Basic Snake HTML Game

Snake is a fun game to make as it doesn't require a lot of code (less than 100 lines with all comments removed). This is a basic implementation of the snake game, but it's missing a few things intentionally and they're left as further exploration for the reader.

Further Exploration

- Score
  - When the snake eats an apple, the score should increase by one. Use context.fillText() to display the score to the screen
- Mobile and touchscreen support
  - Allow the game to be scalled down to a phone size. See https://codepen.io/straker/pen/VazMaL
  - Support touch controls
- Better apple spawning
  - Currently the apple spawns in any random grid in the game, even if the snake is already on that spot. Improve it so it only spanws in empty grid locations

License

Other Basic Games

- Pong
- Breakout
- Tetris
- Bomberman

Support

Basic HTML Games are made possible by users like you. When you become a Patron, you get access to behind the scenes development logs, the ability to vote on which games I work on next, and early access to the next Basic HTML Game.

snake.html

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <style>
  html, body {
    height: 100%;
    margin: 0;
  }
  body {
    background: black;
    display: flex;
    align-items: center;
    justify-content: center;
  }
  canvas {
    border: 1px solid white;
  }
  </style>
</head>
<body>
<canvas width="400" height="400" id="game"></canvas>
<script>
var canvas = document.getElementById('game');
var context = canvas.getContext('2d');
var grid = 16;
var count = 0;

var snake = {
  x: 160,
  y: 160,

  // snake velocity. moves one grid length every frame in either the x or y direction
  dx: grid,
  dy: 0,
```

```
  // keep track of all grids the snake body occupies
  cells: [],

  // length of the snake. grows when eating an apple
  maxCells: 4
};
var apple = {
  x: 320,
  y: 320
};
// get random whole numbers in a specific range
// @see https://stackoverflow.com/a/1527820/2124254
function getRandomInt(min, max) {
  return Math.floor(Math.random() * (max - min)) + min;
}
// game loop
function loop() {
  requestAnimationFrame(loop);
  // slow game loop to 15 fps instead of 60 (60/15 = 4)
  if (++count < 4) {
    return;
  }
  count = 0;
  context.clearRect(0,0,canvas.width,canvas.height);
  // move snake by it's velocity
  snake.x += snake.dx;
  snake.y += snake.dy;
  // wrap snake position horizontally on edge of screen
  if (snake.x < 0) {
    snake.x = canvas.width - grid;
  }
  else if (snake.x >= canvas.width) {
    snake.x = 0;
  }

  // wrap snake position vertically on edge of screen
  if (snake.y < 0) {
    snake.y = canvas.height - grid;
  }
  else if (snake.y >= canvas.height) {
    snake.y = 0;
  }
  // keep track of where snake has been. front of the array is always the head
  snake.cells.unshift({x: snake.x, y: snake.y});
  // remove cells as we move away from them
  if (snake.cells.length > snake.maxCells) {
```

```
    snake.cells.pop();
  }
  // draw apple
  context.fillStyle = 'red';
  context.fillRect(apple.x, apple.y, grid-1, grid-1);
  // draw snake one cell at a time
  context.fillStyle = 'green';
  snake.cells.forEach(function(cell, index) {

    // drawing 1 px smaller than the grid creates a grid effect in the snake body so y
    context.fillRect(cell.x, cell.y, grid-1, grid-1);
    // snake ate apple
    if (cell.x === apple.x && cell.y === apple.y) {
      snake.maxCells++;
      // canvas is 400x400 which is 25x25 grids
      apple.x = getRandomInt(0, 25) * grid;
      apple.y = getRandomInt(0, 25) * grid;
    }
    // check collision with all cells after this one (modified bubble sort)
    for (var i = index + 1; i < snake.cells.length; i++) {

      // snake occupies same space as a body part. reset game
      if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) {
        snake.x = 160;
        snake.y = 160;
        snake.cells = [];
        snake.maxCells = 4;
        snake.dx = grid;
        snake.dy = 0;
        apple.x = getRandomInt(0, 25) * grid;
        apple.y = getRandomInt(0, 25) * grid;
      }
    }
  });
}
// listen to keyboard events to move the snake
document.addEventListener('keydown', function(e) {
  // prevent snake from backtracking on itself by checking that it's
  // not already moving on the same axis (pressing left while moving
  // left won't do anything, and pressing right while moving left
  // shouldn't let you collide with your own body)

  // left arrow key
  if (e.which === 37 && snake.dx === 0) {
    snake.dx = -grid;
    snake.dy = 0;
```

