

COLÉGIO TÉCNICO DE GUARATINGUETÁ - UNESP

ALEXANDRE FREIRE, ARISSA YOSHIDA
DAVI LYRA, KELVIN AUGUSTO
LEANDRO MATSUDA

BLIND'S LIFE

Trabalho de Conclusão de Curso
apresentada ao Colégio Técnico
Industrial de Guaratinguetá UNESP como requisito
para obtenção do título de Técnico
em Informática.
Orientador: Prof. Ricardo Saad

GUARATINGUETÁ - SÃO PAULO
2019

ALEXANDRE FREIRE, ARISSA YOSHIDA
DAVI LYRA, KELVIN AUGUSTO
LEANDRO MATSUDA

BLIND'S LIFE

BANCA EXAMINADORA:

À Lucas Radaelli, pela
orientação e inspiração do
nosso projeto.

AGRADECIMENTOS

Primeiramente à Deus por nos ter dado saúde e força para superar as dificuldades. A Instituição pelo ambiente criativo e amigável que proporcionou por todos estes anos nos ajudando a ter meios para realização do projeto.

Ao Lucas Radaelli, programador da Google Brasil, que nos serviu de inspiração e nos orientou durante todo o processo.

Ao professor Ricardo Wurthmann Saad, pela orientação, apoio e confiança.

A todos professores do curso de informática que nos proporcionaram conhecimento nestes últimos anos de estudo, e a todos que diretamente ou indiretamente fizeram parte de nossa formação para que conseguíssemos desenvolver e finalizar esse projeto.

*“Todos nós fazemos escolhas
na vida, mas no final nossas
escolhas nos fazem”*
Andrew Ryan - BioShock

RESUMO

Blind's Life é um jogo baseado na vida de uma pessoa com deficiência visual, mostrando algumas das dificuldades/desafios na vida da mesma. Os programas utilizados foram o Game Maker 1.4 Professional, Ableton Live 10 Suite, Photoshop e Krita. O objetivo do jogo é ajudar as pessoas a compreender as dificuldades e desafios na vida de um deficiente visual através das experiências virtuais proporcionadas. E também apresentar meios de adaptação para auxílio dessas pessoas, disponibilizando também alguns sites para projetos nacionais que arrecadam doações para ajudar deficientes que não tem condições e criar novos sistemas que podem ser adaptados para uma cidade ou escola. As situações mais comuns que serão exploradas dentro deste jogo serão por exemplo encontrar uma determinada sala dentro de uma escola onde por falta de sinalização adaptada para deficientes visuais termina sendo algo difícil, ou por exemplo tentar atravessar uma rua sem sistema de som onde na vida real isso se aplica à possíveis acidentes.

PALAVRAS-CHAVES: Blind's Life, Deficiente visual, Game Maker, Projetos Nacionais

ABSTRACT

Blind's Life is a game based on the life of a visually impaired person, showing some of the difficulties / challenges in their life. The programs used were Game Maker 1.4 Professional, Ableton Live 10 Suite, Photoshop and Krita. The goal of the game is to help people understand the difficulties and challenges in the life of a visually impaired through the virtual experiences provided. It also presents ways of adapting to help these people, as well as providing some websites for national projects that raise donations to help disabled people and create new systems that can be adapted to a city or school. The most common situations that will be explored in this game will be for example to find a certain room inside a school where for lack of signage adapted for the visually impaired ends being something difficult, or for example trying to cross a street without sound system where in real life this applies to possible accidents.

KEYWORDS: Blind's Life, Visual Impairment, Game Maker, National Projects

Sumário

1. INTRODUÇÃO	1
2. DESENVOLVIMENTO	2
2.1. Fundamentos Teóricos	4
2.1.1. Programas Utilizados.....	5
2.1.1.1. Adobe Photoshop	5
2.1.1.2. Game Maker: Studio.....	6
2.1.1.3. FL Studio 12.....	9
2.1.1.4. Krita.....	11
2.2. Metodologia	12
2.2.1. Criação dos Personagens	12
2.2.2. Música e Efeitos Sonoros.....	15
2.2.3. Programação	15
2.2.4. Exportação do Programa.....	29
3. CONCLUSÃO	31
4. REFERÊNCIAS BIBLIOGRÁFICAS	32

1. INTRODUÇÃO

Cerca de seis milhões de pessoas no Brasil, declaram ter algum tipo de deficiência visual, seja baixa visão, dificuldade de enxergar ou cegueira completa. A importância dada para uma grande parcela da população brasileira indica que a representatividade em território nacional é extremamente pequena, então desde pequenas conscientizações sobre o assunto a entretenimento direcionado a este público, são formas de demonstrar que essa parcela da população é expressiva e deve ter mais espaço no cenário atual de forma geral.

Dentro deste mundo, o projeto desenvolvido se demonstra uma forma de alertar sobre a realidade de muitos e o conhecimento da área por meio de um jogo desenvolvido como não apenas forma de entretenimento, mas sim para abrir os olhos de diversas pessoas para a atual situação que se encontra essa parcela da população, fato que é de enorme importância graças a necessidade de auxílio para o melhor viver dos que precisam por exemplo de um cão-guia e que somente seria possibilitado algo assim com ajuda monetária daqueles que estão dispostos a tal.

Diante dos gastos enormes e necessidades especiais direcionadas a população deficiente visual, a falta de apoio e visão sobre os intempéries tem se mostrado enorme, levando a diferentes iniciativas para que seja possibilitado a todos o mesmo tipo de tratamento e que seus problemas possam se resolver de forma que eles não fiquem debilitados monetariamente também graças ao fato de possuírem uma debilidade visual que muitas vezes é algo com que o indivíduo nasce ou desenvolve na infância e que ficará para o resto de sua vida, sempre com gastos constantes.

Partiu-se da hipótese que a forma de entretenimento utilizada no projeto fosse pouco explorada dentro da utilização para conhecimento de diferentes realidades vivenciadas pelas pessoas no mundo todo e o desenvolvimento se mostrou uma forma divertida e simples de mostrar que é possível dar destaque a necessidades reais e que deveriam ter mais foco dentro do cenário de entretenimento assim como atualmente cada vez mais vemos diferentes formas de distrações não só demonstrando ao público como pessoas com necessidades especiais vivem mas também de forma inclusiva possibilitam a esses indivíduos aproveitar da mesma forma jogos, filmes e outras mídias.

2. DESENVOLVIMENTO

Desde a formação da construção da história e objetivos do jogo, foi essencial a inspiração para de dois pilares do nosso jogo. O primeiro pilar é como chegamos em um acordo para o assunto discutido do jogo, ou seja, sobre deficiência visual e o segundo como chegamos ao estilo desse jogo, ou seja, esse formato RPG.

Como principal influenciador do nosso primeiro pilar, achamos bastante importante destacar a participação de um integrante a mais ao nosso trabalho, seu nome Lucas Radaelli, programador da Google que possui deficiência visual e que faz diversas palestras a respeito disso. Com a ajuda dele, conseguimos ver parcialmente como é a vida de um deficiente visual e aprendemos o quanto é importante os materiais adaptados para o seu cotidiano. Sendo assim, também ficamos abertos para aprender e entender sobre as suas dificuldades e junto com várias notícias e algumas pesquisas, percebemos o quanto é importante a conscientização para ajudar a incluir deficientes de todos os tipos na sociedade.

Sendo assim, nos dando a ideia de utilizar nosso conhecimentos de programação para desenvolver um jogo em que pudéssemos de alguma forma tocar seus usuários assim os conscientizando a respeito desse “problema”, justamente mostrar o mesmo que descobrimos através do contato com um deficiente e grande amigo Lucas Radaelli.

Para o segundo pilar, ou seja, para a resposta de como chegamos a ideia de criar um RPG em 2D com uma história dinâmica, temos três jogos. O primeiro jogo é bastante conhecido, o Undertale, um RPG eletrônico criado pelo desenvolvedor independente norte-americano Toby Fox, que nos serviu de base para a modelagem do nosso jogo, ou seja, criação de personagem e cenários, movimentação, sistemas de falas e história.

