

Animacão de Personagens

Programacão de Jogos

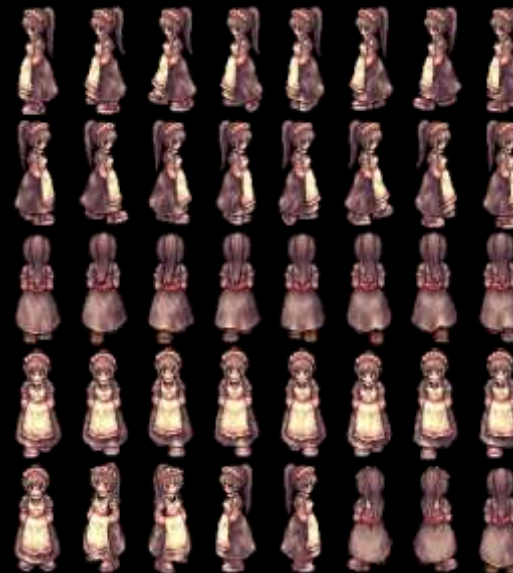
Judson Santos Santiago

Introdução

- ▶ A animação de um personagem se difere de uma animação convencional pela necessidade de guardar o **estado do personagem**

```
// estados do personagem Lady
enum LadyState
{
    STILL,
    WALKUP,
    WALKDOWN,
    WALKLEFT,
    WALKRIGHT
};

// estado atual do personagem
uint state;
```

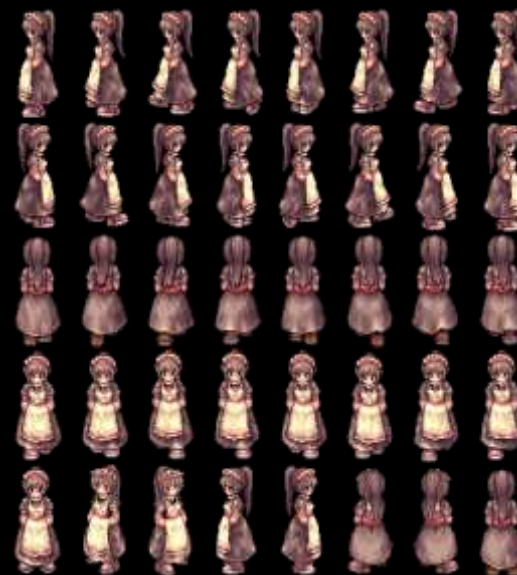


Quadros da Animação

Introdução

- ▶ A animação de um personagem se difere de uma animação convencional pela necessidade de definir **sequências de animação**

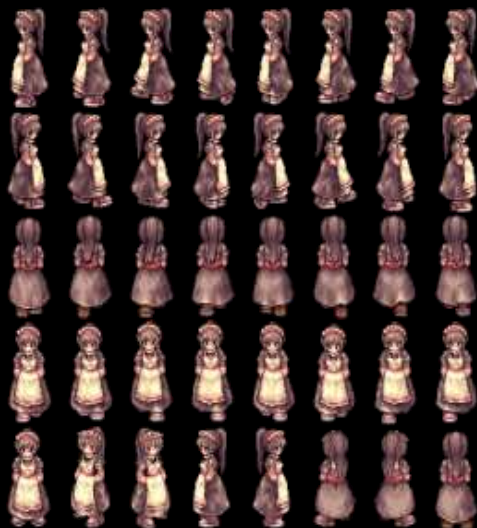
```
// sequências de animação
uint SeqLeft[8]  = {0,1,2,3,4,5,6,7};
uint SeqRight[8] = {15,14,13,12,11,10,9,8};
uint SeqUp[8]    = {16,17,18,19,20,21,22,23};
uint SeqDown[8]  = {24,25,26,27,28,29,30,31};
uint SeqStill[1] = {32};
```



Quadros da Animação

Sistema de Animação

- Um sistema de animação de personagens precisa armazenar as sequências associadas a cada estado do personagem