```
  }
  // up arrow key
  else if (e.which === 38 && snake.dy === 0) {
    snake.dy = -grid;
    snake.dx = 0;
  }
  // right arrow key
  else if (e.which === 39 && snake.dx === 0) {
    snake.dx = grid;
    snake.dy = 0;
  }
  // down arrow key
  else if (e.which === 40 && snake.dy === 0) {
    snake.dy = grid;
    snake.dx = 0;
  }
});
// start the game
requestAnimationFrame(loop);
</script>
</body>
</html>
```

iammanmohit
commented over 1 year ago

Wow!!! It's GREAT!!!

GandalfTheGinger
commented over 1 year ago

There is a problem when I hit the wall I don't die so could you fix that in some way and could you tell me how you fixed cause i'm curious

yotamefr
commented over 1 year ago

GandalfTheGinger it's not a bug. That's how the original game was, I'm pretty sure.

Mr-Alpheous
commented over 1 year ago

woooow wow it nice but score is missing

Tigermagooster
commented over 1 year ago

How to control?

SawyerBx
commented over 1 year ago

Could you add a way to control it on mobile?

andrewreed549
commented over 1 year ago

heres the mobile code

<title>snake</title> <style>
body {
background:blue;
}
canvas {
border:3px solid;
background:#efe;
}

.control {
position:relative;
border-radius:10px;
width:70px;
height:30px;
bottom:-30px;
}

button {
outline:0;
}
</style>

<script> var canvas = document.getElementById('game'); var context = canvas.getContext('2d'); var grid = 10; var colors = "rgb(" +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +")"; var snake = { x: 160, y: 160, dx: grid, dy: 0, cells: [], maxCells: 4 }; var count = 0; var apple = { x: 320, y: 320 }; function getRandomInt(min, max) { return Math.floor(Math.random() * (max - min)) + min; } function loop() { requestAnimationFrame(loop); if (++count < 7) { return; } count = 0;

context.clearRect(0,0,canvas.width,canvas.height); snake.x += snake.dx; snake.y += snake.dy; // wrap snake position on edge of screen if (snake.x < 0) { snake.x = canvas.width - grid; } else if (snake.x >= canvas.width) { snake.x = 0; } if (snake.y < 0) { snake.y = canvas.height - grid; } else if (snake.y >= canvas.height) { snake.y = 0; } snake.cells.unshift({x: snake.x, y: snake.y}); if (snake.cells.length > snake.maxCells) { snake.cells.pop(); } context.fillStyle = 'green'; context.fillRect(apple.x, apple.y, grid-1, grid-1) context.fillStyle = colors; snake.cells.forEach(function(cell, index) { context.fillRect(cell.x, cell.y, grid-1, grid-1); if (cell.x === apple.x && cell.y === apple.y) { snake.maxCells++; apple.x = getRandomInt(0, 30) * grid; apple.y = getRandomInt(0, 30) * grid; } for (var i = index + 1; i < snake.cells.length; i++) { if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) { alert("you lose, your score is") alert(snake.maxCells) colors = "rgb(" +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +")"; snake.x = 160; snake.y = 160; snake.cells = []; snake.maxCells = 4; snake.dx = grid; snake.dy = 0; apple.x = getRandomInt(0, 25) * grid; apple.y = getRandomInt(0, 25) * grid; } } }); } function left(){ if (snake.dx === 0) { snake.dx = -grid; snake.dy = 0; } } function up(){ if (snake.dy === 0) { snake.dy = -grid; snake.dx = 0; } } function right(){ if (snake.dx === 0) { snake.dx = grid; snake.dy = 0; } } function down(){ if (snake.dy === 0) { snake.dy = grid; snake.dx = 0; } } function hack(){ if (snake.maxCells){ snake.maxCells++; } } function hackd(){ if (snake.maxCells){ colors = "rgb(" +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +")"; } } function hacku(){ if (snake.maxCells){ apple.x = getRandomInt(0, 25) * grid; apple.y = getRandomInt(0, 25) * grid; } } document.addEventListener('keydown', function(e) { // prevent snake from backtracking on itself if (e.which === 37 && snake.dx === 0) { snake.dx = -grid; snake.dy = 0; } else if (e.which === 38 && snake.dy === 0) { snake.dy = -grid; snake.dx = 0; } else if (e.which === 39 && snake.dx === 0) { snake.dx = grid; snake.dy = 0; } else if (e.which === 40 && snake.dy === 0) { snake.dy = grid; snake.dx = 0; } requestAnimationFrame(loop); </script>

GandalfTheGinger
commented over 1 year ago

could you please add the scoring

HenryAkaya
commented over 1 year ago

> heres the mobile code

<title>snake</title> <style> body { background:blue; } canvas { border:3px solid; background:#efe; }
.control {
position:relative;
border-radius:10px;
width:70px;
height:30px;
bottom:-30px;
}

button {
outline:0;

}
</style>

<script> var canvas = document.getElementById('game'); var context = canvas.getContext('2d'); var grid = 10; var colors = "rgb(" +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +")"; var snake = { x: 160, y: 160, dx: grid, dy: 0, cells: [], maxCells: 4 }; var count = 0; var apple = { x: 320, y: 320 }; function getRandomInt(min, max) { return Math.floor(Math.random() * (max - min)) + min; } function loop() { requestAnimationFrame(loop); if (++count < 7) { return; } count = 0; context.clearRect(0,0,canvas.width,canvas.height); snake.x += snake.dx; snake.y += snake.dy; // wrap snake position on edge of screen if (snake.x < 0) { snake.x = canvas.width - grid; } else if (snake.x >= canvas.width) { snake.x = 0; } if (snake.y < 0) { snake.y = canvas.height - grid; } else if (snake.y >= canvas.height) { snake.y = 0; } snake.cells.unshift({x: snake.x, y: snake.y}); if (snake.cells.length > snake.maxCells) { snake.cells.pop(); } context.fillStyle = 'green'; context.fillRect(apple.x, apple.y, grid-1, grid-1) context.fillStyle = colors; snake.cells.forEach(function(cell, index) { context.fillRect(cell.x, cell.y, grid-1, grid-1); if (cell.x === apple.x && cell.y === apple.y) { snake.maxCells++; apple.x = getRandomInt(0, 30) * grid; apple.y = getRandomInt(0, 30) * grid; } for (var i = index + 1; i < snake.cells.length; i++) { if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) { alert("you lose, your score is") alert(snake.maxCells) colors = "rgb(" +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +")"; snake.x = 160; snake.y = 160; snake.cells = []; snake.maxCells = 4; snake.dx = grid; snake.dy = 0; apple.x = getRandomInt(0, 25) * grid; apple.y = getRandomInt(0, 25) * grid; } } }); } function left(){ if (snake.dx === 0) { snake.dx = -grid; snake.dy = 0; } } function up(){ if (snake.dy === 0) { snake.dy = -grid; snake.dx = 0; } } function right(){ if (snake.dx === 0) { snake.dx = grid; snake.dy = 0; } } function down(){ if (snake.dy === 0) { snake.dy = grid; snake.dx = 0; } } function hack(){ if (snake.maxCells){ snake.maxCells++; } } function hackd(){ if (snake.maxCells){ colors = "rgb(" +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +"," +Math.floor(Math.random() * 226) +")"; } } function hacku(){ if (snake.maxCells){ apple.x = getRandomInt(0, 25) * grid; apple.y = getRandomInt(0, 25) * grid; } } document.addEventListener('keydown', function(e) { // prevent snake from backtracking on itself if (e.which === 37 && snake.dx === 0) { snake.dx = -grid; snake.dy = 0; } else if (e.which === 38 && snake.dy === 0) { snake.dy = -grid; snake.dx = 0; } else if (e.which === 39 && snake.dx === 0) { snake.dx = grid; snake.dy = 0; } else if (e.which === 40 && snake.dy === 0) { snake.dy = grid; snake.dx = 0; } requestAnimationFrame(loop); </script>
@andrewreed549
This code for mobile did not work.

dehuachen
commented over 1 year ago

@HenryAkaya @SawyerBx
Here is a mobile version:

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title></title>
  <style>
  html, body {
```

```
    height: 100%;
    margin: 0;
  }

  body {
    background: black;
    display: flex;
    align-items: center;
    justify-content: center;
    overflow-y: hidden;
  }
  canvas {
    border: 1px solid white;
  }
  </style>
</head>
<body>
<canvas width="400" height="400" id="game"></canvas>
<script>
var canvas = document.getElementById('game');
var context = canvas.getContext('2d');

var grid = 16;
var snake = {
  x: 160,
  y: 160,
  dx: grid,
  dy: 0,
  cells: [],
  maxCells: 4
};
var count = 0;
var apple = {
  x: 320,
  y: 320
};

function getRandomInt(min, max) {
  return Math.floor(Math.random() * (max - min)) + min;
}

// game loop
function loop() {
  requestAnimationFrame(loop);

  // slow game loop to 15 fps instead of 60 - 60/15 = 4
  if (++count < 4) {
    return;
  }

  count = 0;
  context.clearRect(0,0,canvas.width,canvas.height);

  snake.x += snake.dx;
  snake.y += snake.dy;

  // wrap snake position on edge of screen
```

```
  if (snake.x < 0) {
    snake.x = canvas.width - grid;
  }
  else if (snake.x >= canvas.width) {
    snake.x = 0;
  }

  if (snake.y < 0) {
    snake.y = canvas.height - grid;
  }
  else if (snake.y >= canvas.height) {
    snake.y = 0;
  }

  // keep track of where snake has been. front of the array is always the head
  snake.cells.unshift({x: snake.x, y: snake.y});

  // remove cells as we move away from them
  if (snake.cells.length > snake.maxCells) {
    snake.cells.pop();
  }

  // draw apple
  context.fillStyle = 'red';
  context.fillRect(apple.x, apple.y, grid-1, grid-1);

  // draw snake
  context.fillStyle = 'green';
  snake.cells.forEach(function(cell, index) {
    context.fillRect(cell.x, cell.y, grid-1, grid-1);

    // snake ate apple
    if (cell.x === apple.x && cell.y === apple.y) {
      snake.maxCells++;

      apple.x = getRandomInt(0, 25) * grid;
      apple.y = getRandomInt(0, 25) * grid;
    }

    // check collision with all cells after this one (modified bubble sort)
    for (var i = index + 1; i < snake.cells.length; i++) {

      // collision. reset game
      if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) {
        snake.x = 160;
        snake.y = 160;
        snake.cells = [];
        snake.maxCells = 4;
        snake.dx = grid;
        snake.dy = 0;

        apple.x = getRandomInt(0, 25) * grid;
        apple.y = getRandomInt(0, 25) * grid;
      }
    }
  });
}
```

```
var allowedTime = 200;
var startX = 0;
var startY = 0;

document.addEventListener('touchstart', function(e){
    var touch = e.changedTouches[0]
    startX = touch.pageX
    startY = touch.pageY
    startTime = new Date().getTime()
    e.preventDefault()
}, false)

document.addEventListener('touchmove', function(e){
    e.preventDefault()
}, false)

document.addEventListener('touchend', function(e){
    var touch = e.changedTouches[0]
    distX = touch.pageX - startX
    distY = touch.pageY - startY

    if (Math.abs(distX) > Math.abs(distY)) {
      if (distX > 0 && snake.dx === 0) {
        snake.dx = grid;
        snake.dy = 0;
      }
      else if (distX < 0 && snake.dx === 0) {
        snake.dx = -grid;
        snake.dy = 0;
      }
    } else {
      if (distY > 0 && snake.dy === 0) {
        snake.dy = grid;
        snake.dx = 0;
      }
      else if (distY < 0 && snake.dy === 0) {
        snake.dy = -grid;
        snake.dx = 0;
      }
    }
    e.preventDefault();

}, false)

document.addEventListener('keydown', function(e) {
  // prevent snake from backtracking on itself
  if (e.which === 37 && snake.dx === 0) {
    snake.dx = -grid;
    snake.dy = 0;
  }
  else if (e.which === 38 && snake.dy === 0) {
    snake.dy = -grid;
    snake.dx = 0;
  }
  else if (e.which === 39 && snake.dx === 0) {
    snake.dx = grid;
```

```
    snake.dy = 0;
  }
  else if (e.which === 40 && snake.dy === 0) {
    snake.dy = grid;
    snake.dx = 0;
  }
});

requestAnimationFrame(loop);
</script>
</body>
</html>
```

Bluwinters
commented over 1 year ago

How does the computer know the input for the snake movement what line of code was that

ericsondevan
commented over 1 year ago

it is nice but i need help on how to add score and pause button.

JesseBS2 commented over 1 year ago • edited over 1 year ago

@Bluwinters

It's practically the last line of JavaScript code, right above the final `requestAnimationFrame(loop)`
key up:
`document.addEventListener('keydown', function(e){

if(e.which == 37){
snake.dy= -grid;
snake.dx = 0;
}

})`

that code is all correct(Although 37 Might not be the Up arrow, It could also be Left, Right or Down)

subeshb1
commented about 1 year ago

https://www.sketchalgorithms.com/games/snake-game
Snake Game Tutorial

Jackolanturn0
commented about 1 year ago

I am a beginner at coding, and lots of times I base my code off of this code. It's awesome!

Smt1195
commented about 1 year ago

> woooow wow it nice but score is missing

Smt1195 commented about 1 year ago • edited about 1 year ago

Copy this and enjoy. It has score an highscore added to it

<title></title> <style> html, body { height: 100%; margin: 0; } body { background: black; display: flex; align-items: center; justify-content: center; } canvas { border: 1px solid white; } #p1 { color:red; position:absolute; left:100px; top:10px; } #p2 { color:red; position:absolute; left:100px; top:30px; } #score { color:yellow; position:absolute; left:155px; top:10px; } #high { color:yellow; position:absolute; left:195px; top:30px; } </style>
SCORE:

HIGHSCORE:

<script> var canvas = document.getElementById('game'); var context = canvas.getContext('2d'); var grid = 16; var count = 0; var score=0; var max=0;
var snake = {
x: 160,
y: 160,

// snake velocity. moves one grid length every frame in either the x or y direction
dx: grid,
dy: 0,

// keep track of all grids the snake body occupies
cells: [],

// length of the snake. grows when eating an apple
maxCells: 4
};
var apple = {

```
x: 320,
y: 320
};

// get random whole numbers in a specific range
// @see https://stackoverflow.com/a/1527820/2124254
function getRandomInt(min, max) {
return Math.floor(Math.random() * (max - min)) + min;
}
// game loop
function loop() {
requestAnimationFrame(loop);
// slow game loop to 15 fps instead of 60 (60/15 = 4)
if (++count < 4) {
return;
}
count = 0;
context.clearRect(0,0,canvas.width,canvas.height);
// move snake by it's velocity
snake.x += snake.dx;
snake.y += snake.dy;
// wrap snake position horizontally on edge of screen
if (snake.x < 0) {
snake.x = canvas.width - grid;
}
else if (snake.x >= canvas.width) {
snake.x = 0;
}

// wrap snake position vertically on edge of screen
if (snake.y < 0) {
snake.y = canvas.height - grid;
}
else if (snake.y >= canvas.height) {
snake.y = 0;
}
// keep track of where snake has been. front of the array is always the head
snake.cells.unshift({x: snake.x, y: snake.y});
// remove cells as we move away from them
if (snake.cells.length > snake.maxCells) {
snake.cells.pop();
}
// draw apple
context.fillStyle = 'red';
context.fillRect(apple.x, apple.y, grid-1, grid-1);
// draw snake one cell at a time
```

```
context.fillStyle = 'green';
snake.cells.forEach(function(cell, index) {


  // drawing 1 px smaller than the grid creates a grid effect in the snake body so you can see
  context.fillRect(cell.x, cell.y, grid-1, grid-1);
  // snake ate apple
  if (cell.x === apple.x && cell.y === apple.y) {
    snake.maxCells++;
    score+=10;
    //max=score;
    document.getElementById('score').innerHTML=score;

    // canvas is 400x400 which is 25x25 grids
    apple.x = getRandomInt(0, 25) * grid;
    apple.y = getRandomInt(0, 25) * grid;
  }
  // check collision with all cells after this one (modified bubble sort)
  for (var i = index + 1; i < snake.cells.length; i++)
  {


    // snake occupies same space as a body part. reset game
    if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y)
   {

      if(score>max)
      {
       max=score;
      }
          snake.x = 160;
      snake.y = 160;
      snake.cells = [];
      snake.maxCells = 4;
      snake.dx = grid;
      snake.dy = 0;
          score=0;
      apple.x = getRandomInt(0, 25) * grid;
      apple.y = getRandomInt(0, 25) * grid;
      document.getElementById('high').innerHTML=max;
    }
  }


}

);

}
// listen to keyboard events to move the snake
document.addEventListener('keydown', function(e) {
// prevent snake from backtracking on itself by checking that it's
// not already moving on the same axis (pressing left while moving
// left won't do anything, and pressing right while moving left
// shouldn't let you collide with your own body)
```

```
// left arrow key
if (e.which === 37 && snake.dx === 0) {
snake.dx = -grid;
snake.dy = 0;
}
// up arrow key
else if (e.which === 38 && snake.dy === 0) {
snake.dy = -grid;
snake.dx = 0;
}
// right arrow key
else if (e.which === 39 && snake.dx === 0) {
snake.dx = grid;
snake.dy = 0;
}
// down arrow key
else if (e.which === 40 && snake.dy === 0) {
snake.dy = grid;
snake.dx = 0;
}
});
// start the game
requestAnimationFrame(loop);
</script>
```

noamboy2006
commented about 1 year ago

<script> var canvas = document.getElementById('game'); var context = canvas.getContext('2d'); var grid = 16; var count = 0; var snake = { x: 160, y: 160, dx: grid, dy: 0, cells: [], maxCells: 4 }; var apple = { x: 320, y: 320 }; function getRandomInt(min, max) { return Math.floor(Math.random() * (max - min)) + min; } function loop() { requestAnimationFrame(loop); if (++count < 4) { return; } count = 0; context.clearRect(0,0,canvas.width,canvas.height); snake.x += snake.dx; snake.y += snake.dy; if (snake.x < 0) { snake.x = canvas.width - grid; } else if (snake.x >= canvas.width) { snake.x = 0; } if (snake.y < 0) { snake.y = canvas.height - grid; } else if (snake.y >= canvas.height) { snake.y = 0; } snake.cells.unshift({x: snake.x, y: snake.y}); if (snake.cells.length > snake.maxCells) { snake.cells.pop(); } context.fillStyle = 'red'; context.fillRect(apple.x, apple.y, grid-1, grid-1); context.fillStyle = 'green'; snake.cells.forEach(function(cell, index) { context.fillRect(cell.x, cell.y, grid-1, grid-1); if (cell.x === apple.x && cell.y === apple.y) { snake.maxCells++; apple.x = getRandomInt(0, 25) * grid; apple.y = getRandomInt(0, 25) * grid; } for (var i = index + 1; i < snake.cells.length; i++) { if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) { alert((snake.maxcells-4)+"점"); snake.x = 160; snake.y = 160; snake.cells = []; snake.maxCells = 4; snake.dx = grid; snake.dy = 0; apple.x = getRandomInt(0, 25) * grid; apple.y = getRandomInt(0, 25) * grid; } } }); } document.addEventListener('keydown', function(e) { if (e.which === 37 && snake.dx === 0) { snake.dx = -grid; snake.dy = 0; } else if (e.which === 38 && snake.dy === 0) { snake.dy = -grid; snake.dx = 0; } else if (e.which

=== 39 && snake.dx === 0) { snake.dx = grid; snake.dy = 0; } else if (e.which === 40 && snake.dy === 0) { snake.dy = grid; snake.dx = 0; } }); requestAnimationFrame(loop); </script>
score is here

JesseBS2
commented about 1 year ago

@straker Hello! I am a big fan of this, project. I actually created an AI in pure javaScript to beat the game(because I stink at snake), it still has it's fair share of glitches, and hasn't once beaten the game, but I would still really like it if you checked out the repository here: https://github.com/JesseBS2/Snake-Ai

nedwar01
commented about 1 year ago

Beautiful! Thanks for this guys

Mubarak1234
commented 11 months ago

Hi
how to add score in snake on HTML.
tell me

ismailmuktar
commented 11 months ago

thanks so much.

firdous1
commented 9 months ago

<title></title> <style> html, body { height: 100%; margin: 0; } body { background: black; display: flex; align-items: center; justify-content: center; } canvas { border: 1px solid white; } </style> <script> var canvas = document.getElementById('game'); var context = canvas.getContext('2d'); var grid = 16; var count = 0;
var snake = {
x: 160,
y: 160,

// snake velocity. moves one grid length every frame in either the x or y direction
dx: grid,

```
dy: 0,

// keep track of all grids the snake body occupies
cells: [],

// length of the snake. grows when eating an apple
maxCells: 4
};
var apple = {
x: 320,
y: 320
};
// get random whole numbers in a specific range
// @see https://stackoverflow.com/a/1527820/2124254
function getRandomInt(min, max) {
return Math.floor(Math.random() * (max - min)) + min;
}
// game loop
function loop() {
requestAnimationFrame(loop);
// slow game loop to 15 fps instead of 60 (60/15 = 4)
if (++count < 4) {
return;
}
count = 0;
context.clearRect(0,0,canvas.width,canvas.height);
// move snake by it's velocity
snake.x += snake.dx;
snake.y += snake.dy;
// wrap snake position horizontally on edge of screen
if (snake.x < 0) {
snake.x = canvas.width - grid;
}
else if (snake.x >= canvas.width) {
snake.x = 0;
}

// wrap snake position vertically on edge of screen
if (snake.y < 0) {
snake.y = canvas.height - grid;
}
else if (snake.y >= canvas.height) {
snake.y = 0;
}
// keep track of where snake has been. front of the array is always the head
snake.cells.unshift({x: snake.x, y: snake.y});
// remove cells as we move away from them
```

```
if (snake.cells.length > snake.maxCells) {
snake.cells.pop();
}
// draw apple
context.fillStyle = 'red';
context.fillRect(apple.x, apple.y, grid-1, grid-1);
// draw snake one cell at a time
context.fillStyle = 'green';
snake.cells.forEach(function(cell, index) {

// drawing 1 px smaller than the grid creates a grid effect in the snake body so you can see how long it is
context.fillRect(cell.x, cell.y, grid-1, grid-1);
// snake ate apple
if (cell.x === apple.x && cell.y === apple.y) {
snake.maxCells++;
// canvas is 400x400 which is 25x25 grids
apple.x = getRandomInt(0, 25) * grid;
apple.y = getRandomInt(0, 25) * grid;
}
// check collision with all cells after this one (modified bubble sort)
for (var i = index + 1; i < snake.cells.length; i++) {


    // snake occupies same space as a body part. reset game
    if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) {
      snake.x = 160;
      snake.y = 160;
      snake.cells = [];
      snake.maxCells = 4;
      snake.dx = grid;
      snake.dy = 0;
      apple.x = getRandomInt(0, 25) * grid;
      apple.y = getRandomInt(0, 25) * grid;
    }


}
});
}
// listen to keyboard events to move the snake
document.addEventListener('keydown', function(e) {
// prevent snake from backtracking on itself by checking that it's
// not already moving on the same axis (pressing left while moving
// left won't do anything, and pressing right while moving left
// shouldn't let you collide with your own body)

// left arrow key
if (e.which === 37 && snake.dx === 0) {
snake.dx = -grid;
snake.dy = 0;
```

```
}
// up arrow key
else if (e.which === 38 && snake.dy === 0) {
snake.dy = -grid;
snake.dx = 0;
}
// right arrow key
else if (e.which === 39 && snake.dx === 0) {
snake.dx = grid;
snake.dy = 0;
}
// down arrow key
else if (e.which === 40 && snake.dy === 0) {
snake.dy = grid;
snake.dx = 0;
}
});
// start the game
requestAnimationFrame(loop);
</script>
```

HTMLstar1325
commented 8 months ago

It's great! it is a little too fast to move around without missing the `apples.

kylecumber
commented 7 months ago

so this was coded in html javasctript

arifattal commented 4 months ago • edited 4 months ago

Great work on this.
I would love to minimize the game's size to 400x200, and to replace the arrow keys with a,w,d,s.
I have edited the code as following, however, the controlling keys are still the arrow keys and the snake isn't always able to 'eat' the apples, any suggestions?

<title></title> <style> html, body { height: 100%; margin: 0; } body { background: transparent; display: flex; align-items: center; justify-content: center; } canvas { border: 0px solid white; } </style> <script> var canvas = document.getElementById('game'); var context = canvas.getContext('2d'); var grid = 16; var count = 0;

```
var snake = {
x: 160,
y: 100,

// snake velocity. moves one grid length every frame in either the x or y direction
dx: grid,
dy: 0,

// keep track of all grids the snake body occupies
cells: [],

// length of the snake. grows when eating an apple
maxCells: 4
};
var apple = {
x: 320,
y: 120
};
// get random whole numbers in a specific range
// @see https://stackoverflow.com/a/1527820/2124254
function getRandomInt(min, max) {
return Math.floor(Math.random() * (max - min)) + min;
}
// game loop
function loop() {
requestAnimationFrame(loop);
// slow game loop to 15 fps instead of 60 (60/15 = 4)
if (++count < 4) {
return;
}
count = 0;
context.clearRect(0,0,canvas.width,canvas.height);
// move snake by it's velocity
snake.x += snake.dx;
snake.y += snake.dy;
// wrap snake position horizontally on edge of screen
if (snake.x < 0) {
snake.x = canvas.width - grid;
}
else if (snake.x >= canvas.width) {
snake.x = 0;
}

// wrap snake position vertically on edge of screen
if (snake.y < 0) {
snake.y = canvas.height - grid;
}
```

```
else if (snake.y >= canvas.height) {
snake.y = 0;
}
// keep track of where snake has been. front of the array is always the head
snake.cells.unshift({x: snake.x, y: snake.y});
// remove cells as we move away from them
if (snake.cells.length > snake.maxCells) {
snake.cells.pop();
}
// draw apple
context.fillStyle = 'red';
context.fillRect(apple.x, apple.y, grid-1, grid-1);
// draw snake one cell at a time
context.fillStyle = 'black';
snake.cells.forEach(function(cell, index) {


  // drawing 1 px smaller than the grid creates a grid effect in the snake body so you can see
  context.fillRect(cell.x, cell.y, grid-1, grid-1);
  // snake ate apple
  if (cell.x === apple.x && cell.y === apple.y) {
    snake.maxCells++;
    // canvas is 400x400 which is 25x25 grids

    apple.x = getRandomInt(0, 25) * grid;
    apple.y = getRandomInt(0, 12.5) * grid;
  }
  // check collision with all cells after this one (modified bubble sort)
  for (var i = index + 1; i < snake.cells.length; i++) {

    // snake occupies same space as a body part. reset game
    if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) {
      snake.x = 160;
      snake.y = 100;
      snake.cells = [];
      snake.maxCells = 4;
      snake.dx = grid;
      snake.dy = 0;
      apple.x = getRandomInt(0, 25) * grid;
      apple.y = getRandomInt(0, 12.5) * grid;
    }
  }


});
}
// listen to keyboard events to move the snake
document.addEventListener('keydown', function(e) {
// prevent snake from backtracking on itself by checking that it's
// not already moving on the same axis (pressing left while moving
// left won't do anything, and pressing right while moving left
// shouldn't let you collide with your own body)
```

```
// a arrow key
if (e.which === 37 && snake.dx === 0) {
snake.dx = -grid;
snake.dy = 0;
}
// w arrow key
else if (e.which === 38 && snake.dy === 0) {
snake.dy = -grid;
snake.dx = 0;
}
// d arrow key
else if (e.which === 39 && snake.dx === 0) {
snake.dx = grid;
snake.dy = 0;
}
// s arrow key
else if (e.which === 40 && snake.dy === 0) {
snake.dy = grid;
snake.dx = 0;
}
});
// start the game
requestAnimationFrame(loop);
</script>
```