O segundo jogo nos serviu como uma base para a história, esse jogo não é muito conhecido, pois seu desenvolvedor é anônimo e não tem muitas referências, seu nome é ALEX. O jogo tem aproximadamente uns quinze minutos de duração, com um estilo bem simples de plataforma e foi criado com o objetivo de conscientizar seus jogadores sobre a doença Esclerose Lateral Amiotrófica, conhecida também pela abreviação ELA, mesma doença do famoso físico Stephen Hawking. Após mostrar através de uma simpática porém triste história as consequências dessa doença, ele também compartilha um site que aceita doações para ajudar nas pesquisas para a

cura dessa doença. Assim como esse jogo, nosso trabalho também compartilha links de apoio a cegos.

Uma parte importante de se destacar também é o fato de ser escolhido a criação de um jogo e não a criação de um possível site ou outra plataforma. Percebemos que através de um jogo podemos expressar uma ideia principalmente aos jovens, ou seja, a nova geração e também facilmente compartilhar de maneira dinâmica e sem ser cansativa, sendo também bastante efetivo e sendo compartilhado pelos próprios jogadores como no caso de gameplays e atingindo milhares de pessoas.

Jogabilidade do Jogo ALEX de um usuário do YouTube.



Fonte: Youtube/LubaJoga

(“ Nós esperamos que esse jogo tenha dado pelo menos um leve insight de como é a doença ALS” “Vamos encontrar a cura assim que possível”).

O terceiro jogo, Beyond Eyes, nos deu base para simular, de uma forma que dê para as pessoas jogarem, a vida de uma pessoa com deficiência visual. Com um esquema de que o personagem enxergaria apenas algumas coisas que estaria perto dele, para tentar simular ao máximo como seria a dificuldade dessa pessoa no seu dia a dia.

Juntando essas fontes de inspiração, conseguimos desenvolver e compartilhar assim como o jogo ALEX, uma causa, que ainda tem que ser bastante divulgada para uma mudança melhor na sociedade.

2.1. Fundamentos Teóricos

Além, dos objetivos do nosso projeto, é importante destacar a importância de nossa escolha de porta voz ser o jogo e não por exemplo outro meio de mídia, como sites. Atualmente, podendo ser comprovada por várias pesquisas como a pesquisa do Npd Group, o mercado de jogos é um dos que mais cresce no mundo inteiro, consequência da quantidade de pessoas que procura essa fonte de diversão. Difícilmente nos dias de hoje, as pessoas se interessam por ler jornais ou revistas. Apesar de termos também jornais virtuais, as pessoas em sua maioria preferem mídias que tenham mais interação visual, como por exemplo vídeos no youtube.

Assim sendo, mesmo que a maioria dos jogos não tenham como objetivo informar seus jogadores de um determinado assunto importante a sociedade, ele apresenta como uma boa forma de apresentar projetos de relevância a sociedade de uma maneira significativo tão eficaz quanto seria o jornal nas décadas passadas.

Mercado de Jogos entre 2012 e 2021



Fonte: <https://venturebeat.com/2018/04/30/newzoo-global-games-expected-to-hit-180-1-billion-in-revenues-2021/>

2.1.1. Programas Utilizados

2.1.1.1. Adobe Photoshop

Adobe Photoshop é um software caracterizado como editor de imagens do tipo "raster"(possuindo ainda algumas capacidades de edição típicas dos editores vectoriais) desenvolvido pela Adobe Systems. É considerado o líder no mercado dos editores de imagem profissionais, assim como o programa de facto para edição profissional de imagens digitais e trabalhos de pré-impressão. Para o desenvolvimento do jogo utilizamos esse programa para a criação de alguns cenários, edições de alguns sprites (Em computação gráfica, um **sprite** é um objeto gráfico bi ou tridimensional que se move numa tela sem deixar traços de sua passagem) e testes de animação. Algumas imagens utilizadas no nosso trabalho são de livre acesso na internet, ou seja, sem reivindicações de direitos autorais, mas para utilizarmos de maneira mais "legal", editamos algumas para serem de uso exclusivo do nosso jogo. Portanto o Adobe Photoshop, foi essencial para o desenvolvimento de nosso trabalho.

Logo e Interface principal do Adobe Photoshop.



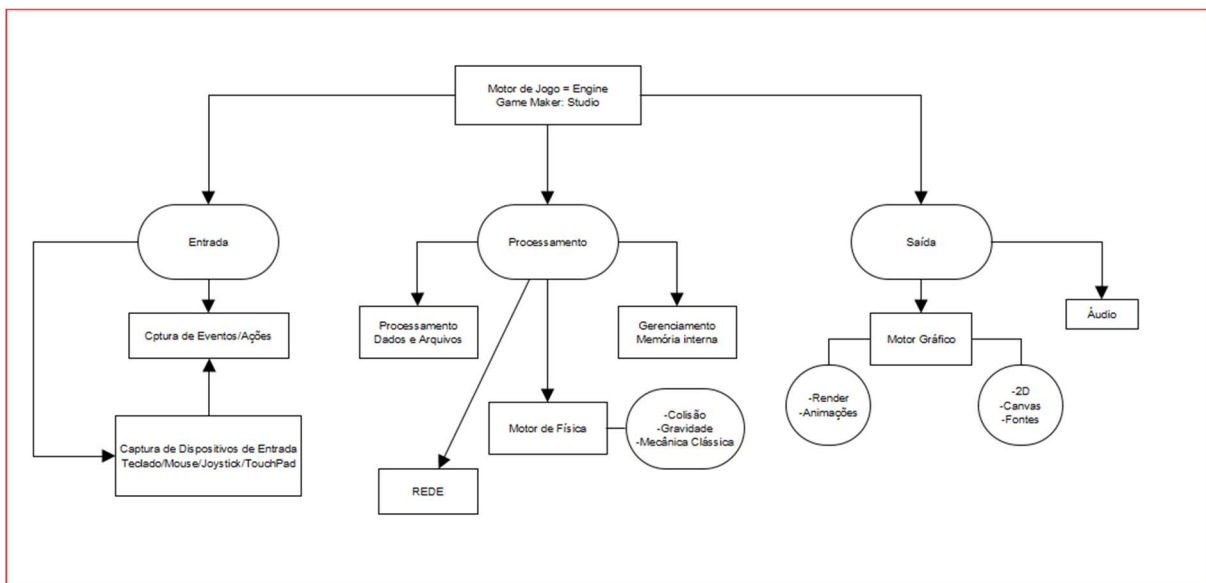
Fonte: www.adobe.com/photoshop

2.1.1.2. Game Maker: Studio

Motores para desenvolvimento de jogos

Atualmente no mercado, existe uma grande variedade de motores para desenvolvimentos de jogos, ou em outras palavras as *game engines*. Os motores são softwares ou um conjunto de bibliotecas capazes de juntar e construir todos elementos de um jogo em tempo real, assim, cada engine tem como função facilitar várias etapas do processo de desenvolvimento do jogo. Com o surgimento das engines, a criação de jogos se tornou um processo acessível para que até mesmo uma única pessoa consiga desenvolver o que na década de 80 necessitava de uma equipe inteira. Exemplos de engine: Unity, Game Maker, RPG Maker, Unreal, etc.

Estrutura de uma *game engine*



O jogo desenvolvido neste trabalho, teve toda sua programação desenvolvida na engine do Game Maker: Studio, na versão 1.4, que apesar de ser uma versão mais antiga, possui todas as ferramentas necessárias. A escolha dessa plataforma, foi também devida a grande facilidade de aprendizagem rápida com o ambiente de programação de jogos, além de suprir nossas necessidades com as animações e programação de um jogo 2D.

Logo do Game Maker: Studio.



Fonte: www.yoyogames.com

O game maker é uma ferramenta que tem a finalidade de construir jogos 2D. Essa ferramenta foi criada por Mark Overmars, que lançou sua primeira versão em 15 de novembro de 1999. Inicialmente o programa foi criado com a intenção de realizar animação em 2D utilizando a linguagem Delphi com o nome Animo, mas ganhou grande quantidade de desenvolvedores de jogos na plataforma e acabou sendo substituído pelo nome de Game Maker. Com as aprimorações e atualização do programa, em 2012 foi lançado o Game Maker: Studio, com suas configurações e interfaces atualizadas facilitando no desenvolvimento de jogos. Uma das vantagens do Game Maker: Studio, são as duas ferramentas disponíveis para programação, a Drag and Drop e pela linguagem GML.

No Drag and Drop (“Arrastar e soltar”), uma ferramenta visual de script, não é necessário o conhecimento de códigos para realizar os comandos aos objetos e ações do jogo, assim servindo de ponto de partida para novos desenvolvedores, onde podem interagir com habilidades comuns e simplificadas de programação e lógica, como a criação de funções, estrutura de dados e instanciamento de objetos.

