

# Design de Jogos Digitais

## Aula 03 - Mecânicas de jogos

# Apresentação

---

Olá, pessoal! Na aula de hoje iniciaremos o estudo dos elementos da Tétrade de *Schell* com enfoque naquele mais próximo da equipe de design: as mecânicas de jogos.

As mecânicas representam os componentes, regras e sistemas que construímos para permitir que o jogador interaja com o mundo do jogo e avance em busca dos objetivos finais. Isso define como o jogo será jogado!

Vamos usar a nossa famosa mecânica de questões-chave:

- O que são mecânicas de jogos? Para que servem?
- Qual a importância da interação do jogador com as mecânicas?
- Quais tipos diferentes de mecânicas existem?

Hora de colocarmos os nossos jogos para funcionar!

## Objetivos

- Conceituar Mecânicas de Jogos
- Apresentar as mecânicas de espaço, objetos/estados e tempo

# Mecânicas de Jogo

---

O que você acha que são mecânicas de jogos? Vamos tentar uma primeira definição:

**Figura 01** - Não são essas mecânicas.



Ei, tipo errado de mecânicas! Então, como poderíamos definir o que são mecânicas de um jogo? Recorrendo à literatura sobre o assunto, existem diversas definições para o termo, então vamos observar algumas e destacar quais os aspectos interessantes que elas abordam.

Rouse (2005) define as mecânicas de jogo de forma bem sucinta e direta: *"mecânicas são as coisas que o jogador pode fazer no mundo do jogo"*. Embora pareça simples, essa definição aponta um aspecto importante das mecânicas: elas representam as ações que o jogador pode executar dentro do jogo e permitem que ele interaja com o mundo virtual.

Já Lundgren e Bjork (2003) apresentam uma abordagem diferente: *"as mecânicas são um sumário das regras que compõem o jogo e definem o que ocorre em uma interação do jogador com o ambiente"*. Ora, por essa definição, as mecânicas também podem ser vistas como as regras ou sistemas de regras que definem os resultados das ações que o jogador efetua.

Essas definições se complementam, evidenciando a importância que as mecânicas de jogos possuem para conferir um poder de transformação ao jogador, ou seja, as ações e seus resultados permitem que o jogador se veja como um elemento

transformador do mundo do jogo e dá significado às suas interações dentro do mesmo.

Cook (2005) diz que as mecânicas podem ser utilizadas para “*permitir que o jogador aprenda sobre o espaço a ser explorado no jogo*”, evidenciando a capacidade das mecânicas de serem utilizadas como ferramentas de instrução para o jogador. Dificilmente o jogador saberá “de cara” todas as formas como pode influenciar o jogo, mas através das mecânicas ele pode ser gradualmente instruído sobre o que pode ou não ser feito. Isso permite que ele construa um conjunto de habilidades gradualmente à medida que progride no jogo.

Essa definição está atrelada ao conceito de *scaffolding*, abordado na aula sobre elementos de jogos: com uma apresentação dosada das possíveis ações do jogo, o jogador passa por um processo de aprendizado sem ser sobrecarregado e evolui de forma natural no jogo, exercitando devidamente cada nova habilidade adquirida antes de ser apresentado a um novo elemento. Assim, quando se depara com mecânicas e comandos mais complexos, ele já está preparado para executar tais ações.

Outra definição interessante é a de *Jarvinen* (2008), ao explicar que as mecânicas de jogos são “formas de direcionar o comportamento do jogador a realizar interações específicas para alcançar objetivos do jogo”. Assim, as mecânicas podem ser utilizadas para criar um certo direcionamento rumo aos objetivos do jogo. Isso pode ser feito mesmo sem retirar a sensação de liberdade e domínio do jogador. Ademais, essa definição ressalta a noção de progressão: sempre que o jogador interage com o jogo através das mecânicas, ele está alterando o estado do jogo, e isso ocorre de forma contínua até que atinja um estado final de vitória ou derrota. Esse aspecto é diretamente relacionado com a expressão original *game mechanics*: é como se as mecânicas fossem, na verdade, as engrenagens do jogo e, ao ativá-las continuamente, o jogador faz com que o jogo funcione como desejado.

Será que dá para sintetizar essas ideias em uma definição mais enxuta? Vamos observar uma caixinha especial para melhor entendermos:

Podemos dizer que as mecânicas de jogos são os elementos que regem a interação do jogador de forma que ele possa alterar o estado em que o jogo se encontra e progredir em direção às condições de vitória do jogo. São as pequenas engrenagens que, quando somadas, definem a jogabilidade do nosso jogo e também servem para educar o nosso jogador sobre o que ele pode e deve fazer. Nada mal, não acha?

Entendeu como funcionam as mecânicas de jogos? Para melhorar a compreensão, vamos indicar alguns exemplos de mecânicas em jogos existentes:

- No jogo do Mario, uma das mecânicas básicas é a de pular. Com ela o Mario pode superar obstáculos (barreiras, buracos), derrotar inimigos e coletar itens. Ou seja, através da ação de pular, o jogador consegue fazer com que o personagem do Mario avance ao longo dos níveis até chegar ao chefe final!

**Figura 02** - Cuidado para não pular da frigideira direto para o fogo, Mario!  
*Jogo: Super Mario World.*



**Fonte:**

<http://www.destructoid.com/blogs/Captain%20of%20the%20Zanarkand%20Abes>

Acesso em: 25 Ago 2015.

- Em um jogo de xadrez, a movimentação de cada peça é uma mecânica específica. O cavalo só se move em padrão de L, enquanto a torre pode se mover nas quatro direções em padrão de cruz. A movimentação define em quais casas a peça pode ser colocada e quais peças ela pode atacar para conquistar.

**Figura 03** - Minha vez!



**Fonte:** <http://www.chess.com/article/view/the-9-best-chess-gifs>. Acesso em: 25 Ago 2015.

- No jogo X-COM, a tomada de decisão ocorre em turnos alternados entre jogador e computador. Essa alternância é uma mecânica de jogo, e as ações que são tomadas em um turno por uma das partes é essencial para que o oponente trace a melhor estratégia de contra-ataque.

**Figura 04** - Em XCOM, humanos e alienígenas alternam os ataques, até que um lado seja vencedor.

**Jogo:** X-COM: Enemy Unknown.



**Fonte:** <https://www.youtube.com/watch?v=y0iNNY2UjZY>. Acesso em: 25 Ago 2015.

- No Street Fighter, a barrinha de golpe especial é uma mecânica a qual permite que o jogador execute um movimento poderoso capaz de reverter uma situação adversa. À medida que o personagem é atingido ou executa golpes especiais, a barra vai incrementando até que fique cheia, liberando a habilidade especial para o jogador.

**Figura 05** - ↓↘→ + Soco.

**Jogo:** *Street Fighter*.



**Fonte:**

[http://streetfighter.wikia.com/wiki/File:Ryu\\_and\\_ken\\_hadoken\\_double\\_hit.gif](http://streetfighter.wikia.com/wiki/File:Ryu_and_ken_hadoken_double_hit.gif).

Acesso em: 25 Ago 2015.

De modo geral, as mecânicas podem ser divididas em dois grandes grupos: mecânicas **primárias** (*core mechanics*) e mecânicas **secundárias** (*lesser mechanics*). As mecânicas primárias são as ações ou regras que constituem o núcleo central da jogabilidade: elas são essenciais para o progresso no jogo e são usadas com bastante frequência. Já as mecânicas secundárias servem para dar suporte às mecânicas primárias, não sendo obrigatórias para o progresso do jogador. As mecânicas secundárias podem, ainda, alterar uma mecânica primária ou afetar outros aspectos do jogo.

Voltando ao jogo de Mario, nós podemos considerar que a mecânica de pulo é uma mecânica principal: sem executá-la, o jogador não consegue avançar nem no primeiro nível do jogo. Já os objetos como cogumelos, flor de fogo e pena podem ser considerados como uma mecânica secundária: apesar de serem úteis e facilitarem a vida do jogador (inclusive permitindo que ele tenha mais chances de errar ao longo do jogo), eles não são itens obrigatórios para que o jogador consiga progredir até as fases finais do jogo. Quando o objeto adquirido é uma pena, a mecânica principal de pulo é alterada, permitindo que o personagem execute movimentos de voos ou pulos mais prolongados. Já a flor de fogo permite que o jogador tenha um novo tipo de ataque, adicionando essa ação ao personagem.

