

GitHub Copilot | VS Code

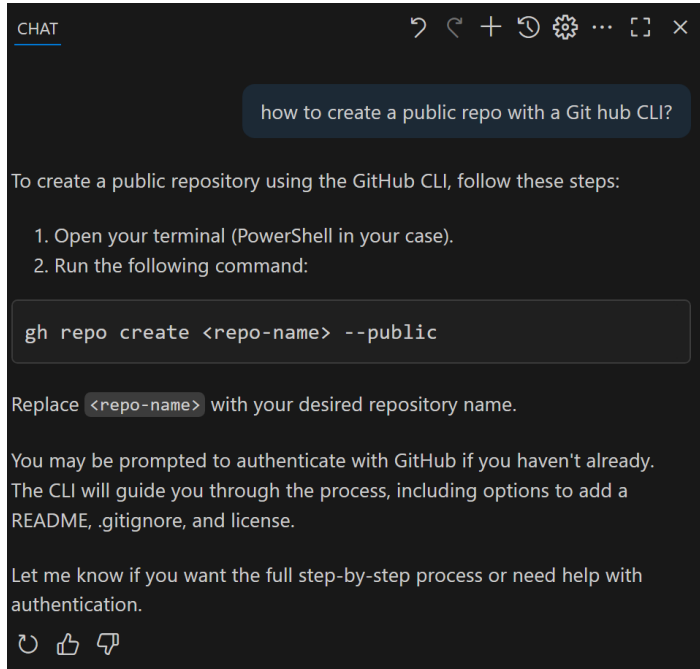
To connect to the GitHub profile from Terminal:

Prompt: `gh auth login`

To show the list of repos:

Prompt: `gh repo list`

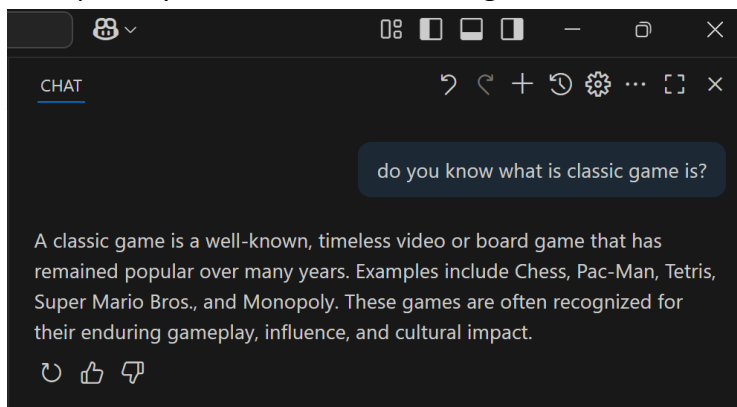
Prompt: How to create a public repo with a Git hub CLI?



Prompt: how to clone the github repo to my local directory?

Command: `git clone <repo-url>`

Prompt: do you know what is classic game is?



Prompt: can you give me the code for a classic snake game in javascript?

can you give me the code for a classic snake game in javascript?

Certainly! Here's a simple implementation of the classic Snake game using JavaScript, HTML, and Canvas. You can copy this into an `.html` file and open it in your browser.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Classic Snake Game</title>
  <style>
    canvas { background: #222; display: block; margin: 0 auto; }
    body { text-align: center; font-family: sans-serif; }
  </style>
</head>
<body>
  <h2>Classic Snake Game</h2>
  <canvas id="game" width="400" height="400"></canvas>
  <script>
    const canvas = document.getElementById('game');
    const ctx = canvas.getContext('2d');
    const gridSize = 20;
    let snake = [{ x: 200, y: 200 }];
    let direction = { x: gridSize, y: 0 };
    let food = { x: 0, y: 0 };

    if (direction.x === 0) direction = { x: -gridSize, y: 0 };
    else if (direction.x === gridSize) direction = { x: gridSize, y: 0 };
    else if (direction.y === 0) direction = { x: 0, y: -gridSize };
    else if (direction.y === gridSize) direction = { x: 0, y: gridSize };

    function gameLoop() {
      moveSnake();
      draw();
      if (!gameOver) {
        setTimeout(gameLoop, 100);
      }
    }

    placeFood();
    gameLoop();
  </script>
</body>
</html>
```

Open this file in your browser to play the Snake game. Use the arrow keys to control the snake.

After this, copy the code, paste it in a text document and save it with `.html` format & then open the file in a browser.



Prompt: can you give code for this game in a JavaScript file?

can you give code for this game in a JavaScript file?

Certainly! Here's the Snake game logic in a standalone JavaScript file. You'll need a simple HTML file to load the canvas and this script.

1. Create the HTML file (e.g., `index.html`):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Classic Snake Game</title>
  <style>
    canvas { background: #222; display: block; margin: 0 auto; }
    body { text-align: center; font-family: sans-serif; }
  </style>
</head>
<body>
  <h2>Classic Snake Game</h2>
  <canvas id="game" width="400" height="400"></canvas>
  <script src="snake.js"></script>
</body>
</html>
```