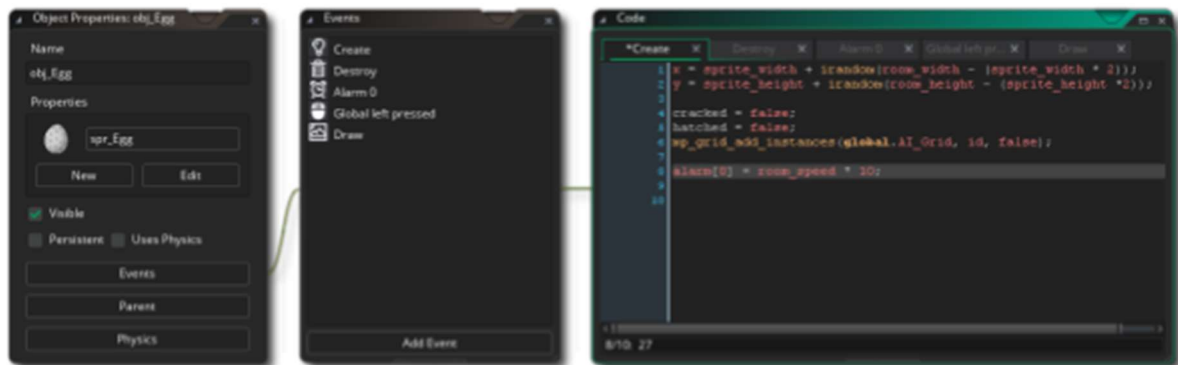
Interface de comando do modo Drag and Drop.



Fonte: Game Maker:Studio

Com o GML (GameMaker Language), uma linguagem de alto nível, baseada em linguagem C, os desenvolvedores podem criar seus jogos através de códigos, sendo necessário um conhecimento maior de programação. Com o uso do GML, o jogo não fica limitado a funções pré-criadas pelo programa, mas também abre espaço para a criação de novas funções e Scripts, sendo mais flexível durante o processo da criação de diferentes tipos de jogos.

Interface do Game Maker, para a criação em código de um objeto.



Fonte: www.yoyogames.com

Após todo o desenvolvimento da programação do jogo, o Game Maker, disponibiliza uma vasta quantidade de plataformas de suporte como: Microsoft Windows, macOS, Ubuntu, HTML5, Android, PlayStation4, Xbox One e outros. Para o compartilhamento e divulgação do nosso projeto, foi disponibilizado nossa programação em duas plataformas: Microsoft Windows e o HTML5.

2.1.1.3. FL Studio 12

O FL Studio é um ambiente de produção musical de software completo ou Digital Audio Workstation (DAW). Representando mais de 20 anos de desenvolvimento inovador, ele tem tudo o que você precisa em um único pacote para compor, organizar, gravar, editar, mixar e masterizar músicas de qualidade profissional. O FL Studio é um dos mais populares DAWs (Digital Audio Workstation) para a produção de música de dance music. Começando como FruityLoops , software de sequenciamento no final dos anos 90, FL Studio amadureceu de forma exponencial ao longo dos anos, e é o DAW de escolha para muitos produtores de EDM hoje. Ele costumava ter uma reputação de ser um software “de principiante”, esta imagem está cada vez mais se tornando apenas uma crítica sem fundamento, porque grandes produtores adotaram o FL Studio como plataforma de sua escolha para criação de seus hits.

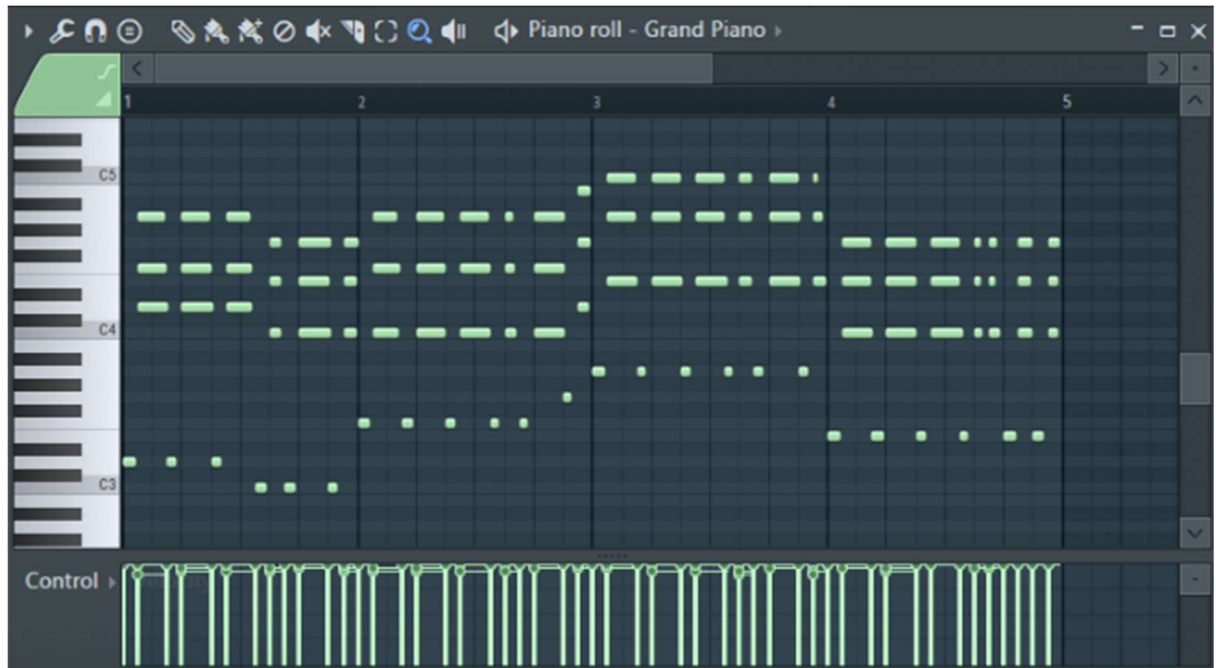
Logo da FL Studio.



Fonte: <https://www.image-line.com/press/presskitblog.php>

O *Piano Roll* da FL Studio tem a merecida reputação de melhor piano do ramo. O piano é usado para enviar dados de anotações e automação para instrumentos de plugin (sequenciamento). Inclui uma ampla gama de ferramentas para ajudar na edição e manipulação complexas de partituras. A maior parte da música é feita no *Piano Roll* pois nele contém uma variada gama de notas musicais das quais o usuário usará para compor suas músicas. Seja o instrumento utilizado, um piano ou uma guitarra, as notas serão sempre as mesmas o que torna o uso de fácil manipulação.

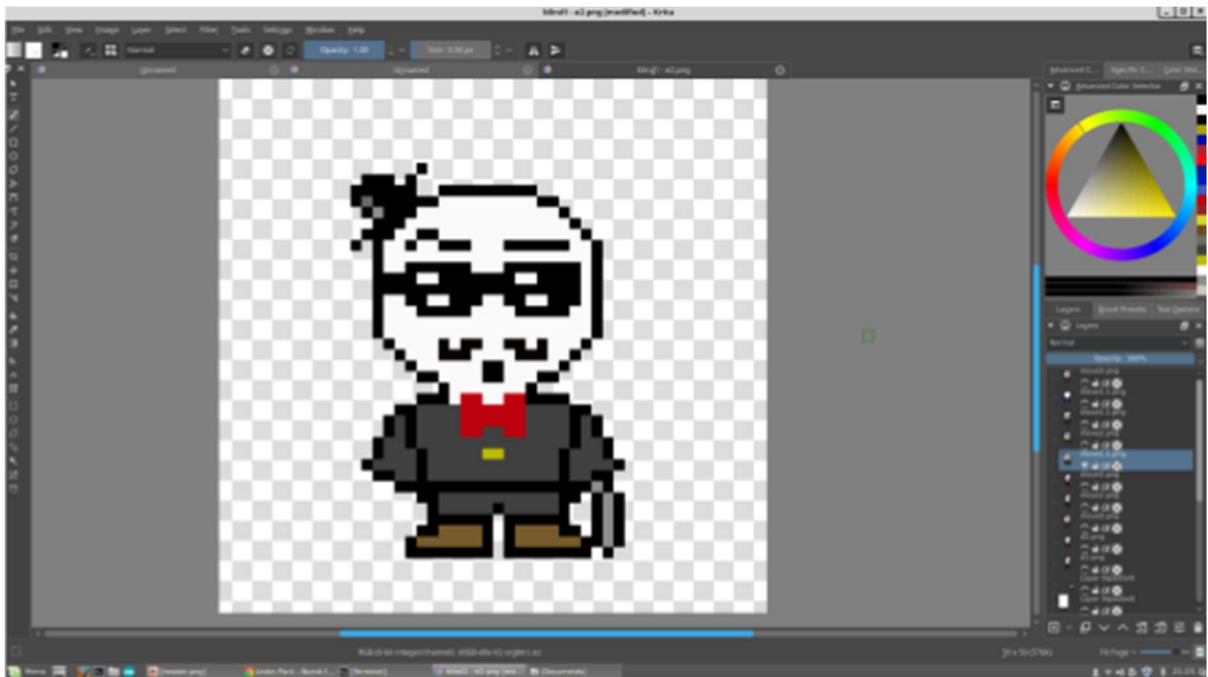
Piano roll FL Studio.