**Figura 06** - As mecânicas secundárias adicionam vários elementos à jogabilidade de Mario.

**Jogo:** *Super Mario World*.



**Fonte:** Esquerda: <http://www.fanpop.com/clubs/super-mario-bros/images/37768740/title/fireball-gif-photo>.

Direita:

[http://78.media.tumblr.com/88d09e81e3895e60aaefc9ac52161a9d/tumblr\\_o8410wf8yK1rrftcdo1\\_500.gif](http://78.media.tumblr.com/88d09e81e3895e60aaefc9ac52161a9d/tumblr_o8410wf8yK1rrftcdo1_500.gif)

Acesso em: 25 Ago 2015.

## Uma taxonomia para as mecânicas

Bom, jogos são uma experiência extremamente interativa, então, era de se esperar que existissem diversas formas de o jogador afetar o estado do jogo. Vamos aqui exibir uma taxonomia de mecânicas baseada em classificações apresentadas pelos autores Schell (2008), Rodgers (2014) e Adams e Dormans (2012). Cada uma das categorias apresentadas corresponde a um tipo de mecânica encontrada em um jogo.

### Espaço

O **espaço funcional** ou ambiente do jogo corresponde ao local onde a ação acontece, ou seja, o mundo de fantasia que existe ao cruzarmos o “círculo mágico”, se assim preferir. Quando falamos de espaço dentro de um jogo, podemos definir vários aspectos diferentes de como o jogador pode perceber e interagir com o mundo do jogo, vejamos a seguir.

O primeiro ponto é com relação à forma como a posição do jogador e dos elementos do jogo são representados dentro do mesmo: o espaço pode ser visto de forma **discreta** ou **contínua**.



Um espaço **discreto** possui posições específicas que podem ser ocupadas tanto pelo jogador como por adversários e objetos. Não existe uma transição suave entre as posições e o espaço é dividido em unidades (células ou “casas”, por isso o nome discreto). Um exemplo clássico desse tipo de espaço são os jogos de tabuleiro, como o xadrez e damas: a posição da peça é determinada pelo quadrado do tabuleiro que ela ocupa.

Cada peça se encontra em um quadrado específico, e a movimentação ocorre de um quadrado para o outro: não existem posições entre quadrados, e a posição da peça dentro de um quadrado não é relevante. Jogos de estratégia que definem o terreno em forma de malhas ou grids de posições também usufruem do espaço discreto como representação do espaço físico em que o jogo acontece.

Esse tipo de espaço facilita a verificação de posições dentro do jogo e permite uma apresentação visual da divisão do espaço virtual do jogo, porém costuma limitar a movimentação dos personagens.

**Figura 07** - O espaço discreto facilita a representação do espaço dividindo-o em unidades lógicas.  
**Jogos:** a) *Battle Chess* b) *Final Fantasy Tactics* c) *Civilization V*.



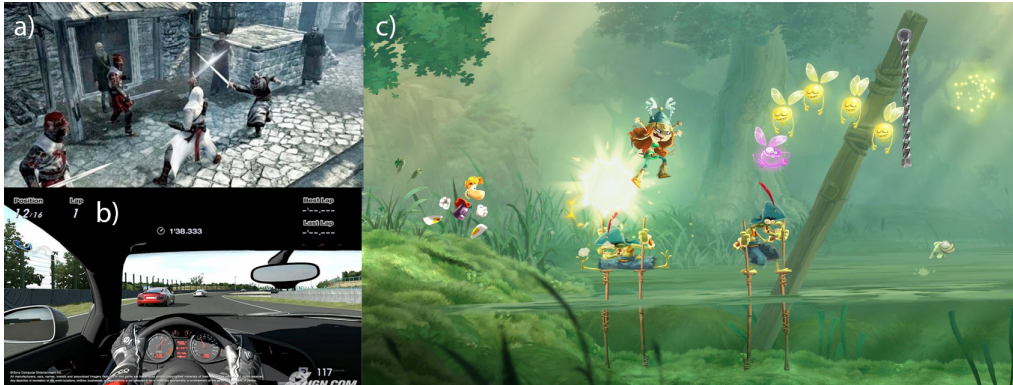
**Fonte:** a) <https://www.myabandonware.com/game/battle-chess-et> Acesso em: 25 Ago 2015  
b) <https://oldgamesftw.wordpress.com/2009/12/22/10-tactics-games-que-eu-recomendo/>. Acesso em: 25 Ago 2015  
c) [http://store.steampowered.com/app/8930/Sid\\_Meiers\\_Civilization\\_V/?l=portuguese](http://store.steampowered.com/app/8930/Sid_Meiers_Civilization_V/?l=portuguese). Acesso em: 25 Ago 2015

Já no espaço **contínuo**, a posição dos elementos varia de forma suave, e cada coordenada é diferente e relevante. Não existe uma divisão em unidades menores e o espaço como um todo pode ser percorrido sem interrupção. A gestão das posições de cada elemento se torna mais complexa, o que dificulta algumas operações, como saber se dois objetos se encontram, porém isso permite que a simulação de movimentos seja mais real.

Esse tipo é muito comum em jogos de ação e simulações. O espaço contínuo também requer um esforço da equipe de programação quando é muito grande, porque ele precisa ser percorrido de forma suave, com o mínimo de interrupções (entenda-se: *loading* ou carregamento de telas) entre as áreas.

**Figura 08** - O espaço contínuo permite uma movimentação mais livre para o jogador.

**Jogos:** a) *Assassin's Creed II* b) *Gran Turismo 5* c) *Rayman Legends*.



**Fonte:** a) <http://news.softpedia.com/news/Mac-Gamers-Get-Your-Copy-of-Assassin-s-Creed-II-Now-157978.shtml>. Acesso em: 25 Ago 2015

b) <http://superstars.kids.sapo.pt/gran-turismo-5-3/>. Acesso em: 25 Ago 2015

c) <https://thegamingdojo.wordpress.com/2014/01/17/rayman-legends-review/comment-page-1/>.

Acesso em: 25 Ago 2015

Podem existir vários tipos de espaços dentro de um jogo. Comumente encontramos jogos que são implementados com espaços contínuos em sua interação principal e espaços discretos nos minijogos, para facilitar a interação com aquela porção específica do jogo. Um exemplo disso são os jogos de RPG que possuem quebra-cabeças para determinar o sucesso na realização de uma tarefa específica, por exemplo, resolver um quebra-cabeça visual para decifrar uma mensagem ou um problema de raciocínio lógico (como torres de hanói) para abrir uma porta/passagem.

**Figura 09** - Os dois tipos de espaço no jogo Bioshock. No modo principal de jogo (a) o espaço é contínuo, porém no minijogo para hackear máquinas (b) o jogador é levado a um espaço discreto de interação.

**Jogo:** *Bioshock*



**Fonte:** a) <https://videogameartreviews.wordpress.com/>. Acesso em: 25 Ago 2015

b) <http://superstars.kids.sapo.pt/gran-turismo-5-3/>. Acesso em: 25 Ago 2015

Outro exemplo são jogos com um espaço muito grande: nesse caso, é comum que o mapa seja discretizado em diversas áreas, e cada área menor seja um espaço de interação contínuo. Dessa forma, o jogo apenas apresenta um espaço/área por vez (o que melhora o desempenho do jogo), e na medida em que o jogador navega pelo cenário, o jogo vai alternando entre os espaços discretos existentes. Isso é muito comum em jogos mais antigos, e tem um impacto direto na navegação do jogador pelo mundo do jogo.

**Figura 10** - Em Zelda, cada área desenhada na tela é um espaço de interação contínuo, porém limitado. Quando o jogador avança em uma das bordas da tela, um novo espaço é carregado. O mapa geral é discretizado em várias áreas para melhorar o desempenho do jogo.

**Jogo:** *Zelda: A Link to the Past*



**Fonte:** <http://modomeu.com/wp-content/uploads/2015/03/Zelda-a-link-to-the-past-1024x764-gameplay.jpg>. Acesso em: 25 Ago 2015

Bom, já que começamos a falar de navegação, dependendo do tamanho do jogo, da estrutura de níveis e da representação do espaço, a navegação pode ser restrita a um único ambiente ou ocorrer entre vários ambientes diferentes. Um jogo organizado em fases pode realizar essa transição simplesmente quando o jogador passa de fase e avança para o nível seguinte (mais comum em jogos do gênero plataforma). Para jogos de FPS, é comum o uso de mecânicas de **portais** para realizar a transição entre os espaços de um nível: cada sala é delimitada como um espaço de interação, e existem pontos de acesso de uma sala para a outra que permitem a livre navegação pelo mapa.