Quadros da Animação

```
// sequência <vetor, tam>  
using AnimSeq = pair<uint*, uint>;
```

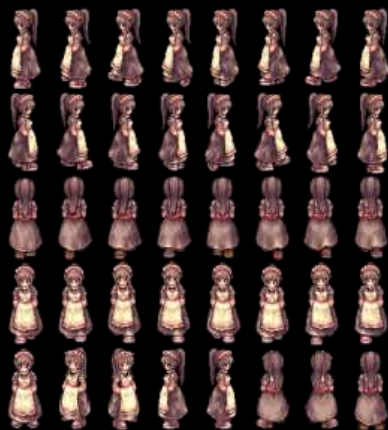
WALKLEFT	0	1	2	3	4	5	6	7	8
WALKRIGHT	15	14	13	12	11	10	9	8	8
WALKUP	16	17	18	19	20	21	22	23	8
WALKDOWN	24	25	26	27	28	29	30	31	8
STILL	32								1

```
// tabela de dispersão <state, AnimSeq>  
using HashTable = unordered_map<uint, AnimSeq>;
```

Sistema de Animação

- Para suportar o sistema de sequências e o sistema antigo é preciso verificar se existe uma sequência ativa na hora de **desenhar uma animação**

```
// desenha quadro atual da animação  
void Animation::Draw(float x, float y, float z)  
{ sequence ? Draw(sequence[frame], x, y, z) : Draw(frame, x, y, z); }
```



	0	1	2	3	4	5	6	7	
WALKLEFT	0	1	2	3	4	5	6	7	8
WALKRIGHT	15	14	13	12	11	10	9	8	8
WALKUP	16	17	18	19	20	21	22	23	8
WALKDOWN	24	25	26	27	28	29	30	31	8
STILL	32								1

Adicionando Sequências

```
void Animation::Add(uint id, uint * seq, uint seqSize)
{
    // cria nova sequência de animação (vetor, tam)
    AnimSeq newSeq(new uint[seqSize], seqSize);

    // copia sequência de quadros da animação
    memcpy(newSeq.first, seq, sizeof(uint) * seqSize);

    // insere nova sequência na tabela
    table[id] = newSeq;

    // seleciona sequência recém inserida
    sequence = newSeq.first;

    // ajusta quadro inicial e final
    iniFrame = frame = 0;
    endFrame = seqSize - 1;
}
```

	0	1	2	3	4	5	6	7	
WALKLEFT	0	1	2	3	4	5	6	7	8
WALKRIGHT	15	14	13	12	11	10	9	8	8
WALKUP	16	17	18	19	20	21	22	23	8
WALKDOWN	24	25	26	27	28	29	30	31	8
STILL	32								1

Selecionando Sequências

```
void Animation::Select(uint id)
{
    const auto & [seq, size] = table[id];

    // se é uma nova sequência
    if (sequence != seq)
    {
        // aponta para nova sequência
        sequence = seq;

        // reinicia a sequência
        iniFrame = 0;
        endFrame = size - 1;

        // se o frame atual está fora da sequência
        if (frame > endFrame)
            frame = 0;
    }
}
```

	0	1	2	3	4	5	6	7	
WALKLEFT	0	1	2	3	4	5	6	7	8
WALKRIGHT	15	14	13	12	11	10	9	8	8
WALKUP	16	17	18	19	20	21	22	23	8
WALKDOWN	24	25	26	27	28	29	30	31	8
STILL	32								1

Resumo

- ▶ Um **sistema de animação** pode ser usado com:
 - Efeitos: explosão, sirene, etc.
 - Interface: barra de progresso, destaque de elementos, etc.
 - Personagens: movimentação, estado, etc.
- ▶ Um sistema de **animação de personagens** precisa:
 - Armazenar sequências de quadros
 - Associar o estado do personagem às sequências
 - Desenhar quadros da sequência selecionada