

Guiando com uma mão invisível: Explorando metroidvanias e sua não-linearidade guiada

Josué Pedro Soares
Departamento de Fotografia e Cinema
UFMG
Curitiba, Brasil
notahazard@gmail.com

Rosilane Ribeiro da Mota
Departamento de Fotografia e Cinema
UFMG
Belo Horizonte, Brasil
rosilane@ufmg.br

Resumo — Este artigo analisa um subgênero específico de jogos, o Metroidvania. Jogos encapsulados por este subgênero são caracterizados por um mapa interconectado, progressão não linear e obstáculos que são ultrapassados por habilidades de personagem. Metroidvanias colocam o jogador no comando da progressão do jogo, dando a ele agência na decisão de onde ir e elevando a sensação de exploração e descoberta. O artigo explora uma pequena seleção de jogos do subgênero, para entender que elementos compõem estes jogos e que conceitos os desenvolvedores destes jogos usam para guiar o jogador de forma indireta, a fim de não tirar a agência dele.

Palavras-chave — metroidvania, plataforma, progressão, exploração, não-linear, subgênero

I. INTRODUÇÃO

Em 1986, *Metroid* foi lançado para NES, onde começaria a experimentar novas ideias e adquirir um público de nicho. Mas é nos anos 90 que a franquia realmente causaria um impacto na indústria com *Super Metroid* (1994) para o *Super Nintendo*. Sendo o terceiro jogo da franquia, ele refina os conceitos e mecânicas que haviam sido apenas experimentados nos primeiros jogos.

Seu impacto, porém, não se dá apenas ao próprio mérito, mas também ao sucesso de *Castlevania: Symphony of the Night* (1997) para o PlayStation 1. Os dois jogos trabalham conceitos similares de exploração e progresso, duas coisas que outros jogos de plataforma da época experimentavam pouco, sem grandes mudanças no padrão previamente estabelecido. Os jogadores que queriam uma experiência similar a *Metroid* encontraram a mesma experiência em *Castlevania* e vice-versa.

Desde então diversos jogos replicaram e expandiram seus conceitos e filosofias de *design*, dando luz a um novo subgênero. Esses jogos seriam chamados pela comunidade de *metroidvanias*, um *portmanteau*¹ dos nomes dos dois jogos considerados precursores do subgênero.

Mas, afinal, o que faz desses jogos tão diferentes ao ponto de criar um sub-gênero novo? Dentre os vários jogos que são considerados metroidvanias, as maiores semelhanças que encontramos são de como estes lidam com a exploração do ambiente e progressão de personagem. Metroidvanias são geralmente caracterizados por um

ambiente interconectado que se expande para todas as direções, onde, ao invés de seguir uma progressão linear, a progressão é definida por habilidades ou ferramentas que o personagem adquire ao longo do jogo. Além disso, os jogos costumam ser, ao menos em parte, não-lineares, dando a liberdade ao jogador de explorar o espaço do jogo na ordem e ritmo que preferir. É comum que esses jogos tenham seu movimento em plataformas 2D, mas isso não é tanto um requerimento e mais um fator recorrente. Esses pontos também variam em prevalência de jogo para jogo, mas de maneira geral, jogos considerados metroidvanias têm a maioria dessas características.

Como então os desenvolvedores proporcionam uma experiência não-linear de exploração sem deixar o jogador completamente perdido no espaço do jogo? E como e quais mecânicas e conceitos deste sub-gênero podem ser aplicados para proporcionar uma experiência dentro do subgênero Metroidvania? Esse artigo tem o objetivo de caracterizar esse subgênero de jogos, identificar o que faz um jogo ser considerado um metroidvania, e discutir como são usados os artifícios de jogabilidade do subgênero para guiar o jogador indiretamente.

Entendendo isso, esse conhecimento pode ser usado para produzir um jogo dentro do subgênero ou até mesmo aplicar seus conceitos em jogos de gêneros diferentes. Se não isto, gerar uma nova apreciação e aprendizado com as decisões de *design* que desenvolvedores do sub-gênero tomaram.

A próxima seção discute as referências e bases teóricas deste artigo, também passando pelo contexto histórico da nomenclatura do subgênero e selecionando os jogos para estudo de caso. Depois disso, o artigo se dedica a analisar as ferramentas, mecânicas e artifícios usados pelos desenvolvedores para proporcionar a experiência de um metroidvania. Após isso, é analisado como os desenvolvedores guiam o jogador de forma sutil apenas com suas decisões de *design*, seguido da conclusão do artigo.

II. APORTES TEÓRICOS E METODOLÓGICOS

Definir o que é um metroidvania não é uma tarefa simples. O próprio termo tem uma origem vaga, e até hoje gera discussões sobre o quão descritivo é para o subgênero. Antes de nos aprofundarmos nos detalhes que compõem um metroidvania, é importante analisar o contexto e origem do termo em si.

¹ Uma palavra que mistura os sons e significado de duas outras, por exemplo motel (de motor e hotel)

A. Dando significado ao termo “Metroidvania”

O termo Metroidvania é um *portmanteau*, uma união de duas palavras, que no caso são os títulos dos dois jogos que são considerados precursores para o sub-gênero: *Metroid* e *Castlevania*.

A origem da união dessas duas palavras, porém, é difícil de retrair. Sabemos apenas que o termo provavelmente surgiu em antigos fóruns de discussão de jogos, que não temos mais como acessar. Existe porém um consenso, de que quem popularizou o termo foi Jeremy Parish, editor-chefe do blog IUP [1]. Jeremy, em um *post* de 2011, explica que o termo inicialmente foi criado para falar dos jogos da franquia *Castlevania* que possuíam um *design* similar à *Metroid*, já que os primeiros *Castlevania* eram muito diferentes [2]. Desde então, o termo passou a encapsular qualquer jogo que tivesse muitas similaridades em seu *design* a esses dois clássicos.

Seria esse, porém, um nome adequado para descrever estes jogos? Vadim Mikhnov (2015), opina em seu blog [3], que o termo não é um bom nome para o subgênero. O termo falha em descrever o tipo de experiência que esses jogos passam, além de ficar preso ao passado dos clássicos que apenas introduziram ideias que hoje foram refinadas e elevadas por outros jogos.

Mas ao mesmo tempo, é difícil se divorciar do termo que hoje é tão bem conhecido, até por pessoas que nunca ouviram falar dos clássicos. Jeremy Parish, no mesmo *post* já mencionado, comenta que também não gosta do termo, mas que apesar disso ele é conciso, direto e fácil de usar. Características que, de acordo com ele, são importantes para um termo sobreviver no instável panorama linguístico atual.

