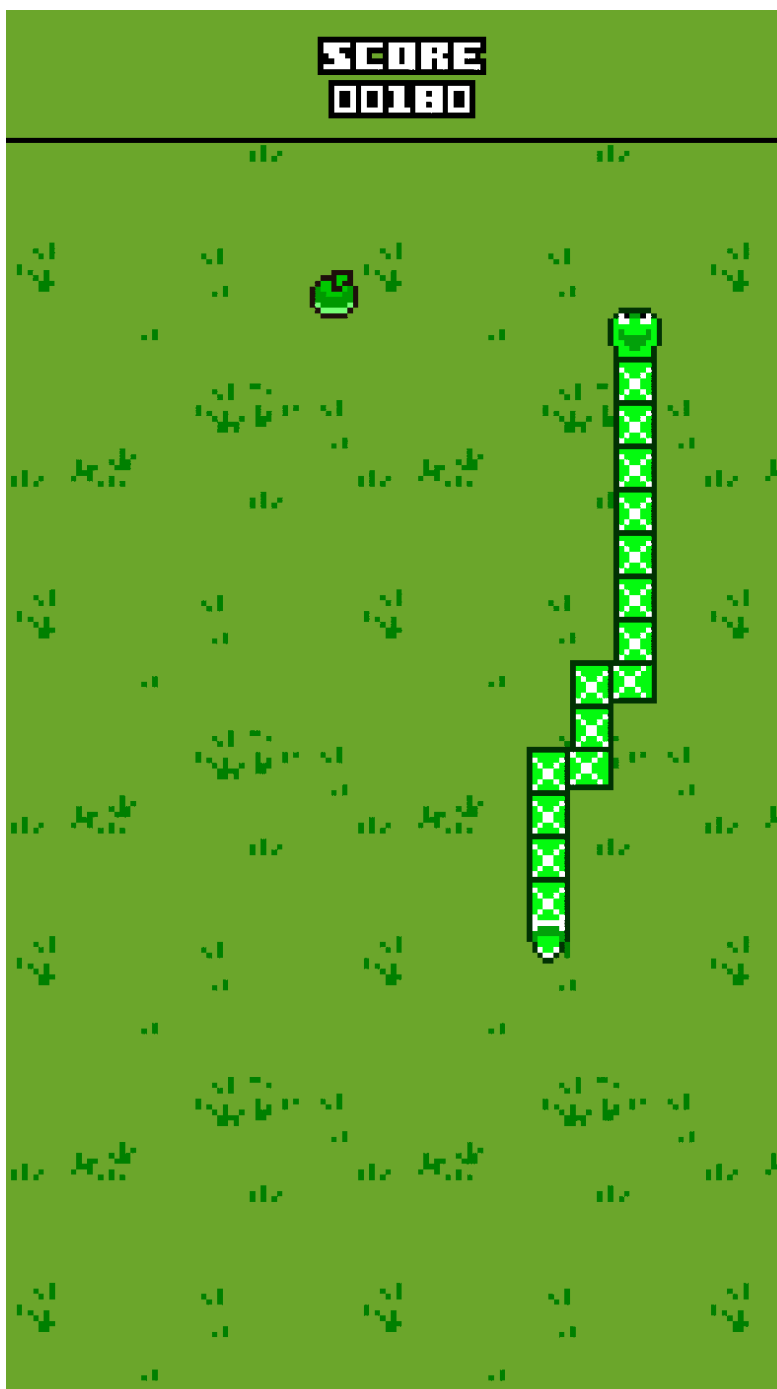

Definição do trabalho da M3 – SNAKE – Part II

Modalidade: em DUPLAS ou em TRIO. (trabalhos individuais perderão 20% da nota)

Visão Geral:



OBSERVAÇÃO

Os requisitos não implementados da primeira etapa terão desconto dobrado na segunda parte.

