

Data Guida: Jogo para o ensino de banco de dados e conscientização ambiental

Breno R. Maia¹, Luisa F. Macedo¹, Leonardo Bravo¹

¹IFSC – Instituto Federal de Santa Catarina – Câmpus Lages
R. Heitor Villa Lobos, 225 - São Francisco, Lages - SC, 88506-400

breno.rocha.maia@gmail.com, luisafariasdemacedo@gmail.com, leonardo.bravo@ifsc.edu.br

Abstract. *Digital RPG games have the capacity to bring immersion by mechanical elements of the gameplay and narratives that instigate the players curiosity, however, they are still underused as learning tools. Although most RPG games address real themes due to its narrative nature, there are unexplored spaces in educational games, or serious games, to create material with the main learning obstacles of students, especially young adults. As an example, in the computational area, the SQL (Structured Query Language) still brings difficulties to students in complex queries, and Database is a prerequisite for programmers in the job market. Another relevant and contemporary theme is the environmental issue, which can be addressed by a narrative of ecological awareness. Using the strengths of the RPG genre, this article seeks to explore the interdisciplinarity of games, bringing a story with an environmental background and challenges using SQL.*

Resumo. *Jogos digitais de RPG possuem capacidade de trazer imersão através de elementos mecânicos de gameplay e narrativas que instigam o interesse dos jogadores, porém, ainda são subutilizados como ferramenta pedagógica. Apesar de grande parte dos jogos de RPG abordarem temas reais por sua natureza narrativa, existem espaços inexplorados em jogos educacionais, ou serious games, para a construção de material envolvendo os principais obstáculos de aprendizado de estudantes, sobretudo jovens adultos. Como exemplo, na área computacional, a linguagem SQL (Structured Query Language) ainda traz dificuldades aos alunos em consultas mais complexas, sendo que Bancos de Dados são pré-requisitos para programadores no mercado de trabalho. Outro tema relevante e contemporâneo é a questão ambiental, que pode ser abordada através de uma narrativa de consciência ecológica. Utilizando os pontos forte do gênero RPG, este artigo busca explorar a interdisciplinidade dos jogos, trazendo uma história com plano de fundo ambiental e desafios utilizando SQL.*

1. Introdução

Um jogo é uma atividade mental ou física envolvendo uma ou mais pessoas, que faz uso de regras predeterminadas e possui condições de vitória ¹(Suits, 2014). O objetivo de um jogo é vasto e não exclusivo, como expressão artística, exercício de habilidades práticas, recreacional

¹Condição de vitória é o elemento que caracteriza “como ganhar”em um jogo, como por exemplo chegar no fim de uma corrida ou sobreviver até o fim de uma noite. Podem ser também indiretos, não caracterizando necessariamente o fim do jogo.

ou educativo Griffiths (2002). Jogo digital, ou eletrônico, é a manifestação desse mesmo conceito utilizando recursos computacionais.

Em 2021, a indústria global dos jogos digitais movimentou US\$ 175,8 bilhões (Pacete, 2022). Essa área costuma ser associada com o entretenimento infantil, no entanto, a Pesquisa *Game Brasil*, realizada pelo Go Gamers (2022) com mais de 13 mil pessoas em 2022, concluiu que a maioria dos jogadores têm entre 16 e 24 anos, um público jovem adulto.

Uma das atratividades comuns em jogos eletrônicos é a presença de elementos que induzem a curiosidade do jogador e estimulem a lógica através da narrativa (Whitton, 2010). *Undertale* (Fox, 2015), por exemplo, baseado no sistema de *Role Playing Game* (RPG) ² e lançado em 2015, levanta questionamentos éticos e morais do jogador com seu sistema de ações, que permite dialogar de maneira pacífica com os monstros do mapa ou atacá-los. Essas decisões impactam a narrativa e faz com que o jogador reflita antes de tomá-las (Rimi, 2020).

No universo dos jogos educacionais, uma abordagem em destaque são os *serious games* que são produzidos com um propósito afora o entretenimento do jogador e podem ser utilizados, por exemplo, como ferramentas de simulação para formação profissional e finalidades educativas dentro da sala de aula (Fedwa Laamarti, 2014) e (Susi et al., 2007). Os produtos lançados como *serious games* ainda possuem escassos investimentos e incentivos fiscais, apesar de ser uma abordagem inovadora de engajamento do estudante. Eles abordam tópicos e problemas atuais com reflexões para as novas gerações nos campos da saúde, indústria, conservação animal e crise climática (Trista Patterson, 2019).

Lançado em 2021 e idealizado pelo Programa das Nações Unidas para o Meio Ambiente (UNEP), o jogo *mobile Reset Earth* traz em sua narrativa uma mensagem ambiental de conscientização a respeito de possíveis ações para um planeta mais sustentável. Com o público alvo voltado para Geração Z³, ele conta também com uma série *online* que acompanha os personagens principais no processo de revitalização do planeta (Trista Patterson, 2019). *Plasticity*, um *puzzle* 3D lançado em 2019, busca também incentivar ações individuais por um planeta mais sustentável, no qual o papel do personagem é mudar o ambiente em um mundo dominado pelo plástico (Game, 2019). Outro exemplo não tão recente é *Abe's Odyssey*, de 1997, que, além das questões ambientais, aborda outros assuntos importantes como exploração de mão de obra barata e péssimas condições de trabalho (Technica, 2019).

Além das questões ambientais, os jogos podem ser utilizados para o aprendizado de conhecimentos técnicos. O ensino em cursos superiores têm buscado cada vez mais inovações pedagógicas, saindo dos métodos tradicionais da docência, em que os alunos não possuem participação ativa no processo de aprendizado. Ainda assim, existem lacunas em universidades para a utilização de ferramentas lúdicas, de aplicação prática, que utilizem processos de aprendizagem ativa e *gamificação*⁴, a fim de instigar a curiosidade e participação de estudantes nesse processo de conhecimento (de Amorim, 2022).

A aprendizagem ativa possui o princípio de colocar o aluno como o centro do processo de aprendizagem, e surgiu com o advento da Escola Nova no final do século XXI. A ideia é

²Gênero de jogo fortemente baseado em narrativa, mais aprofundado na seção referencial teórico.

³Grupo de pessoas nascidas na década de 90 até 2010.

⁴Ato de utilizar elementos de *Game Design* no aprendizado.

que o estudo seja orientado pelo interesse do aluno, e também através da solução de problemas mais práticos que conversam com a experiência do estudante. Uma das maneiras de aplicar esta metodologia é a partir de *game-based learning*, a aprendizagem baseada em jogos, pois eles estimulam o pensamento estratégico do jogador (Pablo Bes, 2019).

Uma das unidades curriculares mais relevantes presentes nos cursos de computação (Ciência da Computação, Sistemas de Informação, entre outros) é Banco de Dados, por conta da sua relevância dentro do mercado de trabalho. Empresas como *Spotify*, *Uber*, *Netflix* e *Airbnb* utilizam bases de dados com *SQL*⁵, sendo uma linguagem relevante e um pré-requisito para qualquer desenvolvedor (Dataquest, 2021). Segundo a pesquisa de 2021 do *Stack Overflow* (StackOverflow, 2021), os três gerenciadores de bancos de dados relacionais mais utilizadas no mercado são o *MySQL*, *PostgreSQL* e *SQLite*, que usam linguagens de *SQL* para realizar consultas.

A disciplina de Banco de Dados possui dados como ponto de partida e envolve conceitos teóricos que devem ser trabalhados com resoluções de problemas práticos do mundo real (Carneiro, 2020). De acordo com os estudos de Seth Poulsen (2020) e Andrew Migler (2020), os alunos ainda têm dificuldades em relação à sintaxe e comandos do *SQL*, principalmente em consultas mais complexas envolvendo múltiplas associações entre as tabelas, como sub consultas, junções, entre outros.

É possível deixar o processo de ensino e aprendizagem mais lúdico através de jogos, além de permitir a associação de assuntos distintos como banco de dados e ensino ambiental, a fim de trazer experiências imersivas e colaborativas no campo da educação que estimulem o jogador na resolução de problemas. Susana Cristina dos Reis (2015) fala sobre como múltiplos ramos de conhecimento e aprendizado podem ser abordados em jogos, envolvendo alunos e professores de forma benéfica, sendo uma mídia fortemente propícia à interdisciplinaridade.