Em um painel da GDC de 2014 [4], Koji Igarashi, diretor de *Castlevania*, comenta sobre o termo. Ele disse que o nome o surpreendeu, pois a sua maior inspiração por trás de *Castlevania Symphony of the Night* (1997) não era *Metroid* (1986) e sim *The Legend of Zelda* (1986). Ele também diz que sente gratidão pelo termo ter sido popularizado, mesmo que impreciso. Quando saiu da Konami, disseram a Igarashi que ele não poderia usar o nome *Castlevania* em nenhuma forma, nem para remeter a jogos similares que ele gostaria de fazer. Metroidvania, porém, é um termo que ele pode usar à vontade.

Com o nome metroidvania sendo tão bem conhecido e utilizado pela comunidade de jogos em geral, é difícil de imaginar o subgênero ser descrito por um novo termo. E se tem algo que os anos que se passaram desde sua concepção indicaram, é que o termo provavelmente é definitivo.

B. O que faz de um jogo um Metroidvania

Agora com este nome incomum contextualizado, resta entender quais as similaridades entre diferentes Metroidvanias. Quais seriam então as características indispensáveis para categorizar os jogos neste subgênero?

Mark Brown descreve o que caracteriza um metroidvania, na segunda temporada de sua série de vídeos *Boss Keys* (2018; 2019). Um metroidvania se passa em um único mapa, contínuo e interconectado, que se expande para todas as direções, se dobra em si mesmo e se conecta por atalhos e passagens.

Em certo momento o jogador se encontra impedido por algum obstáculo intransponível. Ele precisa então explorar outro lugar e encontrar um novo *power up* permanente, que

dá ao seu personagem uma nova habilidade. Isso permite que ele volte a antigos obstáculos e use suas novas habilidades para progredir no jogo.

Essa é uma construção simples, mas que é feita com a intenção de criar um jogo com uma potente sensação de navegação e exploração. Ela dá ao jogador a tarefa de criar um mapa do espaço, seja este físico ou mental, e o recompensa com progresso ou melhorias para o personagem jogável. Quando o jogador lembra onde se encontram os obstáculos que ele antes não podia passar, ele os relaciona à nova habilidade que o ajudará a vencer estes obstáculos.

O website *platformadventure* [5] lista pontos concisos que definem o que é necessário para um jogo ser considerado um metroidvania. Em suma, os principais pontos são:

- Ambiente de jogo interconectado e expansivo;
- Progressão definida por habilidades ou ferramentas que o personagem adquire;
- Não-linearidade, parcial ou completa, da estrutura do jogo (incluindo caminhos não essenciais para completar o jogo);
- Locomoção em plataformas, tradicionalmente em duas dimensões.

Esses pontos podem ser mais ou menos prevalentes dependendo do jogo, mas para ser considerado um metroidvania, geralmente o jogo contém alguma combinação da maioria dos pontos listados.

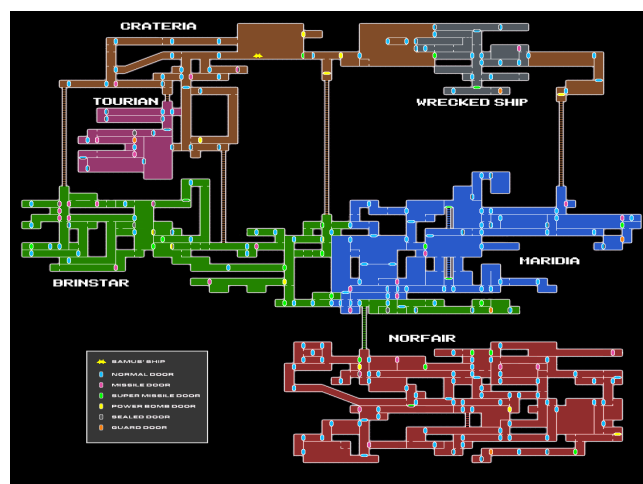


Fig. 1. Exemplo de um mapa interconectado. Fonte: S. Tsubasa [19].

C. Escolhendo jogos para análise

Com tudo isso em mente, temos o necessário para começar a olhar alguns exemplos e selecionar alguns jogos para estudo. É lógico pensar que o ideal seja olhar para os primeiros jogos das respectivas franquias que compõem o nome do subgênero, porém este não é o caso. O primeiro jogo, *Metroid* (1986), apesar de ter concebido vários dos conceitos de *design* que compõem um metroidvania, também é cheio de decisões de *design* que acabam funcionando ao contrário de suas próprias propostas. Coisas como passagem secretas que devem ser destruídas, sem nenhuma indicação, para progredir o jogo podem levar jogadores a se sentir perdidos e pararem de jogar o jogo ou procurar um guia externo.

Metroid II: Return of Samus (1991) também não é um bom candidato para esta análise, pois ele foi lançado para o

console *Game Boy* o que limitou bastante a liberdade de *design* dos desenvolvedores por conta da tela monocromática e pequena. Os desenvolvedores decidiram então fazer um jogo mais linear, que não apresenta motivo para explorar áreas pelas quais o jogador já passou.

O terceiro jogo da série, *Super Metroid* (1994), revisita as raízes estabelecidas no primeiro jogo e as refina. Os desenvolvedores removeram as mecânicas confusas que poderiam deixar os jogadores perdidos e fazer com que o jogo seja mais acessível e fácil de navegar, sem comprometer a proposta e dificuldade do jogo.

Castlevania é um caso ainda mais curioso. Todos os jogos antes de *Castlevania: Symphony of the Night* (1997) apresentaram *gameplay* completamente diferente, focando em ação e plataforma 2D, com fases tradicionais e lineares. Mas *Symphony of the Night*, sob a direção de um novo diretor, Koji Igarashi, experimentou conceitos totalmente diferentes dos apresentados pelos jogos anteriores. Após seu lançamento, todos os jogos da franquia seguiram seus passos ao invés dos primeiros jogos, mudando completamente como a franquia era vista por jogadores até hoje. *Symphony of the Night* contribuiu muito para o subgênero em si, mas ao mesmo tempo despojou de vários conceitos e mecânicas que são menos utilizados por outros jogos.

Vinte anos depois, tanto a Nintendo quanto a Konami, respectivos desenvolvedores de *Metroid* e *Castlevania*, pararam de produzir novos jogos para as suas franquias. Mas isso não quer dizer que o subgênero deixou de ganhar novos jogos. Desenvolvedores independentes começaram a experimentar com as ideias do subgênero e refinaram seus conceitos nos últimos 20 anos. Em 2017, a Team Cherry lançou *Hollow Knight*, um jogo que refinou ainda mais os conceitos do subgênero e adicionou ainda mais profundidade às suas mecânicas.

