ADDETC – Área Departamental de Engenharia Eletrónica e Telecomunicações e de Computadores

LEIM -Licenciatura Engenharia informática e multimédia

Modelação e Simulação de Sistemas Naturais

Trabalho para casa 4

Turma:

LEIM-32D

Trabalho realizado por:

Miguel Silvestre Nº45101

Miguel Távora Nº45102

Pedro Dias Nº45170

Docente:

Arnaldo Abrantes

Data de entrega:22/12/2019

Índice

Índice figuras

Exercício 1

- Na realização do primeiro exercício foram implementadas duas maneiras diferentes na primeira solução foi implementada a tecla **W** para acelarar. Quando se deixa de clicar na tecla nessa mesma tecla o boid desacelara automáticamente. Para realizar esta maneira foi redefinida a função keyReleased() que é chamada assim que se deixa de clicar em alguma tecla.

- A segunda abordagem do programa foi também utilizada a tecla **W** para acelarar, mas para desacelarar é necessário clicar na tecla **S**.

- Também foi adicionado cor ao circulo que o o boid persegue.

Figura 2 - Boid a perseguir o ponto definido

Figura 1 - Segundo exemplo do boid a perseguir o ponto

Exercício 2

- Para implementar o comportamento arrive foi definido o método arrive(), este comportamento consiste essencialmente em diminuir a velocidade quando o boid se aproxima do alvo.

- Para implementar o comportamento de deteção de próximidade do alvo foi utilizada o método inSight().

Figura 4 Comportamento de desacelaramento

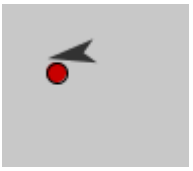

Figura 3 – Segundo comportamento de desacelaramento

Exercício 3

- A implementação do comportamento patrol consiste na existência de três pontos, ou seja, três PVector's, pré-definidos. O boid persegue um dos Pvector's, quando chega perto deste a posição é atualizada a posição para onde o Boid se deve dirigir proseguindo para o próxima posição.

  As posições escolidas foram relativamente parecidas há imagem que se segue:

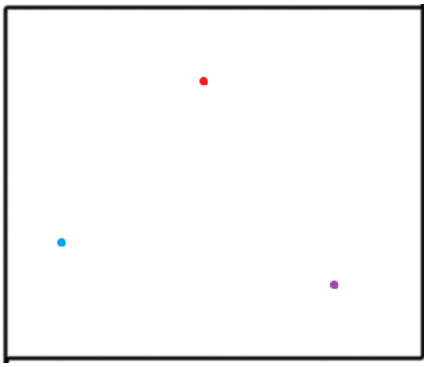

Figura 5 - Disposição dos pontos no comportamento patrol

- Foram escolhidos estes pontos para definir uma tragetória no qual o boid não tivesse de virar para trás para ir para os pontos definidos, para não provocar atrasos.

- Na primeira volta o ponto azul parece mal implementado pois o alcance do range do boid é de 0.5 e ele n chega a tocar exatamente em cima do ponto, porém na segunda volta que o boid realizada este já passa por sima do ponto.

Figura 7 - Comportamento Patrol para o primeiro PVector

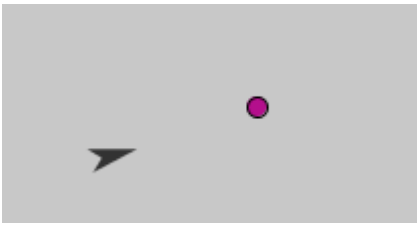

Figura 6 - Comportamento Patrol para o ultimo PVector

Exercício 4

- Para implementar o comportamento wander, foi criado o método wander() que essencialmente gera um novo Pvector para onde o boid se deve dirigir sempre é corrido o método.

- Como o método draw() do Papplet corre diversas vezes em 1 segundo o boid ficava com um movimento pouco fluído e a variar muito.

- Para contornar esse problema foi criado uma espécie de timer mas em forma de contador. Esta metodologia foi devido a que o tempo é pouco preciso, pois por vezes o draw() corre passado 200ms outra vezes mais outra vezes menos. Através do contador é garantido que ele corre sempre o método wander com um desfasamento de tempo pouco relevante. O método wander() é chamado sempre que o contador chega a 50.

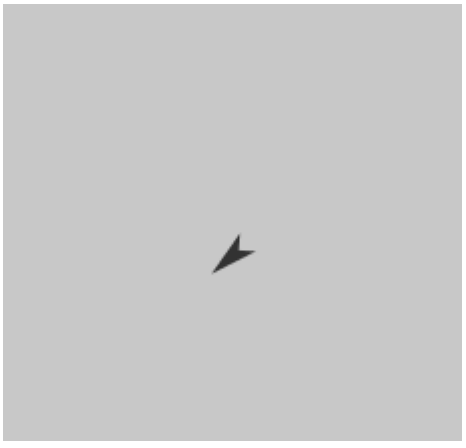

Figura 8 - Comportamento wander

Exercício 5

- Para realizar a comutação de comportamentos foram utilizados dois Boids. Quando um Boid específico encontra outro Boid este foge e caso não encontre fica a realizar o comportamento wander. Para realizar a comutação de comportamentos entre Boids foi utilizada uma estrutura parecida á do exercício 4.

- Para a implementação do comportamento foi acrescentado o método evade, esta função basicamente muda a trajetória para longe do Boid que foi detetado e o método é chamado quando é detetada proximidade entre os Boids. Visto que o Boid demora algum tempo a realizar os cálculos da função evade, e como o boid que é para ser evitado é mais rapido do que esse Boid que realiza o evade nem sempre é possível verificar esse mesmo comportamento.

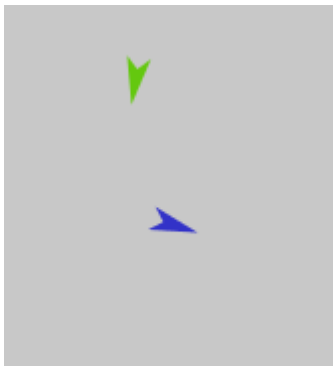

Figura 9 - Comportamento evade

- O Boid que chama o método evade é o Boid azul

Exercício 6

- Para a realização do presente trabalho utilizou-se as classes de Flocking e a classe de boid.

- O conceito do exercício é essencialmente o boid perseguir o conjunto de boids. Para

isso utilizou-se o método persuit() que essencialmente persegue um alvo passado como argumento.

- Como o grupo não conseguio ir buscar individualmente todos os Boids no ArrayList foi utilizado 4 if, quer isto dizer o boid so consegue perseguir 4 boids do flocking

- Outra limitação do trabalho foi não realização da pintura de um boid individual dentro do ArrayList de boids. Pelo que não é possível saber qual será os boids que o boid predador irá perseguir.

- Caso não encontre nenhum desses boids ele realizada o comportamento wander.

Figura 10 - Comportamento pursuit em boids