Console Snake

1. Introduction

The name stands for what it is, it's a simple verion of the original Snake concept, with some changes, running in a console application. The game plays for the same goal, the player have too eat as much food as possible, and fill the map. While it happens, the snake will earn score depending on how fast it reaches the next eatable piece. Next to that, there is also a multiplayer option implemented, it is capable of letting players play on different same sized maps, and on one map together as well.

2. Setup

The application starts with many questins about what to setup. These could be seen in Fig. 1, with example answers included. More on what these exactly are:

1. Number of players*: This is to determine how many snakes are gonna be in play. [int]
2. All snake on one map: If yes, we get one map with all the snakes, if not, we get a map for each snake. Any character after the 1st is ignored and only the 1st is checked. [string]
3. Size of the map*: Map size, boundaries included. [int] [int]
4. Name your snake: Has no effect on the game, will be used if your snake dies or wins. Will be asked for all the players separately. [string]
5. Movement buttons: This is a sequence, that will be used, to set up the movement keys. This plays a role on the road to victory, be careful with mistaking here. Will be asked for all the players separately. Most of the special characters are not accepted. Any character after the 4th is ignored and only the first 4 is checked. [string]

**There is no upper limit, although, the game will most of the time not handle them correctly, if they are too high (e.g. snakes can't fit in the map, maps can't fit in the screen, …).*

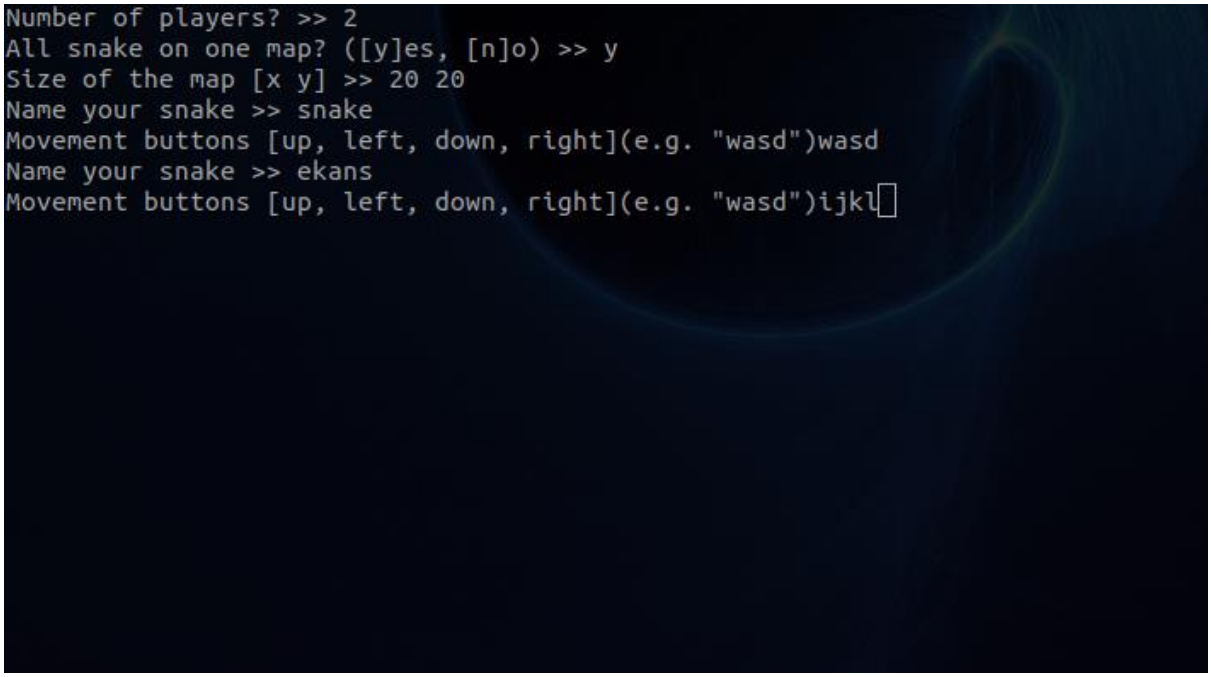

Fig. 1 Example setting for the game.

There is **NO** check on that if an input was correct or not. Most cases wrong input results the game starting with the default settings or cause errors. If you came to play, you better get it right.

