Esse trabalho possui objetivo de ensinar aos jogadores do jogo eletrônico criado maneiras com o meio ambiente de forma divertida e fácil para que mais pessoas se conscientizem e lutem pela causa da salvação do planeta Terra.

**Desenvolvimento (Dividi-lo em subtópicos)**

**Ferramenta escolhida x concorrentes**

Este capítulo visa descrever as etapas e ferramentas utilizadas para a realização da pesquisa. Após um período de estudo das ferramentas de criação de games (conhecidas como *engines)* disponíveis no mercado de softwares, algumas alternativas foram escolhidas para maior aprofundamento.

**Unity3d**: Presente há bastante tempo no mundo dos jogos 2D e 3D, conta com uma ampla comunidade e vários recursos interessantes. Possui suporte para os sistemas operacionais Windows e MacOS, utilizada por grandes empresas de desenvolvimento de jogos, usa linguagem C#, JavaScript e Boo em seus *scripts. Splash-screen* não é personalizável, é gratuita para estudantes e possui versões *pro* com recursos adicionais.

**UnrealEngine**: Utilizado para grandes projetos e jogos com gráficos de alta qualidade, possui cursos online para capacitação do desenvolvedor, seu *script* utiliza linguagem C++ e possui suporte para os sistemas operacionais: Windows e MacOS e é possível desenvolver jogos 2D e 3D. É gratuito para os utilizadores porém quando o jogo atinge um lucro de três mil dólares é preciso pagar 5% à empresa.

**Godot**: Bem nova no mercado de softwares, completamente gratuita e seu código é totalmente aberto. Possui ferramentas para o desenvolvimento de jogos 2D e 3D. Presente para os sistemas operacionais: Windows, Linux, MacOS e FreeBSD. Utiliza a linguagem GDScript para seus scripts, essa linguagem possui bastante familiaridade com Python. Sua comunidade é nova mas cresce a cada dia mais e seus usuários são bem ativos e a engine é promissora.

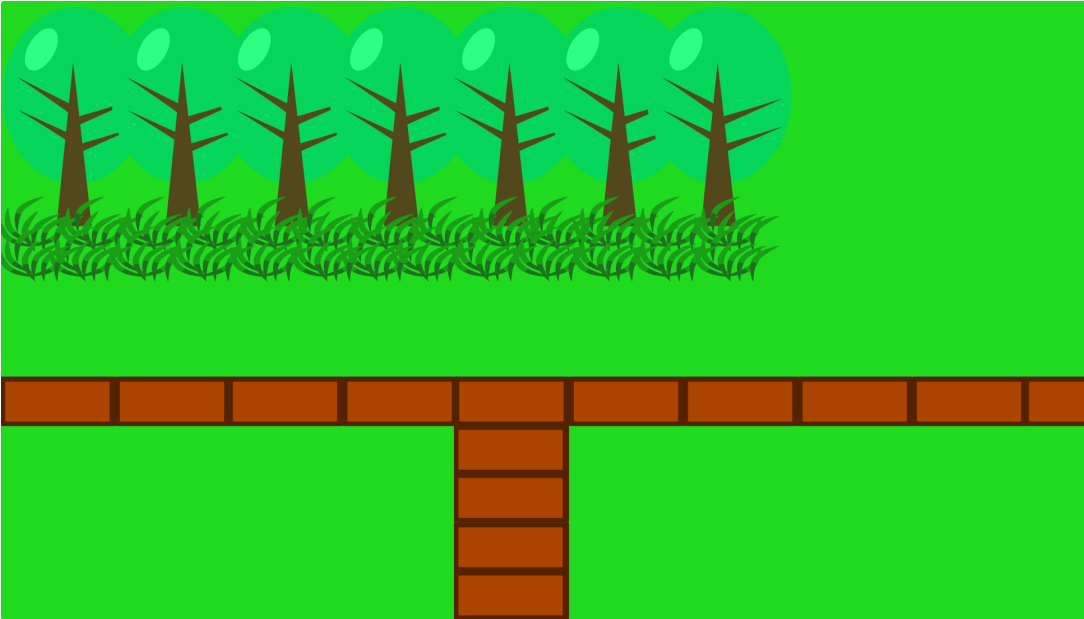
A *game engine* escolhida para o desenvolvimento do jogo eletrônico foi a Godot pelos seguintes pontos: possuir um *software* cliente nativo para *Linux, s*er uma *game engine* muito leve que faz com que funcione muito bem em computadores com pouco poder de processamento e pelo seu futuro promissor que fez com que o pesquisador se aprofundasse na engine.

