

# O QUE É UM JOGO CLICKER E O QUE O TORNA DIVERTIDO

Clístenny Lohran B. Magalhães, MSc Sandir Rodrigues Campos

[lohcli@gmail.com](mailto:lohcli@gmail.com), [sandirr@gmail.com](mailto:sandirr@gmail.com)

**Abstract**—The objective of this document is to show what makes a clicker game fun by presenting most of the mechanics found in clickers already on the market, this material will include from the user experience with the design to the mechanics developed to keep the player entertained even after hours of gameplay. It will also talk a little about how clicker games catch people's attention and what sensations and motivations players enjoy when playing.

**Keywords:** *clicker, sensations, gameplay, games, mechanics, entertained.*

## I. INTRODUÇÃO

Jogos de clicar ou jogos incrementais são jogos que consistem em uma mecânica simples na qual o jogador só precisa clicar repetidamente,

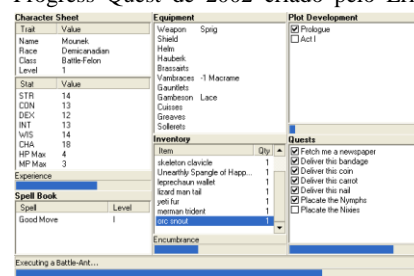
Quais as mecânicas escolhidas pelos desenvolvedores que podem de fato tornar um jogo clicker uma experiência atraente?

Uma das questões levantadas trata-se de que o design para este tipo de jogo tem uma importância realmente relevante, ou somente a jogabilidade, por si só, já se torna fator determinante para a atração do jogador, dentro deste contexto quais são as mecânicas mais procuradas e “obrigatórias” para manter o jogador entretido e interessado?

Quanto mais avançamos no assunto de jogos incrementais, mais somos capazes de encontrar uma divisão de opiniões entre os jogadores que gostam e entre os que não gostam deste tipo de jogo, porém, quais é

importante apontar os motivos que fazem alguns não suportarem este tipo de mecânica enquanto outros simplesmente conseguem passar horas, dias ou até mesmo meses jogando diversos desses jogos ao mesmo tempo.

Este estilo de jogo assim como todos os outros tem uma origem, o considerado primeiro jogo clicker foi atribuído ao Progress Quest de 2002 criado pelo Eric



Fredriksen, este jogo era uma paródia das estatísticas e auto-ataque do MMORPG.

Imagem 1: Interface do jogo Progress Quest de 2002.

Alguns jogos incrementais surgem com este mesmo objetivo até mesmo nos dias atuais, eles se baseiam nas mecânicas de algum jogo e criam uma versão clicker como por exemplo o jogo Grindcraft, este jogo é claramente uma versão incremental do famoso jogo Minecraft, no Grindcraft o jogador pode utilizar do sistema de criação do Minecraft para progredir, quanto mais você clica mais recursos você consegue para progredir no jogo, porém estes jogos também criam mecânicas próprias como por exemplo a possibilidade de utilizar os

**Comentado [sc1]:** Falta uma ligação do parágrafo anterior a esse

aldeões para realizar atividades que no jogo original não é possível, aqui podemos encontrar uma divisão de opinião entre os jogadores, uns defendem que se sentem muito melhores jogando o Minecraft de fato pois passa uma sensação de progresso real, porém o outro lado defende fortemente as mecânicas criadas pelo desenvolvedor do Grindcraft, o que nos leva novamente a mesma pergunta, o que torna um jogo incremental atraente de fato, por quê existe uma divergência de opiniões entre quem gosta e não gosta de jogos incrementais?

## II. O QUE TORNA UM JOGO INCREMENTAL ATRAENTE?

Este tipo de jogo utiliza de um conjunto de fatores que em harmonia proporcionam uma experiência agradável ao jogador, mas a base de tudo isso é o fato dos jogos incrementais serem simples e passarem ao jogador uma sensação de progresso, o conceito do jogo é relativamente simples, você clica consegue dinheiro ou a moeda utilizada no jogo e com este dinheiro você é capaz de comprar melhorias que irão acelerar ou automatizar o seu jogo, a mecânica é relativamente simples, clicar, comprar melhorias e conseguir cada vez mais dinheiro, e por mais que isso pareça muito simples já é o suficiente para prender milhares de jogadores aos jogos incrementais, entretanto alguns desenvolvedores adotam das mais diversas estratégias para transformar os jogos em algo ainda mais viciante e satisfatório, o Cookie Clicker por exemplo utiliza um sistema de “Renascimento”, quando o jogador chega em um certo ponto do jogo fica quase que impossível progredir sem efetuar o renascimento, ao renascer você ganha

bônus e melhorias permanentes em troca de perder todo o seu progresso conquistado ao longo do jogo, porém estes bônus e melhorias não serão perdidas caso você escolha renascer novamente, você começa do zero porém com vantagens que fazem você alcançar o seu progresso anterior com muito mais facilidade do que no caso de não escolher renascer, então a sensação de progresso é mantida e aprimorada tornando o jogo ainda mais interessante.

