Fundação CECIERJ - **Vice Presidência de Educação Superior a Distância**

Curso de Tecnologia em Sistemas de Computação
Disciplina: Computação Gráfica
AP1 - 2° semestre de 2016.

Nome –

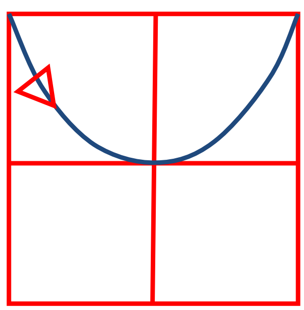
Assinatura –

Observações:

i) Prova sem consulta e sem uso de máquina de calcular.
ii) Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
iii) Você pode usar lápis para responder as questões.
iv) Ao final da prova devolva as folhas de questões e as de respostas.
v) Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

Na última página encontra-se a folha de respostas. Preencha corretamente e sem rasuras. Todas as questões tem o mesmo peso.

1)Em um jogo de celular, uma nave descreve uma trajetória parabólica que vai do canto superior esquerdo passa pelo centro da tela e termina no canto superior direito. Para simplificar, considere que a tela seja um quadrado com canto inferior direito igual a (-1.0,-1.0) e superior direito (1.0,1.0). Considerando que a cada instante de tempo t, a trajetória da nave deve ser atualizada, isto é , a cada frame, descreva um modo de definir os pontos que correspondem a trajetória da nave no tempo. (2.0 pontos)



No enunciado as coordenadas dos cantos foram trocadas.

Para resolver o problema basta primeiramente definir de forma paramétrica a parábola em função do tempo t. Deste modo, iremos descrever uma função que descreve como a partícula se move conforme o tempo varia. Uma parábola padrão, que passa pela origem, é expressa parametricamente como $f(t) = (t,t^2)$.

Em um dado instante de tempo T, durante o jogo, quando a nave aparece, fazemos o tempo inicial de avaliação da trajetória ser $t_i = 0$ e o tempo final ser igual a $t_f = 2.0$. Deste modo para obtermos os pares de coordenadas (x,y) sobre a parábola desejada usamos a função $f(t) = (t-1.0,(t-1.0)^2)$. Avaliamos a função em cada instante de tempo $t_0+\Delta t^k$, onde Δt^k é a quantidade de tempo decorrida desde o início da avaliação até o *frame* de número k, após o início do movimento da nave.

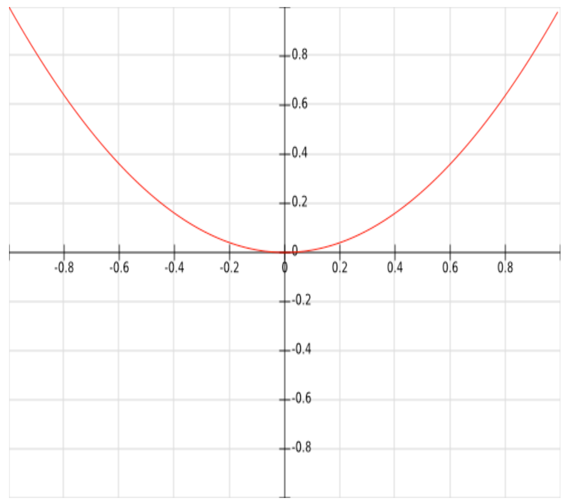


Gráfico gerado com a ferramenta Fooplot em http://fooplot.com

Obs: É possível alinhar a figura da nave com a trajetória usando o vetor tangente da parábola para determinar a matriz de rotação necessária para rotacionar a figura .Este processo não está no escopo desta questão.

2) Defina uma triangulação (2.0 pontos)

Uma triangulação é uma coleção de triângulos T tal que a interseção de dois triângulos T_i e T_j , $i \neq j$, que satisfazem:

- $T_i \cap T_j = \emptyset$.
- $T_i \cap T_j$ é vértice.
- $T_i \cap T_j$ é uma aresta **e** adjacente aos dois triângulos.

Nenhum outro tipo de elemento pode estar na interseção de dois triângulo de T.

3) Compare duas estruturas de dados para representar malhas (2.0 pontos)

Duas estruturas muito simples para representação de malhas são a **lista de faces (ou condificação explícit)** e a **lista de faces-vértices-arestas**.

Na **lista de faces** (**codificação explícita**), apenas são enumeradas as faces da malha e para cada face são listadas as coordenadas explícitas dos seus vértices. Isto faz com que a representação sofra problemas de redundância, possa ter problemas de inconsistência numérica, visto que as coordenadas de um mesmo vértice são replicadas e podem sofrer diferentes operações e acumulo de erro numérico distintos. Além disso, o desenho das arestas é duplicado no processo de renderização das mesmas.

Na **lista de faces-vértices-arestas**, são criadas três listas distintas: uma *lista de faces*, uma *lista de arestas* e uma *lista de vértices*. A *lista de faces* enumera todas as faces, porém, para cada face são descritos somente os índices das arestas que a determinam; índices de onde as arestas se encontram na *lista de arestas*. A *lista de arestas* contém cada uma das arestas da malha, onde para cada aresta são armazenados os índices, na *lista de vértices*, dos dois vértices nos quais ela incide. Finalmente, a *lista de vértices* armazena as coordenadas de todos os vértices da malha. Este tipo de estrutura acaba com o problema de redundância e inconsistência numérica que ocorre na representação usando somente a lista de faces explícita. Ela é a base para construção de estruturas topológicas mais complexas, que armazenam outras informações de adjacência como a Winged-edge, Half-edge, Radial-edge, etc.

4) Que tipo de objeto gráfico é adequado para descrever estruturas internas de um dado de medicina, capaz de permitr a análise de estruturas internas, como nódulos e cistos (2.0).

A representação mais adequada é por objetos espaciais tridimensionais, mais comumente, representações volumétricas uniformes, onde os elementos são voxels. É também comum em representações mais simples, usar imagens que são fatias, (um subconjunto) da estrutura tridimensional, neste caso os objetos gráficos são planares e bidimensionais.

5) Descreva as vantagens e desvantagens de se utilizar um modelo baseado em *voxels* (decomposição espacial) para descrever um sólido (2.0 pontos).

As vantagens estão na regularidade da estrutura, que pode ser descrita computacionalmente como uma matriz. Cada elemento é identificado então com índices i,j,k , tornando fácil sua manipulação. Como tem estrutura matricial, praticamente todas as operações feitas sobre imagens como, aplicação de filtros, métodos de segmentação, e outros, podem ser facilmente estendidas para o dado volumétrico. O cálculo de medidas geométricas como volume, área de regiões também é facilitado pela natureza discreta.

Como desvantagens podemos destacar a necessidade de resolução alta para descrição de estruturas e detalhes finos, o que leva a uma necessidade de espaço de armazenamento considerável. Isto faz com que a complexidade de alguns algoritmos cresça consideravelmente, uma vez que o número de voxels utilizados para descrever o objeto cresce rapidamente conforme a resolução utilizada. Por fim, como toda representação discreta, existe o problema de *aliasing* na representação.