Fonte: <https://www.image-line.com/flstudio/>

2.1.1.4. Krita

O Krita é uma poderosa ferramenta para criação de ilustrações, concept art, histórias em quadrinhos, pinturas digitais, animações, possibilitando também ser usado como um programa de retoques e manipulação de fotografia, conversor de formatos, suportando vários modelos de cores e pintura HDR. Possui uma interface de baixa distração, suporte acelerado de alta qualidade OpenGL, suporte a gerenciamento de cores, mecanismo avançado de pincéis, camadas e máscaras não-destrutivas, gerenciamento de camadas em grupo, suporte a arte vetorial e perfis de customização alternáveis. É software livre e de código aberto e está disponível para Linux, Windows e macOS. Boa parte dos Sprites foi feita no próprio Game maker, por causa da sua ferramenta de edição de sprite e background. Entretanto, foi necessário o uso do Krita em algumas animações, principalmente devido a disponibilidade para o sistema operacional Linux.



Criação de um sprite do jogo, dentro do aplicativo Krita. Fonte:Autoria Própria

2.2. Metodologia

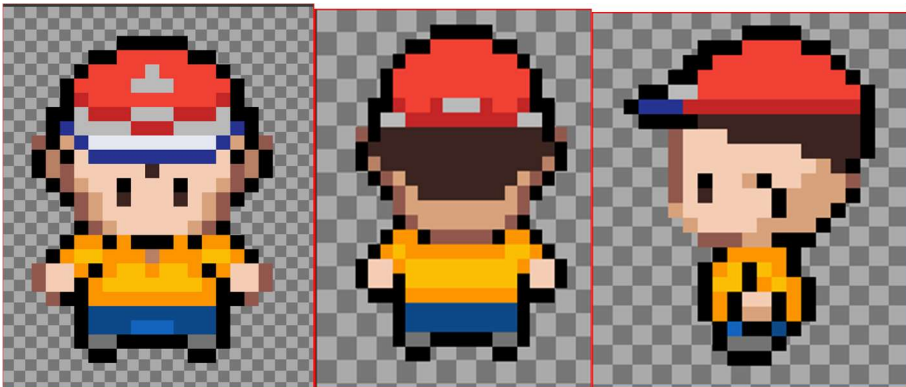
2.2.1. Criação dos Personagens

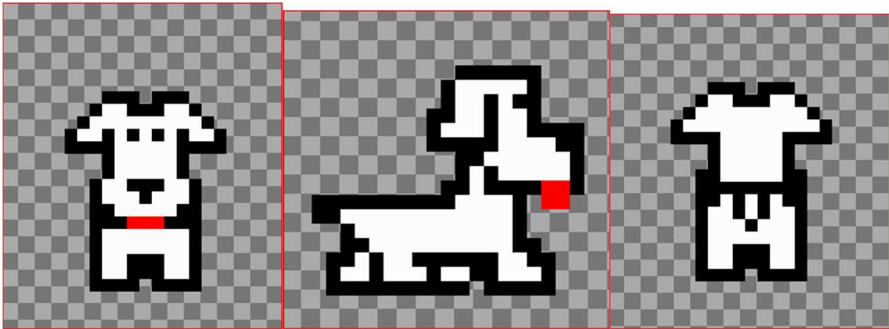
Os Sprites e o Cenário de um jogo é uma parte essencial para o desenvolvimento do mesmo, sem eles não há interface entre o jogo e o jogador. É por meio desses que a transmissão de cada emoção, intenção e cerne dentro do jogo. A música também é conteúdo dessa interface porém é tema de outro tópico.

O desenvolvimento dos sprites e dos cenários ocorreu através de inspirações em alguns já existentes e de alguns jogos antigos e que nos ajudaram a criar a silhueta desejada para os objetos, já que éramos iniciantes, havia dificuldade em expressar detalhes visto que o espaço disponível para trabalho da imagem era pequeno.

O processo de criação foi se desenrolando conforme a necessidade e o surgimento de ideias novas para o projeto. É claro que ocorreram alguns problemas (como a escala dos sprites, coloração, representação de objetos e criaturas da realidade), porém foram todos resolvidos rapidamente com criatividade, programação, e alguns stalks de fotos no Facebook.

Pudemos concluir com a experiência no desenvolvimento de arte que é uma das partes mais trabalhosas e mais divertidas da criação ficando atrás apenas da programação dos objetos.





2.2.2. Música e Efeitos Sonoros

O gênero musical utilizado como música de fundo do jogo *Blind's Life* é baseado no gênero musical Lo-Fi Hip Hop, que diferentemente do gênero musical Hip Hop, o Lo-Fi Hip Hop utiliza-se de acordes, notas e efeitos mais calmos e relaxantes. O Lo-Fi Hip Hop é um novo estilo musical derivado do antigo Lo-Fi (low fidelity), que eram músicas gravadas com técnicas de gravação de baixo custo devido às circunstâncias financeiras dos artistas da época. E pela música ser produzida em lugares amadores, sem microfones profissionais e todo o equipamento, várias pessoas começaram a produzir suas próprias músicas em casa. Com plugins e efeitos sonoros, podiam reproduzir o som do ruído do disco de vinil e outros efeitos nostálgicos, assim conquistando os ouvidos de muitas pessoas e se tornando o famoso Lo-Fi Hip Hop.

Atualmente no YouTube já existem vários canais que transmitem 24 horas por dia, 7 dias por semana de variadas músicas do gênero Lo-Fi Hip Hop, e essas músicas ficaram famosas por serem músicas para estudo e relaxamento. Como o

2.2.3. Programação

Para a programação do jogo foi utilizado no game maker um pouco da ferramenta Drag and Drop e um pouco da programação escrita pela linguagem GML. Para movimentações e mecânicas mais simples como movimento de carros e NPCs, utilizamos o Drag and Drop que não usa códigos de programação e assim facilita e agiliza a construção da estrutura lógica desses objetos no jogo. Entretanto para sistemas mais complexos como a interação de um objeto com outro, utilizamos a linguagem GML, vale destacar, que boa parte dos objetos tinha interação dessas duas ferramentas ao mesmo tempo. Grande parte da linguagem GML é igual a linguagem C, apenas algumas obrigações de sintaxe como ponto e vírgula em alguns casos não são requisitados ou são modificados. O diferencial do GML para o desenvolvimento de jogos, é a ampla quantidade de funções pré criadas pelo programa para facilitar os comandos do jogo. Além dessas funções, também é possível, através do sistema de Script, a criação de novas funções a gosto do desenvolvedor.

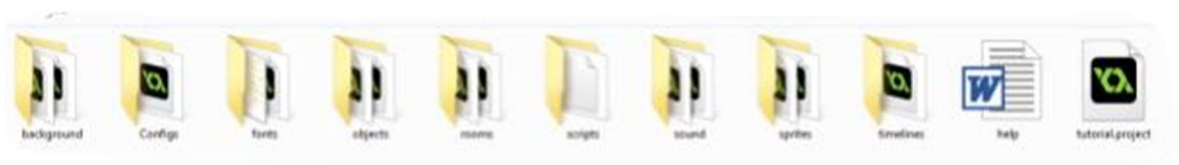
Interface principal do Game Maker: Studio 1.4.



