

# Ozploding bozmbs – Bomberman in Oz

Alexandre Gobeaux and Gilles Peiffer

**Abstract**—In this paper, the methodology and design choices behind our implementation of the Bomberman game in Oz are given. Both the turn-by-turn part of the game controller as its simultaneous part are explained, and an attempt is made to clarify why the authors took certain decisions with regards to how the final product works. On top of that, an algorithm was implemented in order to control the players’ moves, reacting to various messages it receives containing either requests or information about the game. This was done in at various levels of complexity, building increasingly efficient players. On top of the mandatory parts of the project, various supplementary features and embellishments are also included in the implementation.

In order to validate our product, interoperability tests were also carried out with both the reference player and other teams’ players. This paper presents a brief summary of the results of these tests.

## I. INTRODUCTION

**B**OMBERMAN is a famous game.

## II. CONTROLLER STRUCTURE

*A. Turn-by-turn controller*

*B. Simultaneous controller*

## III. PLAYER STRUCTURE

*A. General structure*

**P**LAYERS have to respond to different messages they can get from the controller. In order to handle each of these messages, multiple so-called “handler functions” were created, one for each possible message. One also has to store all relevant information about a player, such as its state, ID, number of lives, position on the board, etc.; in order to do this in a clean manner, the authors opted to use record structures, which can easily be modified with the `AdjoinList` function.

## IV. INTEROPERABILITY

## V. EXTENSIONS

## VI. CONCLUSION

The conclusion goes here.

## REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.