**Figura 11** - Em (a), podemos observar a estrutura de níveis de KOTOR II, na qual cada sala é dividida por portas de acesso que as delimitam (linhas verdes entre os espaços). Em (b), quem precisa de chave quando se tem um sabre de luz?

**Jogo:** a) *Star Wars: KOTOR II*



**Fonte:** a) <http://www.gamebanshee.com/starwarskotorii/locations/residential082east.php>, 25 Ago 2015

b) <https://www.youtube.com/watch?v=EohOVUmCdTw>, 25 Ago 2015

Outro tipo de mecânica importante são as **barreiras**: elementos inseridos dentro do jogo para impedir o acesso a determinadas áreas do mundo até que uma condição específica seja satisfeita. Elas podem ser representadas como portas fechadas, formações geológicas ou naturais (montanhas, árvores, etc.) ou até mesmo um adversário poderoso.

A utilização de barreiras é uma forma de direcionar a interação do jogador, para que, por exemplo, ele seja forçado a realizar uma certa ordem de eventos antes de poder prosseguir para uma determinada área do jogo. O uso desse tipo de mecânica tira um pouco da liberdade do jogador, mas pode ser necessário para construir a experiência desejada.

**Figura 12** - Com licença, eu gostaria de entrar nessa caverna!

**Jogo:** *Pokemon Crystal*



**Fonte:** <https://www.youtube.com/watch?v=i-8NeFTpxtc>. Acesso em 25 Ago 2015

Essas duas mecânicas combinadas costumam gerar mapas com uma navegação mais linear e direcionada, impedindo que o jogador explore livremente o mundo. Uma das vantagens da navegação linear é que ela facilita a apresentação de eventos e sequências importantes para a narrativa, já que o jogador irá percorrer o espaço de forma limitada. Por outro lado, muitos jogos utilizam a abordagem de *sandbox* ou **mundo aberto**. Nesses casos, disponibiliza-se mais liberdade de movimentação para o jogador, que fica livre para explorar o mundo da maneira que ele quiser. É comum em jogos de mundo aberto que os eventos sejam disparados no momento que o jogador atinge uma certa localização, e deve-se tomar bastante cuidado quando se define eventos em sequência ou que dependem de um conjunto de pré-condições para ser iniciado. Uma falha na checagem desses fatores pode levar a um *bug* o qual um evento inicia mas o jogador não tem condições de completá-lo.

Esse tipo de jogo costuma usar a técnica de **espaços aninhados**. Usualmente, existe um espaço macro cujo jogador pode navegar e que representa o mundo do jogo como um todo, e existem espaços menores contidos nele que representam lugares mais detalhados desse mundo: cidades, cavernas, florestas, etc. Quando se trabalha com a noção de espaços aninhados, apenas o espaço atual que o jogador está explorando é desenhado e gerenciado pelo jogo, sendo necessário um tempo para carregar novos espaços quando o jogador transita de um para o outro. O uso de espaços aninhados não só permite a criação de um mundo mais extenso, como também facilita aspectos técnicos de implementação do jogo (afinal, se eu estiver dentro de uma caverna, não preciso me preocupar em desenhar e gerenciar o resto do mundo, concorda?).

**Figura 13** - O mapa de Skyrim ilustra bem a ideia de espaços aninhados: o jogador pode percorrer todo o mapa do mundo (a), mas cada ícone desenhado no mapa representa uma localização que ele pode entrar e explorar mais detalhadamente (b) (cavernas, fortalezas, cidades, etc.).

**Jogo:** *Skyrim*



**Fonte:** a) <http://www.nexusmods.com/skyrim/mods/4929>. Acesso em 25 Ago 2015  
b) <http://www.nexusmods.com/skyrim/mods/22423/>. Acesso em 25 Ago 2015

Outro ponto importante na definição do espaço é a quantidade de dimensões que ele possui: um jogo pode conter um espaço bidimensional, tridimensional ou não conter nenhuma dimensão. Nesse último caso, o jogo ocorre em uma representação abstrata de espaço que permite as interações necessárias. Jogos de simulação de gestão esportiva normalmente possuem grande parte das interações do jogo através de relatórios de dados e telas de informações, não havendo uma representação própria de espaço para interação com o jogo (embora no momento de simulação das partidas exista um campo, o jogador não interage diretamente nele).

A quantidade de dimensões influencia tanto na forma de navegação permitida para o jogador como nos controles de câmera que serão implementados, sem contar que as posições dos objetos devem ser representadas em função de cada dimensão existente.

**Figura 14** - Um exemplo de jogo que tem um espaço físico abstrato. No Football Manager 2015, você deve controlar os aspectos de gestão e treinamento de uma equipe de futebol e levá-la ao sucesso.

#### Jogo: Football Manager 2015



**Fonte:** <http://www.gamenguide.com/articles/14872/20141106/football-manager-2015-editor-explained-our-guide-to-the-free-and-paid-versions-that-let-players-alter-game-data.htm>. Acesso em: 25 Ago 2015.

Um último aspecto interessante de ressaltar com relação ao espaço físico é a forma como ele é criado: o espaço pode ser **pré-definido** ou gerado de forma **procedural**. Um espaço pré-definido é estático e construído na fase de design e implementação do jogo, e todos os elementos que estão contidos no espaço, as interações e eventos programados irão ocorrer de acordo com o que foi definido no momento da construção do nível do jogo. Já o espaço procedural é construído em tempo de execução: através de funções matemáticas, o jogo cria um espaço físico que obedece a um determinado conjunto de regras, mas que será novo cada vez que o jogador iniciar o jogo. A vantagem



dos espaços estáticos é o controle sobre a interação do jogador e sobre os eventos do jogo, enquanto que o espaço procedural ganha no quesito de novidade e imprevisibilidade, o que aumenta o valor de *replayability* (o jogo pode ser jogado várias vezes).

**Figura 15** - Exemplos de jogos que possuem o seu espaço físico definido de forma procedural: FTL e No Man's Sky.

**Jogos:** a) *FTL* b) *No Man's Sky*



**Fonte:** a) [http://www.ign.com/wikis/ftl-faster-than-light/Beacon\\_Map](http://www.ign.com/wikis/ftl-faster-than-light/Beacon_Map). Acesso em: 25 Ago 2015.

b) <http://tec-cia.com.br/2015/06/17/no-mans-sky-e-confirmado-para-pc-e-tera-lancamento-simultaneo-com-o-ps4>. Acesso em: 25 Ago 2015.

## Objetos e Estado

O segundo tipo de mecânica refere-se às mecânicas que definem o estado em que o jogo se encontra a partir de várias informações do mundo do jogo. Essas mecânicas estão relacionadas aos objetos do mundo do jogo e seus atributos.

Imagine um jogo tradicional de *Shoot'em Up*: normalmente você inicia com um tipo de tiro simples e uma proteção inicial. Talvez um ou dois especiais para aqueles momentos de tensão, quando as hordas de adversários começam a encher a tela e você não sabe mais o que fazer. Ao longo da partida, muitas coisas podem mudar: poderá utilizar itens na tela que mudam o seu tipo de tiro; disponibilizam um tiro adicional e/ou uma barreira protetora; você pode ter que usar seus especiais até acabarem ou recarregá-los ao longo da partida; e, ainda, pode aumentar seu número de tentativas/vidas quando atingir uma determinada pontuação. Ou seja, à medida que você joga, o estado dos elementos do jogo (sua nave, o estado do jogo) vão mudando de acordo com as interações realizadas e recursos consumidos/adquiridos.

**Figura 16** - Um tradicional Shoot'em Up japonês, Raiden IV possui vários dos elementos citados no texto acima.

**Jogo:** *Raiden IV*



**Fonte:** <https://www.youtube.com/watch?v=t742aEAC-I>. Acesso em: 25 Ago 2015

A partir desse exemplo podemos começar nossa discussão. O primeiro ponto a se observar é que um jogo possui vários **objetos** que compõem o seu espaço, como o personagem do jogador, outros personagens, os inimigos, obstáculos espalhados pelo cenário, itens e recursos, etc. Cada elemento desse possui um conjunto de **atributos**, características que os definem e os tornam únicos ou parte de um grupo. Esses atributos podem ser tão simples quanto uma posição no mapa ou complexos como o relevo facial



que compõe o rosto do personagem. Os atributos permitem identificar os objetos e também definem o seu **estado** dentro do jogo. Um estado pode ser entendido como uma situação específica relevante para o jogo que é alcançado quando uma determinada combinação de valores dos atributos é atingida. Por exemplo, quando a sua energia chega em zero, o personagem passa do estado ativo para inativo ou quando um item tem em seu atributo condição deteriorado, ele passa a ser menos efetivo.