Fonte: Autoria Própria

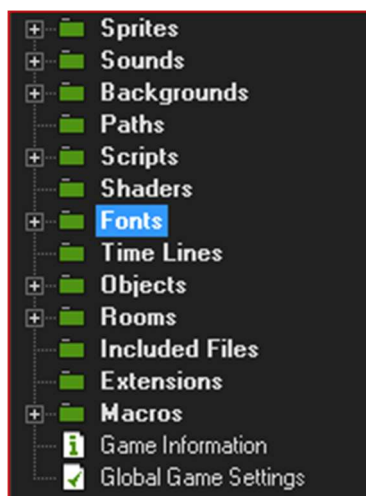
Todo o material utilizado dentro do programa Game Maker: Studio, fica armazenado em um conjunto de pastas gerados após o salvamento do projeto. Na primeira imagem abaixo aparece as pasta como a de sprite, background, etc e ao final o arquivo .gmx, que será responsável por guardar toda a programação e juntar todos os recursos para edição e desenvolvimento do jogo. Na segunda imagem, aparece que a interface que é utilizada dentro do programa.

Pastas geradas pelo Game Maker: Studio



. Fonte: Autoria Própria

Interface das pastas vistas dentro do Game Maker: Studio.



Fonte: Autoria Própria

Explicando as Pastas utilizadas:

Sprites: Nas pastas de Sprite, são encontrados os objetos gráficos, que fazem as animações, por exemplo dos personagens. São apresentadas várias imagens que com o passar do tempo criam animação de movimento.

Sounds: Todos arquivos sonoros do jogo, são encontrados nessa pasta. Como as músicas de fundo e os efeitos sonoros.

Backgrounds: Os Backgrounds são as imagens de fundo do jogo, são o cenário.

Scripts: A pasta de scripts é reservada para arquivos de texto usados para criação de algumas funções e códigos de programação, complementando as ações de um objeto ou sala.

Fonts: A pasta de fonte é onde são armazenadas as diferentes configurações de fontes para cada tipo de texto desejado, sendo chamados por algum objeto. Cada arquivo de fonte define um formato de letra.

Objects: Nas pastas de objetos, estão os arquivos que comandam por exemplo algum sprite a realizar uma ação. Os objetos são todos os elementos programáveis que podem ser inseridos nas rooms (salas).

Rooms: As Rooms, são os ambientes onde serão apresentados os objetos. Nelas são implementados os backgrounds.

A programação ela é feita nos arquivos de objetos, onde podemos utilizar o modo Drag and Drop ou usar a linguagem GML nos Códigos Executáveis. As propriedades dos Objetos são divididas em três partes.

Configuração:

Interface do objeto Player.



Fonte: Autorial Própria

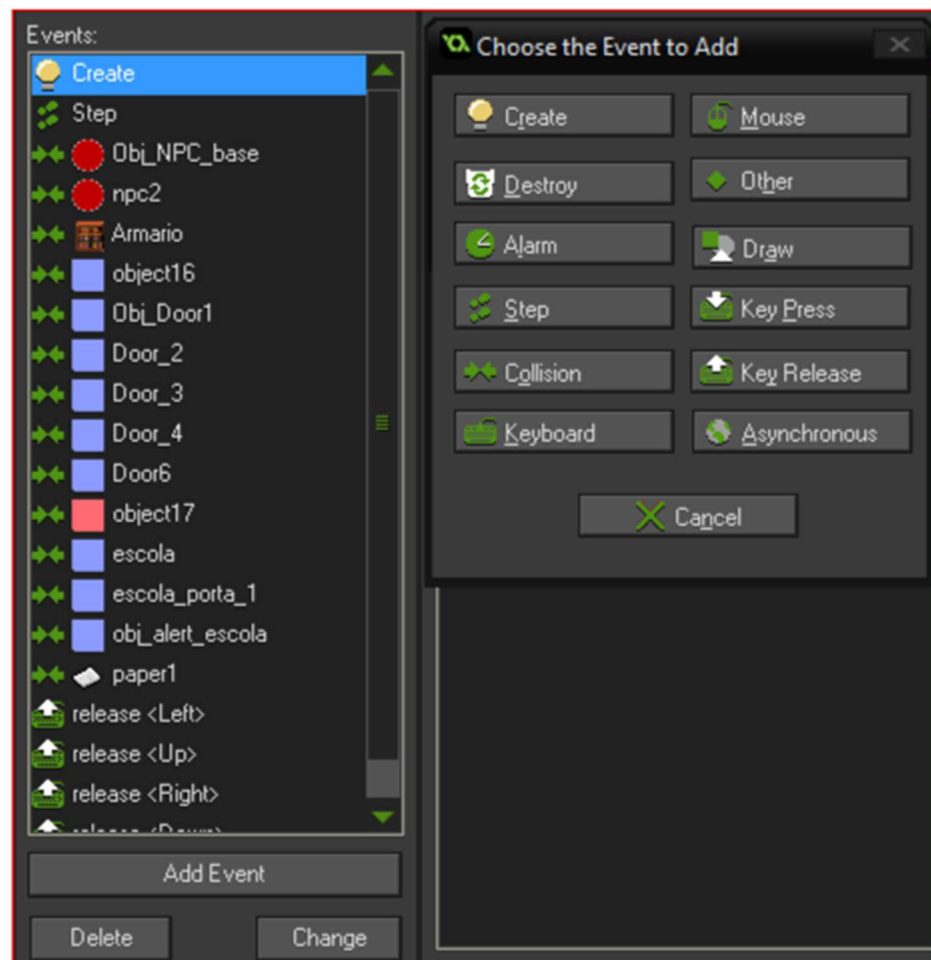
Nessa parte, definimos o nome do nosso objeto (Player), qual o tipo de Sprite que será usado inicialmente (sprite201), e algumas configurações de como ela aparecerá no jogo. No caso, o objeto Player ele será visível e utilizará do sistema de física do Game Maker. As opções que não estão marcadas são de persistência, quando selecionada, ela faz com que esse objeto apareça em todas as Rooms, o que não o caso, e o Sólido, que normalmente é marcado para objetos estáticos para se existir colisão com outros objetos.

Quando habilitado o sistema de física para o corpo do objeto, utilizamos algumas configurações editáveis para representar melhor os tipos de colisão requisitada pelo objeto. Exemplo: Se alterarmos a densidade de um objeto e ele colidir com outro tendo densidade maior que 0.5, ele terá uma reação e se movimentará de acordo com a colisão.

Outras opções possíveis nessa aba, são as configurações de parent que determinam um objeto como parte de uma hierarquia, agrupando vários objetos “filhos” na aba Children, a opção Mask como o próprio nome diz, é onde se escolhe um sprite que irá servir como máscara, podendo se usar essa máscara por exemplo para modificar o campo de colisão do sprite principal, e por último na parte de configuração de objeto temos a parte de de Depth, onde se pode escolher a profundidade do objeto dentro da sala, basicamente representa o eixo Z.

Eventos:

Interface de eventos do objeto Player.



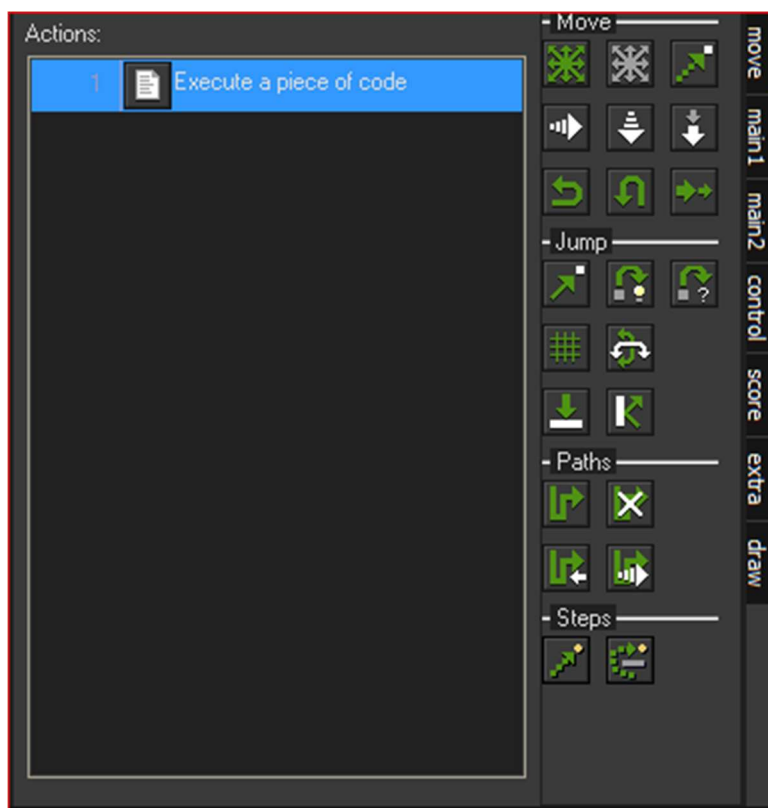
Fonte: Autoria Própria

Durante os jogos, cada objeto do jogo terá um determinado grupo de eventos que estará em Loop running (processo de repetição) desde o início da Room (Sala de execução) até o final dela. Sendo que cada evento possui suas ações. Na imagem é possível ver as configurações de

eventos do objeto Player, um dos eventos por exemplo é a colisão, sempre que por exemplo o objeto Player colidir com o objeto Obj_Door1, alguma ação acontecerá. E essa condição de evento acontecerá em processo contínuo até finalização da sala em que o objeto se encontra. Na aba lateral aparecem todos eventos que podem ser escolhidos de acordo com o desejo de seu desenvolvedor.