O objetivo deste trabalho é desenvolver um jogo de *RPG 2D* para o aprendizado de conceitos de banco de dados e outros desafios lógicos, utilizando como plano de fundo uma narrativa com questões atuais sobre meio ambiente. Os objetivos específicos deste trabalho são:

- Pesquisar referências de jogos e trabalhos relacionados a ensino de conceitos técnicos na computação, sobretudo envolvendo banco de dados;
- Desenvolver a narrativa, focada em conscientização ambiental e resolução de problemas de banco de dados;
- Realizar os passos utilizados para o *design* e desenvolvimento do jogo digital;

Este trabalho está organizado da seguinte forma: Na Seção 1 encontra-se a introdução; na Seção 2, o detalhamento da metodologia; na Seção 3, o referencial teórico; na Seção 4, o desenvolvimento; na Seção 5, resultados e discussões; e na Seção 6, conclusão.

2. Metodologia do Projeto

O trabalho é composto por duas etapas. A primeira etapa consiste na pesquisa bibliográfica de jogos envolvendo ensino, assim como desenvolvimento teórico da narrativa e mecânicas do jogo; e a segunda etapa são as fases para o desenvolvimento do *software*.

⁵Em bancos de dados relacionais, a linguagem de consulta utilizada é a *Structured Query Language* (*SQL*)

A primeira etapa consiste em pesquisar através de plataformas de trabalhos acadêmicos jogos do tipo *RPG*, jogos que ensinem e outras ferramentas que se alinham com a proposta do trabalho. Após isso, elaborar um *Game Design Document (GDD)*, documento teórico com a história do jogo, diálogos entre os personagens, escolhas de *design*, elementos que irão compor o cenário do jogo, entre outros detalhes.

Na segunda etapa, inicialmente foi realizado o estudo das ferramentas utilizadas, sobretudo do *Unity*, uma *game engine* baseada na linguagem *C#*. Assim, foram definidos conceitos no *GDD* relacionados aos requisitos da parte mecânica do jogo, como movimento do personagem, interação com objetos, câmera, realização das consultas no banco de dados, entre outros. Ainda na segunda etapa, foi definida a arquitetura do software e, posteriormente, a implementação de fato. Durante a etapa de implementação foram revistos conceitos, realizados testes manuais, novos requisitos foram criados de acordo com o andamento do jogo e a arquitetura foi refinada; assim, o desenvolvimento não foi linear e seguiu um ciclo de revisão e aprimoramento.

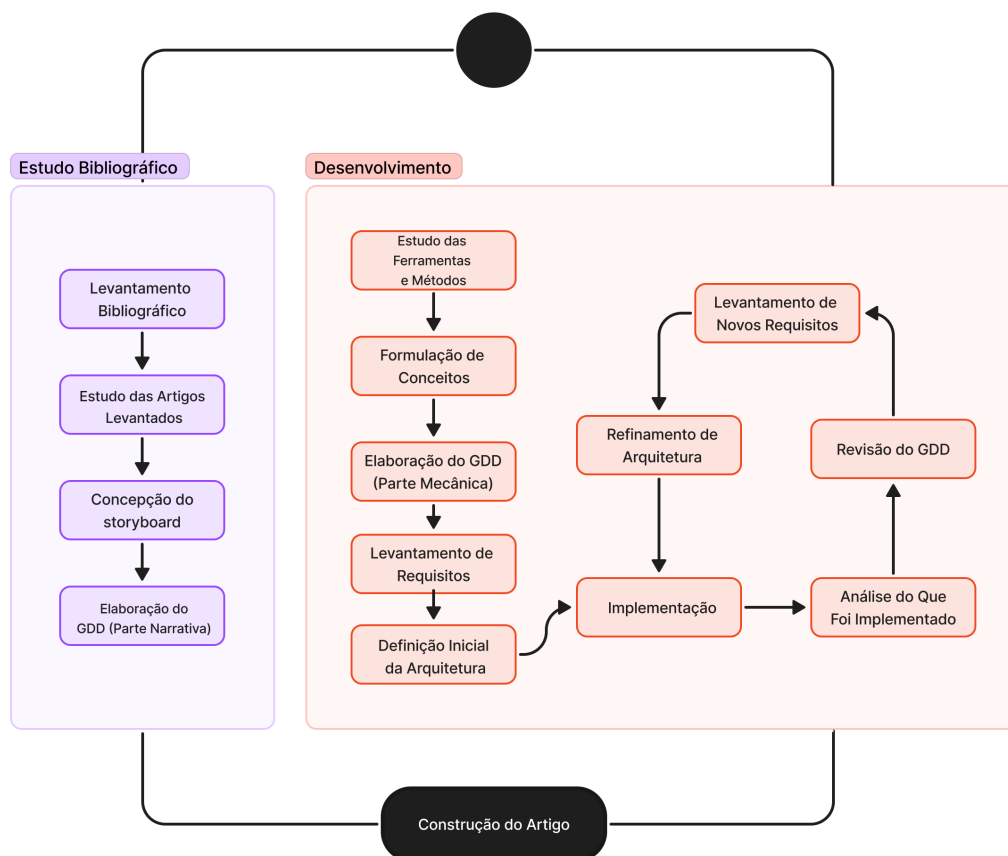


Figura 1. Etapas da Metodologia

Do ponto de vista da natureza, esse trabalho é uma pesquisa aplicada. Quanto à abordagem do problema, trata-se de uma pesquisa qualitativa. Referente aos objetivos, classifica-

se como uma pesquisa exploratória. Considerando os procedimentos técnicos, a pesquisa é bibliográfica.

3. Referencial Teórico

Essa seção aborda o referencial teórico necessário para o entendimento do trabalho, apresenta conteúdos referentes a definição de *Role Playing Game*, questões ambientais, banco de dados, revisão sistemática, trabalhos correlatos e tecnologias utilizadas para o desenvolvimento de jogos digitais.

3.1. RPG

O *RPG* (do inglês *Role Playing Game*, Jogo de Interpretação de Papel) é um gênero de jogo em que se mistura características tradicionais e elementos de *Storytelling*⁶, e em alguns casos, de teatro. Começou com o jogo *Dungeons & Dragons*, criado em 1974, que lança novos produtos até os dias de hoje. Em um *RPG*, o jogador se coloca no lugar de um personagem, pelo qual interage com um mundo ficcional e uma narrativa. A natureza desse mundo e da história que se desenrola é bem variável, desde locais fantásticos até eventos do mundo real.

A grande variedade e versatilidade do que esse gênero pode fazer tornou sua popularidade continuamente ascendente desde sua criação, contando com inúmeras franquias que abrangem diferentes públicos, como *Final Fantasy* (SQUARE ENIX, 1990), *Omori* (OMOCAT, 2020), entre outros (Britannica, 2009). Da mesma forma, o engajamento estimulado naturalmente por esse gênero o tornou propício para ensino de diversas áreas de conhecimento (Anita Raquel, 2008), como em *SCRUME'd* (Schneider, 2015) e *Katakana Wars* (Duck, 2017).

3.2. Questões ambientais

No Brasil, assuntos relevantes como meio ambiente ganharam mais força nos anos 90, quando ocorreu encontro mundial Rio-92 e muitas empresas começaram a se preocupar com sua imagem midiática em relação a sustentabilidade (Martins et al., 2015). Foi nessa época que o Conselho Nacional de Autorregulamentação Publicitária (Conar) começou a monitorar sobretudo propagandas com *greenwashing* ou “maquiagem verde”, prática de manipulação que associa uma marca a apelos sustentáveis, sociais e ecológicos falsos, passando uma imagem ao consumidor de responsabilidade com o meio ambiente que não existe de fato (Garcia, 2019).

Apesar dos questionamentos a respeito do aquecimento global, suscitado muitas vezes por *fake news*, o relatório de 2022 do Painel Intergovernamental sobre Mudanças Climáticas (IPCC) concluiu que os setores da indústria precisam diminuir a emissão de carbono e metano pela metade até 2030 (IPCC, 2021). Entre as consequências das mudanças climáticas estão secas, escassez de água, incêndios, inundações e perda de biodiversidade. De acordo com a Organização das Nações Unidas (2020), a atividade humana tem impulsionado essas mudanças por conta, sobretudo, do uso de energias não renováveis, e também pela pecuária, responsável por 14,5% das emissões dos gases de efeito estufa do mundo (FAO, 2013).

⁶*Storytelling* é a arte de contar histórias, utilizando elementos como personagens e conflito. Dimitri Vieira (2019)

3.3. Banco de Dados e SQL

A *Structured Query Language (SQL)* surgiu nos anos 70 como uma necessidade de linguagem de consulta para o modelo de banco de dados relacional proposto por E. F. Codd e seu aprendizado possui a facilidade de não requerer conhecimento de outros paradigmas de linguagem. Pessoas capazes de trabalhar com bancos de dados relacionais estão sendo cada vez mais procuradas dentro do mercado do trabalho, pois não atuam apenas nas áreas de desenvolvimento de *software*, mas também como analistas de negócios, pesquisadores, contadores, cientistas de dados, *marketing* e finanças (Andrew Migler, 2020) e (Carneiro, 2020).