Existe também um tipo de objeto abstrato que controla informações e, apesar de não estarem fisicamente representadas no mundo do jogo, são essenciais para o controle do que acontece e de como o jogo evolui. Podemos definir esse objeto como o **estado do jogo**. As informações contidas no estado do jogo podem estar relacionadas ao progresso do jogador (quantos pontos ele atingiu, quais regiões ele já explorou, em que ponto das missões ele está) e dos adversários, permitindo identificar que uma condição de vitória ou derrota foi atingida no jogo.

Normalmente existem interações que são realizadas dentro do jogo que ativam a mudança de estado de um determinado objeto. Em um jogo FPS, por exemplo, as armas costumam possuir uma quantidade limitada de munição máxima e/ou de disparos que podem ser efetuados antes da próxima recarga da arma. Inicialmente a arma possui uma quantidade X de balas. Se o jogador efetua um disparo, a quantidade de balas da arma diminui. Caso essa quantidade de balas se torne zero, a arma está descarregada e não consegue efetuar mais nenhum disparo até que ela seja recarregada. Se o jogador não possui mais munição, ele precisa encontrar recursos no jogo para poder adquirir novas balas. Ao encontrar uma munição, o número de balas que a arma contém é acrescido de um valor específico definido pelo projetista do jogo. Se a arma estiver com o número máximo de balas, pegar uma munição não fará nenhuma diferença (ou não será possível pegar o cartucho de balas). Se formos detalhar esse exemplo, podemos perceber que a arma pode se encontrar em alguns estados diferentes:

- Arma vazia: o jogador efetuou um número de disparos que esvaziou o cartucho e precisa recarregar a arma para efetuar novos disparos;
- Arma Carregada: a arma possui balas no cartucho e pode efetuar disparos;
- Munição cheia: o jogador já possui o número máximo de munição possível de carregar.

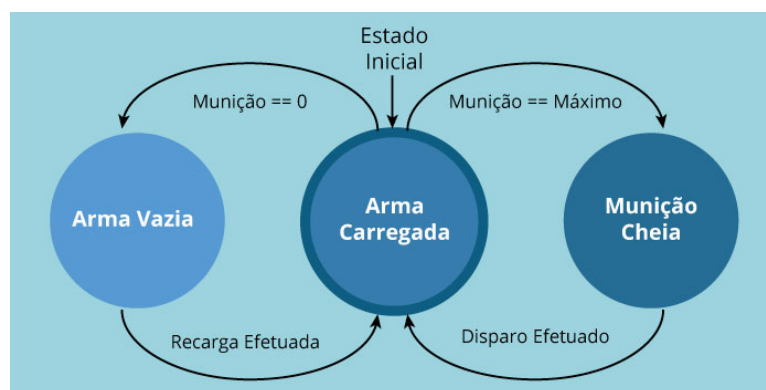
Para cada um desses estados, existem condições específicas que determinam não apenas quando a arma se encontra em algum deles, mas também quando ela muda de um estado para o outro. E basicamente tudo pode ser modelado e controlado da mesma forma! Mas isso não é muito difícil?

A gestão de tantos estados de forma simultânea é uma tarefa complexa e para isso recorreremos a ferramentas conceituais que podem ser programadas dentro do jogo para facilitar o nosso controle. Um dos conceitos que podem ser utilizados é o de **máquina de estado finito** ou *Finite-State Machine* (FSM). Essas máquinas são uma representação em função de estados e transições que permitem modelar como um objeto do mundo do jogo deve se comportar.

De forma intuitiva, podemos descrever o funcionamento de uma máquina de estados da seguinte forma: cada objeto possui um estado inicial, que é a forma como ele inicia no mundo do jogo. Para cada estado existem ações que são possíveis de se realizar, e condições que indicam quando o estado do objeto irá mudar. Cada objeto se encontra em apenas um estado a cada momento, de forma que sempre podemos definir, a partir de uma sequência de interações, qual o comportamento esperado para o objeto. O jogo analisa o estado de forma contínua, e quando as condições para que ocorra uma mudança de estado são atingidas, a máquina realiza uma transição para um novo estado.

É importante notar que existe uma sequência de transições bem definidas, e um objeto não pode passar imediatamente de um estado para outro sem que exista a transição definida na máquina. A principal vantagem da máquina de estados é que ela nos permite elaborar uma representação visual que facilita o entendimento do comportamento que um objeto do jogo deveria ter. Vamos ilustrar a máquina de estado para o caso da arma:

**Figura 17** - Uma máquina de estado simples para ilustrar os estados e transições de um objeto arma ao longo do jogo.



Como denotado na figura, a máquina de estado da arma possui apenas três estados (claro que poderíamos criar uma máquina com mais estados, mas vamos mantê-la simples para facilitar a explicação: Arma Vazia, Arma Carregada e Munição Cheia. O estado inicial para o nosso exemplo é Arma Carregada, denotado pelo círculo duplo no estado. Estamos aqui assumindo que quando o jogador adquire a arma (seja no início ou ao longo do jogo), ela já vem com alguma quantidade de munição disponível para uso. Nesse estado, o jogador pode executar uma série de ações com a arma, como disparar, recolher/comprar mais munição, recarregar a arma quando a quantidade de disparos permitidos pelo cartucho for efetuado, dentre outras possibilidades.

Esse estado pode ser alterado em função de transições ou mudanças definidas, um exemplo possível é quando a quantidade de munição chega a zero, a arma não tem mais balas e passa para o estado **Arma Vazia**. Nesse estado, o jogador não pode efetuar a ação de disparo, que fica desabilitada. Tudo que o jogador pode fazer é adquirir munição e recarregar a arma, para retornar ao estado **Arma Carregada**.

Perceba que o estado do objeto arma influencia nas ações que o jogador pode executar, e esse controle se torna acessível quando temos as transições e estados bem-definidos. Outra possibilidade é que do estado **Arma Carregada** o jogador adquira muita munição e alcance o número máximo de balas (muito comum quando o jogo possui múltiplas armas e o jogador deixa de usar uma delas). Nesse caso, o jogador ainda consegue efetuar as ações de disparo, mas não conseguirá adquirir mais munição. O jogador precisa, então, gastar munição para poder retornar ao estado de **Arma Carregada**.

Muito legal essa máquina de estados, não é? Ela não é exclusividade dos jogos, sendo bastante utilizada quando precisamos modelar o comportamento de sistemas computacionais complexos ou críticos. Sempre é válido aprender a usar uma ferramenta como essa, não acha? Confira o material de leitura complementar se desejar aprofundar seus conhecimentos e entender um pouco mais desse assunto!

Bom, já vimos quais ações podem alterar o estado dos nossos objetos, mas será que existem outras formas? Sim, através de mecânicas específicas para isso!

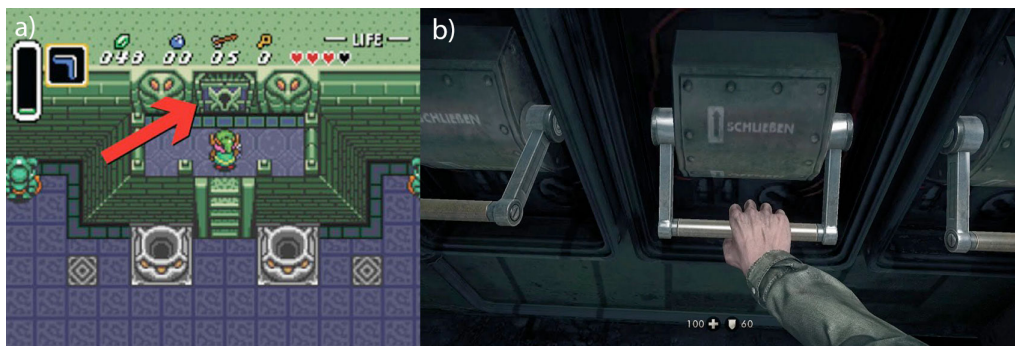
Vamos estudar algumas mecânicas cujo objetivo é alterar o estado de um objeto do jogo. A mecânica mais básica é quando o próprio objeto possui uma característica de comportamento binário: ou ele está ativo ou não está. Podemos chamar essa mecânica de chave-fechadura (*lock and key*) ou botão, já que o objeto tem um comportamento similar a uma porta trancada ou um botão: ele se encontra em um estado de inatividade

até que uma chave seja utilizada, um botão pressionado, uma alavanca movimentada ou um quebra-cabeça resolvido. A partir daí o seu estado passa a ser ativo, mudando completamente as características do objeto em questão.