Ações:

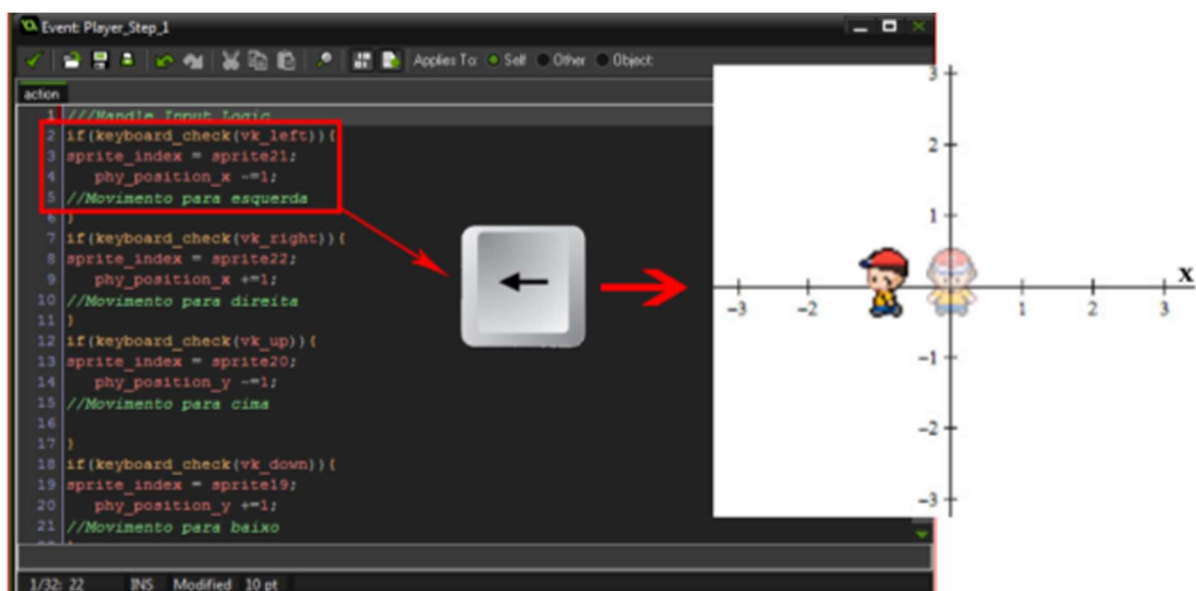
Interface de ações de um evento do objeto Player.



Fonte: Autoria Própria

Na aba de ações, podemos configurar o que o evento do objeto pode fazer, nessa parte temos duas opções de programação, o modo drag and drop, como mostrado na parte direita da imagem acima, onde o que seria comandos em códigos, são apenas figuras visuais de alguma ação. Mas também podemos programar por meio de códigos em linguagem GML, como é visto destacado em azul. Para exemplificar melhor os dois tipos de programação, temos um pequeno esquema abaixo com dois objetos que são executados durante o jogo.

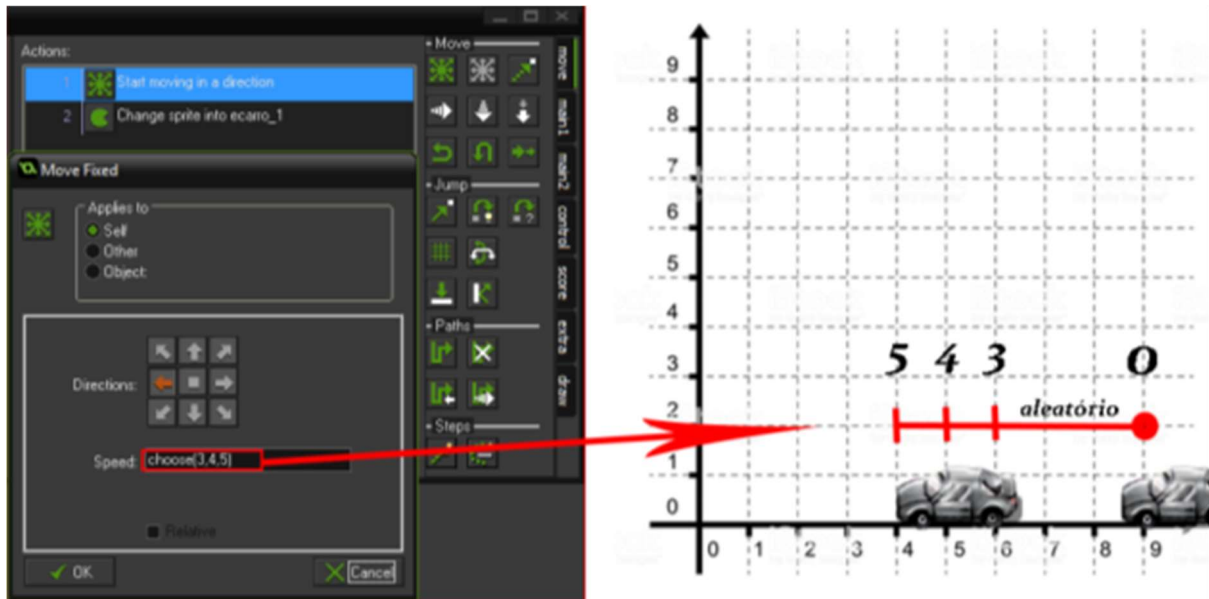
Exemplo de programação em GML



.Fonte: Autoria Própria

Nesta primeira figura, podemos ver uma página de ação de um evento do tipo Step (passo) do personagem principal do jogo. De acordo com os códigos, caso a tecla de seta para esquerda do computador esteja ativada, o objeto Player irá se movimentar para a posição $x = -1$, ou seja, se movimentará para a esquerda e ao mesmo tempo que sua posição mudar dentro da room, o objeto trocará de sprite, sendo agora o seu sprite principal o sprite21, como mostrado na imagem.

Exemplo de programação em Drag and Drop.



Fonte: Autoria Própria

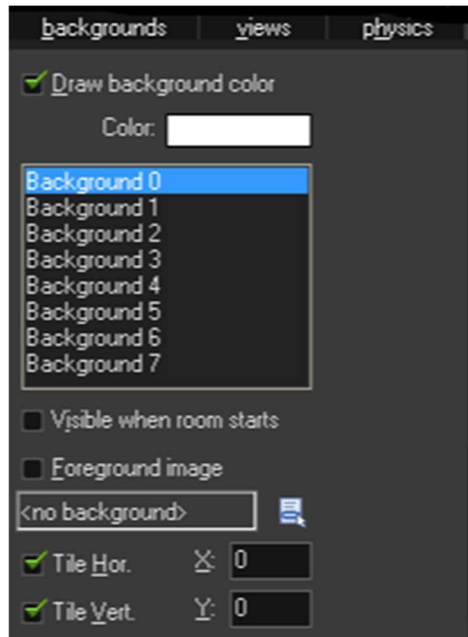
Já nesta segunda figura, podemos ver a aplicação do modo Drag and Drop, onde puxamos da aba lateral direita uma determinada ação e colocamos dentro de um determinado evento. De acordo com a configuração dessas ações, o sprite principal da imagem irá sempre mudar dentro desse evento para o sprite carro_1. E além disso, estará sempre se movimento para esquerda variando sua velocidade aleatoriamente entre os números 3,4 e 5, ou seja, eles estará se movimento dentro do eixo x, entre essas distâncias. Essa ação de aleatoriedade, foi muito importante durante o jogo, para que houvesse uma dinâmica, semelhante a maioria dos jogos estilo Cross Road (atravessar a rua).

