สมาชิกในกลุ่ม

6421653006 นางสาวนภัส เจนพรมราช หมู่ 800

6421650384 นายธนดล แซ่เหล่ม หมู่ 800

โค้ด JavaScript นี้ใช้สร้างเกม เกมทายอะไรเอ่ยย ENG

โดยมีฟังก์ชั่นหลักต่าง ๆ โดยเริ่มจาก  
1. สร้างตัวแปรสำหรับใข้เก็บค่าต่างๆ

* Word : เก็บคำที่จะต้องทายบรรทัดนี้เป็นการสร้างตัวแปร `word` โดยให้มีค่าเป็นคำที่สุ่มมาจากรายการคำที่อยู่ในตัวแปร `words`

`words[Math.floor(Math.random() \* words.length)]` : ด้วยค่าที่ได้จาก `words` นี้จะได้ index สุ่มที่ใช้ในการเข้าถึงคำในอาร์เรย์ `words` สุ่ม index นี้และใช้เพื่อกำหนดคำในตัวแปร `word`

* guessedLetters : เก็บตัวอักษรที่ผู้เล่นทายแล้ว
* guessesLeft : เก็บจำนวนครั้งที่ผู้เล่นสามารถทายผิดได้อีก
* message : ข้อความสถานะเกม

2. `displayWord` : เป็นตัวแปรที่ใช้สำหรับเก็บคำที่จะแสดงบนหน้าเว็บ ถ้าตัวอักษรนั้นถูกทายแล้วจะแสดงตัวอักษร ถ้ายังไม่ถูกทายจะแสดง "\_"

- ลูปผ่านทุกตัวอักษรในคำ `word` และตรวจสอบว่าถูกทายหรือไม่ แล้วกำหนดค่าให้ `displayWord` ตามนั้น

- ใช้ `document.getElementById("word").innerHTML` เพื่อแสดงค่า `displayWord` ลงใน HTML element ที่มี `id="word"`

3. `updateLetters()` : ฟังก์ชั่นนี้ใช้แสดงตัวอักษรที่ผู้เล่นทายแล้วและแสดงบนหน้าเว็บ

- `letters`: เป็นตัวแปรที่ใช้สำหรับเก็บตัวอักษรที่ถูกทายแล้ว

- ลูปผ่านทุกตัวอักษรใน `guessedLetters` และนำมาเพิ่มเข้าไปใน `letters`

- ใช้ `document.getElementById("letters").innerHTML` เพื่อแสดงค่า `letters` ลงใน HTML element ที่มี `id="letters"`

4. `updateMessage()` : ฟังก์ชั่นนี้ใช้สร้างและแสดงข้อความสถานะเกม เช่น "You won!", "You lost! The word was ..." หรือ "Guesses left: ..."

- ตรวจสอบเงื่อนไขว่าผู้เล่นชนะหรือแพ้ แล้วกำหนดข้อความที่เกี่ยวข้อง

- ใช้ `document.getElementById("message").innerHTML` เพื่อแสดงข้อความลงใน HTML element ที่มี `id="message"`

5. `guessLetter(letter)` : ฟังก์ชั่นนี้ใช้ในการปรับปรุงข้อมูลเมื่อผู้เล่นทายตัวอักษร รวมถึงตรวจสอบว่าผู้เล่นชนะหรือแพ้

- ตรวจสอบว่าตัวอักษรที่ทายยังไม่ถูกทายมาก่อนหรือไม่ ถ้าไม่เพิ่มเข้าไปใน `guessedLetters` และลดจำนวน `guessesLeft` ถ้าไม่ถูก.

- เรียกใช้ฟังก์ชัน `updateWord()`, `updateLetters()`, และ `updateMessage()` เพื่อปรับปรุงหน้าเว็บ.

- ตรวจสอบว่าเกมจบหรือไม่ ถ้าจบให้แสดงปุ่ม "Play Again"

6. `resetGame()` : ฟังก์ชั่นนี้ใช้ในการรีเซ็ตเกม เริ่มเกมใหม่

- ทำการรีเซ็ตตัวแปรทั้งหมด เลือกคำใหม่, ล้าง `guessedLetters`, ตั้งค่า `guessesLeft` เป็น 6 และล้างข้อความและแสดงผลใน HTML element ที่เกี่ยวข้อง

- ซ่อนปุ่ม "Play Again"

7. `Event Listener` : ทำการเรียกฟังก์ชั่น guessLetter() เมื่อมีการกดปุ่มบนคีย์บอร์ด

- เมื่อมีการกดปุ่มบนคีย์บอร์ดจะเรียกใช้ `guessLetter()` โดยส่งตัวอักษรที่ถูกกดเป็นพารามิเตอร์

8. เรียกใช้ฟังก์ชัน `resetGame()` :

- เมื่อหน้าเว็บถูกโหลดจะทำการเรียกใช้ `resetGame()` เพื่อเริ่มเกมใหม่

โค้ด CSS ที่ใช้การจัดหน้าตาของหน้าเว็บ เกมทายอะไรเอ่ยย ENG

โดยมีคำอธิบายของแต่ละส่วนดังนี้

1. `@import url("https://fonts.googleapis.com/css2?family=IBM+Plex+Sans+Thai&display=swap");`

- ใช้ `@import` เพื่อนำเข้าฟอนต์ "IBM Plex Sans Thai" จาก Google Fonts

- `&display=swap` ถูกใช้เพื่อแนะนำการใช้งานฟอนต์อย่างเฉพาะเป็นการแลกเปลี่ยน (swap) เพื่อประสิทธิภาพการแสดงผล

2. `body`

- `font-family: "IBM Plex Sans Thai", sans-serif;` : กำหนดฟอนต์ในหน้าทั้งหมดให้เป็น "IBM Plex Sans Thai" หรือหากไม่สามารถใช้ได้ให้ใช้ฟอนต์ที่ไม่มีระบุ

- `text-align: center;` : จัดให้ข้อความทั้งหมดอยู่ตรงกลาง

- `margin-top: 50px;` : กำหนดระยะห่างด้านบนของหน้าเว็บ

3. `#word`

- `font-size: 24px;` : กำหนดขนาดตัวอักษรของคำในเกม

- `margin: 20px;` : กำหนดระยะห่างรอบคำในเกม

4. `button`

- `margin: 5px;` : กำหนดระยะห่างรอบปุ่ม

- `padding: 5px;` : กำหนดพื้นที่ภายในปุ่ม

5. `#message`

- `margin: 10px;` : กำหนดระยะห่างรอบข้อความสถานะของเกม