Outros jogos adotam o sistema de conquistas, muitos jogadores amam ter uma conquista para compartilhar com um amigo ou simplesmente para ter algo a mais em seu currículo gamer, e tem algo melhor para passar a sensação de progresso do que um sistema de conquistas bem implementado?

Os métodos adotados pelos desenvolvedores são muitos, porém nem todos os jogos clicker adotam ou abraçam estas estratégias, devemos admitir que em certo ponto um jogo incremental no qual seu único objetivo é clicar e conseguir melhorias infinitamente se tornaria algo maçante e não atrativo para muitos, porém mesmo estes jogos aparentemente simples e sem graça tem o seu público.

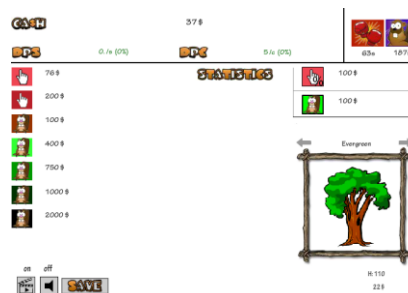


Imagem 2: Jogo clicker com mecânicas simples.

Mesmo que os jogos incrementais com uma mecânica relativamente mais simples não pareçam tão atraentes, grande

parte deles contam com uma mecânica em específico que para muito é o ponto chave do jogo, o fato de que você não precisa de fato jogar para que o progresso ocorra, o Cookie Clicker é um exemplo claro desta funcionalidade, a partir do momento que você compra a primeira melhoria você já nota uma produção de 0.4 cookies por segundo, ou seja você não precisa mais clicar para gerar Cookies, porém com o passar do tempo a quantidade de Cookies gerados por segundo aumenta até chegar a milhões ou até mesmo bilhões, fazendo com que o jogador possa jogar outros jogos clicker ou dormir enquanto o seu progresso aumenta sem que o jogador tenha que de fato estar presente no jogo, e esta mecânica é utilizada até mesmo no IDLE Woodcutting que mesmo sendo um jogo simples adotou esta mecânica base.

### III. O DESIGN TEM RELEVANCIA?

Grande parte dos jogos incrementais de navegador não contam com um design profissional, alguns contam com desenhos que são literalmente rabiscos feitos no paint, porém até quando isso pode afetar a jogabilidade?

O design de um jogo é sim algo que pode se tornar o fator decisivo entre jogar ou não, porém nem sempre um design ruim remete a uma jogabilidade pior ainda, existem jogos que são somente botões e textos e mesmo assim não deixam de oferecer uma jogabilidade atraente, como por exemplo o Swarm Simulator.



Imagem 3: Interface do jogo Swarm Simulator.

Por mais que a interface do jogo seja simples e sem ilustrações ou efeitos, o jogo não deixa de oferecer uma experiência agradável, mas é claro que um design com animações e efeitos atrai ainda mais a atenção, afinal muitos de nós temos o costume de julgar um jogo como bom ou ruim somente pela imagem inicial ou pelos gráficos, com os jogos incrementais não seria diferente, existem estudos muito profundos quanto ao design, e o quanto o mesmo pode ser um fator determinante de extrema importância para o jogador, existem estratégias que quando aplicadas corretamente podem tornar a experiência ainda mais agradável e convidativa.

Ao criar o design de um jogo devemos tomar cuidado com a quantidade de informações que o jogador irá encontrar na tela, podemos encontrar alguns jogos que possuem tantas informações que acabam se tornando assustadores a primeiro momento, algumas pessoas simplesmente desistem deste tipo de jogo enquanto outras se sentem desafiadas, é algo único de cada ser humano, porém informações em excesso sem qualquer tipo de explicação ou guias visuais pode sim ser um fator que afetará na decisão de jogar ou não.