Sistema de Room

Além dos objetos programáveis, existem as Rooms, onde todos os objetos serão colocados, em outras palavras seriam as salas de interface do jogo, normalmente sendo interligadas entre si. Nas configurações de uma Room, temos seis ferramentas, sendo elas:

Background: Onde se pode definir uma imagem para utilizar como fundo de cena, ou também escolher uma cor da sala. Dependendo do tipo de jogo, podem ser utilizadas diferentes backgrounds, sendo assim possível criar mais de um.

Ferramenta Background.

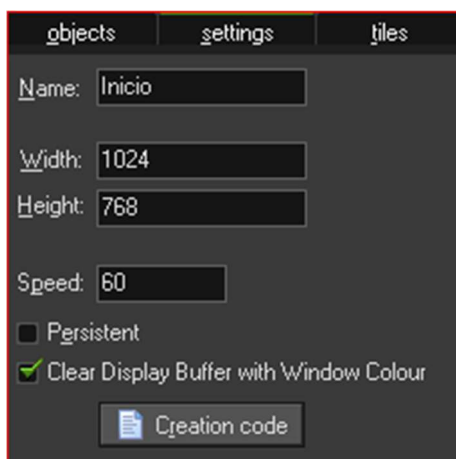


Fonte: Autoria Própria

Objects: Ferramenta, onde o desenvolvedor pode definir os objetos que são utilizados na sala, podendo também pré-definir sua posição inicial em relação aos eixos x e y.

Settings: Nessa opção é possível definir o nome da Room, o seu tamanho, a sua velocidade, ou seja, a quantidade de frames por segundo, e também criar uma programação em código para a Room.

Settings da Room



Fonte: Autoria Própria

Na room **Início**, como mostrada na imagem acima, foi criada uma pequena programação, para que sempre que se inicie essa sala, toque uma música sem repetição e de prioridade 0.

Views: Na Ferramenta view, podemos escolher a área que será visível dentro da Room. Por exemplo, caso a sala tenha um tamanho de 960px de largura e 640px de altura e queremos que o usuário tenha como campo de visão apenas metade, configuramos na aba dessa ferramenta, largura como 480px e altura como 320px. Outra opção existente, é o uso do seguidor de objeto. Com essa opção, podemos definir um objeto com o seu campo de visão sempre sendo movido de acordo com os comandos dados pelo objeto. No jogo essa configuração pode ser vista, quando o objeto principal “Player” se move pela rua.

Tiles: Essa ferramenta, funciona como um adicional da ferramenta background, nessa ferramenta podemos criar o fundo da Room através de recortes de uma imagem de background. Sendo muito utilizados para fazer pisos e outras estruturas que se repetem com frequência na imagem. Também utilizada em processos, onde o desenvolvedor dos sprites de background coloca todas imagens juntas ao mesmo tempo. Como mostrado na imagem a seguir.

Background para recorte do jogo Undertale, sendo algumas partes utilizadas no jogo Blind's Life.



Fonte: Undertale by Toby Fox

Physics: Por fim, essa ferramenta é utilizada na Room para definir se terá o sistema de física agindo nos objetos. Por exemplo se definirmos um objeto com esse sistema de física, não podemos deixar de marcar essa opção para a Room, caso o contrário ocorrerá um erro e o jogo

irá abortar. E também caso seja ativado o sistema de física da sala e o objeto não utiliza física, esse objeto irá se manter estático durante todo momento na sala.

Funções e os principais scripts do jogo

Durante todo o desenvolvimento do jogo, muitas funções e algumas ações do drag and drop foram utilizadas, por isso, apresentaremos dentro desse trabalho escrito apenas os mais utilizados e os mais importantes para que conseguíssemos executar com sucesso a criação do jogo Blind's Life.

- `Keyboard_check()`;

Retorna se a tecla fornecida no teclado está atualmente pressionada. Diferente das funções `keyboard_check_pressed` ou `keyboard_check_released`, o `keyboard_check()`, além de executar o programa caso alguma tecla esteja pressionada, ele mantém executando a ação até que a tecla seja solta. Essa função foi utilizada por exemplo, para o movimento do “Player” principal do jogo.

- `Keyboard_check_pressed()`;

Retorna se a tecla fornecida no teclado acabou de ser pressionada. Essa função foi utilizada dentro da tela inicial de seleção, onde caso seja pressionado a tecla “z”, a opção será selecionada.

- `Keyboard_check_released()`;

Essa função ela verifica se a tecla fornecida já foi acionada. Durante o jogo essa função é vista quando o personagem principal, pare de andar e troca o sprite animado para um sprite estático.

- Keyboard Input (Entrada de teclas)

Dentro das funções de keyboard (teclas), para declarar qual tecla deseja que seja verificada usa-se o termo `vk_(nome da tecla)`, caso a tecla seja alguma letra, aí a sintaxe é diferenciada e se usa o termo `ord('letra da tecla')`.

- `Room_goto(numb);`

Essa função é usada para determinar qual será a Room principal seguinte. Toda vez durante o jogo, que acontece uma troca de cenário, é com ajuda dessa função que a ação é realizada. Por exemplo, caso o personagem esteja na tela de menu e decida iniciar o jogo, a função estaria escrita `room_goto(“nome da sala inicial do jogo”)`.

- `Room_restart();`

Caso essa função seja usada, a sala atual e todos objetos que possuem algum movimento inicial especificado, irão ser reiniciados e o processo se repetirá. No jogo, essa função foi utilizada na travessia de rua, toda vez que o personagem perdia a vida, foi reiniciado a sala.

- `Audio_play_sound(index,priority,loop);`

Para selecionar algum efeito sonoro ou alguma música de fundo. Essa função possui três argumentos: nome da música, número de prioridade e repetição. No primeiro argumento pode ser utilizado uma variável do tipo string ou char, onde será apresentado o nome do som desejado. No segundo argumento é aceito qualquer número do tipo int, onde dependendo do grau de prioridade o som pode sair mais alto ou outro som será interrompido, esse valor pode variar de 0 a 100. Por fim, o último argumento, recebe valores do tipo booleano, ou seja, “true” ou “false”, ele serve como uma ferramenta muito importante de som de um jogo, pois ele define se o som irá ou não repetir. Normalmente, músicas de fundo de jogos, ficam em um ciclo infinito de repetição, pois o tempo que cada jogador levará para terminar determinada fase é indeterminado e varia.

- `Audio_pause_all();`

Toda que vez que essa função aparece, o áudio de fundo da sala ou todos efeitos sonoros que estavam acontecendo durante o momento é pausado e só retorna caso outra função de resumo seja ativada. Essa função foi importante para garantir em algumas partes do jogo não houvesse duas músicas tocando ao mesmo tempo.

Movimentos

No Game Maker, os comandos de movimento são feitos através de coordenadas do plano cartesiano, ou seja, a posição do objeto irá variar entre x e y, podendo ter números positivos ou negativos. Por exemplo, caso eu queira que um objeto vá para a direita e desça cinco pixels, a sintaxe será: `x =+5; y=-5;`. Obs.: Importante destacar que caso o objeto esteja com o sistema de física de corpo ativado, o mesmo comando do exemplo seria: `phy_position_x = +5; phy_position_y =-5;`.