Essa mecânica é muito comum com portas, botões, alavancas e costuma ser bastante utilizada para controlar a navegação do jogador no mundo virtual (ou para habilitar áreas secretas e especiais). Essa alteração de estado pode ser definitiva (existe apenas a transição do estado inativo para o ativo) ou não (existe a transição de ida e volta).

**Figura 18** - Algumas mecânicas de comportamento binário, como portas e alavancas, são bastante utilizadas para direcionar a interação do jogador ou construir quebra-cabeças.

**Jogos:** a) *Legend of Zelda: A Link to the Past* b) *Wolfenstein: New Order*



**Fonte:**

a) [http://www.gamasutra.com/view/feature/134273/evaluating\\_game\\_mechanics\\_for\\_depth.php?print=1](http://www.gamasutra.com/view/feature/134273/evaluating_game_mechanics_for_depth.php?print=1). Acesso em: 25 Ago 2015

b) <https://www.youtube.com/watch?v=jhNPlxIVVY>. Acesso em: 25 Ago 2015

Outro tipo comum de mecânica que altera estado são os **Power-ups**, itens que permitem um incremento das habilidades do jogador, seja por um período de tempo limitado ou não. No jogo clássico Arkanoid, existem itens que mudam sensivelmente as características da bola e da plataforma controlada pelo jogador: aumenta a velocidade da bola, permite que a bola destrua vários blocos sem ricochetejar, aumenta o tamanho da plataforma.

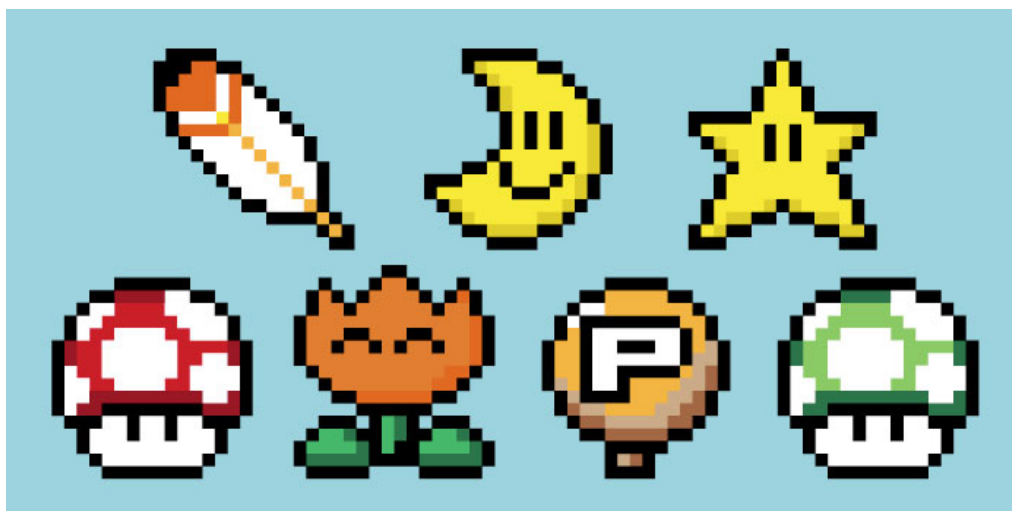
**Figura 19** - Os itens que descem na tela alteram características do jogo, tanto de forma positiva como negativa, e dão variedade à jogabilidade.  
**Jogo:** *Arkanoid*



**Fonte:** <http://giphy.com/gifs/video-games-arkanoid-x2Xcbz96qqEU0>. Acesso em: 25 Ago 2015

Quando um jogo permite a acumulação de vários *power-ups*, costumamos dizer que estamos com um *buffer* de habilidades acrescidas, assim, as vantagens de todas as habilidades adquiridas ficam acumuladas e somam para o desempenho na partida. Apesar de contraditório, existem *power-ups* que fazem justamente o contrário do que o nome expressa: reduzem as habilidades do jogador ou acrescentam uma dificuldade extra na realização de ações, ou seja, diminui sua velocidade ou desabilita alguma ação por um breve período de tempo. Apesar de surtir efeitos contrários a mecânica permanece a mesma, pois o jogador utiliza um item o qual afeta as suas propriedades, por isso o nome é mantido.

**Figura 20** - Em Super Mario, existem diversos itens de power-ups que conferem uma gama variada de poderes e ações especiais para o querido encanador italiano.



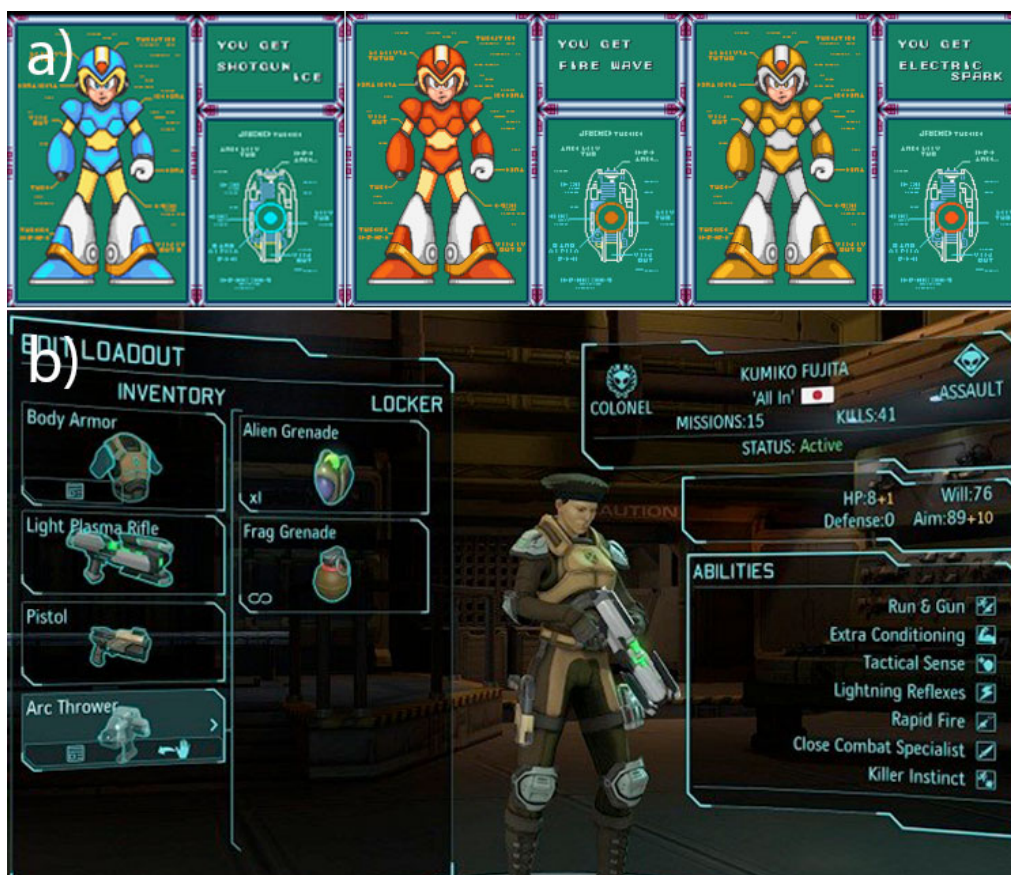
**Fonte:** Adaptado de: <http://www.spritestitch.com/forum/viewtopic.php?f=9&t=1649>. Acesso em: 25 Ago 2015

Uma outra forma de alterar os atributos de um objeto é através do uso de **itens**, que são objetos adquiridos ao longo do jogo e que podem ser acionados pelo jogador para alterar seus atributos (recuperar vitalidade, aumentar uma habilidade). Além dos itens consumíveis (ao utilizá-los são destruídos), existem itens que podem ser equipados pelo jogador para alterar os atributos do seu personagem enquanto ele o possuir. Esse tipo de mecânica é bastante comum em jogos de RPG, mas já pode ser encontrado em diversos outros gêneros. Por exemplo, no jogo Megaman X, a cada chefe de nível que o jogador derrota, ele recebe um item o qual permite utilizar um ataque relacionado ao poder do chefe que derrotou (ataque de gelo, elétrico ou floresta, por exemplo), sendo necessário que o jogador equipe esse ataque para substituir o ataque original do Megaman.



**Figura 21** - A mecânica de equipar itens não só altera os atributos e permite que o jogador ganhe vantagens no jogo, como também deixa o personagem com um visual descolado!

