SUPINFO Academic Dept.

SUPINFO
International University

INSTITUTE OF INFORMATION TECHNOLOGY

Group Projects
A.Sc.1 - Architecture

Contents

2018-2019

Version 1.0
Last update: 09/10/2018
Use: Students/Staff
Author: SAD

TABLE OF CONTENTS

1. Project Overview

Your firm named ByteSmallGame is an editor and game designer using assembler technologies. Your firm imagines, proposes and designs small games only in assembly language. You have no limit and everything you construct is well done with explanations and details. Of course, games are tested.

Write technical and operational documents are your job. This documents systematically present the complete realization of the projects. During a project, the study are completely written (based on building diagrams) with the proposals and creation.

You have responded to an offer to create a maze game. .

2. Functional Expression

2.1. Operation

Application purpose is to design an assembler game to move a hero in a labyrinth. Hero must searching keys to open doors.

In following pictures, a game example

Figure 2.1. First Key

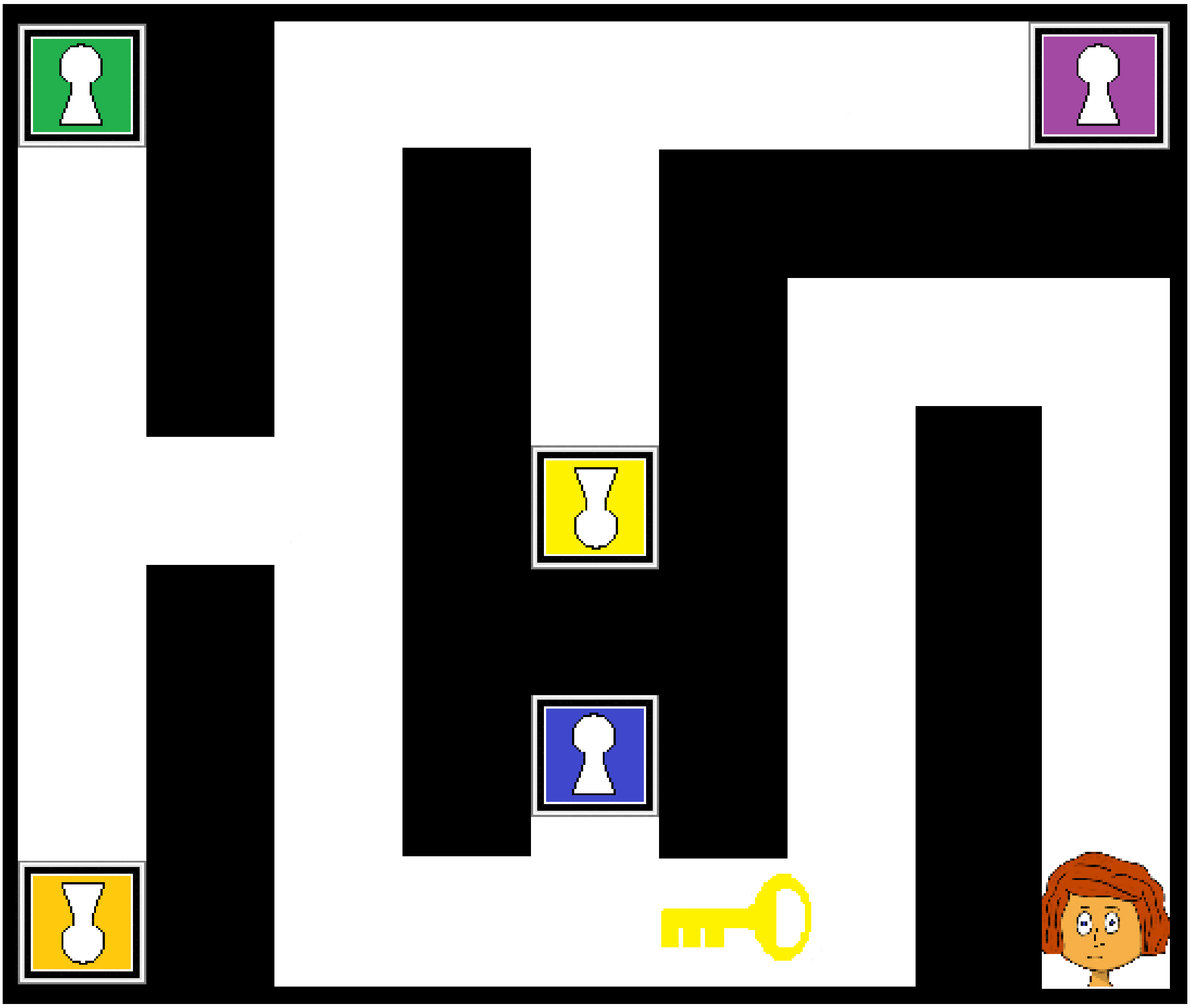


Figure 2.2. Second key, after the first door

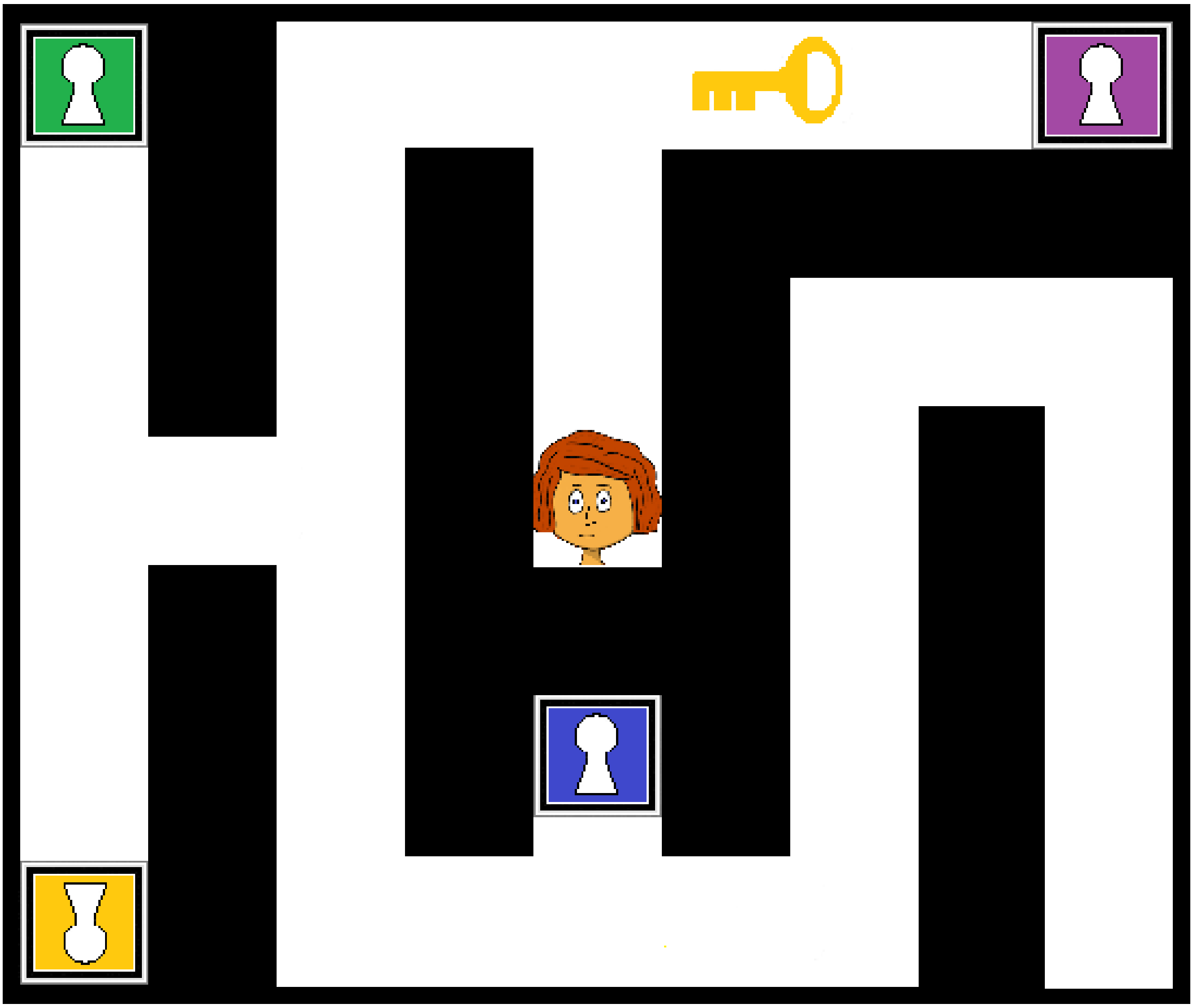


