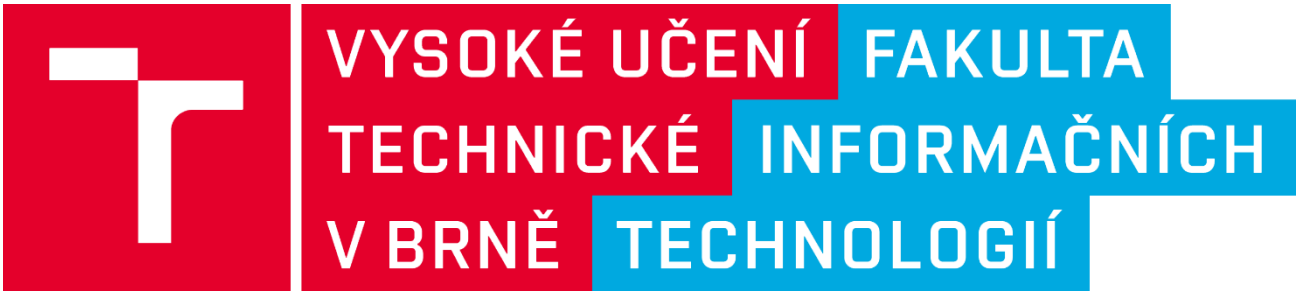

Game “Snake”

FitKit v3 Project

IMP, variant 18
Yahor Senichak (xsenic00)

Task

The task was to make a game using ARM-FITKit v3 with led display extension module. The project was implemented in C++ as it provides stronger abstractions and does not contain any overhead compared to C.

Implementation

The main program is divided into several parts: First – all system initialization Is moved into System.h file. Secondly Port A is frequently used, so it was decided to implement a class to avoid confusion using macros.

`PortA` has 3 immediate initialization functions and delayed mode, which is filled in several steps and then released all at once for better behavior.

After ports are established, the game must be initialized. For this an abstraction over display was made, called `Field`. This class consists of a bit field, which covers pixels of display. They may be switched on using `PutPixel` method, which puts 1 into specified cell. After the field manipulation it may be rendered. Method Render uses bits stored into bitfield to switch on the portA pins. For those helper functions are implemented (`column_select, xrow_select`).

Button input is gathered in main loop using polling, I've tried using IRQ, but for some reason it refused to start. So, both methods are implemented.

`Snake` is the core of the game, it stores field inside of it, which it fills respective to its data.

Game follows the rule of Input->Step->Render. Input is gathered every tick, as well as Render. Step is performed every 2000 cycles, and it advances snake to the next position.

Snake itself consists of `Head`, which is a coordinate pair and an array of its previous states, it's essentially a ring array, because every step a next element is filled, the old one gets deleted. Coordinates are counted respective to head, and rendering, uses before states in reverse order to determine which pixel is to be lit.

`Field` is then proceeded to be rendered in scanline fashion from left to right.

Snake can bite its tail, but then the game is over. To restart the game a bottom button needs to be pressed. And then the game is restarted once again from the original place, using Reset method on `Snake` class.

Demo

https://youtu.be/x0O5vjYTwTg

Conclusion

Specifications are all implemented, as well as logic of the game, it also has extensions of game over logic and reset using button.