**Jogos:** a) *Megaman X* b) *X-COM: Enemy Unknown*



**Fonte:** a) <http://snestalgia.blogspot.com.br/2012/10/analise-detonado-megaman-x-super.html>.

Acesso em: 25 Ago 2015

b) <http://guides.gamepressure.com/xcomenemyunknown/guide.asp?ID=16639>. Acesso em: 25 Ago 2015

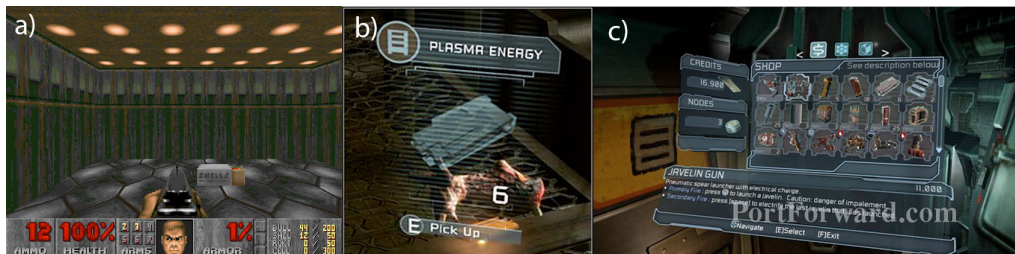
Será que os itens só podem existir para alterar o estado de objetos e serem equipados? Um outro tipo de mecânica possível é o acúmulo de recursos para fins variados: energia vital para manter o personagem ativo, munição para armas, moedas para comprar outros itens, pontos de experiência e habilidade para evoluir personagens ou até mesmo uma pontuação que indique quem é o melhor na tabela de classificação do jogo. A **coleta de recursos** é algo constantemente presente nos jogos e outra possibilidade para utilização de itens.

Um dos motivos para o uso dessa mecânica é a capacidade de prover um feedback instantâneo para o jogador, pois quando ele coleta, o estado do jogo muda. E esses recursos podem ser usados para diversas finalidades: pode balancear a dificuldade do jogo (energia permite que o jogador consiga interagir por mais tempo e se recupere de possíveis erros cometidos ao longo do jogo); pode dar um senso de progressão (pontos

de experiência podem ser utilizados para fazer com que o personagem evolua e se torne mais forte); e até mesmo como a mecânica principal do jogo (em jogos de gestão de recursos, cujos acúmulo e uso de recursos é vital para o sucesso).

**Figura 22** - Normalmente a munição está disponível pelo mapa do nível do jogo ou em estações para ser comprada com recursos do jogo.

**Jogos:** a) *Doom* b) *Dead Space* c) *Dead Space 2*



**Fonte:** a) <http://doom.wikia.com/wiki/Ammo>. Acesso em: 25 Ago 2015

b) <https://guides.gamepressure.com/deadspace/guide.asp?ID=6138>. Acesso em: 25 Ago 2015

c) <http://portforward.com/games/walkthroughs/Dead-Space-2/Chapter-2.htm>. Acesso em: 25 Ago 2015

Alguns recursos disponíveis pelo jogo geram uma espécie de **sistema econômico**, pois se pensarmos em munição e energia como uma moeda, por exemplo, estamos a todo momento consumindo e adquirindo esses elementos ao longo do jogo. Essa troca constante entre jogador e jogo geram algo similar a uma economia e, assim como qualquer sistema econômico, devem ser tratados com cuidado pelo designer do jogo. Então, o que acontece se o seu jogo apresenta uma grande quantidade de itens regenerativos ou armamento para o jogador? É provável que o jogo se torne mais fácil e o jogador não sinta a tensão ou se preocupe em tomar decisões com cuidado, porque facilmente ele pode se recuperar com os itens oferecidos em abundância pelo mundo do jogo.

De forma similar, um jogo que não oferece uma quantidade mínima suficiente de recursos pode se tornar demasiadamente difícil, mesmo para os melhores estrategistas. E se dinheiro/recurso para adquirir itens for algo fácil? É provável que o jogador nem consiga utilizar tudo o que ele coleta e aquele recurso passa a ter menos valor para ele no momento de tomar decisões, deixando o jogo desequilibrado.

O balanceamento da economia de um jogo é algo importante e que deve sempre estar na mente dos projetistas do jogo. Veremos com mais detalhes essa ideia em uma outra aula!



**Figura 23** - Talvez ainda posso colocar mais uma poção aqui no meu bolso.

**Jogo:** *Dungeon Siege*



**Fonte:** <http://www.slcentral.com/c/g/r/microsoft/dungeonsiege/print.php>. Acesso em: 25 Ago 2015

Já falamos de várias mecânicas que alteram o estado dos objetos do jogo, mas também existem algumas mecânicas que são particulares do estado do jogo e servem para facilitar o controle das inúmeras variáveis que podem interferir no andamento da partida.

Uma mecânica muito comum (e debatida!) é a de **salvar** o estado atual do jogo, para que o jogador possa reiniciar a próxima sessão a partir de um ponto específico. Alguns jogos mais antigos não permitiam esse tipo de ação, exigindo que o jogador reiniciasse cada nível do ponto inicial. O problema com essa abordagem é que à medida que um jogador repete muitas vezes o mesmo trecho do jogo, a experiência pode se tornar frustrante e enfadonha.

Atualmente, existem algumas mecânicas específicas para evitar que esse tipo de experiência aconteça: **checkpoints** é uma mecânica baseada em pontos específicos do jogo em que o jogador pode salvar o seu progresso. Dessa maneira, caso ele perca, poderá reiniciar a partir desse ponto específico, em vez de retornar ao início do nível. Outra forma bastante utilizada é o mecanismo de salvar automático em momentos anteriores a desafios ou em pontos específicos do nível, às vezes associado com o progresso da narrativa. Essa mecânica é um pouco controversa porque retira parte da

autonomia do jogador, por realizar uma ação sem o comando direto dele. Outra opção muito comum é permitir que o jogador salve o jogo a qualquer momento, exceto em situações cuja mudança de estado é muito dinâmica (como durante combates, por exemplo).

**Figura 24** - Os checkpoints permitem que o jogador continue o jogo a partir de onde parou, em vez de voltar ao começo da fase.

**Jogo:** *Super Mario World*



**Fonte:** <http://forum.jogos.uol.com.br/como-identificar-um-n00b-jogando-snes-t-2505972>. Acesso em: 25 Ago 2015

Outro ponto importante de se ressaltar com relação ao estado do jogo é o quanto dele é revelado ao jogador. As informações em um jogo podem ser categorizadas da seguinte forma: informações que o jogador pode saber o tempo todo (quantas vidas ele tem, quantos pontos ele fez); informações que são secretas ou escondidas dele, mas conhecidas pelo adversário (a posição dos adversários no mapa, quantidade de recursos que o adversário tem); e informações que são conhecidas apenas pelo estado do jogo, mas que inicialmente não são conhecidas por nenhum dos participantes (a localização de itens e recursos no mapa do jogo). Então, tem coisa que nem mesmo o jogo sabe de antemão, como informações geradas aleatoriamente!

Essa classificação da informação é muito importante no momento de decidir o que cada parte integrante do jogo sabe, onde ela deve ser armazenada e quem deve ser notificado à medida que as mudanças vão ocorrendo dentro do jogo. Só não vale revelar

informações confidenciais aos adversários!

Algumas mecânicas de jogos são implementadas com base nessas informações não conhecidas pelo jogador, como a **neblina de guerra** (*fog-of-war*), por exemplo. Nessa mecânica, o mapa do jogo é ocultado do jogador, e apenas as áreas onde ele já explorou passam a ser visíveis para ele. Além disso, é comum adicionar uma neblina em duas camadas: terrenos já explorados aparecem descobertos no mapa, porém o jogador precisa ter personagens ou elementos sobre o seu controle em uma região para poder visualizar o que realmente ocorre no terreno. Caso não tenha, o jogador não terá informações de movimentações adversárias, mesmo que já tenha explorado o mapa, por exemplo.

**Figura 25** - Nessa imagem, uma neblina de guerra com as duas camadas: a região preta é onde não foi explorado; a região cinza indica o que o jogador já visitou, mas não consegue ver.

**Jogo:** *Age of Empires II*



**Fonte:** <http://www.giantbomb.com/fog-of-war/3015-14/images/>. Acesso em: 25 Ago 2015

## Tempo

Quando falamos de tempo como mecânica temos algumas formas diferentes de interpretar esse elemento dentro do jogo. A forma clássica se refere ao tempo que é permitido ao jogador pensar suas ações e tomar decisões. Aqui temos duas categorias principais, tão clássicas que já são utilizadas até para definir gêneros de jogos: **tempo real** e **turnos**.