Para os objetivos deste artigo os jogos que serão principais referências serão *Super Metroid*, *Castlevania* e *Hollow Knight*, porém os outros jogos também serão mencionados, os quais serão citados na Ludografia.

D. Metodologia

A análise dos jogos, e do subgênero Metroidvania em si, é embasada principalmente no trabalho de Mark Brown [6] [7] [8] [9], que analisou em sua série *Boss Keys* os jogos que ele considera mais icônicos para o subgênero. Brown explora a perspectiva do jogador e do desenvolvedor em seus vídeos, analisando a estrutura de cada jogo de seu início até o fim, comentando sobre as decisões de *design* importantes para o jogo específico.

Brown define em sua análise as técnicas e conceitos que fazem um Metroidvania funcionar, procurando denominar também as falhas e avanços de cada jogo. Suas considerações, então, são usadas para embasar a análise dos jogos que foram jogados para a escrita deste artigo.

Os jogos selecionados também foram jogados para confirmação das informações apresentadas, assim como para analisar o *design* e construção do jogo, dentro das lentes do subgênero.

Também foram lidos diversos artigos de opinião [12] [13] [14], entrevistas com desenvolvedores [10] e análises relacionadas [11], para construir a estrutura das ideias do artigo.

III. EXPLORANDO A COMPOSIÇÃO DE UM METROIDVANIA

Agora que estabelecemos uma base contextual sobre Metroidvanias, podemos observar exemplos dos elementos e mecânicas que os compõem, e se diferem de outros subgêneros.

A. Mapas e caminhos

Jogos *sidescrollers* 2D populares na época, como *Super Mario Bros.* (1985), *Contra* (1987) e *Sonic the Hedgehog* (1991), seguem uma estrutura similar. O personagem se locomove para a direita, e a câmera acompanha o personagem na mesma direção, em alguns casos até mesmo o impedindo de retornar para a esquerda. Esse processo continua até o personagem chegar no final daquela fase e ser transportado para o início de outra fase, onde o processo é repetido novamente.

Comparado a outros *sidescrollers*, o Metroidvania anula completamente o conceito de fases dando ao jogador várias salas e caminhos que se conectam, deixando ele explorar livremente. *Metroid* (1986) e *Super Metroid* (1994) tentam introduzir essa ideia ao jogador fazendo ele encontrar um obstáculo no lado direito, o impedindo de progredir.



Fig. 2. O jogador não consegue entrar na passagem. Fonte: Metroid [20].

O jogador então segue para o lado esquerdo do mapa, onde, após explorar um pouco, encontra a habilidade de *morph ball* que o permite passar pelo obstáculo que encontrou do lado direito no início do jogo. Mesmo se não for o caso e o jogador andar para o lado esquerdo primeiro, após o *power up* o jogador se encontra novamente sem saída, o que o obriga a voltar para o início do mapa e explorar o lado direito, onde utiliza sua nova habilidade para progredir.



Fig. 3. Agora com a Morph Ball é possível avançar pela passagem.

Apesar de *Metroid* (1986) não possuir um mapa, apresentá-lo no jogo em si se tornaria uma característica marcante do subgênero depois de *Super Metroid* (1994) e *Castlevania Symphony of the Night* (1997).



Fig. 4. Mapa de Super Metroid [22].

Castlevania é, na verdade, aquele que mais influenciou os próximos jogos com seu mapa. O mapa de *Super Metroid* era mais obtuso, ele não deixava ver muito do mapa de uma vez só, e também não indicava bem onde existiam portas ou passagens. Já *Castlevania* exibe seu mapa por completo na tela quando aberto e indica de forma mais clara onde é possível passar e onde não.



Fig. 5. Mapa completo de Castlevania: Symphony of the Night [23].

Apesar dessa diferença, os dois mapas são bem similares. Ambos representam o local onde o jogador se encontra no mapa de alguma forma, ambos marcam com ícones salas importantes, como por exemplo as salas onde você salva o jogo. E ambos preenchem o mapa conforme o jogador explora, adicionando novas salas e ícones ao mapa automaticamente.



Fig. 6. Mapa incompleto de Castlevania, exibindo apenas as salas onde o jogador esteve presente.

Ambos os jogos também apresentam um sistema onde é possível ver pedaços seletos do mapa antes de explorá-lo. Em *Super Metroid* você encontra terminais que preenchem pequenos pedaços do seu mapa quando os encontra, e em *Castlevania* você pode comprar mapas de uma loja. Isso revela até mesmo algumas salas e caminhos secretos, com a intenção de incitar o jogador a explorar, sem tirar totalmente a agência dele em completar e explorar o mapa do jogo.

Em *Hollow Knight* o sistema de mapeamento é expandido, e a mecânica adquire mais camadas. Inicialmente o jogador não tem um mapa quando encontra uma nova área, mas em cada área ele pode encontrar um personagem cartógrafo que vende seus mapas para ele. Estes mapas vêm incompletos, tendo informação o bastante só para guiar o personagem no início da área e ativar sua curiosidade de lugares incompletos no mapa. O mapa também não se preenche automaticamente conforme o jogador o explora como nos outros exemplos. Somente

quando o jogador acessa um local seguro para sentar é que seu personagem desenha a porção do mapa que viu. Isso não só deixa mais tensa a exploração de lugares desconhecidos, como também dá mais valor para os lugares seguros que o jogador encontra, além do próprio valor narrativo.

B. Trancas e chaves multiuso

Se desconstruirmos os obstáculos e *power ups* de um metroidvania, ele é um sistema de trancas e chaves. Geralmente cada obstáculo só pode ser “aberto” por uma habilidade específica, que aqui funciona como chave.

Sendo este o caso, é importante comunicar para o jogador de alguma forma quais “chaves” se encaixam em quais “trancas”. *Super Metroid* começa a trabalhar esse conceito de forma bem simples. Uma passagem pequena em que seu personagem não cabe, onde em seguida o jogador ganha a habilidade de se encolher em uma bola, sendo capaz de atravessar agora tais passagens.

Outra forma de comunicar certos obstáculos que é usada em diversos jogos é a de usar uma versão alternativa de um obstáculo que o jogador já consegue atravessar. No caso de *Super Metroid*, são usadas portas literais que podem ser abertas com mísseis. O jogador eventualmente encontra uma porta similar mas com uma cor diferente, e tentando abrir ela da forma costumeira percebe que o seu míssil é inefetivo. Isso cumpre duas funções para os desenvolvedores: passa a mensagem de que o jogador precisa explorar outro lugar e depois voltar naquele, e também comunica que a porta provavelmente é aberta por um míssil ou artefato similar. Quando o jogador encontra o *power up* de super mísseis, ele consegue facilmente fazer a ligação de que esses mísseis podem ser usados nas portas de outra cor.



