

Snake Game part 5

Step 5: Add Food

For this step, we will add food. We want this to be a red square placed in a random grid square on the board. We need a new PVector variable for the food's location.

```
PVector food;
```

Now we will create new methods for placing the food, and showing the food.

```
void newFood() {
  int columns = width/gridSize;
  int rows = height/gridSize;
  int randomColumn = (int)random(columns) * gridSize;
  int randomRow = (int)random(rows) * gridSize;
  food = new PVector(randomColumn, randomRow);
}

void showFood() {
  fill(255, 0, 100);
  square(food.x, food.y, gridSize);
}
```

We will call this method in the `newGame()` method to start the game with food.

```
void newGame() {
  snake = new PVector(width/2, height/2);
  snakeSize = gridSize;
  direction(1, 0);
  newFood();
}
```

Then, we'll use the `mousePressed()` method to move the food to a new location, as a way of testing our code.

```
void mousePressed() {
  newFood();
}
```

Your code will now look like this.

```
int gridSize = 20;

PVector snake;
int snakeSize;
int xSpeed;
```

```

int xSpeed;
int ySpeed;
int tailLength;
ArrayList<PVector> tail = new ArrayList();

PVector food;

void setup() {
  size(600, 600);
  frameRate(10);
  newGame();
}

void draw() {
  background(0);
  showFood();
  moveSnake();
  moveTail();
  showSnake();
  showTail();
}

void newGame() {
  snake = new PVector(width/2, height/2);
  snakeSize = gridSize;
  direction(1, 0);
  newFood();
}

void showSnake() {
  fill(100, 215, 0);
  square(snake.x, snake.y, snakeSize);
}

void moveSnake() {
  snake.x += xSpeed;
  snake.y += ySpeed;

  // This keeps the snake on the board
  snake.x = constrain(snake.x, 0, width-gridSize);
  snake.y = constrain(snake.y, 0, height-gridSize);
}

void moveTail() {
  if (tailLength > 0) {
    if (tailLength == tail.size() && !tail.isEmpty()) {
      tail.remove(0);
    }
    tail.add(new PVector(snake.x, snake.y));
  }
}

```

```

void showTail() {
  for (PVector section : tail) {
    square(section.x, section.y, gridSize);
  }
}

void direction(int x, int y) {
  xSpeed = x * gridSize;
  ySpeed = y * gridSize;
}

void keyPressed() {
  if (keyCode == UP) {
    direction(0, -1);
  } else if (keyCode == DOWN) {
    direction(0, 1);
  } else if (keyCode == RIGHT) {
    direction(1, 0);
  } else if (keyCode == LEFT) {
    direction(-1, 0);
  }
}

void newFood() {
  int columns = width/gridSize;
  int rows = height/gridSize;
  int randomColumn = (int)random(columns) * gridSize;
  int randomRow = (int)random(rows) * gridSize;
  food = new PVector(randomColumn, randomRow);
}

void showFood() {
  fill(255, 0, 100);
  square(food.x, food.y, gridSize);
}

void mousePressed() {
  newFood();
}

```

Challenges

1. Use images of food, instead of a red square. (Search Processing adding images for instructions and tutorials.)