3. Gameplay

After the setup was successful (enough), the game will instantly show the map(s), and wait about 2 seconds to start. The map(s) will have one or more snakes at this point, and a cherry. An example of the default setting is shown on Fig. 2. A starting scene could also look like the ones on Fig. 4 or Fig. 5, depending on the settings.

After that, the snakes will move to the direction of their last eligible choice every 0.5 seconds. A choice can either be changing direction to the sides, or keeping the same direction (this happens, when the button of the current direction, or the opposite is pressed), to be eligible. If there is no input, the snake will just keep the direction from the last move it made.

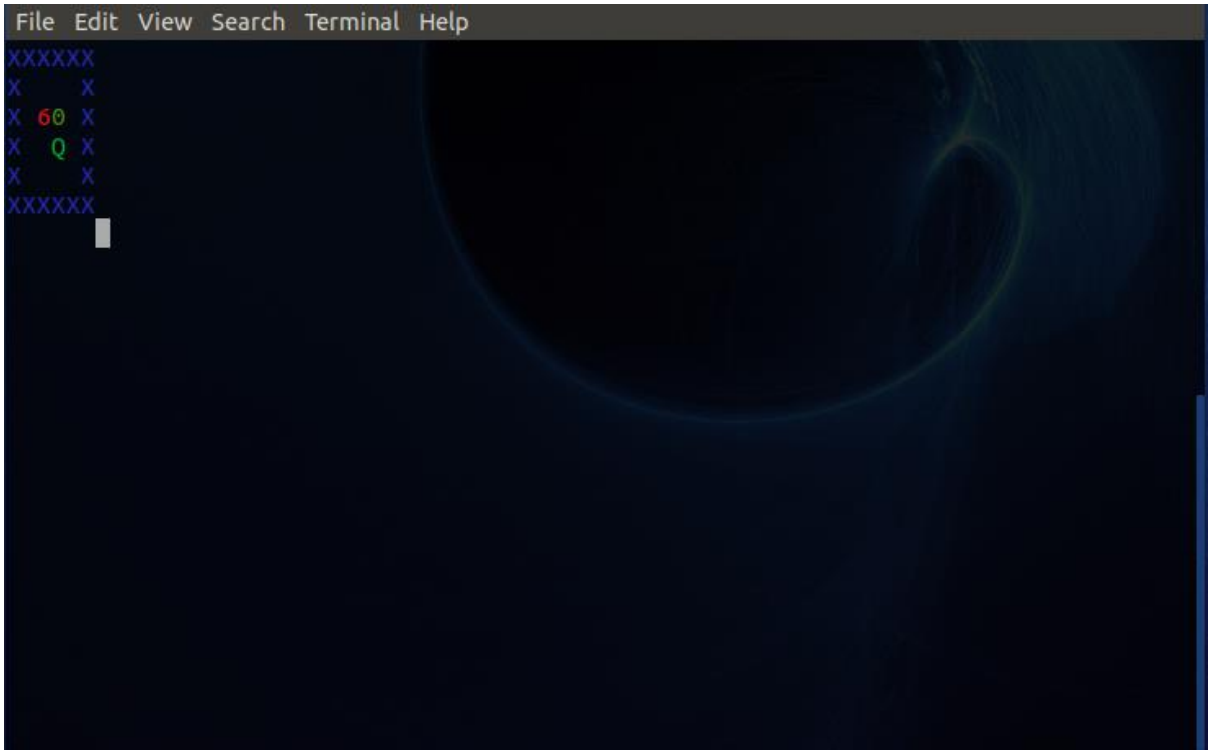

Fig. 2 Default setup start screen.

