

# Computer Systems and -architecture

## MIPS: Extending your Project

*1 Ba INF 2019-2020*

Stephen Pauwels  
stephen.pauwels@uantwerpen.be

### Time Schedule

Exercises are 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. Your final submission consist of two files. The first file (**maze.asm**) contains the solutions for the first two projects. The second one (**maze.extension.asm**) also contains the extension explained in this project.

- Deadline: **January 5, 23u55**

### Evaluation

Your project will be evaluated and assessed using the following table:

Aspect		Points
Keyboard Input	Is keyboard input processed correctly?	5
Bitmap Output	Does the maze gets displayed correctly?	5
Read Maze	Can you read an arbitrary input file?	5
Mips: Standard	Did you use the right assembler constructions and code?	10
Mips: Stackframes	Are stackframes used and created correctly?	5
Overall Game	Does the (manual) game work?	15
Search Extension	Can you automatically find the exit?	5
Correction	Some other errors or inefficient use of constructions?	- ?
Total	Counts for 50% of assignments CS	50

## Assignment

1. **Automatically finding the exit** Create a new asm-file with your basic functionality, but without the logic for processing user input. We will now implement a Depth First Search algorithm that is able to traverse the maze and finds the exit without human intervention. Implement the following function (written in a some kind of pseudo language) in MIPS:

```
dfs(location_row, location_col, visited):
    # First check if exit cell has been reached
    if is_victory(location_row, location_col):
        return "VICTORY"

    for move in [(-1, 0), (1,0), (0,-1), (0,1)]:
        new_location_row = location_row + move[0]
        new_location_col = location_col + move[1]

        if (new_location_row, new_location_col) not in visited:
            update_row, update_col = update_player_location(location_row,
                location_col, new_location_row, new_location_col)

            if (update_row, update_col) != (location_row, location_col):
                dfs(update_row, update_col,
                    visited + [(new_location_row, new_location_col)])

            update_player_location(update_row, update_col, location_row, location_col)
```

In order to create a nice visual representation of your player you have to introduce a sleep time after every player update.

2. **Bonus** Add candies to your maze, until all candies are picked up by the player the exit is not shown. It only will only be displayed when all candies are picked up. You do not have to automate this behaviour.