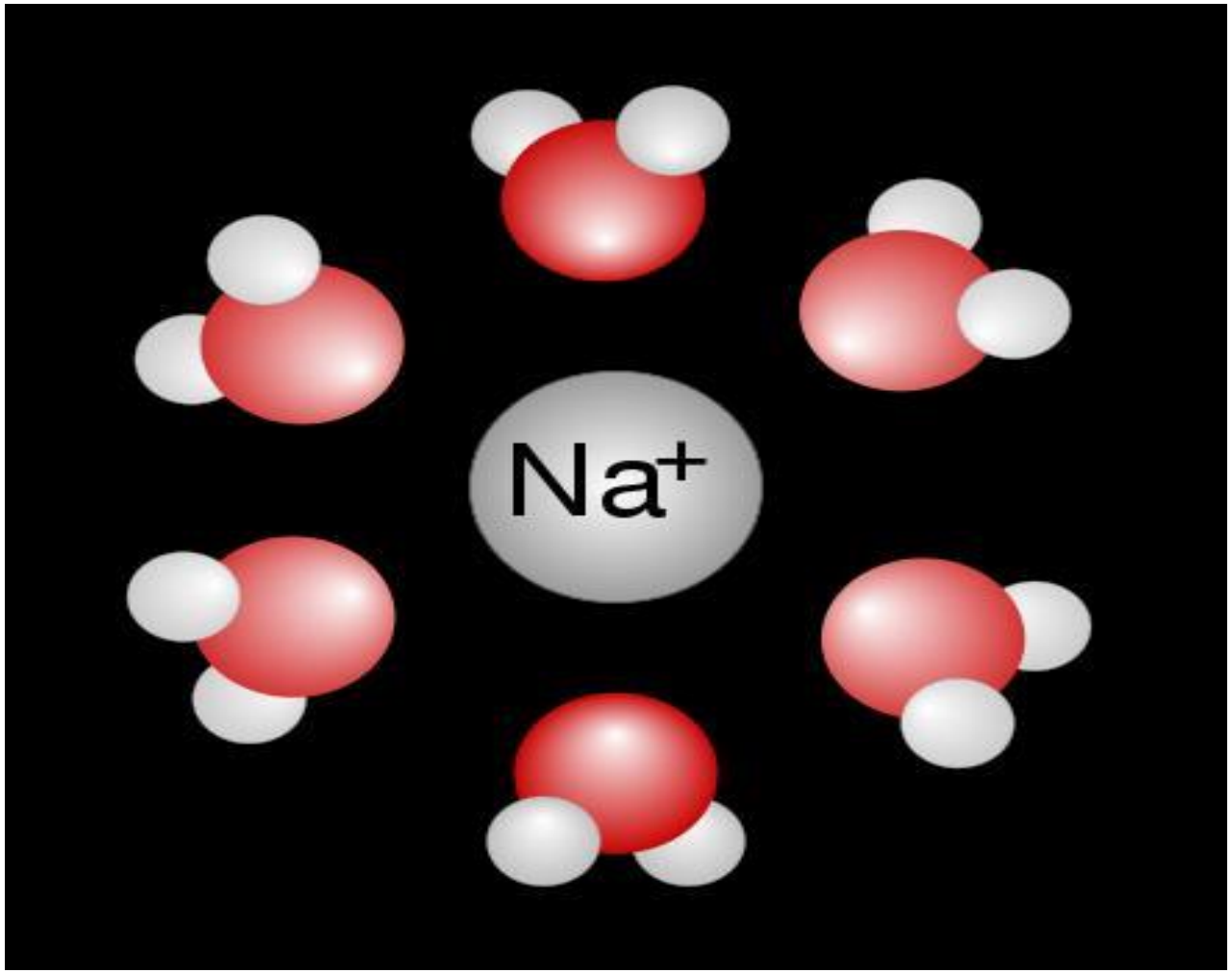
Project development Phase

No.of functional Features Includes the solution:

- An aqueous solution is a solution in which the solvent is water. It is mostly shown in chemical equations by appending (aq) to the relevant chemical formula. For example, a solution of table salt, also known as sodium chloride (NaCl), in water would be represented as $Na^+(aq) + Cl^-(aq)$. The word aqueous (which comes from aqua) means pertaining to, related to, similar to, or dissolved in, water.[1] As water is an excellent solvent and is also naturally abundant, it is a ubiquitous solvent in chemistry. Since water is frequently used as the solvent in experiments, the word solution refers to an aqueous solution, unless the solvent is specified.[2]


Na^+

Code- Layout, Readability, Reusability:

- Code readability is one of the first factors a developer learns, making it a quality one should always master. It merely means writing and presenting your code in such a manner that it can be easily read and understood. Much easier said than done, but it is as important as solving the problem.

Utilisation Of Algorithms:

- In his opinion, the use of algorithms is likely to improve accuracy across a wide range of settings because algorithms will reduce bias and noise. But in important cases, algorithms struggle to make accurate predictions, not because they are algorithms but because they need the necessary data.

Debugging & Traceability:

r.Water.WaterMesh.Enabled

Sets whether the water mesh should be rendered or not. This affects both rendering and the water mesh tile generation.

r.Water.WaterMesh.ShowWireframeAt

Exception Handling:

- Exceptions are the fundamental way how Elements deals with errors, in a consistent way that works the same across all platforms, and works similarly for all languages.
- Exceptions happen asynchronously from the regular application flow. When an error occurs, an exception is “raised” (in Oxygene parlance) or “thrown” (in C#, Swift and Java parlance), and it interups the current execution flow at that point. The exception travels back up the call stack of methods, until it is handled (or “caught”) by an exception handler, or until it reaches the top of stack.
- Exceptions can be caught explicitly, by code in your project, or implicitly, by the surrounding platform or application framework (such as Cocoa, WinForms, or WPF) that provides the execution context for your app
- If an exception reaches the top of its stack, uncaught, it will usually result in termination (or “crash”) of your application.