# Library of games in Vue

Hey Teks or future Teks, welcome to this new session of introduction to code presented to you by Epitech Coding Club! Today we are diving into Web development, more specifically the Vue.js framework.

---

## About Vue.js:

Vue.js is a **progressive JavaScript framework** used to build user interfaces and single-page applications (SPAs). It is designed to be **lightweight, easy to learn**, and **highly flexible**. Vue is particularly known for its **reactivity system** and its ability to scale from small projects to complex web applications.

### Why use Vue.js?

✅ **Simple & Easy to Learn** – Vue has a gentle learning curve, making it beginner-friendly.
✅ **Reactive Data Binding** – The UI updates automatically when data changes.
✅ **Component-Based Architecture** – Applications can be structured using reusable components.
✅ **Lightweight & Fast** – Vue is smaller and faster than many other frameworks.
✅ **Great Ecosystem** – Includes Vue Router for navigation and Vuex (or Pinia) for state management.

### Vue.js in the World of Work 🌍💼

Vue.js is widely used in the **tech industry**, especially for building modern web applications. It is favored by startups, large companies, and freelancers due to its simplicity, flexibility, and performance.

---

# Getting Started with Vue.js 🚀

## Introduction

For each exercise, you will complete and enhance simple **games** built with **Vue.js**. These exercises will introduce you to the key aspects of Vue.js programming, including **reactivity, components, and event handling**.

---

# Exercises

You will work on three different mini-games, each demonstrating a key concept of Vue.js:

## 1. Clicker Game 🖱️

The first game allows users to earn points by clicking a button and purchasing upgrades to increase their point gain per click.

### Objectives:

✅ Understand Vue.js reactivity and data binding.
✅ Use Vue directives ( `@click` , `:disabled` ) to handle interactions.
✅ Implement a simple state persistence mechanism using `localStorage` .
✅ Modify and extend a Vue.js application to add new features.

### Tasks:

- Fix the missing parts of the code to ensure upgrades work correctly.
- Enhance the game by adding new features:
  - A **reset button** to restart the game.
  - Additional upgrade levels that increase the click value even further.
  - A visual effect or animation when clicking the button.
  - A progress bar showing the next upgrade milestone.
- Ensure data persistence so progress is saved even after refreshing the page.

### Bonus Challenge 🎯

- Add **auto-clickers** that generate points over time.
- Introduce **different themes or styles** for a better user experience.
- Implementation of a router.

## 2. Rock Paper Scissors ✊✋✌️

The second game is a simple **Rock Paper Scissors** game where the player competes against the computer.

### Objectives:

✅ Learn about Vue.js computed properties and methods.
✅ Manage user input and display dynamic results.
✅ Use Vue's event handling ( `@click` ) to process choices.

### Tasks:

- Implement a Rock Paper Scissors game where the player selects an option and the computer randomly picks one.
- Display the winner based on the classic game rules.
- Keep track of the player's score and allow them to reset it.

### Bonus Challenge 🎯

- Add a **best of 5 mode** where the first to 3 wins is declared the ultimate winner.
- Implement different animations based on the winner.
- Implementation of a router.

---

## 3. Guess the Number 🔢

The third game challenges the player to guess a randomly generated number within a given range.

### Objectives:

✅ Implement two-way data binding with Vue's `v-model` .
✅ Use conditional rendering to show hints and results.
✅ Manage game state and allow resetting.

### Tasks:

- Generate a random number and prompt the player to guess it.
- Provide hints ("Too high!" or "Too low!") after each guess.

- Display a success message when the player guesses correctly.
- Allow the player to restart the game at any time.

### Bonus Challenge 🎯

- Add difficulty levels (easy, medium, hard) with different number ranges.
- Implement a guess counter and show the player’s performance.
- Implementation of a router.

---

## Final Notes

These exercises will help you grasp the **fundamentals of Vue.js** while building interactive web applications. By the end of this session, you will have a solid understanding of **Vue components, data binding, event handling, and reactivity**.

Good luck, and happy coding! 🚀[[]]