A *SQL* pode ser dividida entre *Data Definition Language (DDL)* - comandos apresentados no Quadro 1; uma não oficial denominada *Data Control Language (DCL)* - comandos apresentados no Quadro 2; *Data Manipulation Language (DML)* - comandos apresentados no Quadro 3; e a *Data Query Language (DQL)* - comandos apresentados no Quadro 4. Na DDL, enquadram-se comandos responsáveis pela criação dos bancos, tabelas, índices, etc. Os principais tipos de dados que podem ser definidos na DDL podem ser observados no Quadro 5. A DML, por sua vez, é responsável pela manipulação e exclusão de dados; DQL envolve a maior parte dos comandos de consulta; e a DCL é responsável pela segurança de acesso dos usuários.

Em relação aos Sistemas de Gerenciamento de Banco de Dados (SGBD), conjunto de *softwares* que permite ao usuário o controle de bancos de dados, podem ser classificados em modelos relacionais, modelos orientados a objeto ou, ainda, modelos objeto-relacionais (Alves, 2014). Para a proposta deste trabalho, será utilizado o modelo relacional.

Quadro 1. SQL - Data Definition Language

Comando	Definição
CREATE DATABASE	Criar uma base de dados.
ALTER DATABASE	Permitir alterações em algumas características da base de dados.
DROP DATABASE	Apagar uma base de dados.
CREATE TABLE	Criar uma tabela no banco de dados.
ALTER TABLE	Realiza alterações em uma tabela existente.
DROP TABLE	Apaga uma tabela de dados existente.
CREATE INDEX	Cria índices em uma tabela.
DROP INDEX	Apaga índices ou um índice específico da tabela.

Quadro 2. SQL - Data Control Language

Comando	Definição
GRANT	Usado para dar privilégios aos usuários.
REVOKE	Remover privilégios ao usuário.

Quadro 3. SQL - Data Manipulation Language

Comando	Definição
INSERT INTO	Insere um novo registro numa tabela.
DELETE FROM	Apaga registros de uma tabela específica.
UPDATE	Atualiza dados da tabela.
SELECT FROM	Selecionar um conjunto de registros de uma tabela.

Quadro 4. SQL - Data Query Language

Comando	Definição
WHERE	Filtra um grupo de registros.
LIKE	Operador que permite filtrar uma cadeia de caracteres.
JOIN	Pode ser de diferentes tipos, como INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN. Usado para relacionar informações das tabelas, unindo-as com alguma condição de junção.
GROUP BY	Agrupar registros semelhantes de uma tabela.
ORDER BY	Ordena de acordo com uma coluna específica.
AND, OR e NOT	Operadores lógicos utilizados junto com o WHERE.

Quadro 5. SQL - Principais Tipos de Dados

Tipo	Definição
NULL	Valor nulo.
INTEGER	Um número inteiro que pode ter de 0 a 8 bytes, no caso do SQLite.
REAL	Um valor real (float) que permite casas decimais.
TEXT	Uma string de textos (UTF-8, UTF-16BE or UTF-16LE).
BLOB	Arquivo que armazena imagens, sons ou vídeos.

3.4. Trabalhos Correlatos

Foram analisados cerca de 25 trabalhos relacionados ao tema, retirados sobretudo da ferramenta do Google Acadêmico. Entre esses, foram elencados os 6 trabalhos mais importantes por meio da leitura do resumo. As *strings* de busca incluíram termos como “desenvolvimento de jogos rpg educacionais”, “jogo educativo meio-ambiente”, “jogo 2d rpg unity”, “jogo educativo sql” e “jogo educativo banco de dados”.

Schneider (2015) desenvolveu *SCRUM'ed*, um jogo de RPG que ensina conceitos da metodologia de desenvolvimento ágil *scrum*. O personagem precisa cumprir missões em um mundo medieval, no qual é demonstrado na prática utilizações e conceitos deste método. Foi feito através da ferramenta *Unity*, e utiliza do componente narrativo proporcionado pelo gênero *RPG* pra envolver o jogador e realizar sua proposta. Assim como *SCRUM'ed*, *Data Guida* busca trazer o aspecto técnico junto com uma narrativa através do gênero *RPG*, além de um sistema de missões e interação com outros personagens.

Schildgen (2014) ensina como manipular dados em um banco de dados relacional através de um *text-based game* chamado *SQI*sland. O jogador se encontra em uma ilha e consegue controlar seu personagem através da linguagem SQL, realizando múltiplas atividades para tentar escapar do lugar e conseguir recursos. Permite a interação com um banco de dados relacional para avançar, como em *Data Guida*. O jogo apresenta todas as tabelas que serão utilizadas previamente, e os personagens são estáticos, ou seja, não há um mapa e o jogador não pode controlar as interações, tendo uma narrativa linear.

Giovani Girardi (2019) fez um aplicativo voltando ao ensino e aprendizagem de linguagens estruturada de consultas, com intuito de ajudar e motivar tanto os alunos quanto os discentes na disciplina de banco de dados. Apesar de ser uma ferramenta de ensino banco de dados, não é caracterizada como um jogo.

de Amorim (2022) realiza um estudo detalhado de ensino através de jogos. Primeiro, foi realizado uma pesquisa extensa sobre o assunto da correlação de jogos e ensino. Após isso, foi criado com a participação dos alunos, um jogo de *RPG* que ensina ecologia. Foi utilizado tanto o jogo final quanto seu processo de desenvolvimento como ferramenta didática, e todo o processo de desenvolvimento foi documentado. O jogo, chamado de *As Origens*, inspirou o trabalho no aspecto educacional, entendendo como o *RPG* poderia ser utilizado em sala de aula para trazer metodologias ativas no aprendizado de SQL.

Beatriz Landigraf (2018) desenvolveu um jogo 3D para ensino de química e boas práticas em laboratórios utilizando a *game engine Unity*. A personagem do jogo, representada pela figura histórica de Marie Curie, tem o objetivo de concluir missões envolvendo a disciplina de química. O jogo, no entanto, não possui uma narrativa, tendo como objetivo principal o ensino técnico e lúdico, porém sem este aspecto que pode adicionar uma profundidade adicional e maior engajamento.

Por fim, Raquel e Tarouco (2008) falam sobre a utilização de jogos na educação, tanto como ferramenta de aprendizagem, quanto para o desenvolvimento de cooperação e criatividade. Enfatiza, principalmente, o gênero dos *RPGs* e sua capacidade de lidar com vários temas harmonicamente, como o caso de histórias envolvendo conscientização ambiental ou temas mais técnicos como banco de dados. O *RPG* permite o desenvolvimento de narrativas

bem elaboradas.

3.5. Tecnologias para Desenvolvimento

O jogo foi desenvolvido através do *Unity*, *game engine* multi-plataforma que utiliza a linguagem de programação C#. Ele possui uma arquitetura baseada em componentes, que define objetos encapsulados por *scripts* independentes, substituíveis e modulares, permitindo assim gerenciar a complexidade e encorajam a reutilização de código. Sua arquitetura e ferramentas facilitaram o desenvolvimento do jogo, que foi feito em 2D (duas dimensões). Dentro do *Unity*, o C# é a linguagem utilizada, e através dela podemos fazer chamadas para a *Unity Scripting API*, um mecanismo que interliga a *game engine* com o código desenvolvido. Cada objeto adicionado em uma cena pode possuir um código que o manipula para um determinado fim. A documentação completa da *Unity Scripting API* pode ser encontrado em *Unity Documentation* (2022).

Foram utilizadas para o jogo artes originais, criadas a partir das ferramentas do *Adobe Photoshop*, um software utilizado para a criação de designs e animações, e *LibreSprite*, um software *open source* de criação de *pixel arts*. Dentre as artes do jogo estão a criação dos personagens, animações, interfaces de menu, interfaces interativas, objetos do jogo, entre outros.

O sistema de banco de dados relacionais escolhido foi o *SQLite*, por conta de sua facilidade de configuração e sua proposta educativa. Ele é apropriado para pequenas aplicações *web*, *mobile*, análise de dados, e para armazenamento de dados individuais, sem vários usuários simultâneos acessando o banco, como é o caso do jogo desenvolvido. Para aplicações maiores, com grandes volumes de dados, recomenda-se outros tipos de bibliotecas e ferramentas; para propósitos educacionais com *SQL*, é uma ferramenta simples e eficiente (Hipp, 2000). Para a criação das tabelas e testes das consultas dos desafios desenvolvidos, foi utilizada a ferramenta *open source DB Browser for SQLite*, que permite o design e edição de bases de dados utilizando o *SQLite*.