ENTREGA:

- *Postar um arquivo texto com o link para o trabalho no Github*

DEFESA:

- *O professor irá realizar a defesa do trabalho com o grupo, não havendo a necessidade de apresentar o trabalho para toda a turma.*
- *O professor poderá fazer perguntas sobre qualquer parte do código a qualquer um dos integrantes do grupo.*
- *Nota da apresentação será individual, além disso todas as notas relativas ao código dependem do desempenho na apresentação. Sem apresentação o trabalho terá nota ZERO.*
- *Para acelerar a apresentação dos trabalhos a mesma será feita em duas etapas:*
 - *Na primeira etapa da apresentação a equipe apresenta junto podendo ser feitas perguntas individuais. (a nota da apresentação ainda é individual)*
 - *Na segunda etapa da apresentação serão solicitadas modificações no código apresentado que deverão ser feitas de forma individual (sem auxílio dos outros integrantes da equipe).*

Critérios de Avaliação: Part 1

- *A Snake se move sozinha para a frente (0.4).*
- *Ao selecionar uma direção o movimento acontece sem delay (0.4).*
- *O score aparece em tela e é atualizado em tempo real (0.4).*
- *A snake morre ao colidir com a parede (0.4).*
- *A snake morre ao colidir no próprio corpo (0.4).*
- *A maçã aparece em posição aleatória em posição válida (0.4).*
- *A condição de vitória de 100 maçãs (0.4).*
- *O corpo da snake se move corretamente conforme movimentos anteriores (0.4).*
- *A snake cresce ao comer uma maçã (0.4).*
- *As paredes, snake e maçã são diferentes visualmente (0.4).*
- *A cobra começa com tamanho de 3 unidades (0.4).*
- *A velocidade da cobra aumenta após consumir 5 maçãs (0.4).*
- *O jogo salvar em arquivo o score (0.4).*
- *Exibir o Rank ordenado por score e nome (0.4).*
- *Deve ser exibido em tela o tempo de jogo (0.4).*

- *Uso de Struct (1.0)*
- *Uso de Sobrecarga e Valores Default (1.0)*
- *Uso de passagem por referência, funções void e de tipo (1.0)*
- *Uso de Arquivo para o rank com score e nome do jogador, data e tempo (1.0)*

1. *Funcionalidades = 60% da nota.*

- Só pontuará se souber explicar como foi implementado.*
- São 15 itens e cada item vale 0.4*

2. Técnicas = 40% da nota.

- a. Só pontuará se souber explicar como foi implementado.
- b. São 4 itens e cada item vale 1.0 pontos

Critérios de Avaliação: Part 2

- A Jogo deve ter ao menos 3 fases (1.2).
- Cada fase deve ter uma arena diferente (0.4)
- Deve-se ter itens especiais (0.4).
- Deve-se ter níveis de dificuldade (0.4).
- Os itens especiais devem contar para o score (0.4).
- Os níveis de dificuldade devem pontuar para o score (0.4).
- Deve-se ter a quantidade de movimentos em tela (0.4).
- A quantidade de movimentos deve contar para o score (0.4).
- Modo Desafio (2.0):
 - A cobra começa no tamanho máximo, se gera maçãs que diminuem e maçãs que aumentam o tamanho da cobra ambas alteram a pontuação.
 - Defina um tempo limite para alcançar a maior pontuação possível.
- Deve-se ter uma opção para o Computador/NPC jogar sozinho o jogo e ser capaz de ganhar e passar as fases (3.0).
- Apresentação para a turma na data da entrega (1.0).

3. Funcionalidades = 60% da nota.

- a. Só pontuará se souber explicar como foi implementado.
- b. São 15 itens e cada item vale 0.4

4. Técnicas = 40% da nota.

- a. Só pontuará se souber explicar como foi implementado.
- b. São 4 itens e cada item vale 1.0 pontos

Obs1: Cópia de soluções de colegas ou da internet acarretarão nota 0 para todos os envolvidos.
Obs2: Trabalhos entregues e/ou apresentados em atraso receberão um desconto inicial de 20% na nota final além de um desconto progressivo de 10% por semana de atraso. Ou seja, apresentar na semana seguinte a data estipulada deixa a nota máxima do trabalho em 70%.