



Snake Ship 3D

- Cenário Marítimo
- Barco representa cobra e tesouros representam frutas
- Conforme o barco conquista tesouros ele cresce
- O objetivo é conquistar o mar inteiro
- O barco não pode colidir consigo mesmo senão afunda
- O barco não pode colidir com a praia senão encalha

ABORDAGEM

- C++11 e OpenGL
- Orientação a Objeto
- Listas
- Vectors
- Threads
- Rand
- Enum

RECURSOS

- GLuint
- gcallList()
- glGenList(1)
- glNewList()
- Viewport
- glLightfv(GL_LIGHT1, GL_AMBIENT, luzAmbiente);
- glLightfv(GL_LIGHT1, GL_DIFFUSE, luzDifusa);
- glLightfv(GL_LIGHT1, GL_POSITION, psosicaoDaLuz);
- glutTimerFunc(100, Anima, 1);
- gluLookAt(0 , yCamPos , -10, 0,0,0 , 0,1, 0);
- GlTranslate(), GlRotate() e GlScale()

ESTRUTURA

- Quatro classes: Barco, Tesouro e Container e Principal
- Barco é um conjunto de containers
- Conforme barco conquista tesouros incrementam-se containers e o barco cresce
- O primeiro, o do meio e o ultimo containers são especiais, pois representam proa, vela e popa respectivamente

OBJETOS

Barco

- direcaoAtual
- ehAtualizado
- pontuacao

Container

- X
- Y
- Z
- SIZE

Tesouro

- X
- Y
- angle

***Todos são inicializados com posições iniciais nos construtores**

MÉTODOS

Classe Principal

- constroiPraia()
- constroiPontas()
- constroiVela()
- desenhaMar()
- detalhesDoBarco()
- redimensionaTela()
- draw()
- specialKeys()
- initGL()
- mouseEvents()
- Anima()
- main()

Classe Barco

- draw()
- move()
- pushBarco()
- update()
- deteccaoDeColisao()
- addContainer()

Classe Container

- drawnormal()
- drawvela()
- drawPonta()

Classe Tesouro

- draw()

DESENHOS

- Barco – Três tipos de containers, o padrão cúbico (GL_QUADS), os das pontas triangulares e o central com haste e vela (GL_POLYGON)
- Praia – Containers cúbicos padrão (GL_Quads)
- Céu - `glClearColor(0.3f, 0.3f, 0.8f, 0.5f);`
- Sol e nuvens - `glutSolidSphere();`

DESENHOS

- Vela – Container especial com haste (GL_POLYGON) e bandeira (GL_TRIANGLES)
- Proa e Popa– Container especial (GL_POLYGON) com ponta representando extremo do barco
- Tesouro - Cubo (GL_QUADS) dourado

ALGORITMO

- Lista de containers: `std::vector<Container*> containers;`
- Inicialização do barco: adiciona no vector de containers os 6 containers iniciais
- addContainer: `containers.push_back(novoContainer);`
- Draw: Percorre vector de containers e os desenha como cubo padrão exceto os identificados como no centro da lista e nos extremos
- push_barco: Conforme o barco movimenta o container a frente assume a posição do de trás e assim sucessivamente
- Update: Atualiza constantemente direção e colisão

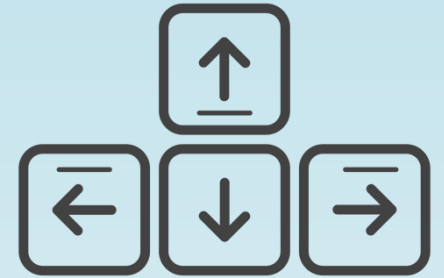
ALGORITMO

- Threads: Auxiliam nos delays de update e game over, exemplo:
`std::this_thread::sleep_for(std::chrono::milliseconds(100));`
- Processo do container vela do barco:
 1. Glint vela;
 2. `vela = glGenLists(1); glNewList(vela, GL_COMPILE);`
 3. `void draw(){ glTranslatef(-x,y,z); glCallList(cubopadrao); }`
 4. `for(int i = 1; i <= containers.size()-2; i++){`
 `Container* temp = containers.at(i);`
 `if (i == (containers.size() - 1) / 2) {`
 `temp->drawvela(); } else { temp-> drawnormal(); }`

ALGORITMO

- Move: movimenta o barco conforme as setas do teclado indicam a direção do Enum, exemplo:

```
if(direcao == D_LEFT && direcaoAtual != D_RIGHT){  
    pushBarco();  
    proa->setX(proa->getX()-0.5f);  
    direcaoAtual = direcao;  
    ehAtualizado = true;
```



- Deteccção de colisão: Pega as posições atuais da proa e compara com a posição do tesouro constantemente se for igual incrementa o score e inicializa de novo o tesouro aleatoriamente

ILUMINAÇÃO

- `GLfloat luzDifusa[] = {1,1,1,1};` - Valores da luz difusa
- `GLfloat luzAmbiente[] = {0.5,0.5,0.5,1};` - Valores da luz ambiente
- `GLfloat posicaoDaLuz[] = {0,3,0};` - Posição da fonte de luz
- `glShadeModel(GL_SMOOTH);` - Sombreamento
- `glEnable(GL_DEPTH_TEST);` - Teste de profundidade
- `glEnable(GL_LIGHTING);` - Ativa a luz e seus cálculos
- `glLightfv(GL_LIGHT1, GL_AMBIENT, luzAmbiente);` - Define parâmetros da fonte de luz
- `glEnable(GL_COLOR_MATERIAL);` - Rastreamento de cor