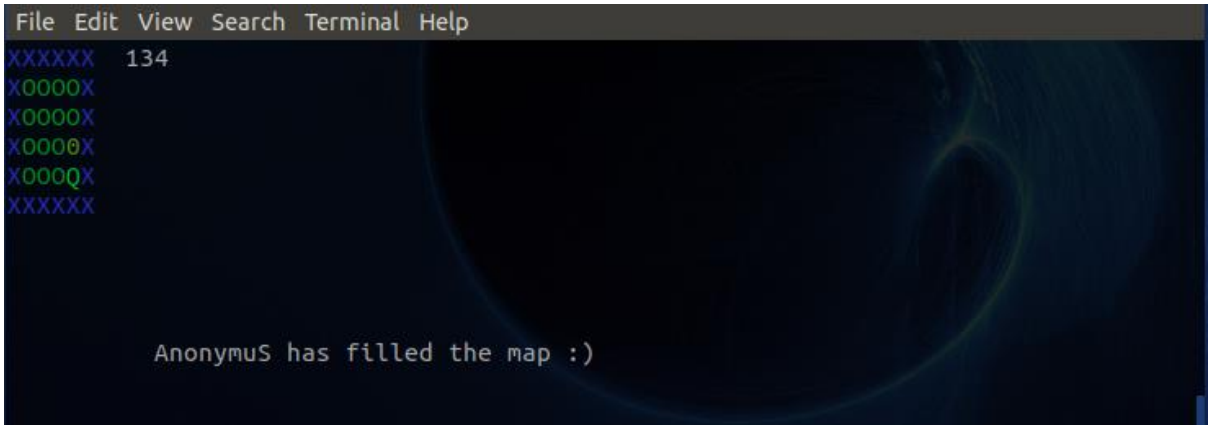

Fig. 3 Default setup victory.

From this point, the objective is filling the map, or getting a high score. This could be achived by eating food. When the character eats one, the snake will grow, and it's score is going to be updated, as well as a new piece of food will be placed on the map randomly. When there is no room left for more cherry, the game will stop, because the given snake filled the map. This is indicated with a text in the middle of the console window as well. This is shown on Fig. 3, with the default setup.

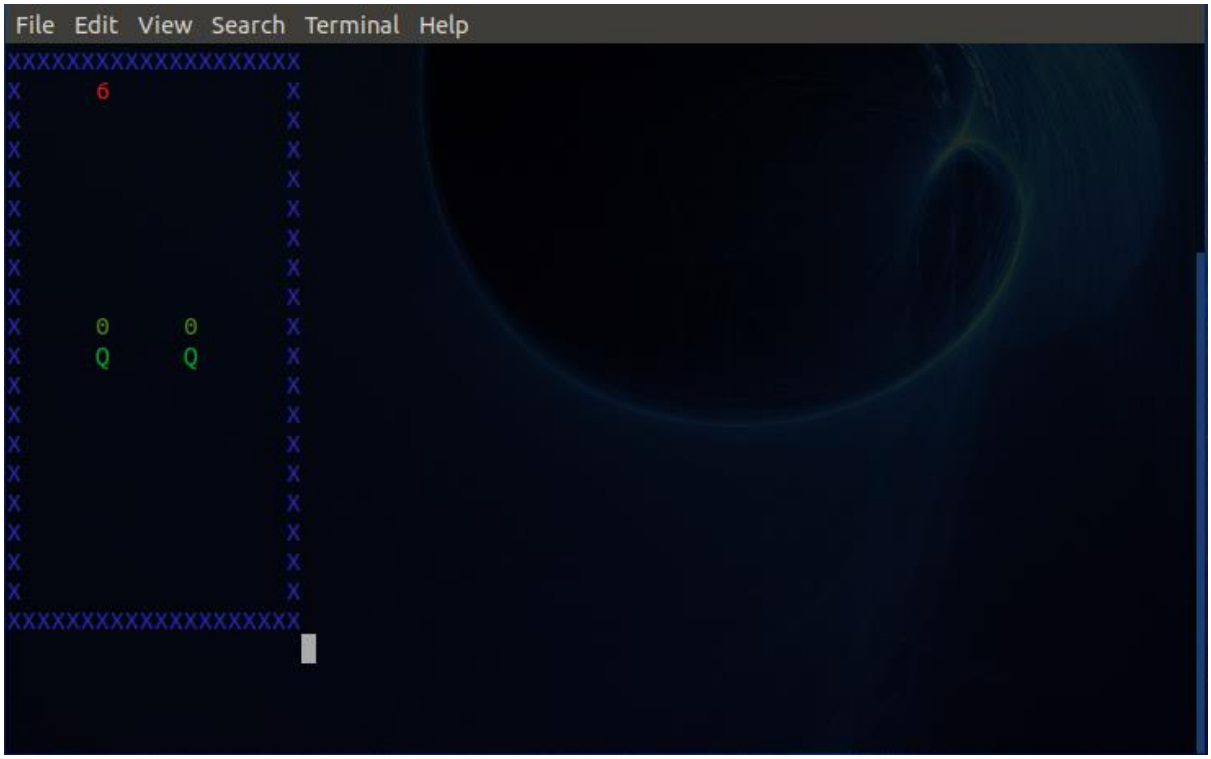

Fig. 4 One map with 2 snakes at start. This is what the setup on Fig. 1 would produce.