### IV. TECNOLOGIAS UTILIZADAS

Quando se trata das tecnologias utilizadas para desenvolver um jogo do tipo clicker, existem diversas ferramentas que podem ser utilizadas, algumas são de uso gratuito e outras cobram um certo preço para que possamos utilizar ou até mesmo comercializar o produto desenvolvido na mesma, dentre as mais comuns estão a Unity e a Unreal Engine, ambos são motores gráficos gratuitos porém a Unity conta com certas restrições que só podem ser quebradas quando adquiridos planos dentro da mesma, porém quando se trata de criar um jogo no estilo Incremental, não existem limites, este tipo de jogo pode ser criado em qualquer linguagem de programação, em qualquer plataforma e com muita facilidade, afinal basta criar um botão ou alguma forma de incrementar um valor em uma variável, a tecnologia utilizada no MoneyClicker é a biblioteca React, normalmente utilizada para criação de sites de maneira mais simples a biblioteca desenvolvida pelo Facebook também oferece diversas facilidades para se desenvolver um jogo Incremental, e sinceramente, por que não?

O React foi criado em 2011 pelo time do Facebook, ele surgiu para otimizar a atualização e sincronização das atividades simultâneas no feed de notícias da rede social, sendo alguns deles o chat os status a listagem de contatos entre outros.



Imagem 4: Logo da biblioteca React.

O React traz como principal vantagem a uso de componentes na criação de páginas, deste modo podemos criar um componente e utilizá-lo em qualquer parte do nosso site, isso faz com que o reaproveitamento de código seja um dos fatores mais importantes da biblioteca, além disso ele pode ser utilizado para construção de interfaces reutilizáveis já que cada um dos componentes pode ser manipulado de forma distinta.

O React assim como outras bibliotecas utiliza o Javascript como linguagem de programação, então para extrair todo o potencial da biblioteca será necessário ter uma sólida base de conhecimento em Javascript.

Outro detalhe importante desta biblioteca é o chamado SPA ou Single Page Application, SPA é uma aplicação que carrega apenas um único documento Web e em seguida, atualiza o conteúdo do corpo desse documento único por meio de APIs Javascript.

De modo geral o carregamento de todos os recursos como CSS, Javascript e HTML ocorre uma única vez que é quando o usuário acessa a aplicação, neste primeiro acesso tudo que deve ser entregue ao usuário já é entregue, a partir deste ponto quando o usuário começar a mudar de páginas ou rotas no site o mesmo não terá de solicitar as páginas ao servidor pois todas elas já estão previamente carregadas e prontas para serem acessadas, por isso se diz que o carregamento das páginas e recursos é de responsabilidade do cliente já que o conteúdo da página é carregado pelo Javascript uma linguagem cliente-side.

Mas entre tantas engines e opções próprias para a criação de jogos, por que utilizar uma biblioteca para criar um jogo web? Um jogo famoso já citado

anteriormente neste documento é o Cookie Clicker, o mesmo utiliza Javascript em sua construção, criar um jogo somente com HTML, CSS e Javascript sem o auxílio de uma biblioteca como o React é totalmente possível, porém como já dito antes, o React facilita e muito no desenvolvimento de qualquer aplicação seja um site ou um jogo, então a pergunta que restou é, por que desenvolver um jogo incremental para web quando existem diversas engines que facilitam este processo? Normalmente estas engines não oferecem qualquer tipo de suporte para a criação de um jogo para navegador, o que é totalmente o contrário quando o assunto é criar um jogo no Javascript que é a linguagem padrão dos navegadores atuais, outra vantagem se diz principalmente na construção das interfaces, para quem não possui habilidades no que se diz criar interfaces em softwares como Photoshop ou Figma o CSS se torna uma opção muito mais amigável e favorável ao desenvolvedor, além do mais o CSS oferece suporte a animações e manipulação de imagens.

## V. COMO SALVAR OS DADOS?

Salvar o progresso do jogador sem dúvidas é uma das principais prioridades, afinal creio que nenhum jogador gostaria de passar horas jogando e ao recarregar a página, perceber que todo o seu progresso foi perdido, porém como salvar estes dados no navegador sem utilizar um banco de dados? É aí que entra o Local Storage do navegador, cada navegador conta com o seu tamanho de Local Storage, o Google Chrome por exemplo oferece um tamanho de 4MB, pode parecer pouco porém vamos entender quais são os tipos de dados gerados pelo usuário, em um jogo incremental em sua grande maioria

trabalhamos unicamente com alteração de números e valores e até mesmo estados, ou seja, tudo que o jogador faz é alterar valores de variáveis que já foram criadas no código fonte, sabendo disso podemos utilizar de recursos como o Local Storage para armazenar o valor dessas variáveis, como por exemplo o dinheiro total do jogador, de tempo em tempo um timer realiza a gravação desses dados no Local Storage, ou seja assim que o jogador entra no Game cabe ao desenvolvedor dizer que os dados devem ser carregados de acordo com as informações gravadas no Local Storage do usuário, é claro que isso possui limitações como por exemplo no caso do usuário mudar de navegador ou limpar os dados existentes no Local Storage, porém existem formas de contornar este problema.