Devido ao grande aumento na capacidade gráfica de computadores, smartphones, tablets e televisores, fez-se necessário que o jogo eletrônico possuísse belos elementos visuais que cativassem a atenção dos jogadores do game. Para o desenvolvimento dos cenários do jogo e dos personagens (conhecidos como sprites), a ferramenta de código aberto **Inkscape** foi escolhida para ser utilizada pois possui software cliente nativo para *Linux* além de consumir poucos recursos do computador em que é instalado.

**Funcionamento do jogo eletrônico**

O funcionamento do jogo eletrônico acredita-se ser a parte mais complicada de se resolver pelo motivo de envolver elementos do meio ambiente e fazer com que os mesmos se tornem algo divertido de ser jogado. Para a primeira fase do jogo eletrônico alguns elementos foram escolhidos para integrar seu funcionamento: lixos recicláveis, coleta dos mesmos, algum elemento de dificuldade e um belo cenário.

O cenário escolhido para o jogo eletrônico foi uma praça devido ao grande número de poluição das mesmas pelas cidades de todo o país. Utilizando-a como cenário crê-se numa conscientização no que diz respeito à praças, bosques, florestas e semelhantes. A figura 1 ilustra como ficou esse cenário depois de pronto na ferramenta Inkscape.

  
Figura 1: Cenário da primeira fase

A primeira fase do jogo ambiental conta com personagem que realiza a coleta de todos os lixos espalhados pelo cenário. A figura 2 é uma representação do resultado desse personagem.

  
Figura 2: Personagem que realiza a coleta

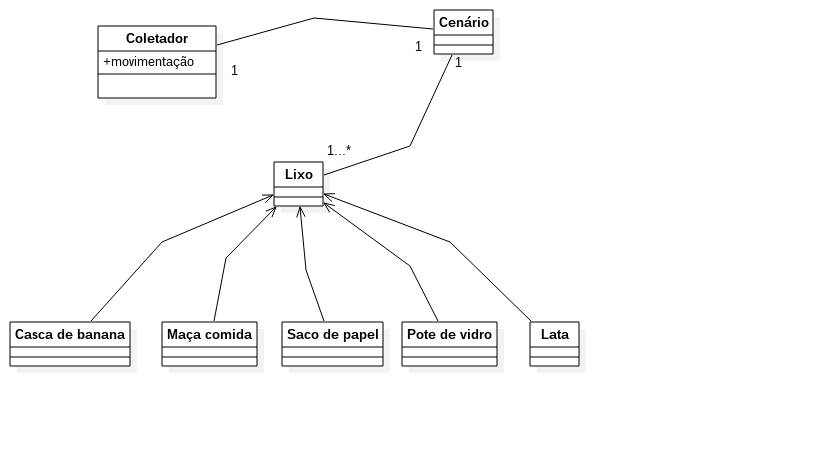
Na lógica do jogo, o controlador deve movimentar o personagem o mais rápido que conseguir e fazer com que ele colete a maior quantidade de lixos espalhados pelo cenário além de desviar de barris explosivos que aparecem aleatoriamente na região superior do cenário. Os barris explosivos foram criados com o intuito de adicionar dificuldade e mais dinâmica ao jogo, sendo assim, o controlador pode ser atingido por no máximo três barris. Ao ser atingido pelo terceiro barril, o jogo reinicia e caso o controlador tenha pontuado com mais de 200 pontos, a segunda fase e disponilizada, caso contrário o controladro pode jogar a primiera fase novamente para tentar alcançar os 200 pontos. A figura 3 ilustra o barril explosivo criado para o jogo eletrônico.

  
Figura 3: Barril explosivo do jogo eletrônico

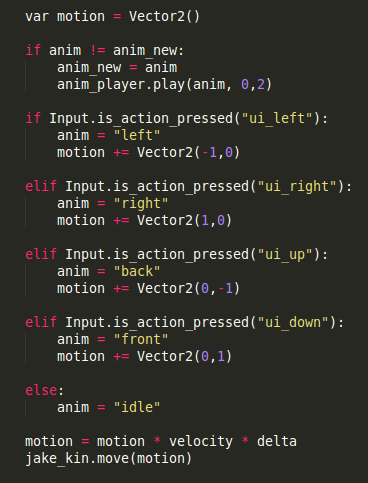
A figura 4 mostra como ficaram disponibilizados todos os elementos do jogo tanto como seu fluxo.