Fig. 7. O jogador consegue utilizar mísseis para abrir a porta azul, porém quando se depara com portas vermelhas percebe que os mesmos mísseis não são capazes de abri-la. Fonte: Super Metroid [22].



Fig. 8. Outros jogos modernos até hoje usam este mesmo tipo de codificação. O objeto da direita é facilmente destruído, enquanto o da esquerda vai necessitar de uma melhoria para a arma do jogador. Fonte: Iconoclasts 2018 [25].

Mas por que abrir portas com mísseis, ou usar obstáculos que você precisa destruir com um ataque especial ao invés de simplesmente usar portas e chaves literais? Os Metroidvanias em geral trabalham com o conceito de habilidades multi-uso, isto é, habilidades que podem servir tanto para atravessar obstáculos como em outros aspectos do *gameplay*. Usando os super mísseis de exemplo, abrir portas é uma função importante, mas ele também é um ataque que o jogador poderá utilizar para derrotar os inimigos em combate. Isso não quer dizer que toda habilidade é útil apenas para progredir e para combate, outro exemplo em *Super Metroid* é o *Speed Booster*. Essa habilidade permite que o jogador corra muito rápido e consiga ignorar os inimigos no caminho. Isso ajuda o jogador a passar por portas que se fecham rapidamente e são essenciais para o progresso, mas também é uma ótima ferramenta para navegar por partes já exploradas do mapa de forma rápida sem preocupação.

Apresentar um uso fora do necessário para progredir no jogo faz com que a habilidade pareça muito mais valiosa quando o jogador a recebe, em comparação a receber uma simples chave para uma porta.

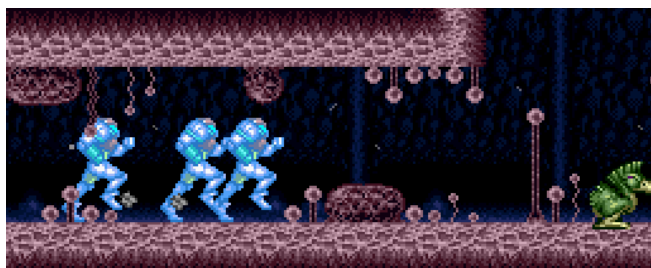


Fig. 9. Speed Booster sendo usado para atravessar rapidamente um corredor.

C. Progressão não-linear

Essa característica varia em prevalência de jogo para jogo, mas a não linearidade apoia imensamente a premissa do subgênero. Se o desenvolvedor, além de dar agência na escolha de onde explorar para o jogador, também deixar para que ele descubra várias formas diferentes de progredir, isso faz com que explorar lugares diferentes dos que são imediatamente óbvios se torne ainda mais valioso para o jogador.

Hollow Knight é talvez o jogo que foi mais longe em desenvolver essa ideia. Depois de um início mais fechado e introdutório, o jogo abre muitos lugares para o jogador explorar, mas vários destes caminhos eventualmente culminam no próximo objetivo. Isso assegura que o jogador não se perca, sem limitar muito as escolhas de exploração.



Fig. 10. Quatro possíveis caminhos diferentes que o jogador pode fazer para progredir em *Hollow Knight*. Fonte: Boss Keys [9].

D. Áreas secretas, conteúdo opcional e backtracking

Um componente comum de se ver em metroidvanias é voltar para áreas que já foram exploradas anteriormente, popularmente chamado de *backtracking*. Normalmente, isso não teria propósito na maioria dos jogos, mas Metroidvanias incentivam o jogador a voltar em áreas já exploradas com novas habilidades, onde encontram conteúdo opcional ou em alguns casos o caminho para progredir. O jogador também tem a oportunidade de sentir que dominou uma área que foi difícil anteriormente quando a explora pela segunda vez.

Outro elemento característico de um Metroidvania é espalhar pelo jogo coisas opcionais para o jogador encontrar. Isso faz com que mesmo quando o jogador explora algum caminho errado, ele ainda receba algo de valor no final. Mas, da mesma forma, o fato de ter algo a se ganhar explorando lugares não essenciais faz com que o jogador tenha um incentivo maior a explorar mais e também voltar em áreas que já foram exploradas, agora com novas habilidades. Os conteúdos opcionais variam, desde recursos, coletáveis, cosméticos, equipamentos, e até personagens e conteúdo narrativo.

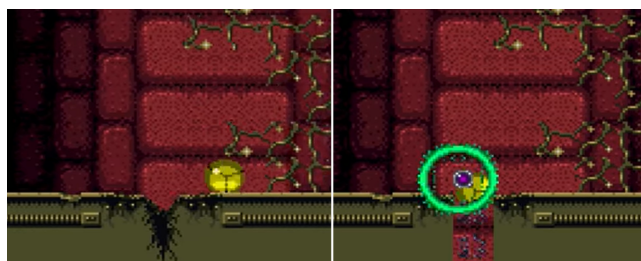


Fig. 11. Se colocada uma bomba na rachadura, é revelada uma passagem para uma área secreta. Fonte: *Super Metroid* [22].

Áreas secretas também são comuns desde o primeiro Metroid. Essas áreas geralmente são escondidas por algum pedaço do cenário que pode ser destruído, ou ultrapassado, exemplos comuns são paredes rachadas e arbustos densos. Essas áreas geralmente escondem apenas conteúdo opcional, para evitar que o jogador não consiga completar o jogo por não encontrar uma área secreta.

E. Movimentos avançados e quebra de sequência

Essa última característica é mais incomum entre os Metroidvanias, mas ainda é bem presente nos maiores nomes do subgênero. Estes jogos apresentam ao jogador opções de movimentação avançadas, muitas vezes presentes desde o início do jogo, sem ensinar elas diretamente ao jogador. Por conta disso, estes jogos não exigem que o jogador saiba usar essas manobras para concluir o jogo, mas os jogadores que as descobrem são recompensados não só com uma nova forma de explorar o cenário do jogo, como também a satisfação de descobrir um segredo sozinhos.

Isso se atrela a outra característica de jogos que apresentam esse tipo de técnica: a possibilidade de quebrar a sequência do jogo. Isto é, explorar áreas em que normalmente o jogador só poderia explorar mais tarde com alguma outra habilidade. Isso eleva ainda mais o nível de liberdade de exploração que os desenvolvedores apresentam para os jogadores. Já como os jogadores podem aprender tais técnicas, será mais aprofundado no próximo capítulo.