- `Instance_create(x,y,obj);`

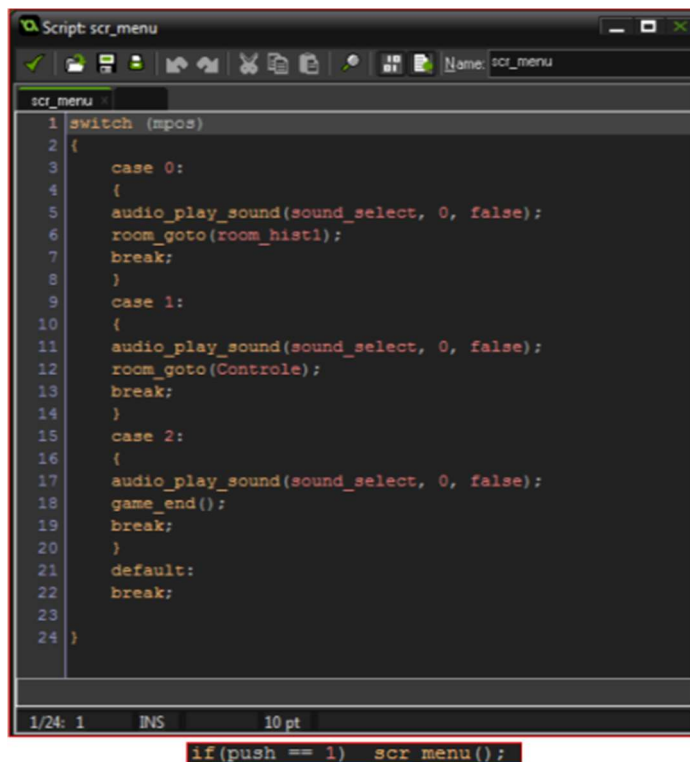
Com esta função, você pode criar uma nova instância do objeto especificado em qualquer ponto da sala. Para essa função, são requeridos argumentos para apresentar qual o objeto e a sua localização pelos eixos x e y. Toda interação com objetos ou NPCs em que haja fala, essa função foi utilizada.

- `Point_distance(x1, y1, x2, y2);`

Para realizar a interação dos outros objetos da Room com o “Player”, foi necessário essa função para saber a distância do “Player” em relação a um determinado objeto e depois fazer uma comparação de posição para ver se estava próximo ou não do objeto para assim, poderem interagir.

- Script da tela inicial:

Script para a função do menu principal.



```
1 switch (mpos)
2 {
3     case 0:
4     {
5         audio_play_sound(sound_select, 0, false);
6         room_goto(room_hist1);
7         break;
8     }
9     case 1:
10    {
11        audio_play_sound(sound_select, 0, false);
12        room_goto(Controle);
13        break;
14    }
15    case 2:
16    {
17        audio_play_sound(sound_select, 0, false);
18        game_end();
19        break;
20    }
21    default:
22        break;
23 }
24 }
```

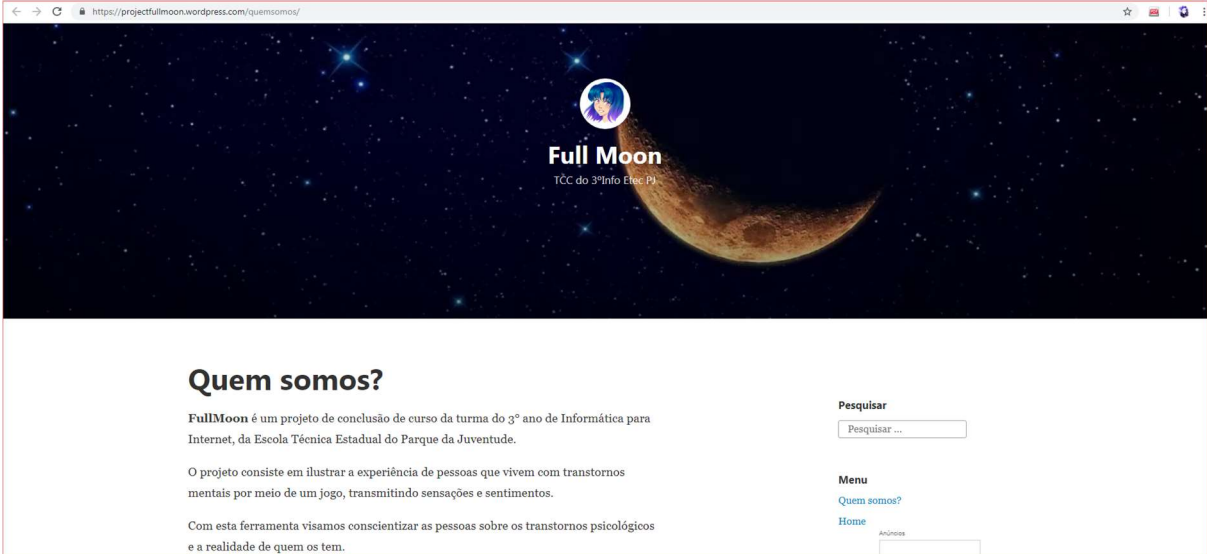
1/24: 1 INS 10 pt

if(push == 1) scr_menu();

Fonte: Autoria Própria

No Game Maker, quando abrimos a pasta de Script, podemos criar funções que posteriormente podem ser utilizadas em alguma ação de um objeto. No jogo Blind's life, foi necessário a criação de uma função, para facilitar a tela de menu. Com essa função criada, `scr_menu()`, é possível direcionar a sala com qualquer uma das três chaves sem precisar estar necessariamente no código do objeto em específico. Essa função no caso, só será ativada caso a condição seja “`push==1`”, caso não seja verdadeiro essa condição, essa função não ocupará por exemplo espaço na memória do jogo.

Para apresentar de maneira resumida e objetiva o sistema de salas do jogo que de maneira geral também resume todo o jogo, criamos um diagrama. Como pode ser visto no diagrama abaixo, as decisões dos usuários em determinadas salas podem alterar o desenvolvimento da história do jogo, formando uma interação diferente e única para cada jogador.



The screenshot shows a web browser window with the address bar displaying `https://projectfullmoon.wordpress.com/quemsomos/`. The website header features a dark space background with a crescent moon and stars. A circular profile picture of a person is positioned above the text **Full Moon**, with the subtitle **TCC do 3º Info Etec PJ** below it.

Quem somos?

FullMoon é um projeto de conclusão de curso da turma do 3º ano de Informática para Internet, da Escola Técnica Estadual do Parque da Juventude.

O projeto consiste em ilustrar a experiência de pessoas que vivem com transtornos mentais por meio de um jogo, transmitindo sensações e sentimentos.

Com esta ferramenta visamos conscientizar as pessoas sobre os transtornos psicológicos e a realidade de quem os tem.

Pesquisar
Pesquisar ...

Menu
[Quem somos?](#)
[Home](#)
[Anúncios](#)

3. CONCLUSÃO

O desenvolvimento do software como forma de jogo possibilitou com que a realidade de milhões de pessoas com deficiência visual fosse demonstrada e as campanhas divulgadas são todas para o auxílio e que de forma benéfica aos necessitados aceita diversas doações que são direcionadas a gastos relacionados aos indivíduos que necessitam deste tipo de auxílio. A pesquisa sobre o tema e a confecção do projeto em si forneceu dados e aumento o grau de conhecimento em diversas áreas, além de criar novas ideias sobre a área.

O jogo de forma geral gera uma nova ideia de apoio a instituições que ajudam pessoas que necessitam, após o término da história dentro do mesmo, é solicitado a atenção do usuário para situações reais onde ele poderia demonstrar auxílio a causa, desta forma gerando fundos que serão direcionados as pessoas certas.

Os resultados no feedback geral demonstram um entendimento da mensagem que foi o objetivo de ser passada, e que uma melhoria em aspectos técnicos e visuais seria apreciada, considerando assim uma boa ideia a ser utilizada para posteriores projetos.

4. REFERÊNCIAS BIBLIOGRÁFICAS

Brasil tem 6 milhões de pessoas com deficiência visual, mas apenas 160 cães-guia. Disponível em: <<https://emails.estadao.com.br/noticias/comportamento,brasil-tem-6-milhoes-de-pessoas-com-deficiencia-visual-mas-apenas-160-caes-guia,10000094416>> Acesso em: 7 fevereiro 2019.

Projeto cão-guia Brasil. Disponível em: <<http://www.caoguiabrasil.com.br/page/oprojeto.asp>>. Acesso em: 10 abril 2019.

Novo projeto para orientar a locomoção do cego Disponível em: <<http://www.acessibilidadebrasil.org.br/joomla/noticias/148-novo-projeto-para-orientar-a-locomocao-do-cego>>. Acesso em: 3 maio 2019.

Cinco aplicativos inovadores para cegos Disponível em: <http://www2.uol.com.br/vivermente/noticias/cinco_aplicativos_inovadores_para_cegos.html>. Acesso em: 20 maio 2019.

Download de programas utilizados para a criação de jogos Disponível em: <<https://www.yoyogames.com/>>. Acesso em: 18 junho 2019.