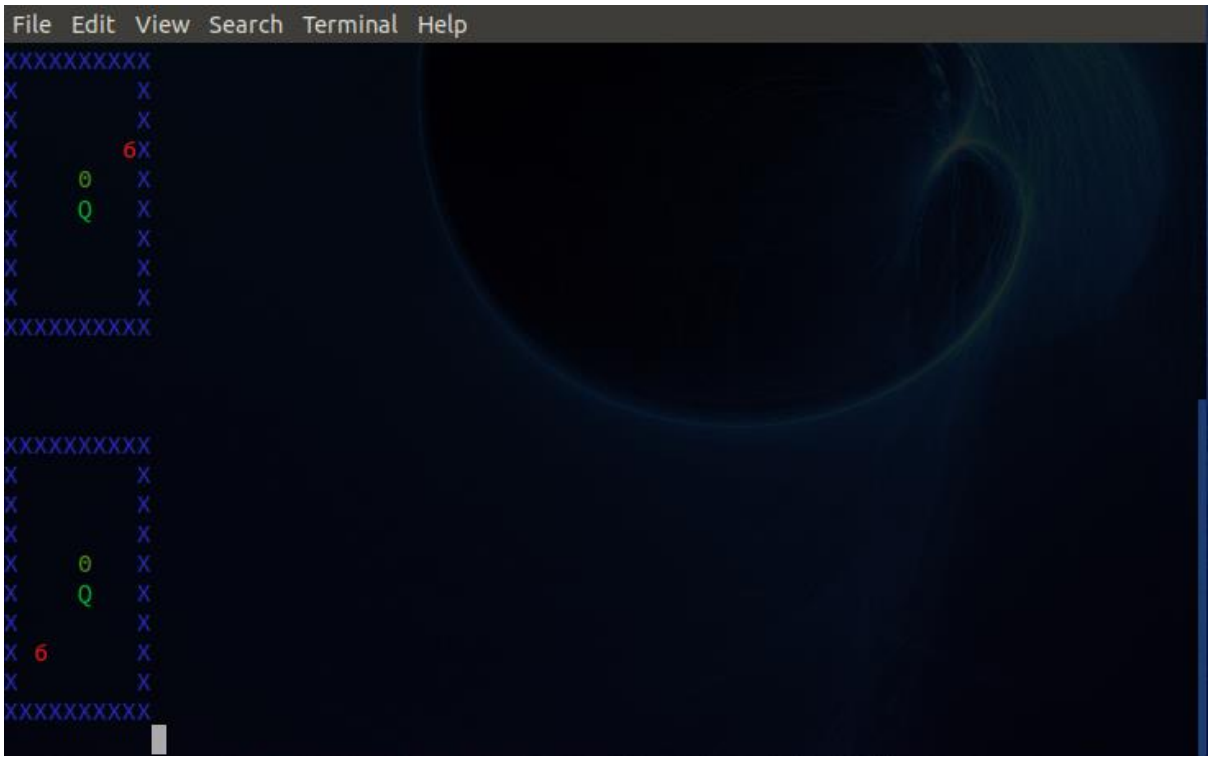

Fig. 5 Setting where each snake has it's own map at start.

Other than the victory, there are other ways to end a round. This happens, when one snake hits something, what is not food. This is shown on Fig. 7 (what features the ending of the gameplay shown on Fig. 6) and Fig. 8. As of the victory, this is also an occasion what features a text in the middle.

