



Co-funded by the
Erasmus+ Programme
of the European Union



User guide for *Assignment 42*

https://store.steampowered.com/app/1641580/Assignment_42/

UPSKILLS Domain Cluster:

- Research-oriented
- Technical (optional as an introduction to programming)
- Transversal

Target Audience:

Undergraduate / postgraduate students taking linguistics & language-related courses

Requirements:

No specific requirements

Gaming Experience:

Intermediate

Skills Targeted:

- Understanding of language technology
- Analytical skills
- Logical skills
- Problem-solving skills
- Teamwork and communication skills
- Working under pressure

Learning Outcomes:

Upon completion of game play, the students are expected to:

- Have acquired a basic knowledge of programming (using visual programming)
- Demonstrate an ability to understand technology terminology and some related tools
- Gain an understanding of data management
- Refine their problem-solving skills through independent learning

Introduction (for academics & lecturers):

This is a game that introduces the concept of programming a bot to carry out a series of tasks leading to the final game mission. The game in itself can take a few hours to play, but the concept of the game can be grasped in the first hour or so of play. This game can be assigned either as a game to be played from start to finish, and thus get more confident with programming and computational thinking skills, or else it can also be played in part to support teamwork, working under pressure, and developing communication skills.

The story unfolds as the students carry out more programming tasks. At first, the instructions are rather scanty, and the game's purpose might not be as revealing. However, this becomes more apparent with more game play.

The overall purpose of playing this game, is that of having students gain a better understanding into a form of algorithmic thinking for the purpose of solving a given task. However, this game can also be used to help them improve on soft skills such as paying attention to detail, independent learning and problem-solving.

Although game play is single-player, if this is given as a group activity, in class or outside class, this can also help develop increased communication skills among the group, through a discussion of the possible solutions to achieve the set missions.

Pre-Game Briefing:

The game you are assigned is called Assignment 42. This is a game where you are tasked with solving a number of programming-related puzzles to solve assignments which will lead you to achieve the final mission. The game's overall story is also described in the first scene of the game, and instructions are also ongoing throughout the game. The basic narrative is that you find yourself in a futuristic research lab and you need to instruct your bot on how to solve specific assignments to deal with problems and challenges that seem to be present in the lab.

The game can take up to a few hours of play. If you are playing this on your own, it is suggested that you focus on how you can make your bot follow instructions to carry out its task. If you are playing this in a team, it is suggested that you balance out your team with people having experience as gamers or as programmers. If this game is played in teams, you are required to record the number of missions you have managed to solve in 30mins. Teams will be challenging each other depending on which team solves the more missions in the given 30mins.

Game de-Briefing (post-play):

- Was it single play or did you play in a team?
- How was game play? Can you describe the game mission and the game rules?
- Did you achieve more than 1 mission?

Possible discussion questions to raise as part of a programming class

- This game uses a type of programming that is not based on a specific language. It is referred to as visual programming.
- Reflect a bit about it. What can you say about it? Is it clear? Is it confusing? What is it trying to achieve?
- What particular terms or terminology have you come across which are also part of the programming terminology?
- What is common between the two (the programming language you are familiar with and this game?)

Possible discussion questions to raise as part of a management class (including and applying to transversal skills)

- What problem were you attempting to solve during the game?
- On a scale from 1-10 with 10 being most difficult, how difficult was it to arrive to a solution and solve the mission?

Possible discussion questions to raise if game is played in teams

For academics and lecturers: It is recommended that if the game is played using teams, it is done as an in-class activity and that the activity is timed (for example teams are given 30mins to see which team solves the most missions). In addition, it is recommended that teams are set up in terms of gaming/programming experience and that the more experienced players or programmers are asked to join different groups. The following are the suggested discussion questions for team play:

- Did playing in teams help solve the game quicker or more efficiently? Why or why not?
- Reflect a bit about your game play in the team. What communication strategies did you use to help make solving missions more efficient and effective?
- Which strategies would you keep or which ones would you remove to be able to play more efficiently next time and complete as many missions as possible?

Conclusion (for academics & lecturers):

The following are a list of questions which you can make use of to evaluate the use of this game as part of your UPSKILLS course curriculum. You can use a journal to keep track of your answers.

- How did this game serve your objective during the UPSKILLS course?
- If you were to use it again, would you change anything in the way the game was inserted in the course program and curriculum?
- Would you have the students play this game once, or play it continuously as part of the ongoing course development?