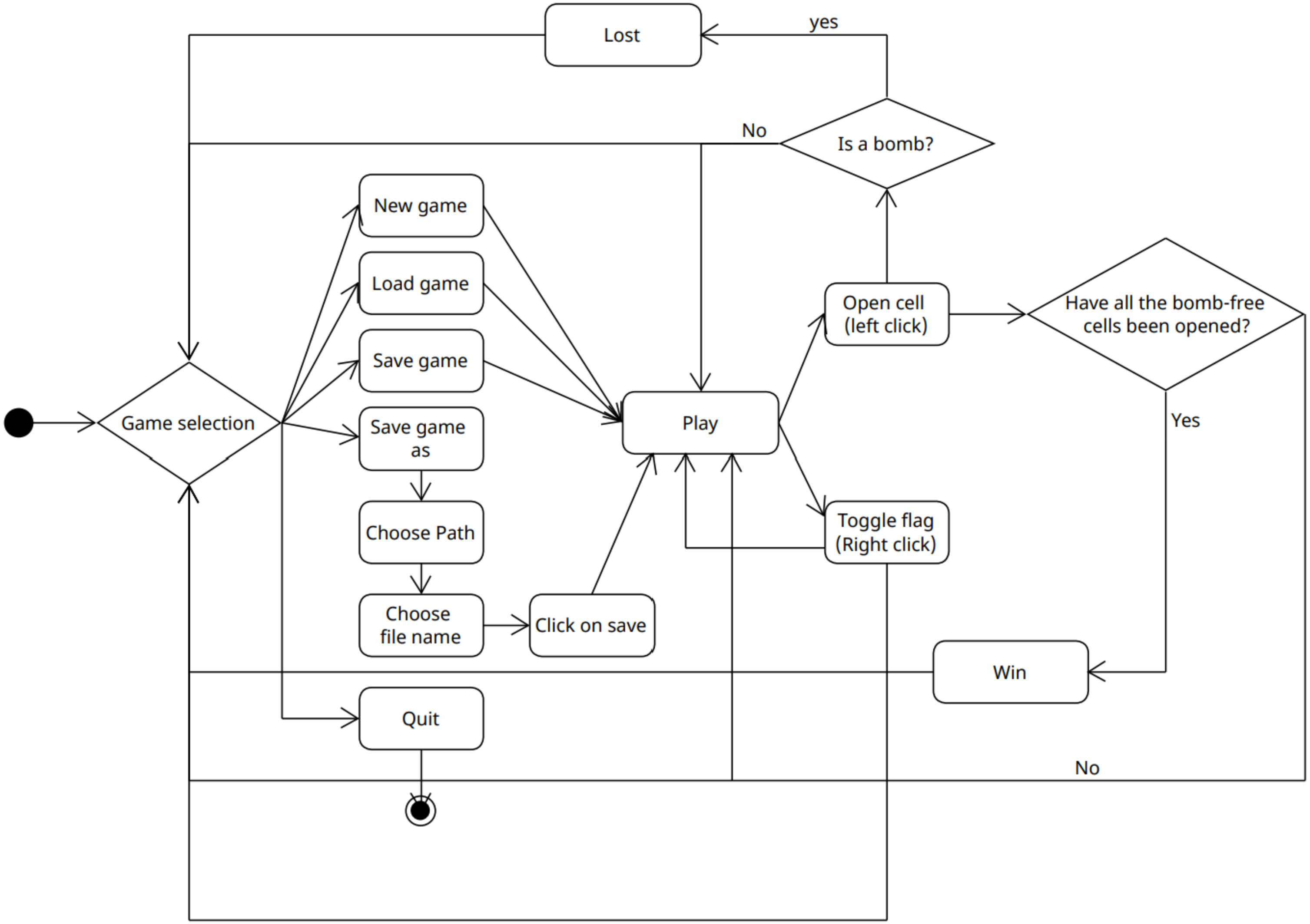


Approach – Design Solutions

- **Allow multi-language translations of the game**
  - .properties files for languages + TranslationsBusinessInterface interface
- **Manage user preferences** (number of bombs, language, etc.)
  - .properties files with fallback to default values
  - Java Properties + access via PropertiesDataAccess

SUPSI

Activity Diagram



Results

⚙ Developing an **up and running multiplatorm program**, tested on:

- **Linux**
- **Windows**
- **MacOS**

DEMO

Conclusions

- Implementation of **MVC** and **Layer Architecture**
- Correct use of **Versioning Tools**
- Improving our **cooperation** and **communication skills**