Computer Systems and -architecture

MIPS: Extending your Project

1 Ba INF 2018-2019

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.

- Deadline: **January 6, 23u55**

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 Dept 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:
            visited.append((new_location_row, new_location_col))

            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)

            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.