

# Assignment 1: Simple Adventure Game

**Due<sup>1</sup>: Week 6 Friday (10/03/2024, 11:59pm Irish time)**

**Mark: 30%**

## 1 Objective

The objective of this assignment is to develop your proficiency in programming by creating a program that. This assignment encourages you to apply basic programming knowledge and skills learned so far in this module, including user input, regular expressions, conditional statements, looping structures, and string manipulation in Java.

## 2 Submission Guidelines

Submission will be soft copy only on **Brightspace**.

- One submission per group.
- One file - JAVA file.
- Submit by Sunday Week 6 (11:59pm Irish time).

## 3 Documents to Submit

You should submit **ONE** file called **ID#\_Assignment1.java**. Replace the **ID#** with your student ID number. For example:

if you are student number is 12345678, then you should name your file:  
**ID12345678\_Assignment1.java**

In the file you should include prominent comments at the start of the code that contain your ID number and your Name. Brightspace will accept any filenames you use but you must adhere to the file naming requirements specified here.

**NOTE:** You will lose marks if you do not adhere to the submission requirements.  
Late submissions will be penalised.

---

<sup>1</sup> Early submission of drafts is advised. The Brightspace assignment will be set up to accept unlimited submissions, however, each one will overwrite the previous one. So each submission must be a single file.

## 4 Assignment Specification: What You Need to Do

You are tasked with creating a simple dialog-based adventure game where the player navigates through a series of scenarios by making choices presented as dialog box options. Each scenario will present the player with multiple choices, each leading to different outcomes and subsequent scenarios. Player's choices will determine the progression and ending of the game.

There are 4 scenarios in total in the game. The execution of these scenarios depends on the user's choices. The game flow including its 4 scenarios, choices and outcomes are shown in Figure 1 below.

### 4.1 Start

The game starts by showing a dialog box to the user. The box should say "Welcome to the adventure game, please choose a name for your player". You should read user's name and use it during the game.

A username should contain only alphanumeric and/or underscore characters, with a minimum length of 2 and a maximum length of 15.

For example, acceptable usernames include: ABC\_1, 2B, \_aG.

Unacceptable usernames include: A?G, B-b, a, ab cd

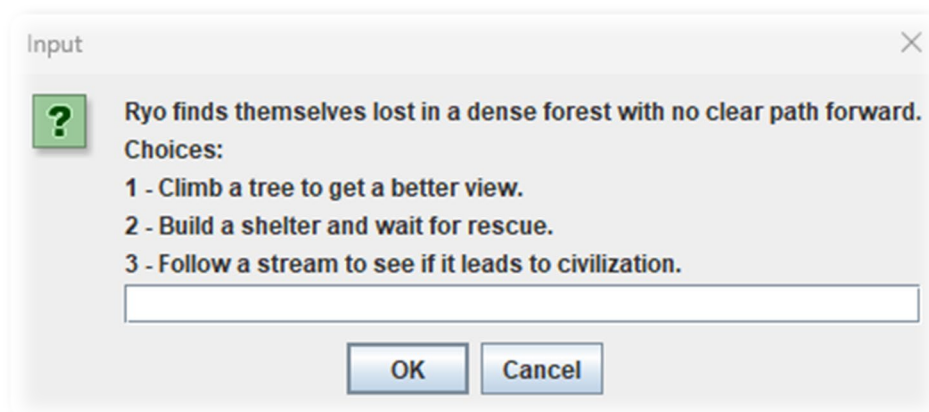
If the user enters an invalid username, the game should show the user a dialog indicating the criteria to accept a username. It can, for example, show the following message:

"Incorrect username. Please enter a username that:

Contains only alphanumeric and/or underscore characters, with a minimum length of 2 and a maximum length of 15."

### 4.2 Scenario 1

The first message displayed to the user is "[**Player Name**]" (this means you need to put the player name read from the **Start** step) finds themselves lost in a dense forest with no clear path forward." With three options below it. See box below – in these examples the player name is assumed to be **Ryo**. User input should be an integer within the range.