Para propósitos de organização do jogo, foi utilizada o *GitHub* para o controle de versionamento, pois permite o acompanhamento da evolução do desenvolvimento. Além disso, foram organizados alguns dos principais requisitos de desenvolvimento na ferramenta *Trello*, como desenvolvimento da narrativa, conceitos dos personagens e seus designs, desenvolvimento das interfaces, criação das artes do mapa, resolução de *bugs*, além do acompanhamento do desenvolvimento de cada tarefa.

4. Desenvolvimento

A seguinte seção contém conteúdos diretamente relacionados ao desenvolvimento do trabalho, e está dividida na seção de requisitos, *GDD* e implementação de requisitos.

4.1. Requisitos do Jogo Digital

O jogo possui os seguintes requisitos:

- Movimentação;
- Janela de diálogo;
- Interação com Objetos;

- Interação com NPCs;
- Interação com Portas;
- Controle de Missões;
- Interações com banco de dados;
- Janelas interativas;
- Desafios de *SQL*;
- História;
- Personagens;
- Cenário e ambiente de jogo;
- Artes (em formato de desenhos digitais) dos personagens e cenários;
- Livros de *SQL* em jogo;
- Janela simulando um sistema de pesquisa onde se desbloqueia conteúdo;
- Menu de Pausa;
- Elaboração do *Game Design Document*.

Os requisitos foram determinados através de reuniões e *brainstorms*⁷ realizados entre os membros participantes do projeto.

O *Game Design Document (GDD)* é um documento importante para desenvolvimento de um jogo (Rogers, 2013). Neste documento estão contidos todos os requisitos teóricos, como decisões importantes de *design*, seu tema, história, personagens, controle, câmera, interface, movimentação, assim como os desafios que serão enfrentados (questões de *gameplay*). O desenvolvimento do *GDD* é um processo iterativo, sendo modificado e adequado durante todo o processo de desenvolvimento.

4.2. Game Design Document

A seguinte subseção contém conteúdo presente no *GDD*, dividido em história, *gameplay*, cenário, personagens, controles, câmera, *kanban* de produção, controle de versão e arquitetura de software.

4.2.1. História

O jogo conta a história de Guida, recém formada na faculdade de Ciência da Computação e que foi trabalhar em sua primeira empresa. Essa empresa produz um multivitamínico, e é conhecida por ajudar o meio ambiente e ser *eco-friendly*.

Esse multivitamínico inicialmente era vendido como um vitamínico comum, mas aos poucos a estratégia de marketing da empresa foi mudando, agora sendo vendido como “tudo que você precisa”. Extremamente popular na cidade, muita gente comia apenas isso e comidas gordurosas.

Ao longo da história, Guida descobre sobre diversas farsas da empresa, como doações para empresas duvidosas, lixo jogado no rio com sua fabricação, embalagens “biodegradáveis” que ainda causam mal, doações fraudulentas para não pagar impostos, alimento sem valor nutricional e viciante. Através das descobertas e resolução dos desafios, Guida decide investigar a fundo.

⁷Técnica de dinâmica de grupo para desenvolvimento criativo.

4.2.2. Gameplay

O jogador controla a personagem Guida dentro do ambiente de uma empresa. É possível caminhar ao redor, explorar o cenário, interagir com objetos e falar com *NPCs*⁸. Seus superiores, dentro da história do jogo, delegam tarefas relacionadas a banco de dados para serem resolvidas, porém, conforme o jogo avança, Guida descobre segredos sobre corrupção da empresa. Cabe ao jogador investigar, através da coleta de pistas, a história dentro do jogo e, principalmente, resolver desafios de *SQL*.

4.2.3. Cenário

A história ocorre em uma cidade ficcional, *SaturnVille*, de aparência fantasiosa e habitantes de diversas formas, com um visual *surrealista e caricato*. Porém, apesar dessa estética, acontece no ano de 2022 com diversos paralelos com o mundo real.

A maior parte da história ocorre dentro dessa empresa ficcional, que possui diversas sessões diferentes, como recepção, área de segurança, área de *marketing*, entre outros. Cada área é populada com respectivos *NPCs*.

4.2.4. Personagens

O jogo possui múltiplos *NPCs*, personagens que trabalham na empresa da narrativa e possuem diálogos diversos que mudam conforme o avanço do jogo. Cada personagem possui personalidades e objetivos distintos na história. A maioria dos nomes foram escolhidos como referências e homenagens a figuras importantes da história da computação.

⁸*Non-Playable Characters*, Personagens Não Jogáveis, termo usado para se referir a personagens em jogos que possuem comportamento automatizado.

Quadro 6. Personagens

Personagem	Função
Guida	Protagonista do jogo. Acabou de entrar na empresa.
Sr. Van Rossum	Chefe da empresa.
Frank Codd	Funcionário da empresa. Secretamente a controla por trás dos panos e é o real antagonista.
Edgar	Gerente da empresa. É o instrutor de Guida.
Srta. Lovelace	Recepcionista da empresa.
Dione	Planta da empresa que alerta Guida que há algo de errado.
George	Funcionário da empresa.
Grace	Funcionária da empresa.
Hopper	Amiga de Guida.

4.2.5. Controles

O jogo foi desenvolvido como um aplicativo para *PC*, e o controle da personagem é realizado por meio de *inputs* de mouse e teclado. Os controles durante jogo são:

- Teclas W, A, S, D ou Setas do teclado: movimentação
- Enter: interagir com os *NPCs* ou objetos.
- Tecla E ou clique de botão: abrir o primeiro caderno de anotações, com informações básicas sobre *SQL*.
- Tecla R ou clique de botão: abrir o segundo caderno de anotações, com informações sobre *SQL* e elementos de história.
- Tecla T ou clique de botão: abrir ou fechar janela de informações *Boogli*⁹, contendo informações relevantes.
- Tecla Esc ou clique de botão: pausar jogo e abrir menu.

Durante janelas interativas e menus, a interação é feita através de cliques e digitação.

4.2.6. Câmera

O jogo utiliza uma *top-down camera*, câmera posicionada acima do cenário que o jogador está e direcionada a ele, permitindo uma visão clara dos arredores. Uma escolha apropriada e comum em *RPG's 2D*, dentro outros gêneros Rimi (2021). O termo câmera dentro de *game design* surgiu pela primeira vez em *Starglider II*, e nada mais é do que o ponto de vista que o jogador vê o mundo do jogo, um ponto que determina quais informação gráficas serão representadas na tela em determinado momento, assim como de qual perspectiva Krichane (2018).

⁹Ferramenta de busca de dentro do universo do jogo, paródia do *Google*.

4.2.7. Kanban de Desenvolvimento

Através do *Trello* foi organizado o desenvolvimento pelo método *Kanban*, separando as tarefas nas categorias *backlog* (tarefas que estão sendo escritas), *backlog selecionado* (tarefas prontas para desenvolvimento), desenvolvimento e finalizado. As tarefas foram catalogadas utilizando etiquetas de suas respectivas áreas: programação, arquitetura, *game design*, arte, *pixelart*, requisito de som, *storytelling* e bloqueada. A Figura 2 representa como foi a organização do quadro.

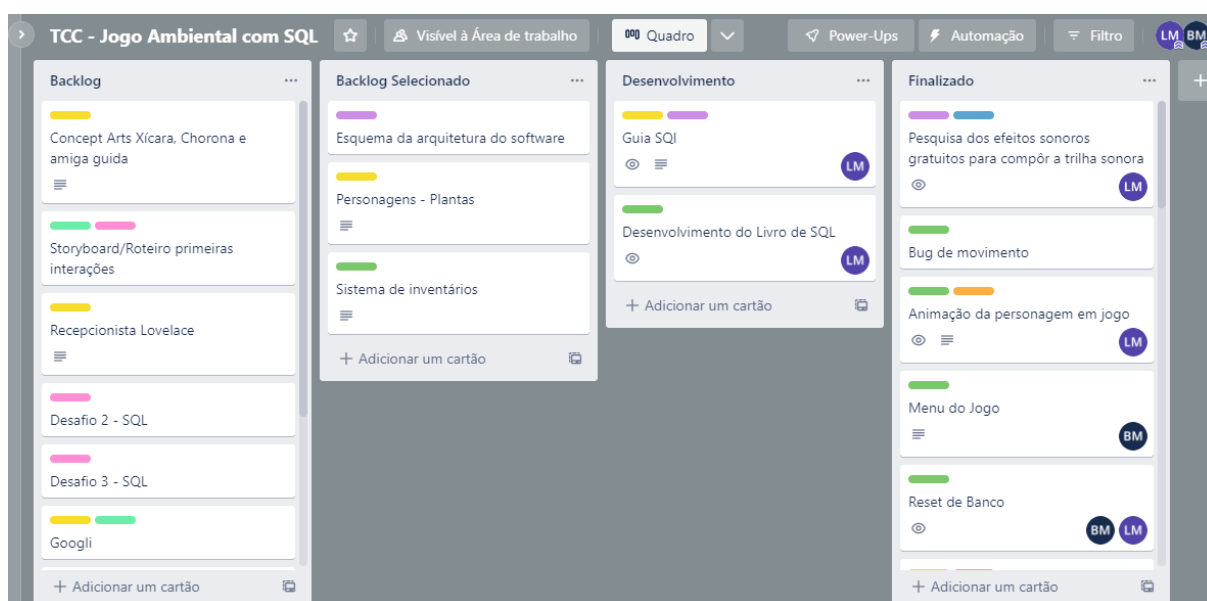


Figura 2. Trello - Kanban de Desenvolvimento

4.2.8. Arquitetura de Software

A arquitetura do jogo é dividida em classes Controladoras, sendo a classe *Player* e *Player-Controller* as principais responsáveis pelo movimento e interação da personagem principal, como pode ser visto na Figura 3. As principais classes serão explicadas nas seguintes seções.

