



# Unity Developer Position Assessment

Mathias Wagner Nielsen  
Simonas Holcmann

May 19, 2021

## 1 Purpose of this document

This is the specification for the Unity Developer Position Assessment. The goal of this document is to facilitate a proper assessment of the applicant's ability to contribute to our Unity Development team. This document is intended as a suggestion for how to demonstrate your fitness for the project to us. If you want to demonstrate it in a better way, you're free to do so, but keep in mind that we have included these items for a reason.

## 2 Rock, Paper, Scissors

The task that we want you to complete is to write the code for a Rock, Paper, Scissors game. We have provided a base repository that you must start from. This base repository includes several packages:

- Zenject
- UniRx
- UniTask

[Zenject](#) is a framework for dependency injection in Unity. It is used extensively in our Unity software. The main goal of using Zenject is to facilitate writing robust, testable code. We have written a Zenject installer which binds the UI components in the scene to different classes that we call **Presenters** (e.g. `PlayerInputPresenter`).

[UniRx](#) is Reactive Extensions (Rx) ported to Unity. One pattern that could be useful here is the [MVRP pattern](#).

[UniTask](#) is a package providing better integration for async/await in Unity. Asynchronous operations with a return value are best represented with a **UniTask** (e.g. HTTP requests).



Figure 1: The player must choose a valid symbol (rock, paper, or scissors)

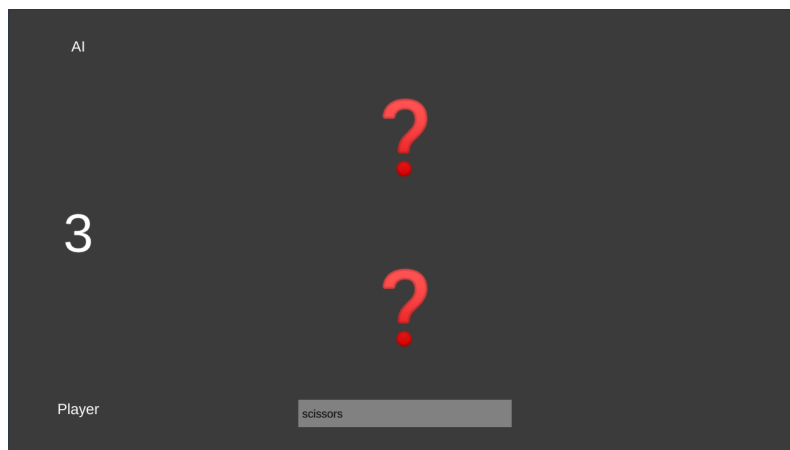


Figure 2: A countdown is displayed while the AI chooses its move

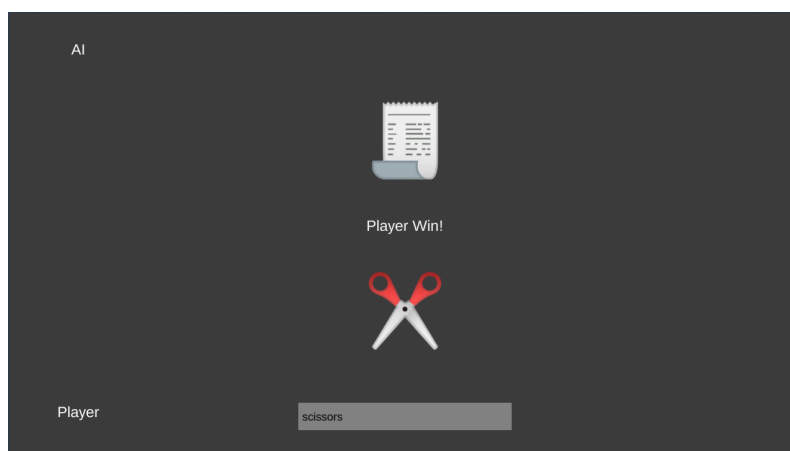


Figure 3: The player wins

### **The flow of the application:**

1. The user inputs one of these values ['rock','paper','scissors'] into the input field.
2. When the user presses 'Enter' on their keyboard, the answer is submitted, and the Player has locked in their chosen symbol. The program decides what symbol should be shown, based on the input. (However, does not show it yet).
3. The AI sends an HTTP request to this website: <http://www.randomnumberapi.com/api/v1.0/random?min=0&max=2&count=1>, and retrieves a random unsigned integer, from 0 (inclusive) to 2 (inclusive) ([0,1,2]).
4. Based on the integer, the program chooses one of the three symbols. (However, does not show it yet).
5. A countdown from 3 to 0 starts. When the timer hits 0, the question mark symbols turn into the respective rock, paper, scissors symbols for both players.
6. After the symbols are shown, there is a 0.5 second delay, and the program announces who won the match.  
If the Player wins, the 'Result Text' should state 'Player Win!'.  
If the AI wins, the 'Result Text' should state 'AI Win!'.  
If it's a tie, the 'Result Text' should state 'Tie!'.
7. The timer, input field value, and the rock paper scissors symbols get reset to their initial values (timer to being disabled, input field to empty, symbol image to question mark sprite).

### **We need to test the application. Write unit tests that prove that:**

1. Input values ['rock','paper','scissors'] return the correct symbol.
2. Matching two symbol values gives a correct output, for example, rock beats scissors, rock is equal to rock, paper beats scissors, etc.
3. The HTTP request made to the [www.randomnumberapi.com](http://www.randomnumberapi.com) returns only values from 0 (inclusive) to 2 (inclusive).