

# Computersystemen en -architectuur

## MIPS Project: Deel 3

*1 Ba INF*  
*2022–2023*

Kasper Engelen  
kasper.engelen@uantwerpen.be

Vrijdag 16 december 2022

## 1 Introductie

In dit document vind je de uitleg voor het derde, en laatste, deel van het MIPS project. De algemene uitleg van het project kan je lezen in de PDF van deel 1.

## 2 Deel 3: automatisch de uitgang vinden

### 2.1 Indienen en beoordeling

Je zal deze opdracht éénmalig moeten indienen op de finale deadline: **vrijdag 23 december 2022, 22:00**. Om deze opdracht in te zenden, dien je je code in via Inginius: “CSA2223” → “MIPS” → “Mips Project Deel 3”. Dit is de code van het volledige spel samen met het algoritme dat je voor deel 3 moet uitwerken. Tijdens de beoordeling zal de code uitgevoerd worden met MARS om te kijken of je programma correct werkt. Zorg er dus voor dat je programma **correct werkt in MARS!** Voor de beoordeling wordt de volgende puntentabel gehanteerd:

Onderdeel	Score
De speler beweegt automatisch naar de uitgang.	4
De speler kan niet worden bewogen met het toetsenbord.	1
<b>Totaal</b>	<b>5</b>

Tabel 1: Puntentabel voor het derde deel.

### 2.2 Opdracht

In het derde en laatste deel van het project zal je het videospel van deel 2 moeten aanpassen zodat de uitgang van het doolhof automatisch gevonden wordt. Deze aangepaste versie zal de speler niet vragen om input van het toetsenbord, maar zal volledig zelfstandig werken. Zodra het pad naar de uitgang berekend is, laat je de speler om de 60ms een stap zetten richting de uitgang. Het zoekalgoritme zal gebruikmaken van depth-first search om de uitgang van het doolhof te vinden. Je kan hierbij gebruikmaken van onderstaande pseudocode:

```
void dfs(location_row, location_col, visited):
    // First check if the exit cell has been reached
    if is_victory(location_row, location_col):
        exit_program()
        return

    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:
            // move towards new position, if possible
            update_row, update_col = update_player_location(location_row,
                location_col, new_location_row, new_location_col)

            // if we have moved, then continue dfs
            if (update_row, update_col) != (location_row, location_col):
                dfs(update_row, update_col,
                    visited + [(new_location_row, new_location_col)])

            // dfs did not yet find the exit, so we take a step back
            update_player_location(update_row, update_col, location_row, location_col)

    return
```