  
Figura 4: Primeira fase do jogo eletrônico

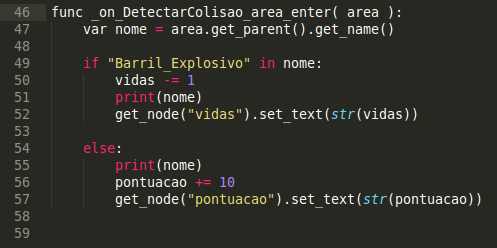
Um diagrama de classes foi criado com o objetivo de ilustrar os relacionamentos de cada item do jogo. A figura 5 mostra esse diagrama.

  
Figura 5: Diagrama de classes da Fase 1

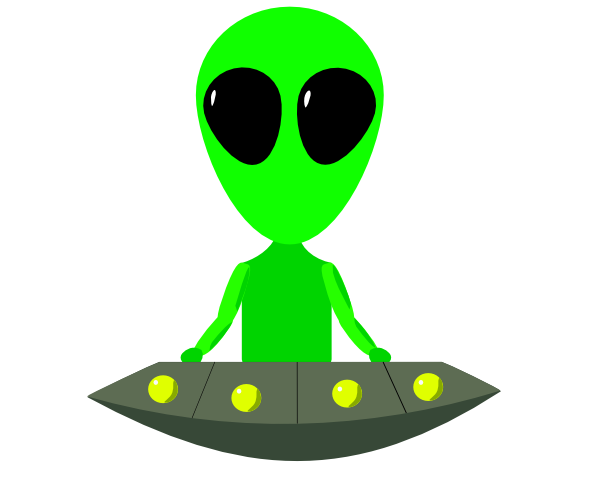
A linguagem de programação utilizada pela Godot se chama GDScript e ela é muito semelhante à linguagem Python. No fluxo de uso da Godot, os scripts (códigos) controlam todo o fluxo do jogo eletrônico, cada item possui seu script individualmente que faz com que eles se comportem da maneira que o desenvolvedor o codifique. A figura 6 mostra todo o código utilizado para realizar a movimentação do personagem.

  
Figura 6: Script de movimentação do personagem

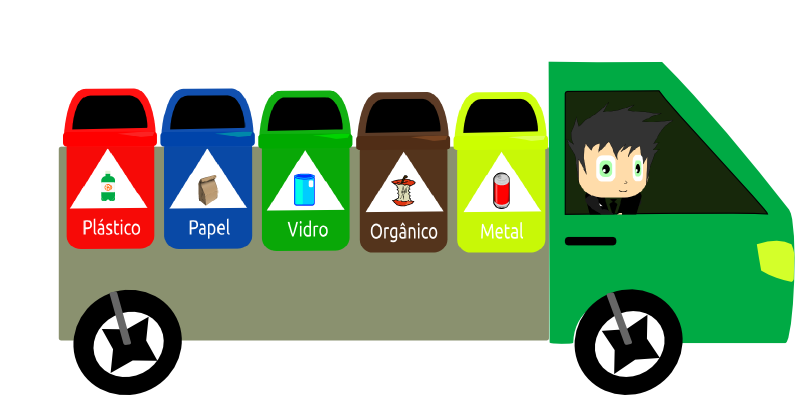
Para fazer o cálculo dos pontos obtidos pelo controlador é necessário identificar quando o personagem colide com os lixos e a cada colisão soma-se dez na variável que armazena a pontuação. A figura 7 demonstra o script que realiza essa função.

  
Figura 7: Script que calcula a pontuação

O objetivo da primeira fase do jogo eletrônico é fazer com o controlador entenda que não é correto jogar lixo no chão onde quer que seja. Na segunda fase os conceitos de separação do lixo são abordados. Um novo personagem foi desenhado para protagonizar a segunda fase, com o intuito de fantasiar o enredo do jogo, um extraterrestre foi desenhado na ferramenta Inkscape. A figura 8 ilustra esse personagem.