Figure 2.3. Third key, after the second door

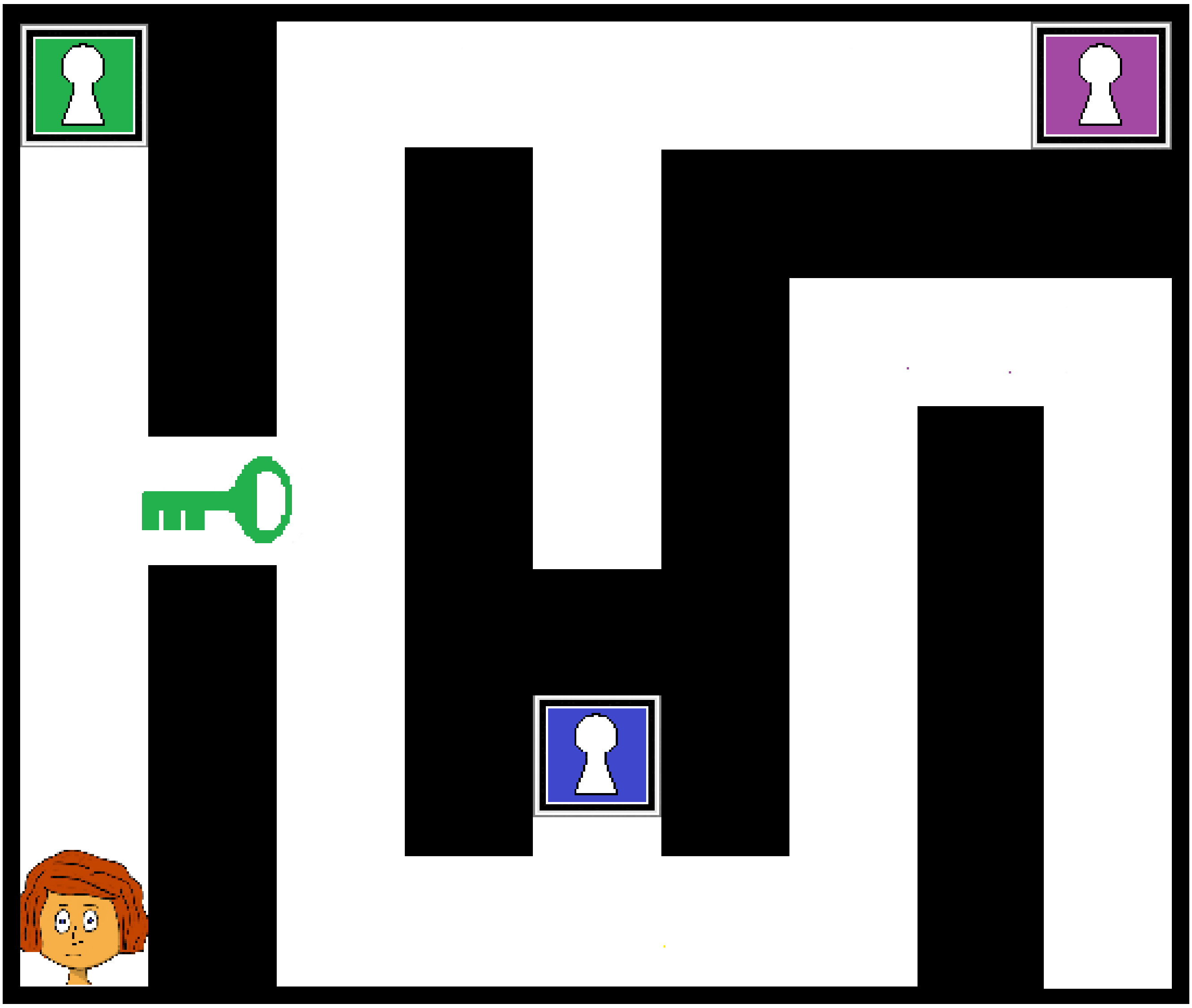


Figure 2.4. Fourth key, after the third door

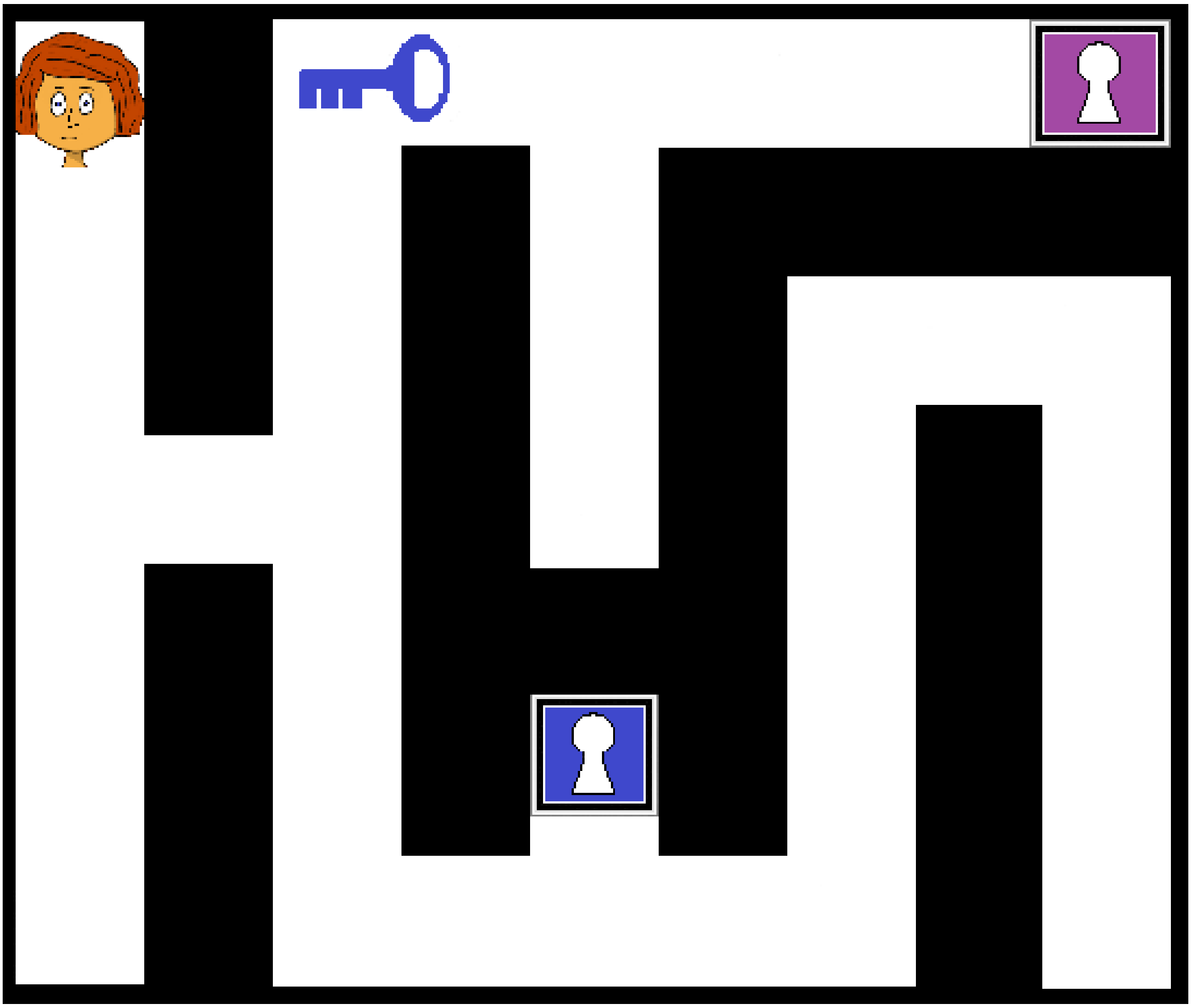


Figure 2.5. Fifth key, after the fourth door

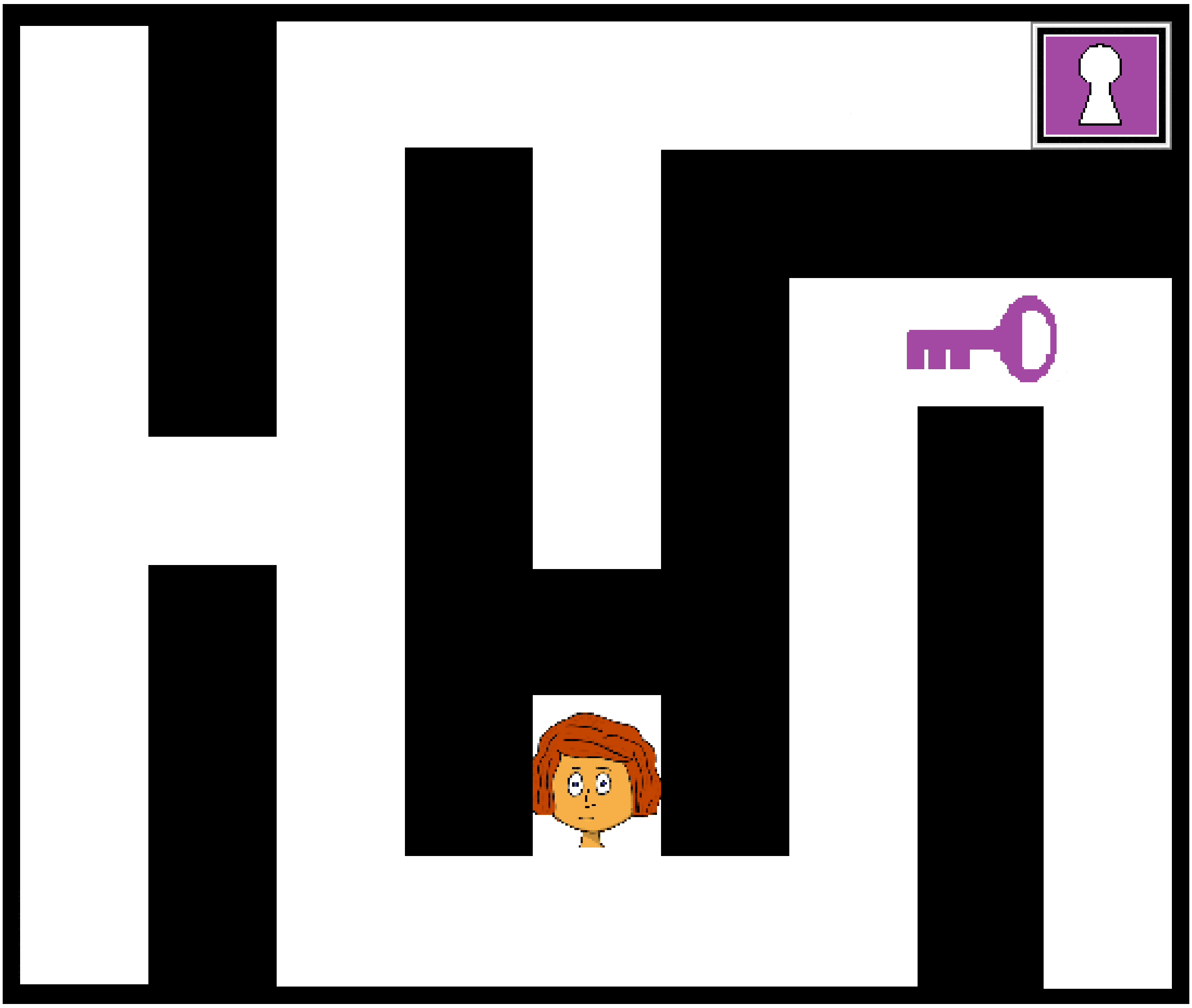
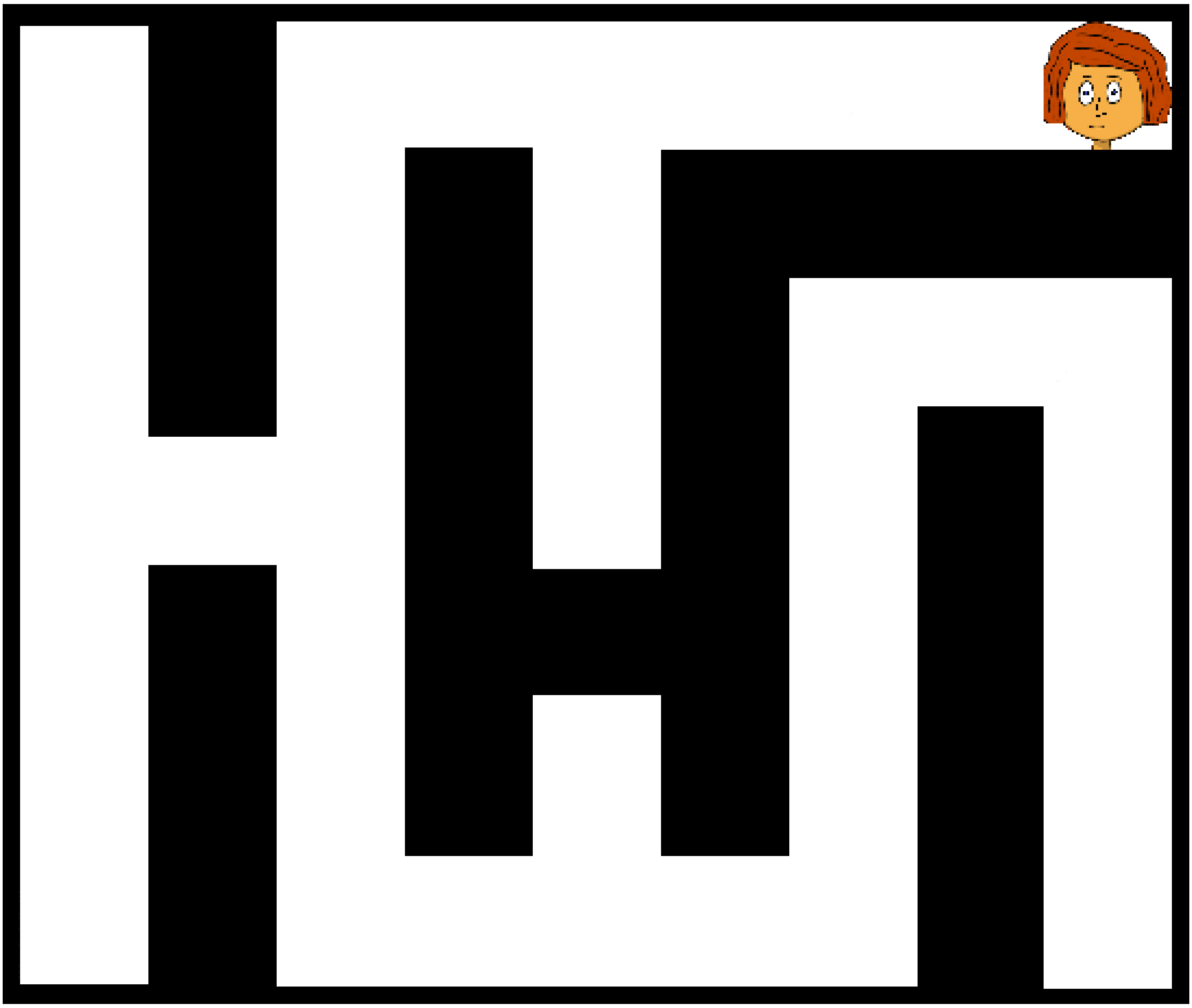