Como os dados ficam gravados no navegador podemos utilizar algumas estratégias para que o jogador possa pegar o seu progresso e transferir de navegador ou guarda-lo em algum lugar para uso futuro.

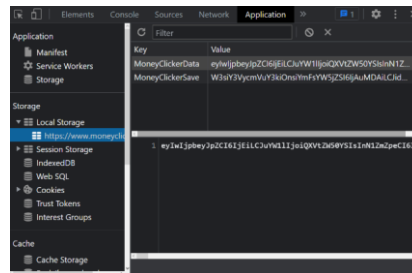


Imagem 5: Exemplo de dados armazenados no Local Storage.

Agora encontramos outro problema, se os dados estão armazenados no lado do jogador, significa que o mesmo pode alterar estes dados e trapacear no jogo? Sim, é exatamente isso, por isso utilizamos formas de dificultar a visualização destes

dados, uma das formas encontradas é criptografar os dados para base64, isso não impede que o jogador vá conseguir trapacear, porém dificulta já que o mesmo teria que ter um conhecimento no assunto de criptografia, infelizmente não existe uma forma de manter os dados 100% seguros sem utilizar um servidor no backend para guardar estas informações, mas qual jogo não é manipulável? Infelizmente sempre vão existir jogadores motivados a trapacear de alguma forma, porém em jogos de um jogador onde os jogadores não estão competindo globalmente, se um decidir trapacear não afetará a gameplay de nenhum outro jogador, somente a experiência daquele que resolver trapacear, então se preocupar com trapaceiros e com formas de dificultar ao máximo não é algo que deve receber uma prioridade, é claro que podemos investir em algumas coisas para dificultar porém atualmente é impossível criar um jogo 100% client-side no qual os dados estejam totalmente seguros e a prova de qualquer tipo de modificação.

## VI. CONCLUSÃO

Criar um jogo bem produzido não é uma tarefa fácil, existem diversos processos e fatores que devem ser levados em conta no desenvolvimento de games sejam quais forem seus estilos, por mais que os jogos incrementais sejam considerados os mais simples por muitos, até mesmos estes jogos possuem o seu valor e a sua complexidade.

Por mais que consigamos encontrar diversas opções de jogos no mercado, cada jogo sempre conta com suas mecânicas e estratégias únicas para manter o jogador entretido e disposto a investir diversas horas do seu tempo atrás de progresso,

conquitas e recompensas, cabe a cada desenvolvedor ser capaz de criar e proporcionar a melhor experiência possível para o jogador.

## REFERÊNCIAS

- [1] Matthew Thomas Payne e Nina B. Huntemann,  
[https://books.google.com.br/books?hl=pt-BR&lr=&id=V7iIDwAAQBAJ&oi=fnd&pg=PA200&dq=clicker+games&ots=\\_aTVJuOdES&sig=epT0lhR14qVQOIZYk3nYMDBIA2k#v=onepage&q&f=false](https://books.google.com.br/books?hl=pt-BR&lr=&id=V7iIDwAAQBAJ&oi=fnd&pg=PA200&dq=clicker+games&ots=_aTVJuOdES&sig=epT0lhR14qVQOIZYk3nYMDBIA2k#v=onepage&q&f=false)  
2019
- [2] NachoMeister,  
<https://www.kongregate.com/games/NachoMeister/idle-woodcutting>  
2014
- [3] Katta Spiel, Sultan A. Alharthi, Andrew Jian-lan Cen, Jessica Hammer, Lennart E. Nacke, Z O. Toups, Theresa Jean Tanenbaum.  
<https://dl.acm.org/doi/abs/10.1145/3311350.3347180>  
2019
- [4] Devmedia Wladimir,  
[https://www.devmedia.com.br/react-spa-entendendo-o-conceito-de-uma-single-page-application/43490#:~:text=SPA%20\(Single%20Page%20Application\)%20é,por%20meio%20de%20APIs%20JavaScript.](https://www.devmedia.com.br/react-spa-entendendo-o-conceito-de-uma-single-page-application/43490#:~:text=SPA%20(Single%20Page%20Application)%20é,por%20meio%20de%20APIs%20JavaScript.)  
2022
- [5] Kenzie Ugo Roveda,  
<https://kenzie.com.br/blog/react/16/11/2020>