IV. GUIANDO COM UMA MÃO INVISÍVEL

A. Observando o lado oposto do espectro

Se observarmos como outros jogos lidam com exploração do seu ambiente e como ela é atrelada à sua progressão, existe um número crescente de exemplos que fazem uso de *waypoints* e mapas com ícones de objetivo.



Fig. 12. Uma seta aponta constantemente para o próximo objetivo do jogo. Fonte: BioShock 2 [26].

Waypoints são ícones dentro do jogo que apontam de alguma forma para o próximo objetivo do jogador. Eles são boas ferramentas de acessibilidade, mas também são usados como parte do *design* do jogo em si. Usar eles dessa forma ajuda a manter o interesse do jogador, impedindo que ele se perca e sempre tenha algum objetivo para se direcionar e ver mais conteúdo do jogo. A desvantagem de usar essa mecânica é que ela tira a agência do jogador de explorar o cenário do jogo, fazendo com que ao invés dele tomar decisões de onde ir, o jogo simplesmente dá a resposta para ele, diminuindo a sensação de exploração que o jogo poderia passar.

Para jogos com conteúdo não linear, existem exemplos que usam uma combinação de *waypoints* e ícones em um mapa que indicam os diversos objetivos do jogador, deixando ele selecionar seus próprios objetivos e incentivando a exploração. A quantidade de objetivos e atividades diferentes pode, porém, ser um pouco opressiva e confusa para alguns jogadores, deixando eles mais perdidos do que direcionados.



Fig. 13. Mapa com ícones de objetivo e atividades que o jogador pode ir atrás. Fonte: Assassins Creed: Odyssey [27].

Esses dois exemplos - como mostrado na Fig. 11 e Fig. 12 - são muito úteis para direcionar o jogador, e são ideais para certos tipos de jogos mesmo com suas desvantagens. Mas como *Metroidvânias* lidam com guiar o jogador se a proposta do subgênero é dar a agência da exploração para ele? Indicar a direção que o jogador deve ir de forma tão

direta é prejudicial para a experiência que um *metroidvânia* quer passar. Por outro lado, dar completo controle de onde ir e onde explorar para o jogador por si só pode facilmente fazer com que ele se sinta perdido, sem saber onde explorar ou o que fazer.

Para resolver isso então, *Metroidvânias* usam ferramentas sutis para direcionar o jogador, mas tentam manter a ilusão de liberdade e agência para o jogador.

B. Ensinando organicamente

Um exemplo icônico de como os desenvolvedores ensinam o jogador a jogar de forma indireta é quando em *Super Metroid* o jogador encontra sua primeira habilidade: a *Morph Ball*. Quando o jogador encontra um círculo brilhante, seu primeiro instinto provavelmente é descer pelo buraco e ir atrás dele, sem pensar muito nas consequências. Quando ele faz isso, logo percebe que não é capaz de voltar pelo mesmo lugar que entrou.



Fig. 14. Jogador vindo da direita encontra um círculo diferente. Fonte: Super Metroid [22].

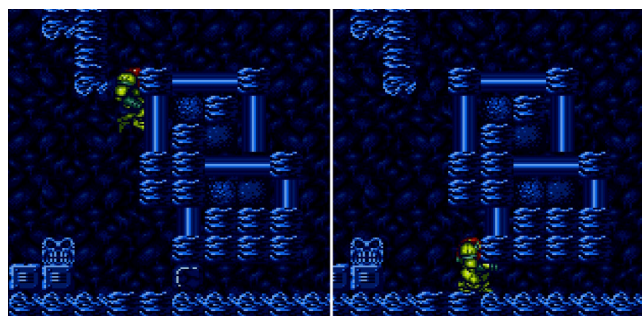


Fig. 15. O jogador não consegue pular alto o bastante nem se esgueirar pelo buraco embaixo.

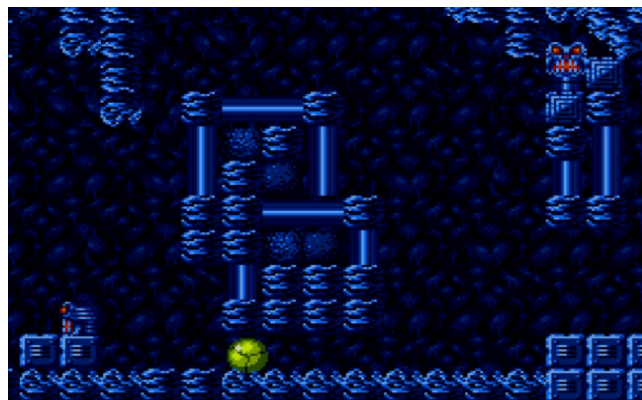


Fig. 16. Usando a habilidade de Morph Ball é possível escapar.

A única saída desse local é uma passagem estreita demais para o jogador passar. Porém quando o jogador adquire a nova habilidade tocando no símbolo brilhante a única mensagem que aparece no jogo é qual botão deve ser pressionado para utilizar a nova habilidade, no caso a *Morph Ball* apresentada na Fig. 15. O jogador naturalmente tentará pressionar o botão indicado e vai notar imediatamente que o seu personagem se encolhe em uma pequena esfera, pequena o bastante para passar pelo pequeno buraco e escapar da pequena sala.

O que isso cumpre é que garante que o jogador aprenda a como usar suas novas habilidades efetivamente antes de voltar a desbravar o cenário do jogo. Não só isso, como ensina para o jogador ao menos um dos contextos em que sua nova habilidade pode ser utilizada. O melhor exemplo desse tipo de ensinamento sutil é quando se adquire o *Ice Beam*, uma ferramenta em *Metroid* capaz de congelar projéteis e pequenos inimigos. Após receber a nova habilidade, o jogador precisa congelar um inimigo para atravessar a lava. Mas para chegar ali, inevitavelmente o jogador precisa cair por uma sala repleta destes mesmos inimigos, sem ter como voltar. Após usar o *Ice Beam* no mesmo inimigo, é provável que o jogador faça a conexão de que pode usar essa habilidade para escalar até o topo da outra sala, usando os inimigos como degraus.



Fig. 17. Na esquerda, logo depois de receber o Ice Beam, o jogador precisa congelar esse inimigo para sair da sala. Na sala da direita o jogador precisa congelar os inimigos e usá-los como escada para ir até outra área.

Essa técnica de ensinar o jogador a usar sua nova habilidade em um contexto relativamente seguro, logo após recebê-la, se torna comum entre vários Metroidvanias.

