Animação Comportamental de personagens com máquina de estados

Eduardo Eidelwein Berlitz

Introdução

- Máquina de estados levando em consideração estado interno (necessidades, emoções, personalidade etc) - "The Sims"
- Adaptar o primeiro trabalho do GB
 - disparar um conjunto animações conforme atributos internos do personagem
 - Utilizar uma máquina de estados

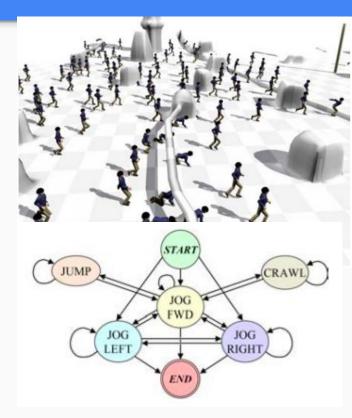
Trabalhos relacionados

- Lau, Manfred and James J. Kuffner. "Behavior planning for character animation." Symposium on Computer Animation (2005).
- Ho, Edmond S. L. and Taku Komura. "A finite state machine based on topology coordinates for wrestling games." Journal of Visualization and Computer Animation 22 (2011): 435-443.
- Bourse, Yoann. "Artificial intelligence in the Sims series." (2012).

Trabalhos relacionados - Behavior planning for character animation

Explora a utilização de uma máquina de estados em conjunto com a descoberta de um caminho para chegar de um ponto A a B animando de forma coerente o personagem. Como desviando de obstáculos, se inclinar, engatinhar, caminhar, correr, pular.

Meu trabalho não faz uso dos algoritmos de **Path Finding** portanto também não precisa explorar a possibilidade de animações referente a estados e transições conforme o melhor caminho. Mas faço uso da **máquina de estados** para realizar as **animações** porém de acordo com **propriedades do personagem**.



Lau, Manfred and James J. Kuffner. "Behavior planning for character animation."

Symposium on Computer Animation (2005).

Trabalhos relacionados - A finite state machine based on topology coordinates for wrestling games

O trabalho utiliza de máquina de estados para definir as diferentes posições de dois personagens de luta livre de forma que os jogadores possam ter uma certa liberdade na escolha dos próximos movimentos de forma que a visualização dos personagens seja o mais real possível.

Se relaciona com o meu trabalho da mesma forma do artigo anterior.

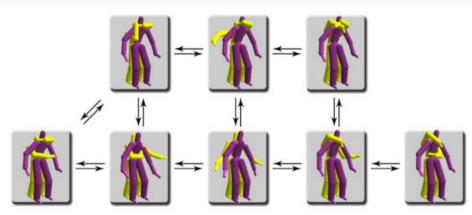


Figure 5. A finite state machine of two people wrestling when one character at the back of the other.

Ho, Edmond S. L. and Taku Komura. "A finite state machine based on topology coordinates for wrestling games." Journal of Visualization and Computer Animation 22 (2011): 435-443.

Trabalhos relacionados - Artificial intelligence in the Sims series

Segundo estudo o The sims utiliza de atributos internos para a tomada de decisões de cada personagem.

No meu trabalho, utilizo do mesmo conceito porém apenas com dois atributos, sendo eles **Felicidade** e **Energia**.

Modeling human needs

8 basic needs evolving through time, under the influence of circumstances (sleeping? eating?):

Physical

- Hunger (eating)
- Comfort (sitting/laying down)
- Hygiene (bathing)
- Bladder (urinating)

Mental

- Energy (sleeping)
- Fun (playing)
- Social (interacting with others)
- Room (architecture, furniture)

Bourse, Yoann. "Artificial intelligence in the Sims series." (2012).

Metodologia

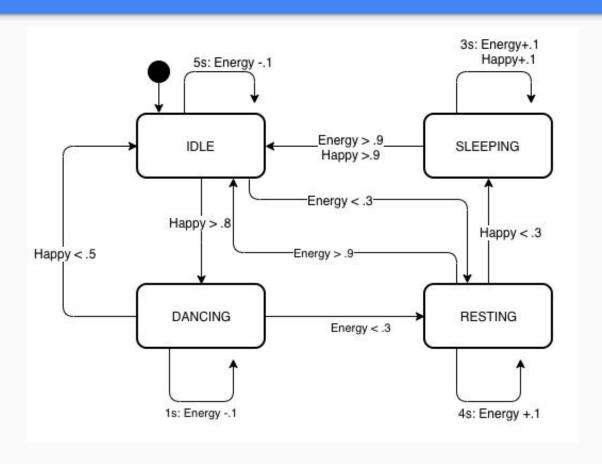
Ferramentas

- Three.js
- javascript-state-machine
- Mixamo

Etapas

- Planejar a máquina de estados
- Baixar modelo do personagem 3D
- Baixar as animações
- Programar a máquina de estados
- Configurar as transições para iniciar cada tipo de animação