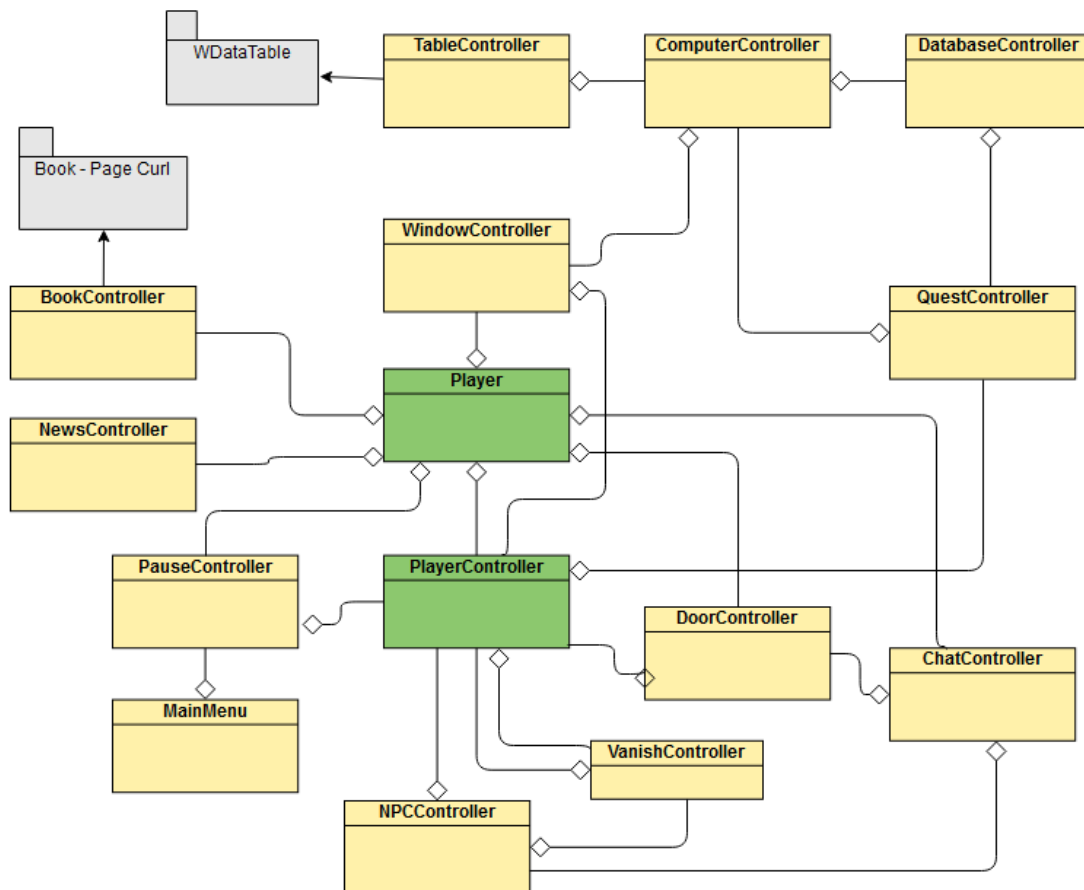


Figura 3. Abstração da Arquitetura do Jogo

4.3. Implementação

Nesta subsecção são mostrados detalhes do desenvolvimento.

4.3.1. Sprites, Mapa e Colisão

O jogo utiliza *Sprites*, imagens usadas pra formar todos os elementos gráficos apresentados ao jogador. Durante o desenvolvimento, foram utilizados *placeholder Sprites*¹⁰ que foram substituídos por *Sprites* originais criados através das ferramentas do *AdobePhotoshop* e *LibreSprite*. Na Figura 4, podemos observar o conjunto de imagens que determina o movimento da personagem principal, Guida, criada através do *LibreSprite*. Na Figura 5, podemos observar o mapa do jogo visto de cima e seus ambientes.

¹⁰Recursos temporários utilizados durante o desenvolvimento de um jogo no lugar dos recursos finais que ainda não foram desenvolvidos.



Figura 4. Sprites da protagonista do jogo: Guida

Os *Sprites* são atrelados a objetos e separados em camadas que determinam qual é posicionado à frente. Com objetos na mesma camada, isso é determinado pelo *Pivot*, posição do ponto mais baixo dele.

Nesses mesmos objetos podem ser colocados *BoxColliders2D*, que determinam sua área de colisão. A colisão impede o jogador de atravessar objetos, como uma parede. Também são usados para ler a presença do usuário, como, por exemplo, saber se ele está perto de um *NPC* específico para iniciar o diálogo.

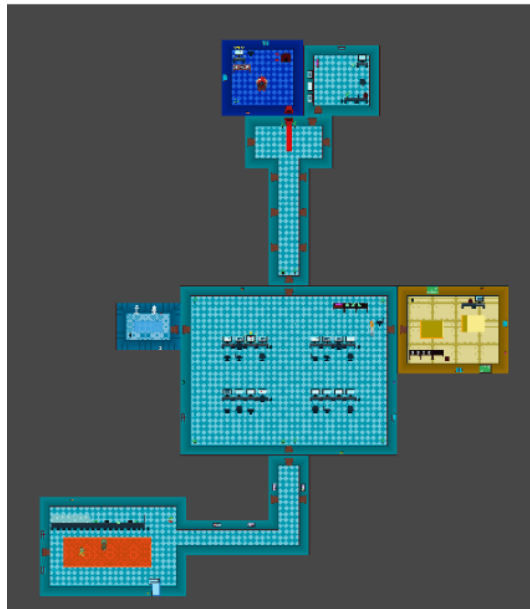


Figura 5. Mapa do jogo

4.3.2. Animação do Personagem

Existem dois recursos essenciais para a criação das animações no *Unity*: os clipes de animação (ou *Animation*) e o *Animator Controller*. Enquanto os clipes de animação determinam a sequência de *Sprites* a serem executadas para criar a ilusão de determinado movimento (por exemplo, um personagem andando para a direita), o *Animator* é responsável por mudar os estados do personagem (por exemplo, mudar o movimento de esquerda para a direita; mudar o estado parado para correndo, entre outros). O *Animation* é mostrado na Figura 6.

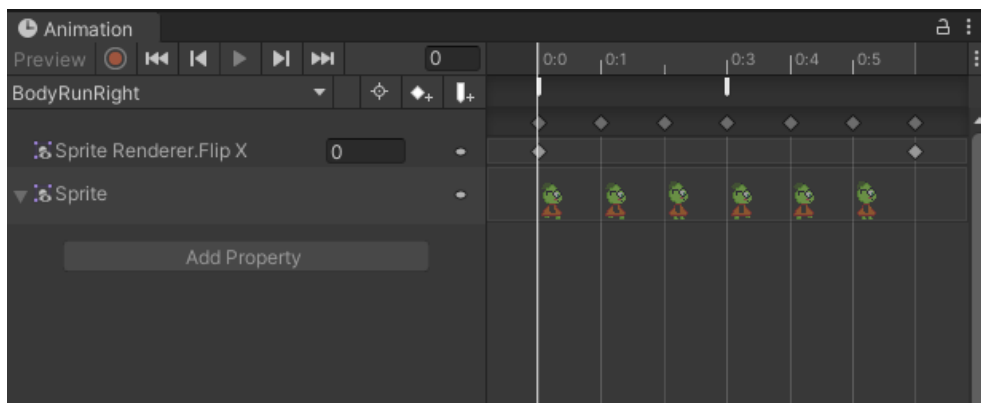


Figura 6. Exemplo de um clipe de animação no Unity

O *Animator Controller* é, em resumo, um conjunto de clipes de animação, responsável por mudar seus estados e transições de acordo com regras pré-determinadas. Como mostrado na Figura 7, dentro do jogo os estados da personagem se alteram entre correr (*Run*), andar (*Walk*) e parado (*Idle*), sendo o estado parado também o estado inicial do jogador ao iniciar a partida.

