Computer Systems and -architecture

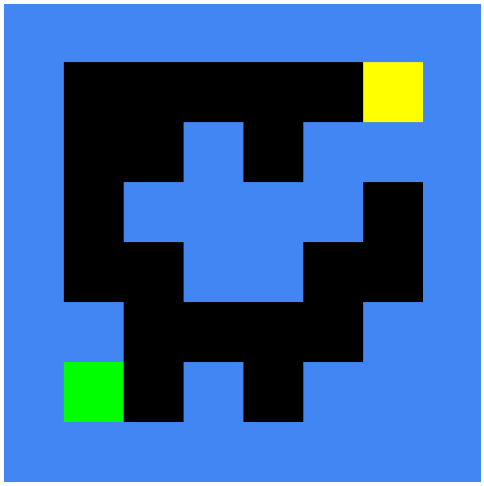
MIPS: Building your Projects' Building Blocks

1 Ba INF 2018-2019

Stephen Pauwels
stephen.pauwels@uantwerpen.be

Project

The goal of this project is to implement a maze-based game with influences from Pack-man. The user starts inside a maze and has to find the exit. When the user is able to reach the exit he wins the game. A maze looks like the one bellow:



Walls are in blue, passageways are in black, the player is yellow and the exit is green. A player has to move through the maze, using the keyboard (*hint: use z, q, s and d*), to the exit. During the last exercise sessions we will gradually build our little game:

- Week 8: Intro and creating the different building blocks (capturing keyboard input, displaying a bitmap, reading a file)
- Week 9: Combining all the different building blocks into one program using functions
- Week 12: Add algorithm to automatically find the exit

You will be asked to hand in intermediate results after the different stages of the project (every intermediate result will be graded). At the end, your end product itself will also be graded.

Time Schedule

The assignment is made individually. Put all your files in a tgz archive, as explained on the course's website, and submit your solution to the exercises on Blackboard.

- Deadline: **November 22, 23u55**

Assignment

Do not put all these programs into one file, make different files for the different tasks!

1. Bitmap
   (a) Write a program that sets every pixel in a 32 x 32 bitmap to red. Make sure you test your program with appropriate settings in the Bitmap Display tool.
   (b) Extend the previous program to set the color of the edges to yellow. Use counters to keep track of your current position in the bitmap.
2. Keyboard input
   (a) Write a program that reads once per 2 seconds from the keyboard. When a character was entered you print this character otherwise you print "Please enter a character".
   (b) Write a program that keeps reading a character. That displays the following when encountering these characters:
   - z : "up"
   - s: "down"
   - q: "left"
   - d: "right"

   When x is entered the program quits.
3. File input
   (a) Write a program that reads a textfile and prints the content of this file to screen.
4. From logical coordinates to memory addresses
   (a) Write a program that takes logical coordinates ((0,0), (0,1), . . .) as input and returns the actual address in memory for that particular pixel. Make sure you take the size (number of rows and columns) of the logical field into account.