Exercise: Create a Simple “Spaceship Shooting Game” in ReactJS

Prerequisites:

- Basic knowledge of JavaScript, HTML, and CSS.
- Familiarity with ReactJS basics (components, state, props).

Step 1: Setting Up the Project

1. **Install Node.js and npm**: Ensure Node.js and npm are installed on your computer. You can download them from nodejs.org.
2. **Create a New React Project**: Open your terminal and run the following command to create a new React project:

   ```
   npx create-react-app spaceship-game
   cd spaceship-game
   ```

3. **Start the Development Server**:

   ```
   npm start
   ```

Step 2: Create the Game Board

1. **Create a Game Component**: In the `src` directory, create a new file called `Game.js` and add the following code:

   ```
   import React from 'react';
   import './Game.css';

   const Game = () => {
       return (
           <div className="game-board">
               <h1>Spaceship Shooting Game</h1>
               <div className="board"></div>
           </div>
       );
   };

   export default Game;
   ```

2. **Update the App Component**: In `src/App.js`, import and render the `Game` component:

   ```
   import React from 'react';
   import Game from './Game';
   import './App.css';

   function App() {
       return (
           <div className="App">
   ```

```
            <Game />
        </div>
    );
}

export default App;
```

3. **Add CSS for the Game Board**: In `src/Game.css`, add the following styles:

```
.game-board {
    text-align: center;
}

.board {
    width: 800px;
    height: 600px;
    border: 1px solid #000;
    margin: 0 auto;
    position: relative;
    background-color: black;
}
```

Step 3: Create the Spaceship

1. **Add Spaceship State**: Update `Game.js` to include state for the spaceship:

```
import React, { useState } from 'react';
import './Game.css';

const Game = () => {
    const [spaceship, setSpaceship] = useState({ x: 390, y: 550 });

    return (
        <div className="game-board">
            <h1>Spaceship Shooting Game</h1>
            <div className="board">
                <div
                    className="spaceship"
                    style={{
                        left: `${spaceship.x}px`,
                        top: `${spaceship.y}px`,
                    }}
                ></div>
            </div>
        </div>
    );
};

export default Game;
```

2. **Add CSS for the Spaceship**: In `src/Game.css`, add the following styles:

```
.spaceship {
    width: 20px;
    height: 20px;
    background-color: blue;
    position: absolute;
}
```

Step 4: Move the Spaceship

1. **Handle Key Presses**: Update `Game.js` to handle arrow key presses and move the spaceship:

```
import React, { useState, useEffect } from 'react';
import './Game.css';

const Game = () => {
    const [spaceship, setSpaceship] = useState({ x: 390, y: 550 });

    useEffect(() => {
        const handleKeyDown = (event) => {
            setSpaceship((prevPosition) => {
                const newPosition = { ...prevPosition };
                switch (event.key) {
                    case 'ArrowLeft':
                        newPosition.x = Math.max(0, prevPosition.x - 10);
                        break;
                    case 'ArrowRight':
                        newPosition.x = Math.min(780, prevPosition.x + 10);
                        break;
                    default:
                        break;
                }
                return newPosition;
            });
        };

        document.addEventListener('keydown', handleKeyDown);
        return () => {
            document.removeEventListener('keydown', handleKeyDown);
        };
    }, []);

    return (
        <div className="game-board">
            <h1>Spaceship Shooting Game</h1>
            <div className="board">
                <div
                    className="spaceship"
                    style={{
                        left: `${spaceship.x}px`,
                        top: `${spaceship.y}px`,
                    }}
                ></div>
            </div>
        </div>
    );
};

export default Game;
```