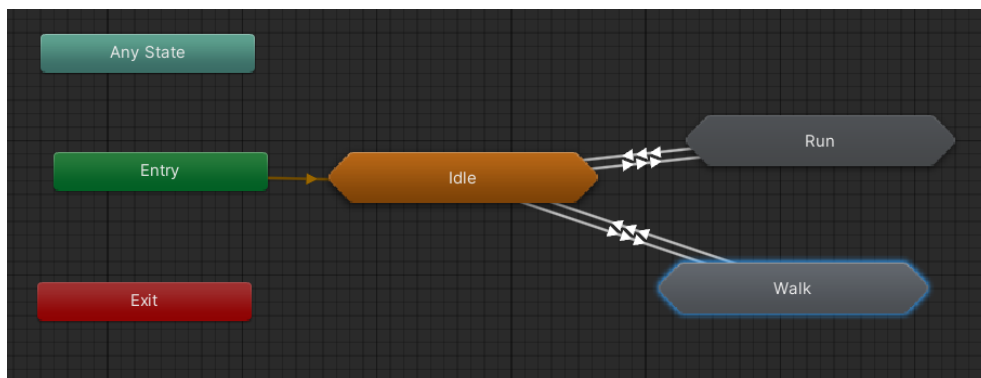


Figura 7. Animator Controller

4.3.3. Movimentação do Personagem

No contexto do *Unity*, a movimentação está ligada a um objeto, por sua vez ligado a um *Sprite* específico que é acionado de acordo com a direção do personagem. A classe *Player*,

mostrada na Figura 8, que é a classe principal do jogador, possui parâmetros que verificam a movimentação do personagem, como, por exemplo, a direção dentro do plano cartesiano; se ele está caminhando, correndo ou parado; qual a velocidade de movimento; verifica se ele pode se mover ou não, entre outras verificações de direção.

A classe *Player* herda da classe que implementa o *Design Pattern* chamado *Singleton*, um padrão de projeto que permite apenas uma instância de uma classe no jogo. Este padrão garante a existência de apenas um objeto *Player* no código.

```
public class Player : SingletonMonobehaviour<Player>
{
    public float xInput;
    public float yInput;
    private bool isWalking;
    private bool isRunning;
    private bool isIdle;
    private bool idleUp;
    private bool idleDown;
    private bool idleLeft;
    private bool idleRight;

    public bool firstInteraction = false;
    public bool canMove = true;
    public PlayerController playerController;
    private float movementSpeed;
}
```

Figura 8. Classe Player

O método *PlayerMovementInput* faz com que, como padrão, o jogador se mova na velocidade da corrida, definida na classe *Settings*. A classe *Settings* possui vários parâmetros de configuração e conversão utilizados no jogo. O método *PlayerMovementInput* também verifica qual a direção do personagem (esquerda, direita, abaixo ou a cima) de acordo com o plano cartesiano. Se o usuário não clica para mover o personagem, ele fica no estado denominado *Idle*, ou seja, parado.

Se o usuário clicar nas teclas de movimentação e simultaneamente na tecla *Shift*, a velocidade do personagem diminui, e ele começa a caminhar ao invés de correr. Esta lógica foi realizada pelo método *PlayerWalkInput*.

Por padrão o *Unity* executa o método *Update* a cada *frame* do jogo, permitindo uma atualização de *frames* mais otimizada. Os métodos *PlayerMovementInput* e *PlayerWalkInput* foram explicados anteriormente, e são acionados dentro dessa classe; por fim, a classe *EventHandler* chama o método responsável por acionar as animações das *Sprites* através de um evento que detecta o movimento do jogador, utilizando o padrão de projeto *Observable* (Figura 9).

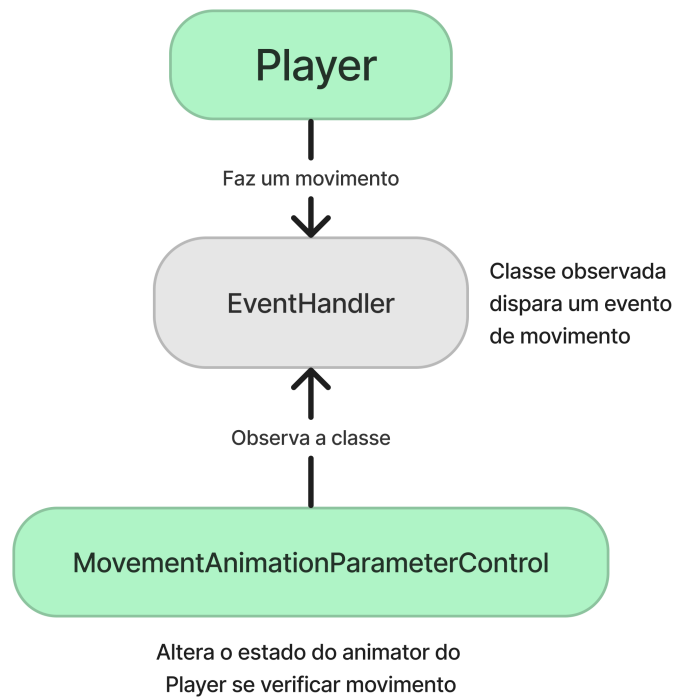


Figura 9. Padrão de projeto Observable

O jogador possui uma câmera filha o seguindo, que por sua vez é acompanhada de uma *CinemachineVirtualCamera*, que serve de controladora para onde vai com maior precisão, assim como uma *CinemachineConfiner*, que impede que mostre áreas onde não há mapa, "parando" a câmera nos limites.

4.3.4. Controle de interações

O *PlayerController* é a classe controladora do *Player* e responsável por maior parte das interações do jogo, como o início da caixa de diálogo, a interação com o computador, a interação com as portas e também a abertura do menu de pausa. As principais variáveis e métodos podem ser verificados na Figura 10.

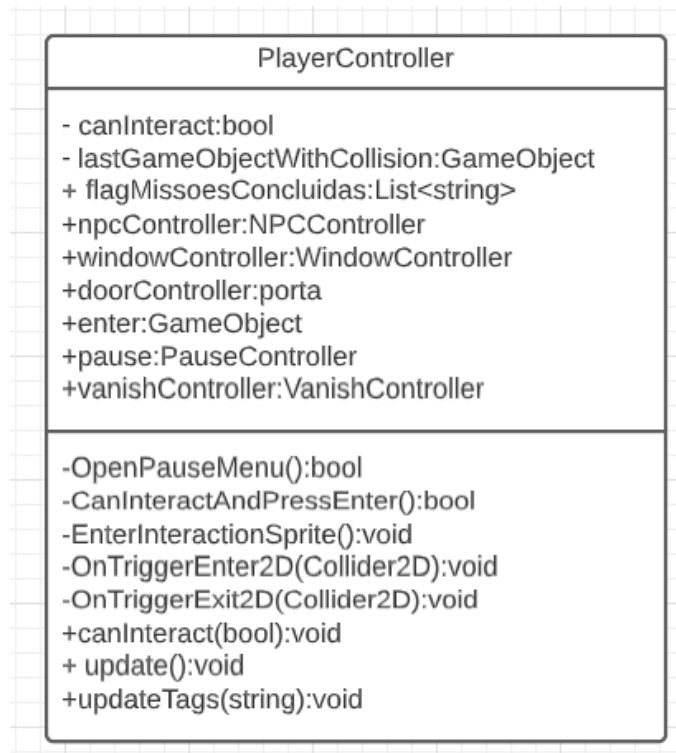


Figura 10. Principais variáveis e métodos

A classe faz isso por meio da interação com a tecla *Enter* ou *Esc* (no caso do menu de pausa). Quando o jogador está próximo de um NPC, um computador ou uma porta, ele ativa um colisor e a possibilidade de interação, como mostrado no diagrama da Figura 11. Isso é verificado por meio do seu método *Update*.