After the user inputs a correct number, the game should display a dialog box indicating an **outcome**. In this case, the outcome for all the choices is a message saying “**Ryo** continues their journey”.

#### 4.3 Scenario 2

In this scenario you should display the following: “While wandering through the forest, **Ryo** encounters a mysterious stranger who offers them assistance. **Ryo** must decide whether to trust the stranger.” With the following three options: “1 - Accept the stranger's help, 2 - Politely decline and continue on your own, 3 - Confront the stranger and demand answers”.

Similar to scenario 1, the user must input an integer within the correct range (1-3).

**Choice 1 Accept the stranger’s help.** If the user selects this choice (i.e. 1), then there are two possible outcomes:

**Outcome 1 (70%):** message displayed is “**Ryo** is saved!”. There is a 70% chance of this outcome occurring.

or

**Outcome 2 (30%):** message displayed is “**Ryo** is killed!”. There is a 30% chance of this outcome occurring.

One will be executed but both outcomes will **end the game**.

**Choice 2 Politely decline and continue on your own.** If the user selects this choice (i.e. 2), then the outcome is a dialog with the message “**Ryo** continues their journey”.

**Choice 3 Confront the stranger and demand answers.** If the user selects this choice (i.e. 3), then the outcome is a dialog with the message “The stranger offers **Ryo** to play a game”.

#### 4.4 Scenario 3

A dialog with the message “**Ryo** comes across a wide river blocking their path. **Ryo** must figure out how to cross it safely” with three choices: “1- Attempt to swim across the river, 2- Look for a shallow area to wade through, 3- Build a makeshift raft to cross the river”. User input should be an integer within the range.

There are **two outcomes**:

**Outcome 1:** if the user selects **choice 1**, then the game shows a dialog to the user with the message “**Ryo** drowns!”. This is considered a [Sad End].

**Outcome 2:** if the user selects **choice 2 or 3**, then the game shows a dialog to the user with the message “**Ryo** is saved!”. This is considered a [Happy End].

#### 4.5 Scenario 4

**Guess Dialog.** A dialog is shown to the user with the message “The game is to guess the number hidden in this mysterious box. It should be between 1 and 10 (inclusive). You have

THREE tries.” The dialog should be an **input dialog**. You should ensure that the user inputs an integer in the range of 1 to 10.

There are **three possible outcomes**:

**Outcome 1 (Win)**: the user correctly guesses the number. A dialog is shown to the user saying “**Ryo** is saved!”. After that a dialog to restart the game is shown to the user.

**Outcome 2 (Lose)**: the user fails to guess the right number three times. A dialog is shown to the user saying “**Ryo** is killed!”.

**Outcome 3 (Incorrect)**: the user attempts to guess the number, but fails to guess right. A dialog is shown to the user saying “INCORRECT. You should aim [LOWER/HIGHER] in your next try, **Ryo**”. [LOWER/HIGHER] gives the user a hint for their next try, for example if they guessed 3, but the right number is 7, then the message will include HIGHER (“You should aim HIGHER...”). After that, a dialog with the same message as the **Guess Dialog** above is shown, except that the number of tries should be changed, e.g., should be TWO instead of THREE, and if one try is left, it should say “...You have ONE FINAL try left”.

#### 4.6 Play Again?

When the game reaches an end at any of the outcomes indicated above, your game should offer the user to play again. It should ask: “Do you want to play again? (Y/N)”. The user should input either Y/y to play again or N/n to end the game. If the user enters Y/y, then your game should start from the beginning. If the user enters N/n, then your game should show a goodbye dialog and finish. Otherwise, the dialog should stay up until a correct input is given.

## 5 Your Implementation

Please make sure to:

- Use meaningful variable names and include comments to explain the logic where necessary.
- Ensure that your code is well-structured and follows best practices.
  - Do not implement everything in the main or a single method.
  - Provide a meaning full structure to your methods.
- Test thoroughly.
  - Ensure there are no errors or exceptions.
  - Ensure that the game follows the flow indicated in this document (i.e. in Fig 1).