Step 5: Shoot Projectiles

1. **Add Projectiles State**: Update `Game.js` to include state for projectiles:

```
import React, { useState, useEffect } from 'react';
import './Game.css';

const Game = () => {
    const [spaceship, setSpaceship] = useState({ x: 390, y: 550 });
    const [projectiles, setProjectiles] = useState([]);

    useEffect(() => {
        const handleKeyDown = (event) => {
            if (event.key === ' ') {
                setProjectiles((prevProjectiles) => [
                    ...prevProjectiles,
                    { x: spaceship.x + 10, y: spaceship.y - 10 },
                ]);
            }
            setSpaceship((prevPosition) => {
                const newPosition = { ...prevPosition };
                switch (event.key) {
                    case 'ArrowLeft':
                        newPosition.x = Math.max(0, prevPosition.x - 10);
                        break;
                    case 'ArrowRight':
                        newPosition.x = Math.min(780, prevPosition.x + 10);
                        break;
                    default:
                        break;
                }
                return newPosition;
            });
        };

        document.addEventListener('keydown', handleKeyDown);

        return () => {
            document.removeEventListener('keydown', handleKeyDown);
        };
    }, [spaceship]);

    useEffect(() => {
        const moveProjectiles = () => {
            setProjectiles((prevProjectiles) =>
                prevProjectiles
                    .map((projectile) => ({ ...projectile, y: projectile.y - 10 }))
                    .filter((projectile) => projectile.y > 0)
            );
        };

        const intervalId = setInterval(moveProjectiles, 100);
        return () => clearInterval(intervalId);
    }, []);

    return (
        <div className="game-board">
            <h1>Spaceship Shooting Game</h1>
            <div className="board">
                <div
                    className="spaceship"
                    style={{
                        left: `${spaceship.x}px`,
                        top: `${spaceship.y}px`,
                    }}
                ></div>
                {projectiles.map((projectile, index) => (
                    <div
                        key={index}
                        className="projectile"
                        style={{
                            left: `${projectile.x}px`,
                            top: `${projectile.y}px`,
                        }}
                    ></div>
                ))}
            </div>
        </div>
    );
};

export default Game;
```

2. **Add CSS for Projectiles**: In `src/Game.css`, add the following styles:

```
.projectile {
    width: 5px;
    height: 10px;
    background-color: red;
    position: absolute;
}
```

Step 6: Add Targets

1. **Add Targets State**: Update `Game.js` to include state for targets:

```
import React, { useState, useEffect } from 'react';
import './Game.css';

const getRandomPosition = () => ({
    x: Math.floor(Math.random() * 780),
    y: Math.floor(Math.random() * 200),
});

const Game = () => {
```

```
const [spaceship, setSpaceship] = useState({ x: 390, y: 550 });
const [projectiles, setProjectiles] = useState([]);
const [targets, setTargets] = useState([getRandomPosition(), getRandomPosition(), getRandomPosition()]);

useEffect(() => {
    const handleKeyDown = (event) => {
        if (event.key === ' ') {
            setProjectiles((prevProjectiles) => [
                ...prevProjectiles,
                { x: spaceship.x + 10, y: spaceship.y - 10 },
            ]);
        }
        setSpaceship((prevPosition) => {
            const newPosition = { ...prevPosition };
            switch (event.key) {
                case 'ArrowLeft':
                    newPosition.x = Math.max(0, prevPosition.x - 10);
                    break;
                case 'ArrowRight':
                    newPosition.x = Math.min(780, prevPosition.x + 10);
                    break;
                default:
                    break;
            }
            return newPosition;
        });
    };

    document.addEventListener('keydown', handleKeyDown);

    return () => {
        document.removeEventListener('keydown

        handleKeyDown);
    };
}, [spaceship]);

useEffect(() => {
    const moveProjectiles = () => {
        setProjectiles((prevProjectiles) =>
            prevProjectiles
                .map((projectile) => ({ ...projectile, y: projectile.y - 10 }))
                .filter((projectile) => projectile.y > 0)
        );
    };

    const intervalId = setInterval(moveProjectiles, 100);
    return () => clearInterval(intervalId);
}, []);

return (
    <div className="game-board">
        <h1>Spaceship Shooting Game</h1>
        <div className="board">
            <div
                className="spaceship"
                style={{
                    left: `${spaceship.x}px`,
                    top: `${spaceship.y}px`,
                }}
            ></div>
            {projectiles.map((projectile, index) => (
                <div
                    key={index}
                    className="projectile"
                    style={{
                        left: `${projectile.x}px`,
                        top: `${projectile.y}px`,
                    }}
                ></div>
            ))}
            {targets.map((target, index) => (
                <div
                    key={index}
                    className="target"
                    style={{
                        left: `${target.x}px`,
                        top: `${target.y}px`,
                    }}
                ></div>
            ))}
        </div>
    </div>
);
```

```
};

export default Game;
```