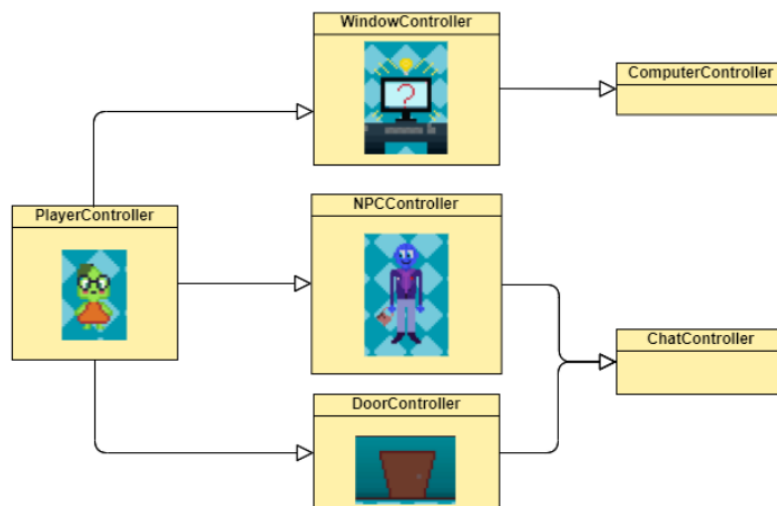


Figura 11. Interação do PlayerController com outras classes

Essencialmente, o método verifica se foi clicado *Enter* e se o jogador pode interagir. Assim, ele verifica o colisor mais próximo do jogador, verifica a *tag* do objeto do jogo, busca o objeto, inicia a interação e pausa o movimento do personagem (caso a interação seja com *NPC* ou Computador). Caso a interação seja com a porta, o jogador não é parado e é simplesmente transportado para outro cenário.

Existem outros métodos de interação no jogo utilizando botões que abrem outras telas, como é o caso do guia de *SQL* e o *Boogli*, que serão explicados nas seguintes subseções.

4.3.5. Sistema de Diálogo com NPCs

O jogo possui um sistema de diálogo em que, ao se aproximar de um *NPC*, é possível iniciar uma interação, como mostrado na Figura 12. Cada *NPC* possui uma lista de diálogos diferentes, os quais são liberados de acordo com o progresso do jogador e conclusão das missões. Esses diálogos são armazenados em cada *NPC* individualmente em seu objeto da classe *NPC-Controller*, que é responsável por iniciar o diálogo e por fim chamar o *ChatController*.



Figura 12. Interação de Diálogo

O *NPC* abre uma janela de diálogo, e altera para próxima fala toda vez que a tecla *Enter* é pressionada. A classe *ChatController* é responsável por controlar essa janela, assim como manter o jogador imóvel enquanto todas as falas aparecem em tela. Como mostrado na Figura 13, o método *Update* é chamado a cada *frame* e verifica as falas do personagem, imprimindo-as no *ObjetoTexto*, e a caixa de diálogo é desativada e o jogador volta a se movimentar em tela ao finalizar.

```

void Update()
{
    if (Input.GetKeyDown(KeyCode.Return) && Player.Instance.firstInteraction)
    {
        indexOfText = indexOfText < textos.Length ? ++indexOfText : indexOfText;

        if (indexOfText == textos.Length)
        {
            gameObject.SetActive(false);
            Player.Instance.PlayerReturnInteraction();
        }
        else
        {
            ObjetoTexto.text = textos[indexOfText];
        }
    }
}

```

Figura 13. Classe ChatController

O método *IniciaDialogo* é um método do *ChatController* que verifica se certa fala deve ser mostrada com base nas missões já concluídas do jogador. Assim, certos diálogos são desbloqueados apenas depois da conclusão das missões. Cada *NPC* possui um retrato específico associado, que será mostrado na tela junto a caixa de diálogos.

Pra determinar qual fala é utilizada, o *ChatController* recebe *flags* do jogador, que determina quais missões ele já realizou e com quais falas ele já interagiu. Após iniciar cada diálogo, ele chama uma classe *QuestController*, que é responsável por distribuir essas *flags* sempre que apropriado.

O *QuestController* também monitora cada consulta com o banco de dados, pra distribuir *flags* ao completar um objetivo.

4.3.6. Sistema de Interação com o SQL

O sistema de interação com a linguagem *SQL* é controlado através da classe *ComputerController*. Essencialmente, ela se comunica com uma classe *DatabaseController* que é responsável por realizar as consultas e pegar as colunas e os valores da base de dados criada previamente. Após isso, ele encaminha para um objeto *DataTable*, da biblioteca *WDataTable*, e retorna os dados na forma de tabela para serem mostrados na tela.

O componente *WDataTable* facilita a visualização das informações do banco de dados, mostrando as colunas e os dados de maneira organizada e contendo funções que auxiliam na manipulação da tabela (como adição de linhas, exclusão, atualização, *scrollbar* e possibilidade de alterar o tamanho dinamicamente), como se pode observar na Figura 14. Além disso, a janela do computador, possui outras funções, como o botão de *reset* à direita, que permite recuperar os dados do banco caso a tabela seja alterada, garantindo que o usuário não perca os dados para concluir a missão.

Os métodos colunas e dados da classe *DatabaseController* recebem a consulta, escrita pelo jogador, e o nome do banco de dados que será consultado. Através de métodos de

iteração, as listas das colunas e dos dados são preenchidos e enviados para a classe *Table-Controller*, no qual o componente *WDataTable*, mostrado pelo computador, é preenchido de fato.

Caso a consulta possua algum problema de sintaxe ou busque algo não presente no banco de dados, uma mensagem de erro amigável é exibida ao usuário. Caso o usuário deseje, ele pode clicar no botão de interrogação ao lado direito da tela, para ver a mensagem real de erro emitida pelo banco de dados.



Figura 14. Tabela sendo consultada

4.3.7. Livro de SQL

Durante o jogo, o jogador consegue verificar consultas de banco de dados e aprender outros conceitos práticos de SQL através da história e de dois livros, que serão dado a personagem logo no começo do jogo. O propósito é possuir conceitos técnicos passados de maneira divertida.

Os livros foram criados utilizando um Asset gratuito disponibilizado pela comunidade do Unity. Assets são recursos e objetos que podem ser utilizados no jogo, sejam pagos ou gratuitos; em resumo, são recursos de uma biblioteca. Cada página do livro foi configurada a partir de uma Sprite, e são escritos simulando serem escritos por um personagem fictício do universo do jogo. Há dois livros, um que possui elementos mais detalhados de bancos de dados, e outro que apresenta o conteúdo de forma direta e resumida. Isso foi feito para que o jogador não precise ler muito conteúdo teórico pra avançar no jogo, porém conforme ele vai se interessando, possui mais material para ler.



Figura 15. Uma página do livro de SQL

4.3.8. Boogli

Ao mesmo tempo que se desbloqueiam os livros, o jogador tem acesso ao Boogli, um sistema de pesquisa ficcional dentro do jogo e uma paródia do *Google*. Ao longo do jogo, vários conceitos na história são descobertos, como nome de empresas ficticiais, produtos, notícias e pessoas. Esses conceitos são adicionados no Boogli, em que se pode acessar e ler, como se fossem páginas da *web* ficticiais. Sempre que algo é adicionado dinamicamente nessa ferramenta, fica visível um símbolo de exclamação em cima do seu ícone indicando uma nova notícia até ele ser acessado novamente, como mostrado na Figura 16.



Figura 16. Tela do Boogli

4.3.9. Sistema de Missões

Para realizar o controle das missões, o *QuestController* é acionado e verifica se o jogador teve o resultado esperado da consulta através do método *verifyData*, mostrado na Figura 17. Foi determinado que a conclusão de uma missão será definida pela comparação do resultado esperado com o resultado obtido, pois diferentes tipos de consultas *SQL* podem resultar na mesma saída de dados.

No método *verifyData*, o comando *SQL* da missão é criado previamente, e a consulta é realizada num banco de dados de *backup*, ou seja, um banco de dados reserva que o usuário não terá nenhum controle. O resultado esperado é armazenado nas variáveis de linhas esperadas (*expectedLines*) e colunas esperadas (*expectedColumns*) no retorno, e comparado através do método *compare* com o resultado das colunas e linhas obtidas pelo jogador. Assim, se o jogador obteve o resultado correto, a *flag* da missão é atualizada, para que ele possa passar para o próximo desafio; caso contrário, ele continuará na mesma missão até conseguir completá-la.


```

private bool verifyData(List<string> columns, List<IList<object>> lines, string query)
{
    bool isMissionCompleted = false;

    expectedLines = database.dados(query, bancoReserva);
    expectedColumns = database.colunas(query, bancoReserva);

    if (compare(columns, lines))
    {
        Debug.Log("Missão concluída");
        isMissionCompleted = true;
        return isMissionCompleted;
    }
    else
    {
        Debug.Log("Missão não concluída");
        return isMissionCompleted;
    }
}