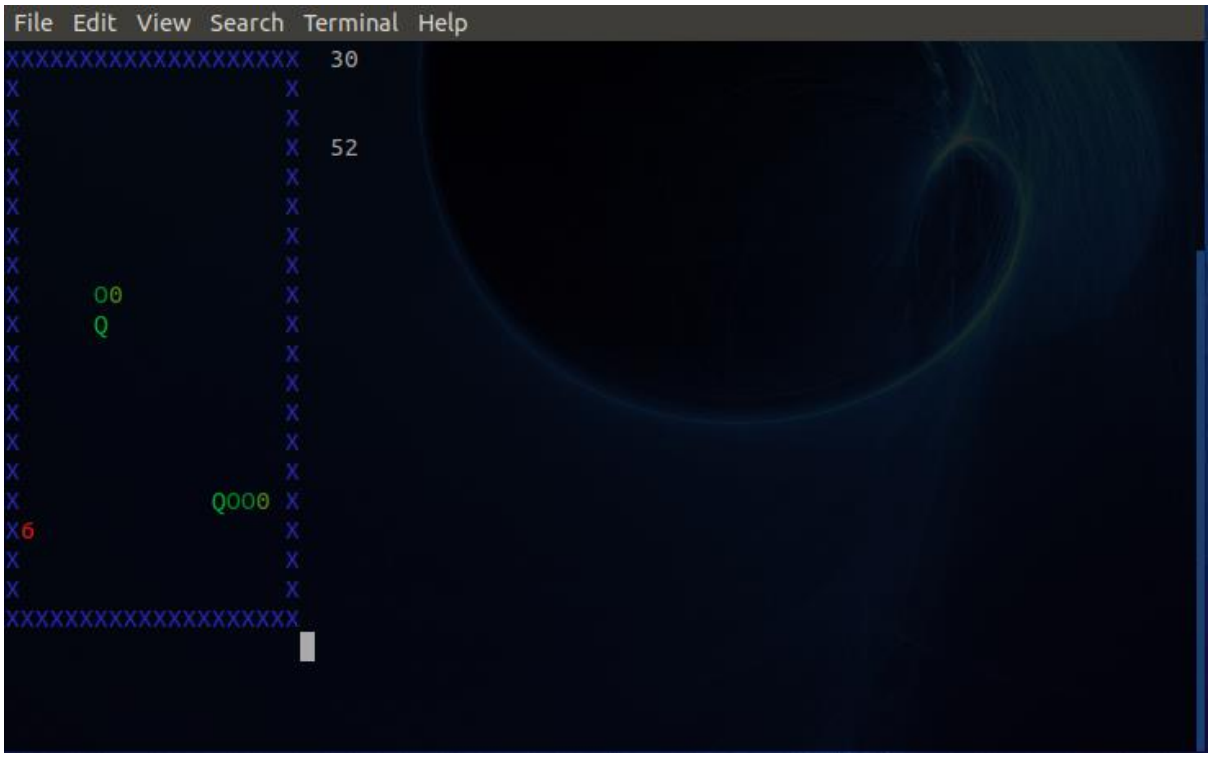

Fig. 6 Picture of the gameplay, where snakes have eaten some cherry already.