2. Create the JavaScript file (e.g., `snake.js`):

```
const canvas = document.getElementById('game');
const ctx = canvas.getContext('2d');
const gridSize = 20;
let snake = [{ x: 200, y: 200 }];
let direction = { x: gridSize, y: 0 };
let food = { x: 0, y: 0 };
let score = 0;
let gameOver = false;

function randomPosition() {
  return {
    x: Math.floor(Math.random() * (canvas.width / gridSize)),
    y: Math.floor(Math.random() * (canvas.height / gridSize))
  };
}

function placeFood() {
  food = randomPosition();
  while (snake.some(segment => segment.x === food.x && segment.y === food.y)) {
    food = randomPosition();
  }
}

function draw() {
  ctx.clearRect(0, 0, canvas.width, canvas.height);

  // Draw snake
  ctx.fillStyle = 'lime';
```

Prompt: can you provide comments for this code to make it easier to read?

```
CHAT

great! can you also provide comments in this code to make it easier to read?

# snake.js:16

> Used 1 reference

// Get the canvas and its drawing context
const canvas = document.getElementById('game');
const ctx = canvas.getContext('2d');

// Set the size of each grid cell
const gridSize = 20;

// Initialize the snake with one segment at the center
let snake = [{ x: 200, y: 200 }];

// Set the initial movement direction (moving right)
let direction = { x: gridSize, y: 0 };

// Food position
let food = { x: 0, y: 0 };

// Player's score
let score = 0;

// Game over flag
let gameOver = false;
```

Output after saving the files:



Prompt: How to create a obstacles in the game?

```
CHAT

how can i add obstacles in the game?

# index.html:16

> Used 1 reference

To add obstacles to your Snake game, follow these steps:

1. Define obstacles: Create an array to store obstacle positions.
2. Draw obstacles: Render them in the draw() function.
3. Collision detection: End the game if the snake's head collides with any obstacle.
4. Avoid placing food or obstacles on the snake or each other.

Here's how you can update your # snake.js :

// ...existing code...

// Array to hold obstacle positions
let obstacles = [];

// Function to place obstacles randomly
function placeObstacles(count = 5) {
  obstacles = [];
  while (obstacles.length < count) {
    let pos = randomPosition();
    // Avoid placing on snake or food or other obstacles
    if (
      !snake.some(segment => segment.x === pos.x && segment.y === pos.y) &&
      !food.x === pos.x && food.y === pos.y &&
      !obstacles.some(ob => ob.x === pos.x && ob.y === pos.y)
    ) {
      obstacles.push(pos);
    }
  }
}
```

Insert the new code, Update it and save it.

Output:



Prompt: How can I make the snake go through the walls without finishing the game?

```
How can I make the snake go through the walls without finishing the game?
```

Used 1 reference

To allow the snake to go through the walls (wrap around), update the wall collision logic in your `moveSnake()` function. Instead of ending the game, make the snake appear on the opposite side when it crosses a boundary.

Replace this block in `moveSnake()` :

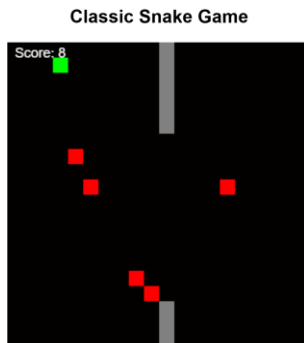
```
// ...existing code...
// Check collision with walls
if (
  head.x < 0 || head.x >= canvas.width ||
  head.y < 0 || head.y >= canvas.height
) {
  gameOver = true;
  return;
}
// ...existing code...
```

with this:

```
// ...existing code...
// Wrap snake position if it goes through walls
if (head.x < 0) head.x = canvas.width - gridSize;
else if (head.x >= canvas.width) head.x = 0;
if (head.y < 0) head.y = canvas.height - gridSize;
else if (head.y >= canvas.height) head.y = 0;
// ...existing code...
```

Now, the snake will reappear on the opposite side when it crosses a wall.

Output: the snake passes through the walls without finishing the game



Prompt: Make my game look more prettier - greet the user like Welcome Guest! & add instructions like play using arrow keys, avoid obstacles and you can escape through the walls.

Changed the snake.js code entirely

Output:

