

Game Designing

Main Menu Page:

Snake Math Game

Instruction

Instructions

To play the game please use the arrow keys on your keyboard



The snake math game educates young scholars by implementing math concepts into a simple game

Once you access the game the coordinates will appear

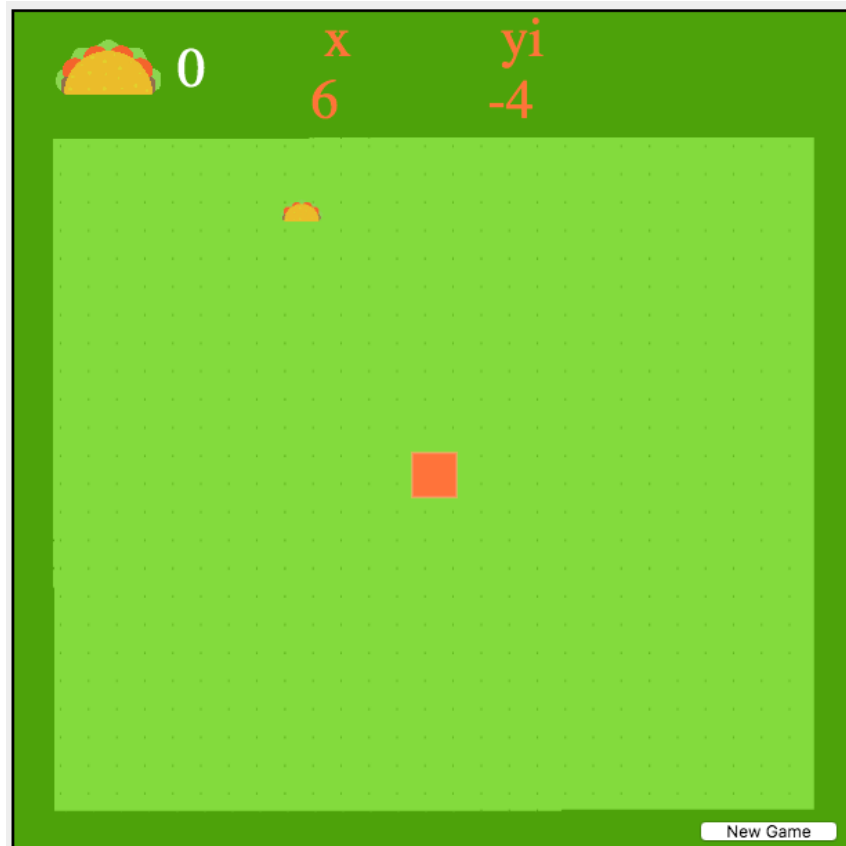
Please pay close attention to the coordinates

This is the fourth quarter where x numbers are positive and y numbers are negative

If you feel ready to enjoy the magic, please click the button PLAY below

[PLAY](#)

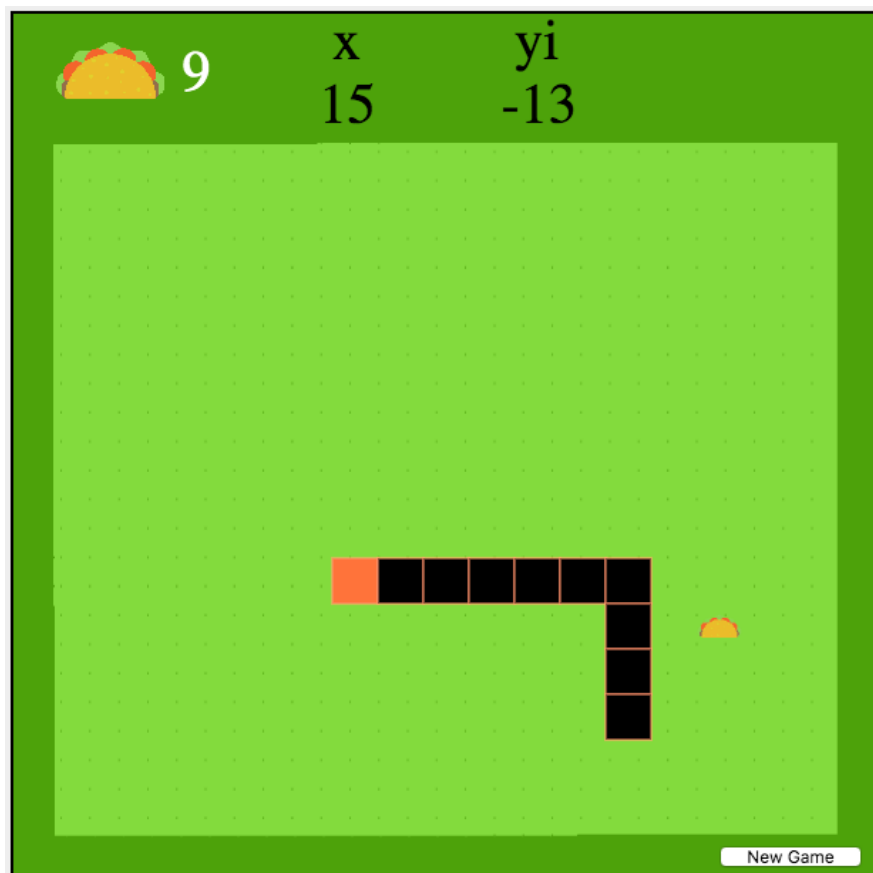
(MAIN_PLAY Page) START:



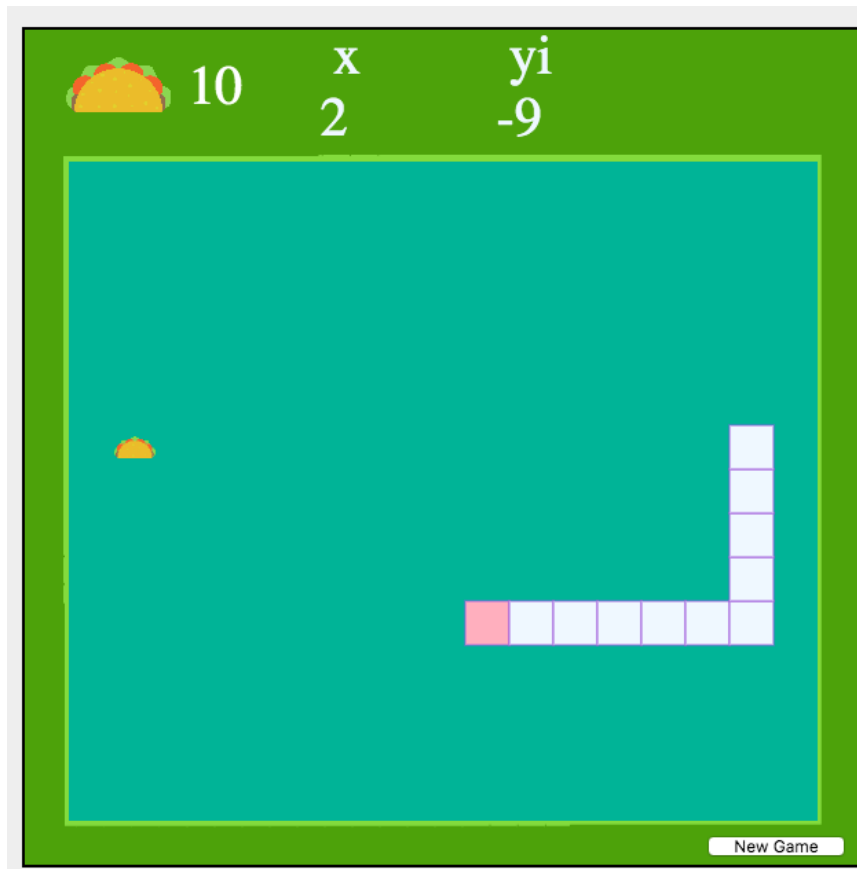
(MAIN_PLAY Page) MOVING:



(MAIN_PLAY Page): EATING THE SNACK:



(MAIN_PLAY Page): SCORE = 10



(MAIN_PLAY Page): GAME OVER



Solution Overview:

Canvas:

Before the object creation the canvas for the field should be set:

```
screen_height = 608 px
screen_width = 608 px
```

First, the snake field should be constructed. A snake field is a place, where the snake can freely move and eat snacks. The field takes a big part of the canvas, however, it is not the whole canvas that is illustrated as an are of 608x608px. The field is confined by the range:

```
x.min = 32px
x.max = 544px
y.min = 96px
y.max = 544px
```

The main object of the game is a snake. The snake is represented as a two-dimensional(2D) array - an array that has two values: x and y. As an array the first element of the array is the head. Positions of the head are snake_array[0].x and snake_array[0].y. The values of the snake and its initial position are defined:

```
snake_array = 32px
snake.x = 288px
snake.y = 320px
```

The snake can move UP, RIGHT, LEFT, DOWN. The movement of the snake should be connected to four different keys. Moreover, the snake cannot travel to the opposite side without turning. For example, if the snake is moving up, then it can't move down. The keys that are connected to the movement of the snake are arrow keys:

```
IF keys.get_pressed("LEFT") AND snake_movement != "RIGHT":
    snake_array[0].x = snake_array[0].x + 32px
ELIF keys.get_pressed("RIGHT")AND snake_movement != "LEFT":
    snake_array[0].x = snake_array[0].x - 32px
ELIF keys.get_pressed("UP")AND snake_movement != "DOWN":
    snake_array[0].y = snake_array[0].y + 32px
ELIF keys.get_pressed("DOWN")AND snake_movement != "UP":
    snake_array[0].y = snake_array[0].y - 32px
END IF
```

The snake cannot go over the field, hence the field should be set up. It is also important to know that the snake bumping into the wall that is going to be created results in a game over:

```
IF snake_array[0].x <32 OR snake_array[0].x >544 OR
snake_array[0].y < 96 OR snake_array[0].y > 544
    game_over = true
```

The snack is another important part of the game:

```
snack_value = 32
```

The position of the snack is defined as the all available space, where the snake can freely move, however space free from the snake itself.

When a snake collides with the snack the program needs to check whether the positions of the snack are the same as the positions of the snake's head. Furthermore, with each snack, the snake eats the snake becomes bigger and the score increases by 1:

```
score = 0
IF snack.x == snake[0].x AND snack.y == snake[0].y
    score++
    snake.length = snake length + 1
END IF
```

The snake then needs to be drawn. To draw the snake the loop that goes through the whole array is used. The function is similar to bubble sort. In addition, with a bigger snake, the need to check the collision between a snake and the snack is needed to be checked:

```
FOR(i=0; i<array.length; i++)
    IF(snake[0].x == snake[i].x && snake[0].y == snake[i].y)
        collision = true
    ELSE
        collision = false
    END IF
END FOR
```

Test Plan

	What is being tested	Test Methods and Expected Results
1	Whether the menu works properly	Click on all the links(button)- the instruction should be a text.
2	To have a score at the top of the canvas, that will be updating as soon as snake eats the snack.	Once the collision between snack and snake is happening the score should increase by 1.
3	Whether the game reloads after pressing the reload button	After a collision with the tailor, the collision with a tailor, wall the game should be reloaded when the user presses the button. It can be also checked by pressing the reload button and seeing the different locations of snacks.
4	Whether the snack spams randomly	By colliding with a snake or refreshing the page or pressing the button that refreshes the page the snack should appear in different places to ensure interest to the game. The snack should spam in different places without any pattern.
5	Whether the snake moves correctly	Check on the corresponding up and down and right and left keys and if one of them is pressed the opposite direction is declined in execution. It is checked by moving the snake with arrow keys and if the snake moves and doesn't move the opposite position when pressing an opposite key then the function succeeds.
6	Whether the snake goes over the field	It is impossible for the snake to go over the field. The way to test it is to control the snake and direct it towards barriers. If the game is going to stop then the function works properly.
7	Whether coordinates change with a new snack spawn	When a snake collides with a snack new coordinates for the new snack appear. To test that the page can be refreshed manually or with the use of a button and then if coordinates change in response to snack change then the function is tested.
8	Whether the snake can go	When the snake eats its own tale it

	through itself	should go through it due to a special request from the client. To check that the snake needs to be long enough and when eating the tail the snake should go its own tail.
9	The color of coordinates, canvas, and the snake when the score hits 10	For this to work the score of the snake should be 10 then the snake should frequently change its color until the player scores 11. If the player score 10 and the color changes and then after 11 the color returns to normal then the function works properly.
10	The snake should grow as soon as it eats the snack	Whenever the snake eats the snack the tail should extend immediately. The way to check it is to collide the snake with the snack and to see if the tail changed or not.
12	The banner game over should appear after game over	When game over happens - collision with the wall the banner with the game over should appear. The only way to test it is to collide snake with the wall and when it collides with the wall then canvas should change and a new canvas will appear as a banner.