Figure 1 Figure 2 A diagram showing the flow of the game.

## 6 Assessment Criteria

The following table provides a breakdown of the marks allocated to various components of your submission.

Component	Marks
Start	5
Scenario 1	3
Scenario 2	5
Scenario 3	5
Scenario 4	6
Play Again?	6
Total	30

## 7 Need Help?

We fully expect that some of you will use Brightspace, WhatsApp, email or even talk to each other about this assignment. In fact, we would be disappointed if you didn't because it would mean that you were failing to use one of the best resources for learning that is available to you – **your peers**. If a group is trying to create a useful tool, then it would be madness if they didn't talk to each other about it and consider various approaches and advice and stories of mistakes.

Remember, the **ICT centre** provides support for students unsure about how to proceed with a programming task or why something is not working the way it is expected to. All of the people involved have in-depth knowledge of the challenges that face students and can help. So we would encourage you to use those resources if you think you need to.

However, we need to emphasise that whilst we are wholeheartedly supporting the exchange of ideas we cannot support the appropriation of code and its unattributed use in a submission. So for clarity please be aware that if you find yourself unable to tackle some part of the assignment or are finding something particularly tricky to get working, then by all means **ask us, or ask your friends, or ask someone you know who has solved the problem, for advice and guidance**, but do not just take their solution and attempt to create the impression that it is your work.

If you genuinely cannot solve some part of the assignment, then tell us and we will help you so that you get a chance to learn something. You may lose some marks but you will gain knowledge and confidence. The alternative is catastrophic. The university has severe penalties for plagiarism up to and including expulsion. More significantly, when you are applying for jobs and doing interviews it will become obvious very quickly that your knowledge level is not what you claim it is. In other words, it's just not worth it!

## 8 Sample Solution

A short while after the submission deadline, we will circulate a sample solution to the assignment. We will try to include sufficient comments to assist you with understanding it but if you have any questions about the solution please email and ask. All of the questions we receive will be anonymised and our proposed answers shared with everyone involved with the module. So as always, no one knows who asked the question and everyone gets the opportunity to benefit from the answer.

If you wish you can incorporate the sample solution into your own code if you think it helps you make it better or more understandable or neater or solves a problem in a fashion that suits your purpose. If you decide to do that **you should make it clear in subsequent submissions that you are including code from the sample solution**. You won't lose any marks for including whatever parts of the sample solution you find useful but you could lose marks for not highlighting the fact that you are including the code in your solution.

By the way, if you have submitted a better solution to the part than the one contained in the sample solution then don't be afraid to point that out to us. This is a university. Everyone is here to learn – including us. We have a lot of the answers but we don't have all of them. We will be very happy to share your innovative solution with everyone involved with the module and with your permission we would be delighted to include your name.



## 9 Note on Plagiarism and Cheating

Plagiarism and cheating are **serious academic offenses** that undermine the integrity of the learning process and the reputation of both individuals and institutions. It is crucial that all group members understand and adhere to the principles of academic honesty throughout this assignment:

1. **Original Work:** All work submitted must be original and created by the student. Copying, reusing, or paraphrasing content from external sources without proper attribution is **strictly prohibited**.
2. **Proper Citation:** If you refer to external sources, including textbooks, online materials, or classmates' work, ensure that you provide appropriate citations. Failure to do so can be **considered plagiarism**.
3. **Collaboration vs. Copying:** Collaboration between students is encouraged and expected, but there is a clear distinction between collaborative work and copying. Collaboration involves sharing ideas and discussing concepts to enhance understanding. Copying involves duplicating another person's work or ideas without adding your own input.

**Consequences for plagiarism and cheating may include a failing grade for the assignment, academic probation, or more severe disciplinary actions as per university policies.** Please take this note seriously and maintain the highest standards of academic integrity in your work. If you have any questions or uncertainties about what constitutes plagiarism or cheating, seek clarification from your instructor before submitting your assignment.