

Universidade de São Paulo

Instituto de Matemática e Estatística
Departamento de Ciência da Computação

MAC0346 - Programação para Jogos Digitais

Relatório EP1

Andrés Eduardo Marchetti Colognesi - 9838161
Arthur Pires da Fonseca - 10773096

SÃO PAULO - SP

Sumário

1. Descrição dos módulos	3
1.1 Main	3
1.2 AuxLoader	3
1.3 AuxLayer	4
1.4 Matrix	4
2. Tarefas completadas	4
3. Observações adicionais	5

1 Descrição dos Módulos

Nosso programa consiste de quatro módulos. Eles são os arquivos `main.lua`, `auxLoader.lua`, `auxLayer.lua` e `matrix.lua`.

1.1 Main

Este módulo chama a tabela provinda do arquivo `test.lua` de MAP.

Usa as funções dos módulos `auxLoader` e `auxLayer` como apoio e possui 3 funções próprias:

- `love.load`: Carrega os Quads dos blocos feitos no Tiled e dos *sprites*. Também carrega as câmeras. Todos esses elementos estão descritas no arquivo `test.lua`

- `render`: Renderiza o que foi carregado em `love.load`, fazendo uso das funções implementadas nos módulos `auxLoader` e `auxLayer`.

- `love.draw`: Desenha na tela a cada *frame* os blocos e os *sprites*, chamando a função `render`. Muda os eixos de desenho da tela e ativa o modo de tela cheia para a visualização integral do mapa.

1.2 AuxLoader

Este módulo possui 5 funções:

- `AUXLOADER.format`: Formata uma string de modo a descrever o local exato no projeto löve onde um certo arquivo será acessado.

- `AUXLOADER.blocks`: Para um bloco de índice *index* na *tilesheet*, devolve as coordenadas de posicionamento dos *tiles*.

- `framesTable`: Função local, auxiliar da `spriteQuads`.

-AUXLOADER.spriteQuads: A partir da tabela MAP, devolve uma tabela completa com os Quads de cada *sprite*, indexados por seus nomes próprios.

-AUXLOADER.objects: A partir das camadas do mapa MAP, retorna duas tabelas, *sprites* e *cameras*, que indexam os objetos de forma ordinal e nominal, respectivamente.

1.3 AuxLayer

Chama o módulo matrix.lua para auxiliar o posicionamento dos elementos do mapa.

-AUXLAYER.tilelayer: Desenha os *tiles* em suas posições, auxiliando a função render do módulo main.

-AUXLAYER.sprite: Desenha os *frames* de uma *sprite* em sua posição no mapa, mudando o *frame* apenas quando os campos newTime e oldTime do objeto obj diferirem do tempo de um *frame*. Adiciona uma sombra embaixo de cada *sprite* também.

-AUXLAYER.camera: A partir de uma câmera, translada o sistema de coordenadas para permitir a visualização das partes do mapa.

1.4 Matrix

Possui 1 função:

-MATRIX.linearTransform: Função de apoio para o módulo auxLayer, realizando a projeção do mundo 3D do mapa do jogo para a tela 2D da löve.

2 Tarefas Completadas

Foram realizadas as tarefas A1, A2, A3, R1, R2, R4, Q1, Q2, Q3 e Q4.

3 Observações Adicionais

A função `love.draw`, do módulo `main.lua`, translada o eixo de coordenadas e configura a löve para o modo de tela cheia para que o mapa inteiro possa ser visualizado, já que as câmeras não funcionam.

O *sprite* das tartarugas não é recortado corretamente em todos os *frames*, pois recortamos de forma generalizada as imagens dos *sprites*, e a imagem que contém as tartarugas não é regular quanto à altura de cada linha, portanto alguns cortes pegam uma parte do frame de cima na imagem, o que cria um risco a mais onde a tartaruga está na tela.

A função `AUXLAYER.sprite` não conseguia posicionar corretamente as *sprites*, portanto transladamos os eixos de coordenadas empiricamente até que as *sprites* estivessem ajustadas como no *gif* do enunciado.

A função `AUXLAYER.camera` não tem efeito algum no cenário, pois translada o sistema de coordenadas e desfaz essa transformação logo em seguida, portanto não pode ser percebido na mudança dos *frames*.

Implementamos a passagem do tempo entre frames usando a função `love.timer.getTime`, que foi usada na função `AUXLAYER.sprite` na linha 55 do módulo `auxlayer.lua`.

Acreditamos que a tarefa Q3 foi realizada parcialmente, pois todas as funções que fizemos possuem em torno de 15 linhas. Algumas excedem levemente o limite pois têm espaços vazios, que lá estão para leitura fluida do código. As únicas funções que realmente extrapolam os limites são `auxLayer.sprite`, com 35 linhas, a qual não poderia ser menor pois tem várias variáveis locais que melhoram a leitura do código e `AUXLOADER.spriteQuads`, a qual ficou com 40 linhas

porque não conseguimos desacoplar as funcionalidades que ela promove.