C. Usando elementos visuais para memorização

Supondo que o jogador aprendeu efetivamente como e onde usar suas novas habilidades, o próximo desafio que o desenvolvedor encontra é como fazer com que ele lembre onde usar essa habilidade. Uma forma de fazer isso, que é amplamente utilizada em Metroidvanias, é marcar o lugar na memória do jogador com algum recurso visual destacado.

Em *Super Metroid* talvez o exemplo mais icônico seja o cenário da Fig. 17, onde o jogador se depara com um rosto alienígena, que se difere de todo o resto de cenários apresentados anteriormente. A singularidade visual do rosto é feita para ativar a curiosidade do jogador e também marcar este local na sua memória. O jogador não consegue pular alto o bastante para acessar o rosto ainda, mas no futuro, quando adquirir a habilidade de pular mais alto, ele provavelmente irá se lembrar onde poderá utilizar sua nova habilidade.



Fig. 18. O jogador não consegue pular alto o bastante para subir até o rosto sinistro.

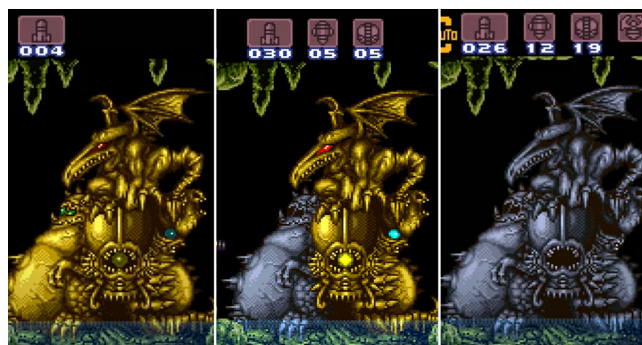


Fig. 19. O totem exibido em uma das áreas iniciais representa os 4 chefes do jogo. Quando um deles é derrotado, sua estátua fica cinza.

Super Metroid também usa outra forma elegante de comunicar ao jogador o seu objetivo final apenas com elementos visuais do cenário. Na Fig. 18 é representada uma estátua que fica na região inicial do jogo, quando o jogador a encontra pela primeira vez ele não possui nenhuma ideia do significado dela além de representar em seu topo a figura de Ridley, o principal antagonista do jogo, que é introduzido na primeira parte. A estátua por si só já é distinta o bastante de tudo o que o jogador já viu no jogo antes para ficar gravada na memória.

Eventualmente o jogador derrota algum dos chefes e talvez lembre de que sua aparência é idêntica à da estátua que viu antes. Caso ele volte até a sala da estátua verá o chão tremer e a estátua dourada se quebrar e ficar cinza. Isso cria a ideia de que quando o jogador derrotar os 4 chefes representados algo irá acontecer naquela situação, dando um objetivo para o jogador sem tirar dele a agência.

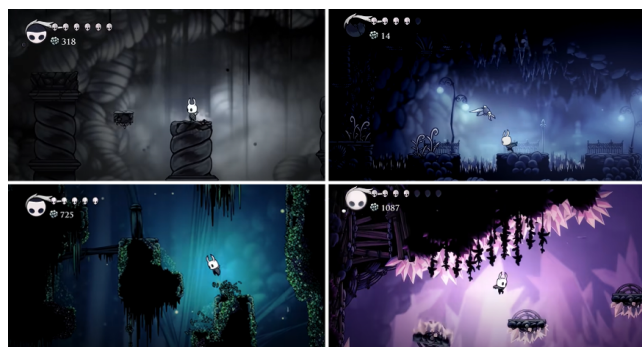


Fig. 20. Em *Hollow Knight* cada área tem uma paleta de cores e visuais completamente diferentes uma da outra. Fonte: Boss Keys [9].

Também é comum que, para facilitar a navegação pelo vasto mapa de um Metroidvania, o mapa seja dividido em zonas, onde cada uma delas tem um visual e estética completamente diferente uma das outras, ajudando o jogador a se localizar sem ter que abrir seu mapa o tempo todo.

D. Limitando e liberando a área explorável

Apesar de Metroidvanias apresentarem a ilusão de que o jogador está livre para explorar todo o cenário do jogo assim que começa, na realidade os desenvolvedores controlam sutilmente o quanto do mundo ele pode explorar. O quanto é possível de se explorar nunca é explícito para o jogador, e como seus obstáculos são em sua maioria orgânicos, como delineamos nas seções anteriores do artigo, eles não parecem bloquear seu progresso arbitrariamente. Mesmo os jogos que tem não-linearidade mais prevalente, como *Hollow Knight*, estes começam com uma pequena área disponível para o jogador explorar, que aumenta progressivamente conforme ele recebe novas habilidades.

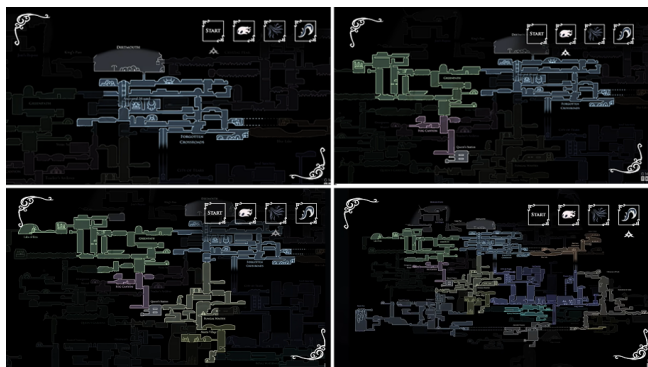


Fig. 21. A figura ilustra o quanto do mapa o jogador pode explorar, a cada iteração de habilidades que ganha, no caso as 4 primeiras. Fonte: Boss Keys [9]

Limitar a quantidade de conteúdo que o jogador tem acesso impede que ele se sinta sobrecarregado com a quantidade de opções. Isso é importante especialmente no início, mas conforme o jogador se acostuma mais com o jogo e o cenário, os desenvolvedores liberam progressivamente mais lugares para serem explorados. *Hollow Knight*, por exemplo, começa bem contido, e cada habilidade libera novas áreas de tamanho similar. Essas partes são não-lineares, mas ainda são bem diretas, específicas em termos de seu objetivo. Depois que o jogador encontra a *Mantis Claw* porém, o jogo abre exponencialmente, apresentando ao jogador diversas opções de exploração as quais ele pode escolher em qualquer ordem. Neste ponto, o jogador provavelmente já está acostumado tanto com o cenário quanto com suas habilidades para se sentir confortável em explorar livremente.