morganpage
commented 3 months ago

Nice! Did a Phaser version here

AndreWWWolf
commented 3 months ago

> Great work on this.
> I would love to minimize the game's size to 400x200, and to replace the arrow keys with a,w,d,s.
> I have edited the code as following, however, the controlling keys are still the arrow keys and the snake isn't always able to 'eat' the apples, any suggestions?
>
> <title></title> <style> html, body { height: 100%; margin: 0; } body { background: transparent; display: flex; align-items: center; justify-content: center; } canvas { border: 0px solid white; } </style> <script> var canvas = document.getElementById('game'); var context = canvas.getContext('2d'); var grid = 16; var count = 0; var snake = { x: 160, y: 100,
> // snake velocity. moves one grid length every frame in either the x or y direction
> dx: grid,

```
dy: 0,

// keep track of all grids the snake body occupies
cells: [],

// length of the snake. grows when eating an apple
maxCells: 4
};
var apple = {
x: 320,
y: 120
};
// get random whole numbers in a specific range
// @see https://stackoverflow.com/a/1527820/2124254
function getRandomInt(min, max) {
return Math.floor(Math.random() * (max - min)) + min;
}
// game loop
function loop() {
requestAnimationFrame(loop);
// slow game loop to 15 fps instead of 60 (60/15 = 4)
if (++count < 4) {
return;
}
count = 0;
context.clearRect(0,0,canvas.width,canvas.height);
// move snake by it's velocity
snake.x += snake.dx;
snake.y += snake.dy;
// wrap snake position horizontally on edge of screen
if (snake.x < 0) {
snake.x = canvas.width - grid;
}
else if (snake.x >= canvas.width) {
snake.x = 0;
}

// wrap snake position vertically on edge of screen
if (snake.y < 0) {
snake.y = canvas.height - grid;
}
else if (snake.y >= canvas.height) {
snake.y = 0;
}
// keep track of where snake has been. front of the array is always the head
snake.cells.unshift({x: snake.x, y: snake.y});
// remove cells as we move away from them
```

```
if (snake.cells.length > snake.maxCells) {
snake.cells.pop();
}
// draw apple
context.fillStyle = 'red';
context.fillRect(apple.x, apple.y, grid-1, grid-1);
// draw snake one cell at a time
context.fillStyle = 'black';
snake.cells.forEach(function(cell, index) {


  // drawing 1 px smaller than the grid creates a grid effect in the snake body so you car
  context.fillRect(cell.x, cell.y, grid-1, grid-1);
  // snake ate apple
  if (cell.x === apple.x && cell.y === apple.y) {
    snake.maxCells++;
    // canvas is 400x400 which is 25x25 grids
    apple.x = getRandomInt(0, 25) * grid;
    apple.y = getRandomInt(0, 12.5) * grid;
  }
  // check collision with all cells after this one (modified bubble sort)
  for (var i = index + 1; i < snake.cells.length; i++) {

    // snake occupies same space as a body part. reset game
    if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) {
      snake.x = 160;
      snake.y = 100;
      snake.cells = [];
      snake.maxCells = 4;

      snake.dx = grid;
      snake.dy = 0;
      apple.x = getRandomInt(0, 25) * grid;
      apple.y = getRandomInt(0, 12.5) * grid;
    }
  }


});
}
// listen to keyboard events to move the snake
document.addEventListener('keydown', function(e) {
// prevent snake from backtracking on itself by checking that it's
// not already moving on the same axis (pressing left while moving
// left won't do anything, and pressing right while moving left
// shouldn't let you collide with your own body)

// a arrow key
if (e.which === 65 && snake.dx === 0) {
snake.dx = -grid;
snake.dy = 0;
}
```

```
// w arrow key
else if (e.which === 86 && snake.dy === 0) {
snake.dy = -grid;
snake.dx = 0;
}
// d arrow key
else if (e.which === 68 && snake.dx === 0) {
snake.dx = grid;
snake.dy = 0;
}
// s arrow key
else if (e.which === 83 && snake.dy === 0) {
snake.dy = grid;
snake.dx = 0;
}
});
// start the game
requestAnimationFrame(loop);
</script>
```

//the numbers called in the arrow key functions are key values, 37, 38, 39, 40 are the arrow keys. so replacing those values with the ones above as i edited will use the w,a,s,d keys instead of arrows. look up "keyboard key values html" to see a full list.

erosramses18
commented 3 months ago

i download the game

nhat-nam
commented 3 months ago

> How does the computer know the input for the snake movement what line of code was that the eventListener part

---

Nyal1
commented about 2 months ago

The two issues are that there is no score counter, and the snake doesnt die when it hits the wall. To add a score counter:

