

Unity Dev Test Assignment

The Test Assignment aims to:

- Acquaint the hiring party with the potential employee's preferred coding style and practices
- Assess the potential employee's program architecture approach
- Assess the potential employee's ability to work with requirements to achieve business goals

Snake Game

You are asked to implement a simple “[Snake](#)” game with the following requirements:

- The game can be visually simple (e.g. boxes or squares) however has to be visually readable:
 - a. Elements should be visible (e.g. avoid typing black text on black background);
 - b. Colors should not be distracting (e.g. avoid “toxic” neon colors); etc.
- “Edible elements” should be randomly generated on the game field over time and should have several possible effects:
 - a. Increase the snake length by 1;
 - b. Decrease the snake length by 1;
 - c. Temporarily slow down the snake movements;
 - d. Temporarily speed up the snake movements;
 - e. Reverse the snake (head becomes tail, tail becomes head, the movement direction reverses as well);
- The field should be “looped” — e.g. by moving through the right border of the game field the snake should continue its movement from the left border of the game. Top and bottom borders should follow the same logic.
- When the snake collides with itself the game restarts.
- There should be a score somewhere on the screen displaying how many “edible elements” the snake has “eaten” over the current game round.

Context & Assumptions

- Freely add more rules and / or assumptions to the implementation as long as the core requirements are fulfilled and all significant rules and assumptions are documented (in any format you prefer).
- Assume that the game is a PoC / UX demo that should be rapidly delivered (if it takes a full day or longer to implement - you are most likely over-engineering).
- Assume that the purpose of the game is to display the gameplay/ux, not the visual design.
- NB! Assume that the code and the game will evolve later on in a series of rapid gameplay prototypes (which is not included in the scope of this task). The potential client's possible wishes to experiment with numeric parameters ("edible element" rates and probabilities, effect durations and severities, game speed, etc.) or the chance of them asking for new effect types.
- Assume that the core snake game mechanics will remain unchanged and will never lose its initial concept or transform into any other type of game (e.g. Tetris or Slots) - in its core it will definitely stay a snake game no matter what.
- Assume that you start with one target platform (of your choice), but will want to port it to a variety of other platforms later on.

You are welcomed to ask for missing details, but you are encouraged to make decisions yourself and document them as you do.