Figure 2.6. The final



Pictures show an example of game play. There are 6 screens to show the game play.

Your game must include the following:

- One hero
- 5 doors
- 5 keys (which are discovered one by one to open the doors)
- One maze

The game is :

- The hero move in maze, with keyboard keys.
- The hero cannot cross walls.
- When hero pass over a key, this key is recovered. The result is the key is fading from maze.
- When hero pass over a door (corresponding to the recovered key color), the door is opening and it's fading from the maze and another key appears in the maze (not in walls).
- When hero pass over a door, but the color door does not match with color key, it is impossible to open the door and nothing happens. The door is always visible after the hero departure.
- When hero pass over the last door, the game is completed.

2.2. Technical requirements

Your game can be done with only a language in **Assembler programming**. The 8086 Assembler language will be preferred, but it is possible to use any **ASM and their libraries**.

2.3. Bonus features

A bonus is for example a simulation to indicate if a door is valid from key, or a dialogue messages front the doors, such as: "You do not have the right key". All other additions which unlike of basic elements imposed, such as a menu to start game or music….

3. Deliverables

Students should include the following elements in their final delivery:

- A zip archive with the project source code. The source code must also come with the build system used (Project file, autotools, libraries, ...), if any.
- Project documentation.
  - Technical documentation explaining your choices and/or implementation choices/details on the following items (at least):
    - Component operation
    - Graphics management
  - User manual

The first document is an academic document. Address the reader as a teacher, not a client. This document can be in French or in English, at your option. On the other hand, user manual must be understandable by the client.

4. Graded Items

The project will be graded as follows, on a 230/270 scale:

- Documentation : 50 points
  - User documentation (20 points)
  - Technical documentation (30 points)
- System and architecture : 90 points
  - Creating the keys (40 points)
  - Creating the doors (30 points)
  - Creating the maze (20 points)
- The graphic simulation : 90 points
  - Disappearance management (30 points)
  - Collision management (30 points)
  - Displacing management (30 points)
- Bonus : 40 points
  - Music (10 points)
  - Text (10 points
  - Menu (10 points)
  - Simulation (10 points)