ANIMAÇÃO E TRANSFORMAÇÕES

Animação Giratória do Tesouro

- tesouro -> draw();
- angle+=2.0f;
- glutPostRedisplay();
- glutTimerFunc(10, Anima, 1);
- glutTimerFunc(100, Anima, 1);



Transformações

- Rotação do sol e das nuvens
- Translação no movimento do barco
- Escala do tesouro, da proa e popa para ajustes

CÂMERA E PROJEÇÃO

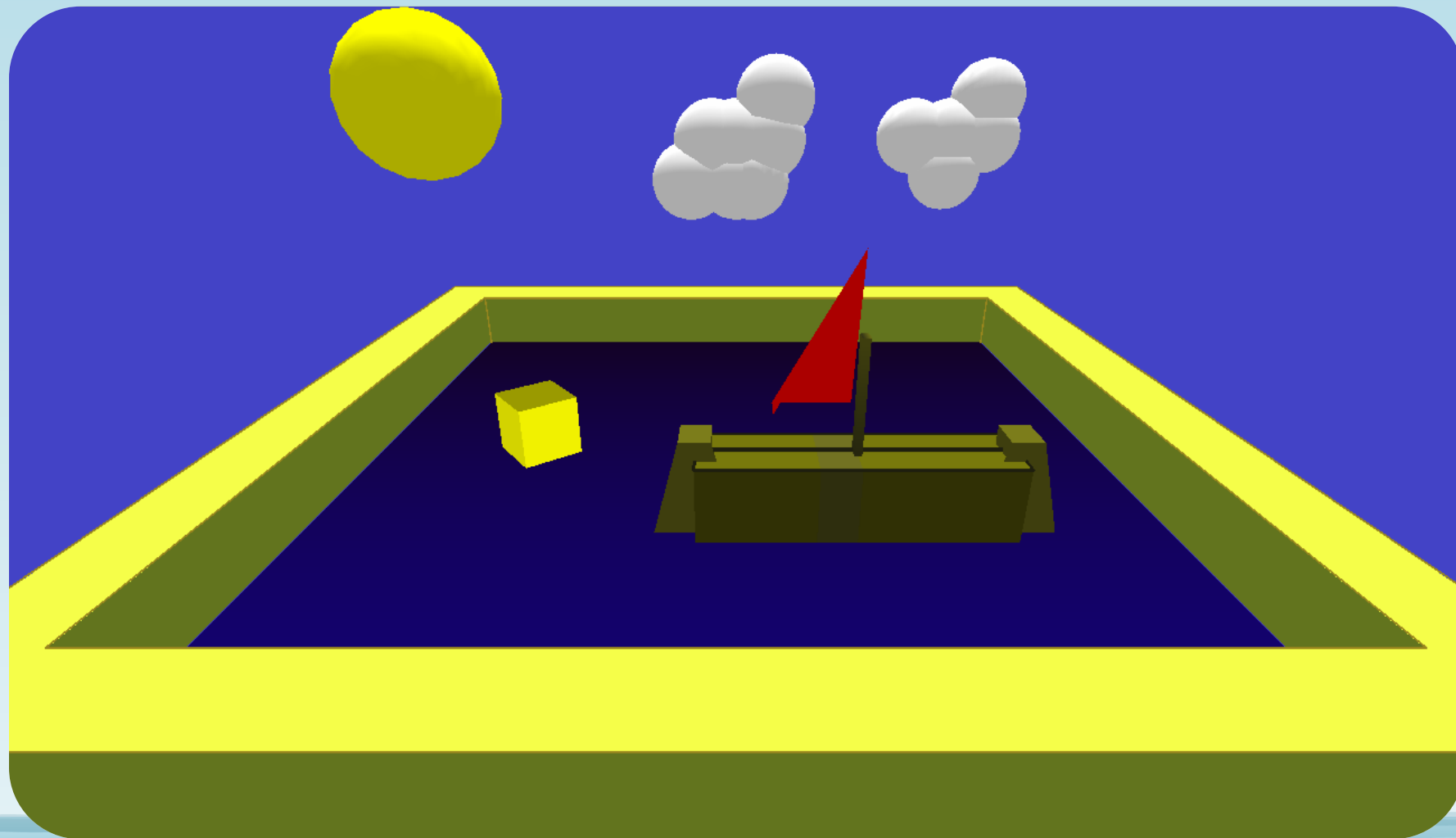
Configura a Câmera

- `gluLookAt(0 , yCamPos , -10, 0,0,0 , 0,1, 0);`
- Mouse em scroll para cima: `yCamPos -= 0.5;` (zoom in)
- Mouse scroll para baixo: `yCamPos += 0.5;` (zoom out)

Mar em Perspectiva

- `glMatrixMode(GL_PROJECTION);` - Especifica pilha de projeção
- `glViewport(0,0,w,h);` - Recorte para o mar
- `gluPerspective(60,raio,0.5,60);` - Configura matriz de projeção

JOGO EM EXECUÇÃO



CONSIDERAÇÕES FINAIS

- Referência aos Kuja Pirates
 - Referência ao mar cáspio (Ásia Ocidental)

