The Bomberman

INGI1131 Course project (2010)

1 Introduction

It is almost winter and we are looking for something hot to warm a bit all these cold days and nights. The contribution from the INGI1131 staff members is a warm-to-hot project statement. This time we are going to *play* an explosive game: *the bomber man*. You probably have heard about it but if haven't, do not worry, this is a modified version with our own rules.

Continue reading this document until the end and read it again until you have a clear idea about what are we asking you to design and implement. Be aware about all the rules, do not be shy to be creative and enjoy this project.

2 The paranormal bomber man à-la INGI1131

From the introduction you probably think that this game is about explosions and bombs. If you do, you are right but there are more things: it is about players, teams, food, and tricky things.

The game consists of a world in which there are players of two different teams. The players are starving and are looking desperately for food. When they step in a place of the world containing food they immediately eat it and their power is increased.

As life is not always fair players neither. Players minds are very irrational and to get as much food as possible they try to kill others by placing bombs in the world. Here you see the relation with the game's name!. They place a bomb and some short time after the explosion takes place (this time is only to run away). How bigger is the explosion depends on the power of the player placing it. There can be from level one to level four explosions, being the last ones the more dangerous.

The bombs in this game are different from conventional weapons in two ways: the affected zone is not radial, actually when there is an explosion, it only propagates vertically and horizontally affecting everyone it encounters; the second difference is that bombs do not kill people all the time. When a player in the game is reached by an explosion it gets killed if the bomb causing the explosion was placed by a player of its own team. If the bomb was placed by an adversary then the affected player will become part of the other team.

There is only one *fair* way to protect against explosions and is by using walls. Walls are placed around the world and they allow players from any team to be protected.

Not everything is good for players with have eaten a lot of food. As they eat foot they are converting it in power and this is a heavy process. This is precisely the reason why players get slower when they eat food. They recover the speed when they place the bombs. Concerning the food there is another issue: it appears randomly in the world but for some amount of time. After this time it disappears and will appear somewhere else.

This game is just fiction! however in every game there should be a winner. A team wins when all opponent's players are extinct.

2.1 Rules' summary

The rules can be summarized in:

- Bombs are placed by players and their power depends on the amount of food of the player at the time of putting it.
- Explosions propagate in a cross pattern. Only vertical and horizontal places are affected by an explosion. A level 3 bomb will affect only three positions in a cross pattern.
- A player is killed only when it is reached by a bomb placed by a teammate. If the bomb was placed by an adversary the affected player is *converted* to the other team.
- Food appears for some time and then disappears and is placed randomly in the world.
- Every time a player steps on a place in the board containing food its power gets increased. The more food the slow the player becomes: a player able to place bombs of type four is four times slower than an starving fast player.
- Walls are the only way for players to protect against explosions.
- When there are only players of one team in the game that team becomes the winner.

3 The real statement!

After all this prose about world food and bombs you may be thinking about the relation with a course on applied concurrency as is the case of INGI1131. Well

the short answer is: you should find the relationship!. To help you to establish this relation we ask you to provide an implementation of the game described above. This implementation will not be user interactive, instead it will be like a simulation where the different elements of the game interact according with the given description.

For implementing this project you have of all the knowledge acquired during the course, so remember all the abstractions you have because some of them will be very useful here. As an advice, remember to think before start coding. A good design is very valuable and will make your life easier during the development. To encourage you even more to design we ask you to present a short report of maximum two pages with the design of your project. This report has to be written in English and has to be submitted along the project. The report will contain the most important things of your design including the decisions, some diagrams explaining the concurrency and the main problems you faced during the project.

To provide an easy way to grade your project we ask you to implement a user graphical interface. This is very straight forward in Mozart-Oz by using QTk and we will send you a practical example for you to reuse and adapt.

There are some practical issues that need to be defined and that influence your project. It was not stated in the description of the game but the world has a size. This size is given at the beginning (as a parameter) and consists of the height and width of the world. Also the positions with walls are given as an Oz list of pairs (2 size tuples) with the coordinated when there is a wall.

The positions of the players at the beginning are in the top and the bottom of the board (like in chess).

4 The challenge

To encourage good projects we have decided to make a challenge. In order to participate on it we will define a common interface for the teams to play. There will be a module called *the mind*. This component has a very simple interface that encapsulates the way players of one team interact in the game. Given a position it returns the environment in the game around this position including: bombs, walls and other players. However, the received environment only reflects the state of the game for the contiguous positions. After calling *the mind* with the desired position it will return the corresponding action.

5 Advice to submit and to win the challenge

As an advice remember that graphical user interfaces (GUIs) are just intended to observe the interaction among all the actors in the game. It can be simple or

elaborated, this is your choice but please take into account that the main purpose of this course is about applying concurrency abstractions and not GUI design.

Regarding the contest: game strategies can be distinct for all the groups and on this we have to make it clear that this is not a course on artificial intelligence so we do not expect algorithms implementing complex strategies. If you have time to spent on this part do it but be sure your project fulfills all the requirements.

At the end, the best advice to win in this game and the contest is to recall Charles Darwin's quote: *It is not the strongest of the species that survive, nor the most intelligent, but the one most responsive to change.*

5.1 How to submit

The project is in groups of maximum **two** people. And in order to submit it you have to send an email to *gutierrez.gustavo@uclouvain.be* with the subject "`[INGI1131-2010] lastname1-lastname2`" replacing `lastname` by the last name of the people in the group. Additionally attach a zip or tgz file called "`lastname1-lastname2.zip`" with the project sources and the document. As an example let's suppose that Boris Mejías and Gustavo Gutiérrez are submitting the project, the subject will be: `[INGI1131-2010] mejias-gutierrez` and the attached file's name must be `mejias-gutierrez.zip`. Please, **do not** send any file in proprietary format. For instance, avoid .doc files for the reports.