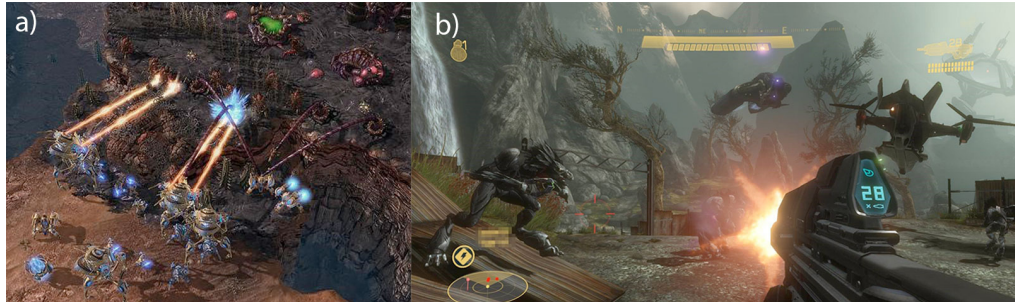
Jogos em tempo real simulam uma atividade contínua e ininterrupta, não permitindo que o jogador pare e reflita longamente as suas decisões. A ação corre o tempo todo e força o jogador a rapidamente avaliar suas opções e tomar uma decisão, mesmo que



não seja ótima para a situação. São jogos que priorizam o raciocínio rápido, reflexos e tempo de resposta dos jogadores.

**Figura 26** - Jogos em tempos reais forçam o jogador a tomar decisões de forma mais instintiva, e possuem um senso maior de urgência.

**Jogos:** a) *Starcraft II* b) *Halo: Reach*



**Fonte:** a) <http://www.tault.com/starcraft-2/sc2-zerg>. Acesso em: 25 Ago 2015

b) <http://www.thegameraccess.com/news/360-news/brand-new-halo-reach-screens/>. Acesso em: 25 Ago 2015

Já os jogos que são baseados em turnos permitem que o jogador possa raciocinar mais e avaliar melhor a condição do jogo antes de tomar uma decisão. Nesses jogos, os turnos entre as ações do jogador e do computador (ou outro jogador) são alternados, e existe uma sinalização por parte do jogador de que ele terminou de executar todas as ações possíveis.

**Figura 27** - Em jogos por turnos, o jogador pode traçar uma estratégia mais elaborada e considerar os possíveis cenários futuros antes de tomar uma decisão.

**Jogos:** a) *King's Bounty: The Legend* b) *Crusader Kings II*



**Fonte:** a) <https://br.gamersgate.com/DD-KBDSSE-NEW/king-s-bounty-dark-side>. Acesso em: 25 Ago 2015

b) [https://assets.nuuvem.com/image/upload/t\\_screenshot\\_full/v1/products/557dbc5369702d0a9c3fe600/screenshots/w5gjecim0v8k2pi7pm9u.jpg](https://assets.nuuvem.com/image/upload/t_screenshot_full/v1/products/557dbc5369702d0a9c3fe600/screenshots/w5gjecim0v8k2pi7pm9u.jpg). Acesso em: 25 Ago 2015

A diferença entre essas duas abordagens costuma definir se o jogo é mais reativo (exige menos planejamento) ou mais cognitivo (exige mais planejamento). O tempo também pode ser visto como a duração de um nível como um fator restritivo para as ações do jogador: o jogo pode requerer que o jogador realize uma ação em um prazo

limite máximo, sob pena de perder o jogo caso contrário. Quando se utiliza essa mecânica, é importante que o jogador saiba quanto tempo tem para finalizar seus objetivos, e isso pode ser expresso visualmente na tela e/ou através do som do jogo. Essa mecânica costuma ser comum em jogos de luta e jogos mais antigos de plataforma.

**Figura 28** - Em Street Fighter, ganha quem derrotar o adversário ou retirar mais energia dele antes que o tempo se esgote.

**Jogo:** *Ultra Street Fighter V*



**Fonte:** <http://www.techtudo.com.br/noticias/noticia/2015/10/nova-lutadora-de-street-fighter-5-pode-ser-brasileira-e-se-chama-laura.html>. Acesso em: 25 Ago 2015

Uma variação desse tempo limite é bastante encontrada em jogos de **sincronia**, como Guitar Hero e outros jogos musicais. Nesse caso, a mecânica exige que um botão seja pressionado em um curto espaço de tempo de forma correta, e a não execução da ação resulta em perdas de pontos pelo jogador. Os **Quick-Time Events** ou **QTE** são mecânicas de sincronia nas quais o jogador deve pressionar um ou mais botões em sequência para executar ações especiais. Esse tipo de mecânica é muito usado para emular ações que seriam difíceis com o esquema natural de controle do jogo, como na batalha contra adversários fora do padrão ou no combate contra um número excessivo de adversários.

Um ponto negativo dessa mecânica é justamente este: como ela altera a sequência natural de comandos que o jogador executa ao longo do jogo, quebra um pouco da imersão e pode rapidamente se tornar desinteressante à medida que o usuário falha, repetidas vezes, em acertar a sequência de comandos.

**Figura 29** - Com os QTEs, é possível associar ações únicas a um ou mais comandos no controle, permitindo experiências diferenciadas da jogabilidade normal do jogo.

**Jogo:** a) *Heavy Rain* b) *God of War 3*



**Fonte:** a) <http://atarireel.blogspot.com.br/2010/08/heavy-rain.html>. Acesso em: 25 Ago 2015  
b) <http://www.mobygames.com/game/ps3/god-of-war-iii/screenshots/gameShotId,557057/>. Acesso em: 25 Ago 2015

Até agora falamos de tempo no sentido de duração, porém existe uma outra forma de abordar tempo – o que se passa dentro do jogo. É cada vez mais comum que os jogos simulem a **passagem de tempo**, alternando períodos de dia/noite nos cenários onde o jogador interage. Além de trazer mais realismo ao jogo e permitir o desenvolvimento estético dos cenários, a mecânica de passagem de tempo pode ser utilizada para habilitar ações/eventos ao longo do jogo, como missões que só podem ser executadas à noite ou personagens que estão em diferentes locais do jogo, dependendo da hora do dia.

Em alguns jogos, como Pokemon, o horário define inclusive qual tipo de monstro o jogador pode encontrar, existindo monstros diurnos e noturnos. Um detalhe sobre o Pokemon é que a passagem de tempo no jogo é atrelada à passagem de tempo no mundo real, forçando o jogador a ter várias sessões de jogo em horários diferentes se quiser coletar todos os monstros possíveis.



**Figura 30** - No Pokemon, as horas do dia definem os eventos e criaturas que você encontra no jogo. Menos esses dois da figura, que passaram o dia na rua!

**Jogos:** *Pokemon Diamond & Pearl*

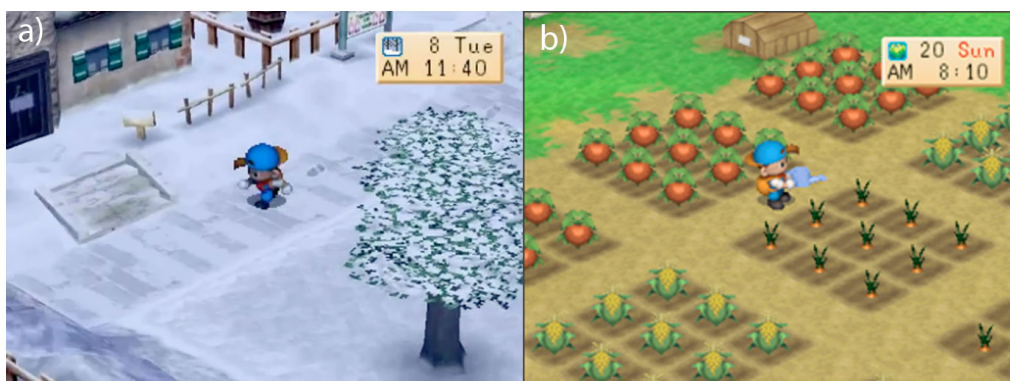


**Fonte:** <http://tvtropes.org/pmwiki/pmwiki.php/Main/InUniverseGameClock>. Acesso em: 25 Ago 2015

No simulador de fazenda Harverst Moon, além da passagem do dia, também há a passagem das estações, o que implica diretamente no tipo de vegetais que você pode plantar. Outros jogos também utilizam ações que dependem da passagem de tempo no mundo real, como o Zombie Tsunami, onde existem missões as quais exigem que o jogador inicie uma sessão de jogo em um horário específico do dia. Jogos que utilizam os períodos do dia ou estações do ano apenas como uma ambientação para os seus cenários não utilizam o tempo como mecânica, mas apenas de forma temática.