<style> canvas { border:1px solid #d3d3d3; background-color: #f1f1f1; } </style> <script>

```
var myGamePiece;
var myObstacles = [];
var myScore;

function startGame() {
myGamePiece = new component(30, 30, "red", 10, 120);
myScore = new component("30px", "Consolas", "black", 280, 40, "text");
myGameArea.start();
}

var myGameArea = {
canvas : document.createElement("canvas"),
start : function() {
this.canvas.width = 480;
this.canvas.height = 270;
this.context = this.canvas.getContext("2d");
document.body.insertBefore(this.canvas, document.body.childNodes[0]);
this.frameNo = 0;
this.interval = setInterval(updateGameArea, 20);
},
clear : function() {
this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
},
stop : function() {
clearInterval(this.interval);
}
}

function component(width, height, color, x, y, type) {
this.type = type;
this.width = width;
this.height = height;
this.speedX = 0;
this.speedY = 0;
this.x = x;
this.y = y;
this.update = function() {
ctx = myGameArea.context;
if (this.type == "text") {
ctx.font = this.width + " " + this.height;
ctx.fillStyle = color;
ctx.fillText(this.text, this.x, this.y);
} else {
ctx.fillStyle = color;
ctx.fillRect(this.x, this.y, this.width, this.height);
}
}
```

```
this.newPos = function() {
this.x += this.speedX;
this.y += this.speedY;
}
this.crashWith = function(otherobj) {
var myleft = this.x;
var myright = this.x + (this.width);
var mytop = this.y;
var mybottom = this.y + (this.height);
var otherleft = otherobj.x;
var otherright = otherobj.x + (otherobj.width);
var othertop = otherobj.y;
var otherbottom = otherobj.y + (otherobj.height);
var crash = true;
if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft) || (myleft > otherright)) {
crash = false;
}
return crash;
}
}

function updateGameArea() {
var x, height, gap, minHeight, maxHeight, minGap, maxGap;
for (i = 0; i < myObstacles.length; i += 1) {
if (myGamePiece.crashWith(myObstacles[i])) {
myGameArea.stop();
return;
}
}
myGameArea.clear();
myGameArea.frameNo += 1;
if (myGameArea.frameNo == 1 || everyinterval(150)) {
x = myGameArea.canvas.width;
minHeight = 20;
maxHeight = 200;
height = Math.floor(Math.random()(maxHeight-minHeight+1)+minHeight);
minGap = 50;
maxGap = 200;
gap = Math.floor(Math.random()(maxGap-minGap+1)+minGap);
myObstacles.push(new component(10, height, "green", x, 0));
myObstacles.push(new component(10, x - height - gap, "green", x, height + gap));
}
for (i = 0; i < myObstacles.length; i += 1) {
myObstacles[i].speedX = -1;
myObstacles[i].newPos();
myObstacles[i].update();
```

```
}
myScore.text="SCORE: " + myGameArea.frameNo;
myScore.update();
myGamePiece.newPos();
myGamePiece.update();
}

function everyinterval(n) {
if ((myGameArea.frameNo / n) % 1 == 0) {return true;}
return false;
}

function moveup() {
myGamePiece.speedY = -1;
}

function movedown() {
myGamePiece.speedY = 1;
}

function moveleft() {
myGamePiece.speedX = -1;
}

function moveright() {
myGamePiece.speedX = 1;
}

function clearmove() {
myGamePiece.speedX = 0;
myGamePiece.speedY = 0;
}
</script>
```

UP

LEFT RIGHT

DOWN
The score will count one point for each frame you manage to "stay alive".

Code-With-Me-YT
commented 22 days ago

Can someone please edit this so that it closes the window if you hit the wall. Thankyou.

```
var canvas = document.getElementById('game');
var context = canvas.getContext('2d');
```

```
var grid = 16;
var count = 0;

var snake = {
x: 160,
y: 160,

// snake velocity. moves one grid length every frame in either the x or y direction
dx: grid,
dy: 0,

// keep track of all grids the snake body occupies
cells: [],

// length of the snake. grows when eating an apple
maxCells: 4
};
var apple = {
x: 320,
y: 320
};
// get random whole numbers in a specific range
// @see https://stackoverflow.com/a/1527820/2124254
function getRandomInt(min, max) {
return Math.floor(Math.random() * (max - min)) + min;
}
// game loop
function loop() {
requestAnimationFrame(loop);
// slow game loop to 15 fps instead of 60 (60/15 = 4)
if (++count < 4) {
return;
}
count = 0;
context.clearRect(0,0,canvas.width,canvas.height);
// move snake by it's velocity
snake.x += snake.dx;
snake.y += snake.dy;
// wrap snake position horizontally on edge of screen
if (snake.x < 0) {
snake.x = canvas.width - grid;
}
else if (snake.x >= canvas.width) {
snake.x = 0;
}
```

```
// wrap snake position vertically on edge of screen
if (snake.y < 0) {
snake.y = canvas.height - grid;
}
else if (snake.y >= canvas.height) {
snake.y = 0;
}
// keep track of where snake has been. front of the array is always the head
snake.cells.unshift({x: snake.x, y: snake.y});
// remove cells as we move away from them
if (snake.cells.length > snake.maxCells) {
snake.cells.pop();
}
// draw apple
context.fillStyle = 'red';
context.fillRect(apple.x, apple.y, grid-1, grid-1);
// draw snake one cell at a time
context.fillStyle = 'green';
snake.cells.forEach(function(cell, index) {


  // drawing 1 px smaller than the grid creates a grid effect in the snake body so you can see
  context.fillRect(cell.x, cell.y, grid-1, grid-1);
  // snake ate apple
  if (cell.x === apple.x && cell.y === apple.y) {
    snake.maxCells++;
    // canvas is 400x400 which is 25x25 grids
    apple.x = getRandomInt(0, 25) * grid;
    apple.y = getRandomInt(0, 25) * grid;
  }
  // check collision with all cells after this one (modified bubble sort)
  for (var i = index + 1; i < snake.cells.length; i++) {

    // snake occupies same space as a body part. reset game
    if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) {
      snake.x = 160;
      snake.y = 160;
      snake.cells = [];
      snake.maxCells = 4;
      snake.dx = grid;
      snake.dy = 0;
      apple.x = getRandomInt(0, 25) * grid;
      apple.y = getRandomInt(0, 25) * grid;
    }
  }


});
}
// listen to keyboard events to move the snake
document.addEventListener('keydown', function(e) {
// prevent snake from backtracking on itself by checking that it's
```

```
// not already moving on the same axis (pressing left while moving
// left won't do anything, and pressing right while moving left
// shouldn't let you collide with your own body)

// left arrow key
if (e.which === 37 && snake.dx === 0) {
snake.dx = -grid;
snake.dy = 0;
}
// up arrow key
else if (e.which === 38 && snake.dy === 0) {
snake.dy = -grid;
snake.dx = 0;
}
// right arrow key
else if (e.which === 39 && snake.dx === 0) {
snake.dx = grid;
snake.dy = 0;
}
// down arrow key
else if (e.which === 40 && snake.dy === 0) {
snake.dy = grid;
snake.dx = 0;
}
});
// start the game
requestAnimationFrame(loop);
```