Metodologia - Máquina de estados



Metodologia - Implementação

- Animações obtidas pelo Mixamo
 - Ocioso
 - Dançar
 - Deitar e levantar
 - Sentar e levantar
- Utilizado o FBXLoader do Three.js para carregar o personagem 3D e as animações
- Utilizado a classe AnimationClip do Three.js para configurar as animações (loop infinito ou não, morphing).
 - Apenas as animações Ocioso e Dançar, ficam animando em um loop infinito, as demais apenas são executadas durante a transição dos estados.
- Utilizado a classe StateMachine do javascript-state-machine para descrever os estados e transições assim como as animações que devem ser iniciadas

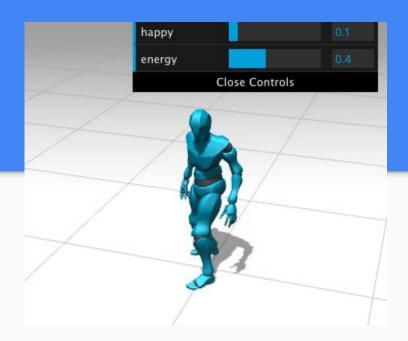
Metodologia - Implementação StateMachine + AnimationAction

```
var fsm = new StateMachine({
 init: 'idle',
 transitions: [
    { name: 'dance', from: 'idle', to: 'dancing' },
   { name: 'sitFromDancing', from: 'dancing', to: 'resting' },
   { name: 'idleFromDancing', from: 'dancing', to: 'idle' },
   { name: 'sitFromIdle', from: 'idle', to: 'resting' },
   { name: 'stand', from: 'resting', to: 'idle' },
   { name: 'goSleep', from: 'resting', to: 'sleeping' },
    { name: 'wakeUp', from: 'sleeping', to: 'idle' },
 methods: {
    onLeaveState: () => disposables.splice(0).forEach(a => a()),
   onEnterSleeping:() => {
      disposables.push(every(3000, () => {
       if (innerState.energy <= 0.9) {
          innerState.energy += 0.1;
       if (innerState.happy <= 0.9) {
          innerState.happy += 0.1;
     })):
      return actions.lav();
    onLeaveSleeping:() => actions.stand2(),
```

```
async function updateAnimations() {
  if (fsm.is('dancing') && innerState.energy < 0.3) {
    fsm.sitFromDancing();
} else if (fsm.is('idle') && innerState.energy < 0.3) {
    fsm.sitFromIdle();
} else if (fsm.is('resting') && innerState.happy <= 0.3){
} else if (fsm.is('resting') && innerState.energy >= 0.9) {
    fsm.stand();
} else if (fsm.is('idle') && innerState.happy >= 0.8) {
    fsm.dance();
} else if (fsm.is('dancing') && innerState.happy < 0.5) {
    fsm.idleFromDancing()
} else if (fsm.is('sleeping') && innerState.energy >= 0.9 && innerState.happy >= 0.9){
    fsm.wakeUp();
}
```

```
activeAction
   .reset()
   .setEffectiveTimeScale(1)
   .setEffectiveWeight(1)
   .fadeIn(duration)
   .play();
```

Resultados



Demo:

https://eberlitz.github.io/animations/TGB2/dist/

• Repo:

https://github.com/eberlitz/animations/tree/master/TGB2

Considerações Finais

Pontos fortes:

 Relativamente simples de implementar para um bom ganho no comportamento do personagem

pontos fracos:

- Morphing é falho
- Adicionar mais atributos internos ao personagem requer muitas transições e novos estados - possivelmente utilizar

possíveis pontos a serem melhorados no futuro:

- Morphing entre as animações
- Integrar com a movimentação do personagem no espaço

Referências

- Bourse, Yoann. "Artificial intelligence in the Sims series." (2012).
- Lau, Manfred and James J. Kuffner. "Behavior planning for character animation." Symposium on Computer Animation (2005).
- Ho, Edmond S. L. and Taku Komura. "A finite state machine based on topology coordinates for wrestling games." Journal of Visualization and Computer Animation 22 (2011): 435-443.

https://threejs.org/

https://github.com/jakesgordon/javascript-state-machine https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867