2. **Add CSS for Targets**: In `src/Game.css`, add the following styles:

```
.target {
    width: 20px;
    height: 20px;
    background-color: green;
    position: absolute;
}
```

Step 7: Collision Detection

1. **Detect Collisions**: Update `Game.js` to check for collisions between projectiles and targets:

```
import React, { useState, useEffect } from 'react';
import './Game.css';

const getRandomPosition = () => ({
    x: Math.floor(Math.random() * 780),
    y: Math.floor(Math.random() * 200),
});

const Game = () => {
    const [spaceship, setSpaceship] = useState({ x: 390, y: 550 });
    const [projectiles, setProjectiles] = useState([]);
    const [targets, setTargets] = useState([getRandomPosition(), getRandomPosition(), getRandomPosition()]);

    useEffect(() => {
        const handleKeyDown = (event) => {
            if (event.key === ' ') {
                setProjectiles((prevProjectiles) => [
                    ...prevProjectiles,
                    { x: spaceship.x + 10, y: spaceship.y - 10 },
                ]);
            }
            setSpaceship((prevPosition) => {
                const newPosition = { ...prevPosition };
                switch (event.key) {
                    case 'ArrowLeft':
                        newPosition.x = Math.max(0, prevPosition.x - 10);
                        break;
                    case 'ArrowRight':
                        newPosition.x = Math.min(780, prevPosition.x + 10);
                        break;
                    default:
                        break;
                }
                return newPosition;
            });
        };

        document.addEventListener('keydown', handleKeyDown);

        return () => {
            document.removeEventListener('keydown', handleKeyDown);
        };
    }, [spaceship]);

    useEffect(() => {
        const moveProjectiles = () => {
            setProjectiles((prevProjectiles) =>
                prevProjectiles
                    .map((projectile) => ({ ...projectile, y: projectile.y - 10 }))
                    .filter((projectile) => projectile.y > 0)
            );
        };

        const intervalId = setInterval(moveProjectiles, 100);
        return () => clearInterval(intervalId);
    }, []);

    useEffect(() => {
        const detectCollisions = () => {
            setProjectiles((prevProjectiles) => {
                return prevProjectiles.filter((projectile) => {
                    return !targets.some((target, targetIndex) => {
                        const isHit = projectile.x >= target.x && projectile.x <= target.x + 20 &&
                                      projectile.y >= target.y && projectile.y <= target.y + 20;
                        if (isHit) {
                            setTargets((prevTargets) => prevTargets.filter((_, index) => index !== targetIndex));
                            return true;
                        }
                        return false;
                    });
                });
            });
        };

        const intervalId = setInterval(detectCollisions, 50);
        return () => clearInterval(intervalId);
    }, [targets]);

    return (
        <div className="game-board">
            <h1>Spaceship Shooting Game</h1>
            <div className="board">
                <div
                    className="spaceship"
                    style={{
                        left: `${spaceship.x}px`,
                        top: `${spaceship.y}px`,
                    }}
                ></div>
                {projectiles.map((projectile, index) => (
                    <div
                        key={index}
                        className="projectile"
                        style={{
                            left: `${projectile.x}px`,
                            top: `${projectile.y}px`,
                        }}
                    ></div>
                ))}
                {targets.map((target, index) => (
                    <div
                        key={index}
                        className="target"
                        style={{
                            left: `${target.x}px`,
                            top: `${target.y}px`,
                        }}
                    ></div>
                ))}
            </div>
        </div>
    );
};

export default Game;
```

Bonus Challenges

1. **Add More Targets**:
   - Add more targets to the game and make them move randomly.
2. **Add Scoring**:
   - Add a score counter that increments every time a target is hit.
3. **Game Over Condition**:
   - Add a game-over condition if the spaceship collides with a target.
4. **Improve Graphics**:
   - Use images for the spaceship, projectiles, and targets to make the game more visually appealing.