# Lab 03 – React

#### **Creating a React Project for Games**

Today, we will create a new React project and build several simple components related to video games. We will focus on **component state and adding custom styles**. This project will be developed throughout the semester to create a larger, more complex application.

### 1. Initializing a New Project

Let's start by creating a new React project following the instructions from the previous class.

- 1. Open WebStorm.
- 2. Click Create New Project.
- 3. Select React App.
- 4. Choose the project location.
- 5. Set the project name: games-project.
- Click Create.
   Make sure WebStorm automatically runs: npx create-react-app games-project
- 7. In the terminal, enter: cd games-project
- 8. To run the application, type: npm start

Now, our application should be available at localhost:3000. We should see the default React starter project in the browser.

#### 2. Project Structure

In the src/ directory, create a folder called components/, where we will store our components. Add a components/ folder inside src/.

By the end of today's class, our project structure should look like this:

src/

- |-- components/
- | |-- Header.js
- | |-- GameList.js
- | |-- LoadMore.js
- | |-- GameCard.js
- | |-- Footer.js
- | |-- FavoriteGames.js
- | |-- ThemeToggle.js
- | |-- RandomGame.js

In the src/ directory, also create a **styles**/ folder where we will store component styles.

Now, let's move on to creating the necessary components!

# 3. Creating the Header Component

We will start by creating a **header component** for our application. This element will contain the project name and a welcome message for visitors.

Create a new file: Header.js inside the components/ folder.

# Also create a file:

Header.css in the styles/ folder for styling.

## components/Header.js

Now, place the Header component inside App.js.

## styles/Header.css

```
.header {
```

```
background: #282c34;
```

color: white;

```
padding: 20px;
```

```
text-align: center;
```

```
}
```

Finally, add an **import statement** for Header.css in Header.js. import '../styles/Header.css';

Check how the styled component looks in the browser.

## 4. Creating the GameCard Component

Now, let's create a component that **receives a single game as a prop** and displays its **title and genre**.

Create a new file: GameCard.js inside the components/ folder.

```
};
```

```
export default GameCard;
```

Now, create a new file: GameCard.css inside the styles/ folder.

```
.game-card {
border: 1px solid #ddd;
padding: 10px;
margin: 10px;
border-radius: 5px;
background-color: #f9f9f9;
transition: 0.3s;
```

```
}
```

```
.game-card:hover {
    background-color: #e0e0e0;
}
```

Import the CSS file inside GameCard.js.

## 5. Creating the GameList Component

Now, we will create a **component that displays a list of games** using GameCard. The list of games will be stored in a separate file and **passed as a prop** to GameList.

Create a new file: GameList.js inside the components/ folder.

### components/GameList.js

Create a new file: GamesData.js inside the src/ directory.

const gamesData = [

{ id: 1, title: 'The Witcher 3', genre: 'RPG' },

{ id: 2, title: 'Cyberpunk 2077', genre: 'Sci-Fi RPG' },

{ id: 3, title: 'Dark Souls III', genre: 'Action RPG' },

];

export default gamesData;

Now, place the GameList component inside App.js and pass gamesData as a prop.

# 6. Creating the "Load More" Button

Now, let's create a component that **allows users to load more games** into the list. The games will be loaded **after clicking the button**.

This component **changes its behavior based on state** (useState). In the example below:

const [loaded, setLoaded] = useState(false);

- loaded is the **state variable**.
- setLoaded is the function to update state.
- useState(false) sets the **initial state to false** the list remains hidden until the button is clicked.

Create a new file: LoadMore.js inside the components/ folder.

```
import React, { useState } from 'react';
```

import GameList from "./GameList";

const extraGames = [

{ id: 4, title: 'Elden Ring', genre: 'RPG' },

{ id: 5, title: 'Horizon Zero Dawn', genre: 'Action RPG' }

];

```
Additional Tasks:
```

- 1. Create a Footer component for the application's footer.
- Create a FavoriteGames component that displays a list of favorite games when clicked.
- 3. Create a **ThemeToggle** component that switches between two background colors.
- 4. Create a **RandomGame** component that displays a random game from the list when clicked.