Similarmente, em *Super Metroid*, a área explorável começa relativamente pequena e aumenta conforme o jogador avança. Porém, ele usa uma tática diferente em conjunto; em alguns momentos-chave o jogador passa por uma passagem que corta o acesso do local de onde ele veio. Isso tem dois usos diferentes.

O primeiro é o mesmo que abrir aos poucos a área explorável; os desenvolvedores fazem com que o jogador perca acesso a uma área do jogo momentaneamente para

que suas opções diminuam no início do jogo e para que o incentive a pensar em como voltar.

O outro uso funciona em menor escala. Ao fechar o jogador em uma área para que ele não pense que é incapaz de resolver o obstáculo desta área em específico, ele não corre o risco de pensar que está indo para um lugar errado e que deve voltar depois. Eventualmente, *Super Metroid*, assim como *Hollow Knight*, fornece liberdade de escolha para o jogador e libera a maior parte do mapa para a exploração.

E. Descoberta natural de técnicas avançadas

Foi comentado na Seção III que alguns Metroidvanias têm opções de movimento avançadas que não são requeridas para avançar até o fim do jogo, nem são ensinadas diretamente. Como então o jogador aprende esses movimentos avançados? E como ele os utiliza?

Super Metroid tenta ensinar de forma simples. Em alguns dos locais opcionais que o jogador pode encontrar no jogo, existem salas com animais alienígenas não hostis. Assim que o jogador se aproxima, eles percorrem um caminho alternativo pela área. O jogador então pode tentar imitar os animais e notar que ele também sempre foi capaz de fazer estes mesmos movimentos para explorar o cenário do jogo.



Fig. 22. Os alienígenas símios pulam de parede em parede até alcançarem o topo. Fonte: *Super Metroid* [22].

Hollow Knight é mais sutil em comparação ao *Super Metroid*. Nele o jogador pode atacar em 3 direções quando se desloca pelo chão, mas no ar ele pode atacar também para baixo. Em qualquer uma das direções, o ataque faz com que o personagem do jogador recue quando atingir qualquer coisa sólida. Porém, quando ele faz isso no ar, o seu ataque o faz ganhar mais altura, o que pode ser usado para atravessar vários obstáculos. O jogo introduz sutilmente nos primeiros combates inimigos que defendem acima de suas cabeças, ajudando o jogador a perceber tanto que ele pode atacar inimigos por cima quanto que superfícies metálicas o fazem ganhar altura com o recuo.

Além dessas dicas sutis, uma das áreas do jogo é feita com a intenção de ajudar o jogador a entender naturalmente como usar essa técnica. As *Fungal Wastes* são uma área repleta de cogumelos, alguns deles saltitantes. Se o jogador atacar os cogumelos, estes o rebatem uma distância grande para trás. Quando então o jogador se encontra numa situação que não consegue alcançar a altura desejada apenas pulando nos cogumelos ele então os ataca no ar, efetivamente aprendendo a usar a técnica.

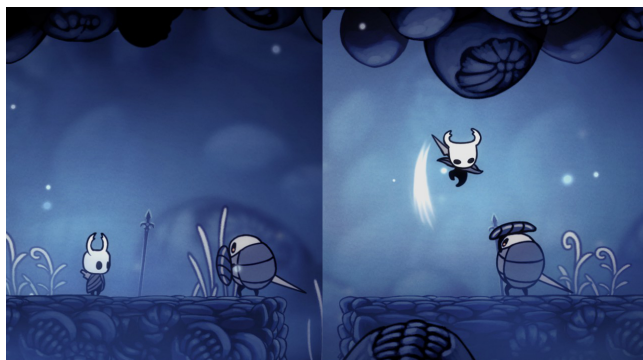


Fig. 23. Inimigo defende acima quando o jogador pula. Fonte: Hollow Knight [24].

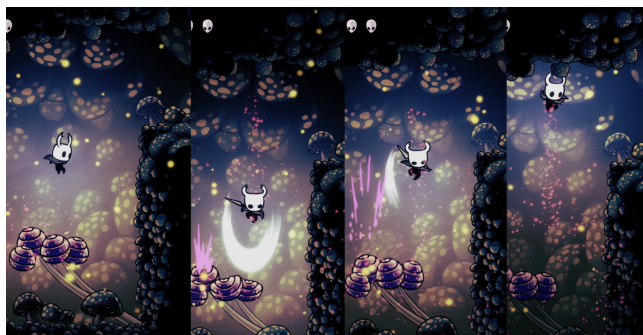


Fig. 24. Exemplo do ataque sendo usado para ganhar impulso vertical, batendo nos cogumelos da Fungal Wastes.

Os desenvolvedores então apresentam situações onde essa técnica pode ser usada para explorar lugares novos e encontrar segredos, apresentando ao jogador a oportunidade de sentir que descobriu sozinho grandes segredos usando uma técnica de movimento avançada. Além disso, como essas técnicas não são melhorias dadas ao personagem, e sim coisas que ele sempre pode fazer, isso recontextualiza todo o jogo, caso o jogador queira rejogá-lo do início em algum momento.



Fig. 25. Para passar por esse lago de ácido é necessário atacar a dura carapaça deste inimigo repetidamente, enquanto ele anda até o outro lado.

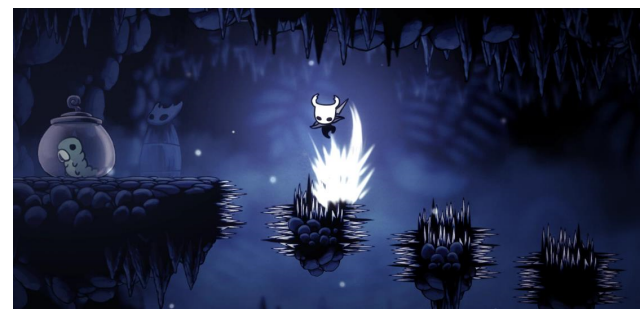


Fig. 26. Só é possível alcançar essa área usando a técnica para pular nos espinhos.

V. CONCLUSÃO

O Metroidvania é um subgênero que busca dar agência para o jogador na exploração, isto é, deixar que o jogador decida para onde ir sem nada dizer explicitamente sobre a direção correta. Jogos do subgênero realizam isso aplicando não-linearidade em seu *design*, dentro de um espaço interconectado, fazendo com que o jogador não só explore novos lugares, como também volte para lugares que já explorou com uma nova perspectiva, após avançar mais no jogo. Ele também recompensa o jogador por explorar avidamente o ambiente com conteúdo opcional, novas habilidades e progresso no jogo.