  
Figura 8: Extraterrestre da segunda fase

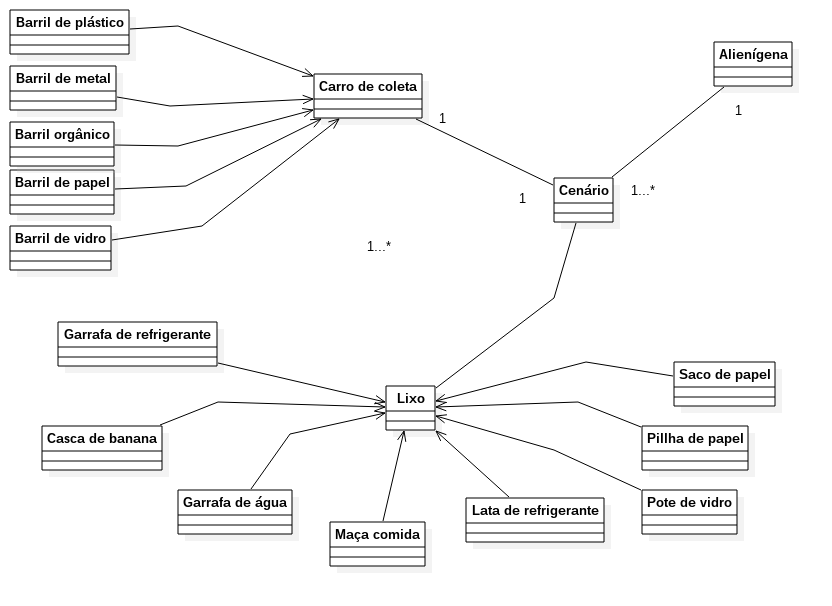
Na introdução da segunda fase, este extraterrestre se apodera de todo o lixo coletado na primeira fase e começa a soltá-los novamente no chão. Neste ponto, a tarefa do controlador é pegar todos os lixos que o extraterrestre solta antes que eles atingam o chão. Mas desta vez o controlador deve dirigir um caminhão de coleta para quando for coletar o lixo jogado pelo extraterrestre encaixá-lo no latão correto. A figura 9 ilustra esse caminhão depois de acabado na ferramenta Inkscape.

  
Figura 9: Caminhão de coleta da fase 2

A cada lixo coletado corretamente, a pontuação do controlador soma um ponto e a cada lixo coletado de forma errônea faz com que a pontuação diminua em um ponto. O controlador pode errar até dez vezes, caso ele gaste as dez chances de errar, uma tela de fim de jogo é mostrada e ele pode, se assim desejar, recomeçar a fase. A figura 10 mostra como ficaram disponibilizados todos os itens da segunda fase tanto com seu fluxo.

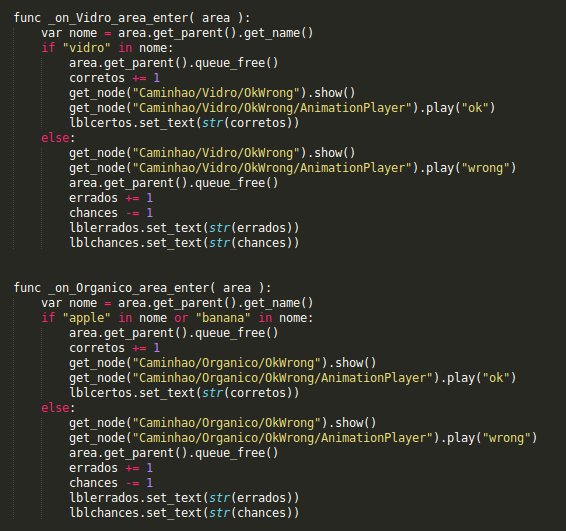
  
Figura 10: Funcionamento da fase 2

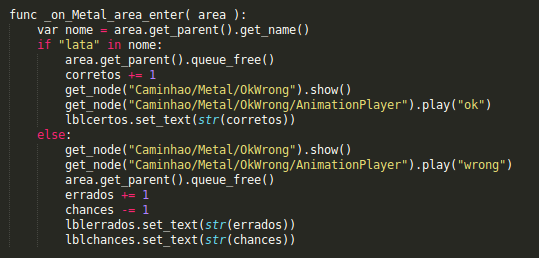
Um diagrama de classes foi criado para representar os relacionamentos presnetes na fase 2. A figura 11 ilustra esse diagrama.

  
Figura 11: Diagrama da fase 2

Para realizar a contagem dos pontos foi necessário identificar a colisão dos objetos e se eles eram passíveis de serem reciclados por cada latão que o controlador indicava. A figura 12, 13 e 14 mostram o código usado para realizar essa detecção.

  
Figura 12: Código de detecção de colisão parte 1

  
Figura 13: Código de detecção de colisão parte 2

  
Figura 14: Código de detecção de colisão parte 3