**Figura 31** - Em Harverst Moon, além do ciclo dia/noite, também ocorre a passagem de estações, que define o que você pode plantar na fazenda e eventos especiais sazonais.

**Jogo:** *Harverst Moon: Back to Nature*



**Fonte:** a) <http://npmarshall.com/2012/04/20/the-harsh-world-of-harvest-moon-back-to-nature/>.

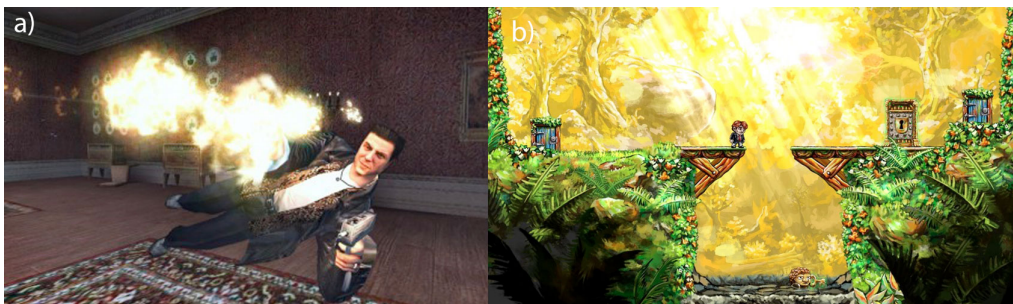
Acesso em: 25 Ago 2015

b) <http://harvestmoonbacktonatureguide.com/crops.html>. Acesso em: 25 Ago 2015

Um outro uso importante do tempo envolve mecânicas relacionadas à **manipulação do tempo** no jogo. Em *Max Payne*, o jogador pode reduzir a velocidade em que o jogo ocorre utilizando a mecânica de *Bullet Time*, esta deixa o jogo lento e melhora a chance do jogador desviar de projéteis e mirar nos oponentes. Quem não gosta do Efeito *Matrix* no jogo, não é verdade? Ainda mais interessante é a aplicação de manipulação de tempo encontrada em *Prince of Persia* e *Braid*: nesses jogos a manipulação do tempo foi implementada como uma mecânica a qual permite que o jogador refaça ações erradas quase que instantaneamente, não forçando a repetição de grandes porções do jogo para tentar novamente. No caso de *Braid*, essa manipulação do tempo ocorre tanto para trás (*rewind*) como para frente (*fast-forward*), e é essencial para a resolução da maior parte dos desafios e *puzzles* apresentados pelo jogo. Além, é claro, de reforçar o tema central do jogo, que é uma reflexão sobre segundas chances!

**Figura 32** - Jogos com a manipulação temporal permitem alternativas estratégicas quando o jogador está enfrentando um problema sem solução.

**Jogos:** a) *Max Payne* b) *Braid*



**Fonte:** a) <http://www.osel.cz/6346-technologie-bullet-time-obranou-rozvodnych-siti.html>. Acesso em: 25 Ago 2015

b) <http://tig.wikia.com/wiki/Braid>

E vamos parando por aqui! Ainda faltam mais tipos de mecânicas para discutirmos, mas essas ficarão para a próxima aula. Até lá!



# Pontos-chave

---

E passamos por mais uma fase do nosso curso. Como vimos nessa aula, as mecânicas são os blocos fundamentais com os quais montamos o nosso jogo. Vamos destacar alguns pontos sobre o que vimos:

- **Mecânicas de Jogos** são componentes do jogo com os quais o jogador interage para transformar o estado do jogo, e através dessas interações ele consegue progredir até atingir os objetivos de vitória (ou ser derrotado).
- Mecânicas **primárias** são mecânicas que definem o núcleo do jogo e sua jogabilidade, e são essenciais para o progresso do jogador na partida. Mecânicas **secundárias** são mecânicas que auxiliam e dão suporte às mecânicas principais, podendo inclusive alterar alguns elementos da jogabilidade.
- O **espaço funcional** refere-se ao espaço do jogo onde as interações acontecem. Esse espaço pode ser discreto ou contínuo, dependendo da forma como ele interpreta as posições dos elementos no mundo do jogo.
- Existem mecânicas que atuam no espaço como uma forma de **limitar ou permitir a navegação** do jogador por determinadas áreas, como as barreiras e os portais. Os **espaços aninhados** permitem a existência de uma área de jogo macro contendo várias áreas menores nas quais o jogador pode transitar e explorar.
- O espaço do jogo pode ser gerado de antemão ou criado no momento em que o jogo iniciar, de forma **procedural**.
- O jogo é composto por **objetos** que possuem um conjunto de atributos que os definem. Esse conjunto de **atributos** determina o **estado** em que o objeto se encontra.
- A **máquina de estados finitos** é uma técnica que permite a modelagem do comportamento de vários objetos no mundo do jogo.
- O **estado do jogo** refere-se a um conjunto de informações que são importantes para controlar o progresso, tanto do jogador como dos adversários. Existem mecânicas que atuam diretamente no estado do

jogo, seja para salvar o seu conteúdo ou para determinar o quanto de informação sobre o estado cada parte envolvida no jogo possui.

- **Power-ups, itens e equipamentos** são mecânicas que permitem a alteração dos atributos de um objeto, seja de forma temporária ou permanente.
- O **tempo** pode ser abordado dentro do jogo de várias formas: à quantidade de tempo disponível para o jogador tomar uma decisão, ao tempo que passa dentro do mundo do jogo (muitas vezes chamado de ciclo de tempo) ou a uma manipulação direta do tempo pelo jogador.

## Leitura Complementar

---

Estes são alguns links interessantes para vocês se aprofundarem ou observarem exemplos de alguns tópicos abordados nessa aula!

- [Alguns exemplos de mecânicas de jogos.](#)
- [Um vídeo sobre geração procedural em jogos.](#)
- [Uma aula sobre máquina de estados finitos.](#)
- [Várias formas de ataque de Megaman X, exemplificando a mecânica de equipamento.](#)
- [Exemplos de jogos com manipulação temporal.](#)

## Autoavaliação

---

1. Considerando o que estudamos até o momento, observe o último jogo que você jogou e tente listar quais as suas mecânicas primárias e quais as suas mecânicas secundárias.
2. Se você fosse desenvolver um jogo do gênero FPS sobre a Guerra de Canudos, como você definiria o espaço funcional do jogo?
3. As máquinas de estado finito são ótimas para modelar comportamentos. Como você representaria o seu próprio comportamento em um restaurante através de uma máquina de estados?

4. Se você fosse contratado para fazer um jogo sobre viagem no tempo, como você poderia explorar o uso das mecânicas temporais dentro do jogo?
5. Saber analisar e dissecar um jogo é uma característica importante para um designer de jogos. Tente pegar um jogo que você jogou recentemente ou está jogando e descreva quais mecânicas relacionadas ao espaço, tempo e estado dos objetos vocês conseguem observar.

## Referências

---

ADAMS, Ernest; DORMANS, Joris. **Game mechanics**: advanced game design. New Riders, 2012.

BRATHWAITE, Brenda; SCHREIBER, Ian. **Challenges for game designers**. Cengage Learning, 2009.

COOK, Daniel. **What are game mechanics**. Disponível em: <<http://lostgarden.com/2006/10/what-are-game-mechanics.html>>. Acesso em 12 ago. 2015.

JÄRVINEN, Aki. **Games without frontiers**: Theories and methods for game studies and design. Tampere University Press, 2008.

LUNDGREN, Sus; BJORK, Staffan. Game mechanics: Describing computer-augmented games in terms of interaction. In: The PLAY studio, The Interactive Institute (Org.). **Proceedings of TIDSE**. Göteborg, SWEDEN, 2003.

ROGERS, Scott. **Level Up!** The guide to great video game design. John Wiley & Sons, 2010.

Rouse III, R. **Game Design Theory and Practice**. Plano, Texas: Wordware Publishing Inc., 2005.

SALEN, K.; ZIMMERMAN, Eric. **Rules of Play**. Game Design Fundamentals. Cambridge, Massachusetts: The MIT Press. 2004.

SHELL, Jesse. **The Art of Game Design**: a book of lenses. CRC Press, 2008.

SICART, Miguel. Defining game mechanics. **Game Studies**, v. 8, n. 2, p. 1-14, 2008.