Para produzir um jogo que crie essa sensação, porém, é necessário ter cuidado para que o jogador não se sinta perdido. Metroidvanias lidam com esse problema usando diversos artifícios que servem para guiar o jogador de forma sutil e orgânica, a fim de não quebrar sua imersão no jogo, nem tirar sua agência. Por conta dessas decisões de *design* únicas, os Metroidvanias são mais caracterizados por elas do que por suas temáticas e *gameplay*.

Diante do que foi definido ao longo deste artigo, vemos que as ferramentas e conveniências estabelecidas nos últimos vinte anos ainda têm espaço para serem experimentadas dentro do subgênero. Vemos, também, que o que define o subgênero é menos um mote ou uma narrativa e muito mais uma lógica de *design* que pode ser desmembrada de forma a oferecer novas ferramentas a outros tipos de jogos, coisa que de fato tem acontecido nas últimas décadas. O material é rico o bastante para expandir o debate, aprofundando não só a pesquisa dentro do próprio subgênero - dando mais atenção ao próprio *Castlevania: Symphony of the Night*, por exemplo -, como também analisando quais as influências que essa proposta de *game design* sobre outros recortes da indústria.

VI. REFERÊNCIAS

- [1] J. Parish. "Castlevania: Symphony of the Night Deep Dive, Part 3: The Metroidvania Lie." USgamer.net. [https://www.usgamer.net/articles/design-in-action-\[-\]-symphony-of-the-night-and-the-metroidvania-lie](https://www.usgamer.net/articles/design-in-action-[-]-symphony-of-the-night-and-the-metroidvania-lie) (Acesso em: 4 Dez. 2019).
- [2] J. Parish. "The Metroid Legacy." 1Up.com. <http://www.1up.com/features/the-metroid-legacy.html> (Acesso em: 4 Dez. 2019).
- [3] V. Mikhov. "On metroidvanias." <https://sqncbrk.com/metroidvanias> (Acesso em: 4 Dez. 2019).
- [4] C. Nutt. "The undying allure of the Metroidvania." Gamasutra. https://www.gamasutra.com/view/news/236410/The_undying_allure_of_the_Metroidvania.php (Acesso em: 4 Dez. 2019).
- [5] Platform Adventure. <https://platformadventure.weebly.com> (Acesso em: 4 Dez. 2019).
- [6] M. Brown. Boss Keys - The World Design of Metroid 1 and Zero Mission. (31 Jul. 2018). Acesso em: 4 Dez. 2019. [Vídeo Online]. Disponível em: <https://youtu.be/kUT60DKaEGc>
- [7] M. Brown. Boss Keys - The World Design of Super Metroid. (3 Set. 2018). Acesso em: 4 Dez. 2019. [Vídeo Online]. Disponível em: <https://youtu.be/nn2MXwplMZA>
- [8] M. Brown. Boss Keys - The World Design of Castlevania: Symphony of the Night Youtube. (15 Nov. 2018). Acesso em: 4 Dez. 2019. [Vídeo Online]. Disponível em: <https://youtu.be/a1hHOVIkrcc>
- [9] M. Brown. Boss Keys - The World Design of Hollow Knight. (31 Out. 2019). Acesso em: 4 Dez. 2019. [Vídeo Online]. Disponível em: <https://youtu.be/7ITtPPE-pXE>
- [10] GDC. There and Back Again: Koji Igarashi's Metroidvania Tale (English Translation). (8 Abr. 2016). Acesso em: 4 Dez. 2019. [Vídeo Online]. Disponível em: <https://www.youtube.com/watch?v=gLyjAWYK2Kg>
- [11] T. Wahlberg, "Blockades in the Metroidvania genre of games: A examination of backtracking." B.S. Thesis, Game Des. Dept., Uppsala

- Univ., Uppsala, Uppland, Suécia. 2015. [Online]. Disponível em: <http://www.diva-portal.org/smash/get/diva2:862195/ATTACHMENT01.pdf>
- [12] J. Bycer. "The Foundation of Metroidvania Design." Gamasutra. https://www.gamasutra.com/blogs/JoshBycer/20171019/307837/The_Foundation_of_Metroidvania_Design.php (Acesso em: 4 Dez. 2019).
- [13] J. Bycer. "Making Sense of Metroidvania Game Design." Gamasutra. https://www.gamasutra.com/blogs/JoshBycer/20140623/219621/Making_Sense_of_Metroidvania_Game_Design.php (Acesso em: 4 Dez. 2019).
- [14] C. Nutt. "Castlevania's Koji Igarashi offers advice to today's Metroidvania devs." Gamasutra. https://www.gamasutra.com/view/news/238553/Castlevanias_Koji_Igarashi_offers_advice_to_todays_Metroidvania_devs.php (Acesso em: 4 Dez. 2019).
- [15] A. Wawro. "How the Hollow Knight devs mapped out their 'Metroidvania.'" Gamasutra. https://www.gamasutra.com/view/news/303127/How_the_Hollow_Knight_devs_mapped_out_their_Metroidvania.php (Acesso em: 4 Dez. 2019).
- [16] A. Webster. "The Enduring Influence of Metroid." The Verge. <https://www.theverge.com/2017/9/14/16303016/metroid-nintendo-influence-legacy>. (Acesso em: 4 Dez. 2019).
- [17] R. Frushstick. "Modern Metroidvanias: How Dead Cells and Hollow Knight elevated a genre." Polygon. <https://www.polygon.com/videos/2018/8/13/17684628/modern-metroidvanias-how-dead-cells-and-hollow-knight-elevated-a-genre> (Acesso em: 4 Dez. 2019).
- [18] D. Milner. "The Making of Hollow Knight." Game Informer. <https://www.gameinformer.com/2018/10/16/the-making-of-hollow-knight> (Acesso em: 4 Dez. 2019)
- [19] "Super Metroid randomizer portal door guide." <https://maplequeensaku.weebly.com/news/super-metroid-randomizer-portal-door-map-guide> (Acesso em: 24 Out. 2020).
- [20] Metroid. (1986). Nintendo. [Nintendo Entertainment System].
- [21] Metroid II: Return of Samus. (1991). Nintendo. [Game Boy].
- [22] Super Metroid (1994). Nintendo. [SNES].
- [23] Castlevania: Symphony of the Night. (1997). Konami. [Playstation].
- [24] Hollow Knight. (2017). Team Cherry. [Steam, Nintendo Switch].
- [25] Iconoclasts. (2018). Bifrost Entertainment. [Steam]
- [26] Bioshock 2. (2010). 2K Games. [Playstation 3, Xbox 360, Steam].
- [27] Assassins Creed: Odyssey. (2018). Ubisoft. [Playstation 4, Xbox One, Steam]