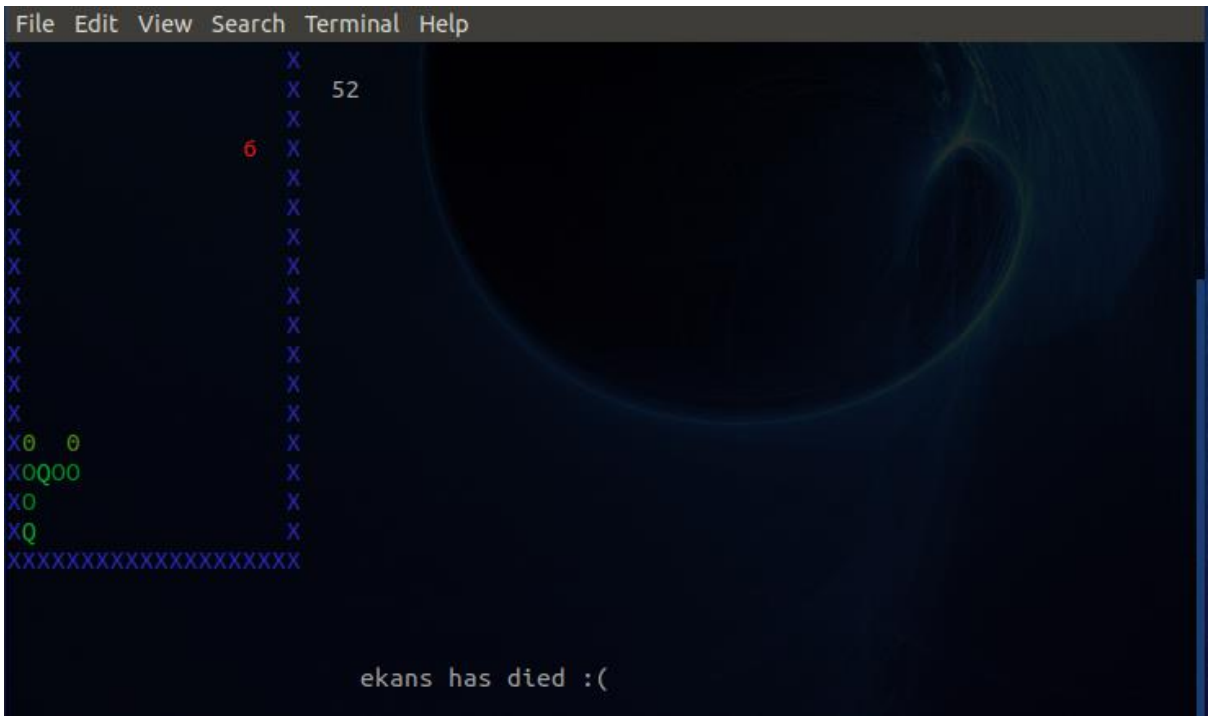

Fig. 7 The second player has died hitting the other players snake.

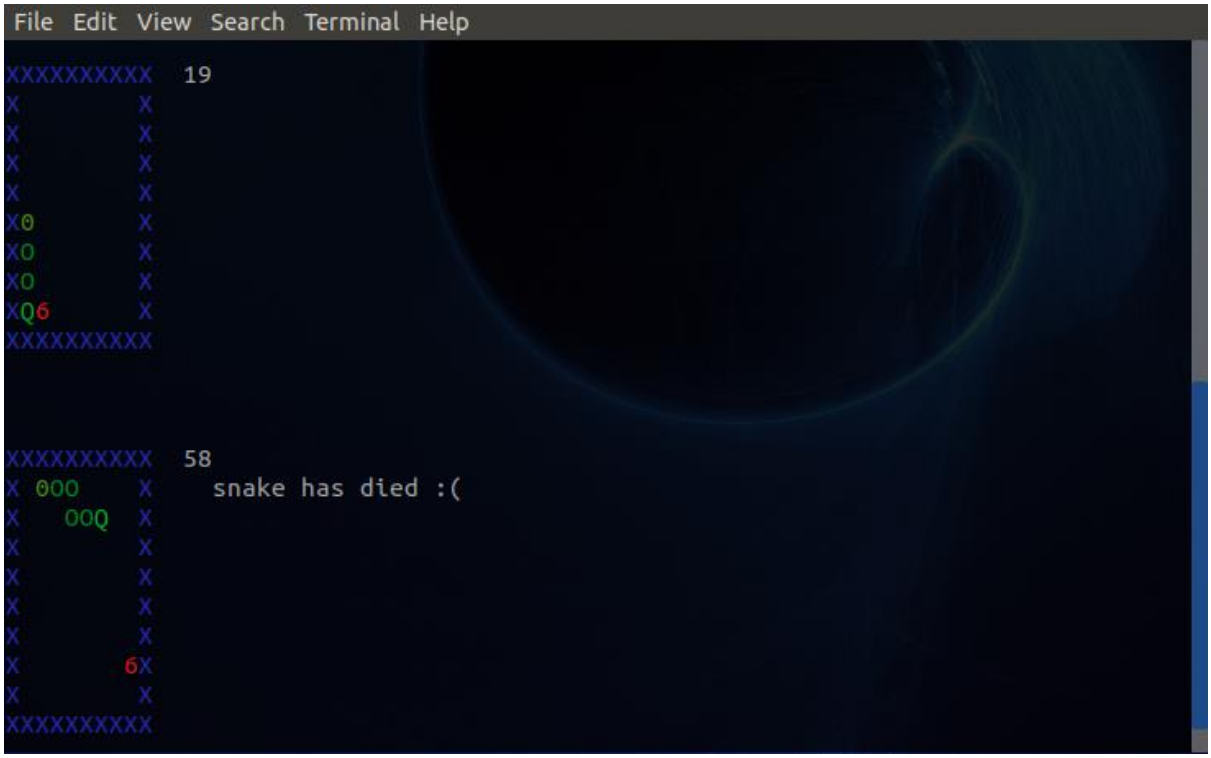

Fig. 8 The first player died hitting a wall, in a multimap setup.

4. Summary

The code of the game was created using C++ (including C++11 elements as well). There are elements, that are not part of the standard C++ library, these are necessary mostly for colours and element printing optimizatios. The code was designed, to be usable on both linux and windows platforms without any changes to the source. This goal was tested on the latest (as of 15th of June, 2018) ubuntu and windows versions, and seemed to be satisfied. There are although diffenerces on the two operating systems, because of the layout of them, the two is offseting maps differently. Mostly, this change is only visual, so no effort was yet put in to make changes here.

Hopefully, this sums up what the application is about, and how to use it properly. Good luck! :)