```

Figura 17. Método de verificação da conclusão da missão



Figura 18. Jogador Caminhando até um NPC

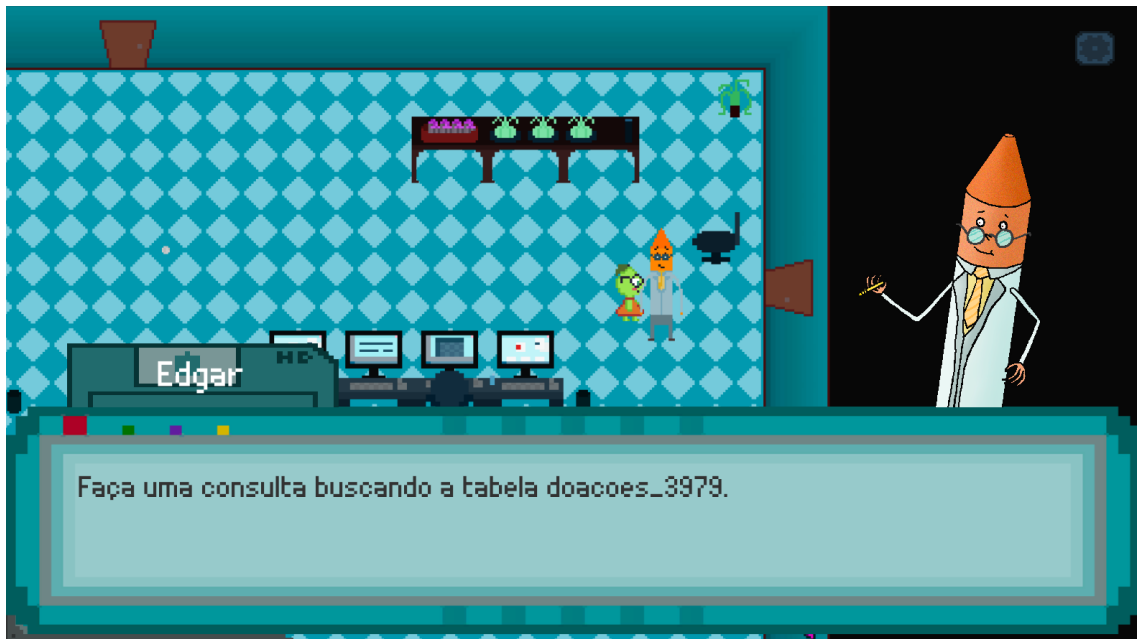


Figura 19. Recebendo missão de um NPC

4.3.10. Menu do jogo e Pausa

O jogo possui um sistema de menu inicial que controla as cenas e inicializa o jogo, assim como acessa uma janela de configurações, uma de créditos e um botão pra sair do jogo. Na janela de configurações, é possível alterar o volume das músicas do jogo, as quais foram escolhidas de acervos de uso livre. Todo o menu inicial é controlado pelo *script MainMenu*. Dentro do jogo em si, há um menu de pausa semelhante, acessível por um botão ou pressionando *esc*, controlado pelo *PauseController*.



Figura 20. Menu Principal



Figura 21. Tela de Pause

5. Conclusão

O objetivo do projeto foi desenvolver um jogo de RPG que ensina *SQL* e banco de dados aos alunos para que tenham uma interação ativa do aprendizado, e trazer como aspecto narrativo a conscientização ambiental. Como mencionado na Introdução, jogos educacionais ainda

possuem pouco investimento, e temas técnicos, como banco de dados, junto a discussões relevantes como sustentabilidade podem ser abordados dentro desse gênero através da interdisciplinaridade.

A equipe realizou o estudo bibliográfico de jogos que envolvem ensino técnico, criou o *Game Design Document* para a parte narrativa e também desenvolveu do jogo, que possui a primeira fase finalizada, com desafios que ensinam banco de dados e uma narrativa de conscientização ambiental.

Uma das dificuldades encontradas foi a interação do jogador com o banco de dados, pois foi necessária a criação de uma classe específica para resgatar os dados de forma apropriada. As colunas e linhas foram percorridas para serem mostradas na tela através do componente *WDataTable*. Além disso, houveram outras dificuldades envolvendo possíveis erros que o jogador poderia cometer durante o jogo e a interação com o banco, solucionados a partir dos botões de *reset* (permitindo a recriação do banco) e dica (com dicas de possíveis erros de consulta).

Outro desafio envolveu a parte de colisões, que são utilizadas para interação do personagem com o jogo. O jogador não conseguia interagir quando havia dois objetos próximos e com interação (como *NPCs* próximos a porta), e isso foi solucionado a partir da mudança de lógica, verificando qual o colisor mais próximo para realizar a interação.

Por conta da complexidade da criação de um jogo de RPG, e a falta de conhecimento da equipe em relação a ferramenta do *Unity*, foi necessário realizar treinamento e cursos externos, que facilitaram o desenvolvimento e consequentes dificuldades.

Dentre as limitações da aplicação está o uso do *SQLite* como banco de dados. A facilidade de configuração no banco permitiu uma otimização do tempo, porém, seria interessante usar um banco de dados mais completo, envolvendo autenticação, tipos de dados mais complexos e outras funcionalidades para a posterior criação de fases mais complexas no jogo.

As atividades futuras incluem avaliações de alunos de computação para um *feedback* de melhorias e ideias; adição de novas fases e amplificar a narrativa com novos desafios, deixando mais claro a história ambiental; criar um sistema para salvar o estado do jogador e permitir que ele volte a jogar a partir de uma fase específica; ampliar a trilha sonora e efeitos sonoros; entre outras melhorias relacionadas a arquitetura do software e otimização do jogo.

Referências

- Alves, W. P. (2014). *Banco de Dados*. Editora Saraiva, 1th edition.
- Andrew Migler, A. D. (2020). Mapping the sql learning process in introductory database courses. *Disponível em: <https://dl.acm.org/doi/pdf/10.1145/3328778.3366869>*. Acessado em: junho de 2022.
- Anita Raquel, L. M. (2008). O uso de jogos educacionais do tipo rpg na educação. *Disponível em: https://www.researchgate.net/publication/333278268_O_Uso_de_Jogos_Educacionais_do_Tipo_RPG_na_Educacao*. Acessado em: junho de 2022, page 11.
- Beatriz Landigraf, L. B. (2018). Laboratório curie: Jogo para o ensino de química experimental e boas práticas no laboratório.

- Britannica (2009). role-playing video game. Disponível em: <https://www.britannica.com/topic/role-playing-video-game>. Acessado em: junho de 2022.
- Carneiro, J. (2020). A review on tools for teaching database systems online. Disponível em: <http://www.kriativ-tech.com/wp-content/uploads/2020/09/Learning-Databases-Online-JOAO-CARNEIRO.pdf>. Acessado em: junho de 2022.
- das Nações Unidas, O. (2020). Climate action fast facts: What is climate change? Disponível em: <https://www.un.org/en/climatechange/science/key-findings>. Acessado em: julho de 2022.
- Dataquest (2021). Want a job in data? learn sql. Disponível em: <https://www.dataquest.io/blog/why-sql-is-the-most-important-language-to-learn/>. Acessado em: junho de 2022.
- de Amorim, D. C. (2022). Aprendizagem baseada em jogos digitais rpg no ensino superior: uma experiência na disciplina de ecologia. *Tese (Doutorado em Educação) - Programa de Pós-Graduação em Educação, Centro de Educação, Universidade Federal de Alagoas, Maceió, 2021*, page 44.
- Dimitri Vieira (2019). O que é storytelling? o guia para você dominar a arte de contar histórias e se tornar um excelente storyteller. Disponível em: <https://rockcontent.com/br/talent-blog/storytelling/>. Acessado em: junho de 2022.
- Duck, S. (2017). Learn japanese to survive katakana wars. Disponível em: https://store.steampowered.com/app/554600/Learn_Japanese_To_Survive_Katakana_War/. Acessado em: junho de 2022.
- FAO (2013). Mitigation of greenhouse gas emissions in livestock production: A review of technical options for non-co2 emissions. Disponível em: <https://www.fao.org/news/story/en/item/197623/icode/>. Acessado em: julho de 2022.
- Fedwa Laamarti, Mohamad Eid, e. A. E. S. (2014). An overview of serious games. Disponível em: <https://dl.acm.org/doi/pdf/10.1155/2014/358152>. Acessado em: junho de 2020.
- Fox, T. (2015). Undertale. Disponível em: <https://undertale.com/>. Acessado em: junho de 2022.
- Game, P. (2019). Plasticity. Disponível em: <https://plasticitygame.wixsite.com/about>. Acessado em: junho de 2022.
- Garcia, S. (2019). *A propaganda e sua relação com a sustentabilidade*. Blucher, 1th edition.
- Giovani Girardi, Matheus Conceição C. F. Guimarães, A. S. F. (2019). Ifsql: Plataforma de ensino e aprendizagem de linguagem estruturada de consultas.
- Go Gamers, S. G. (2022). Pesquisa game brasil - 9ª edição gratuita. Disponível em: <http://pesquisagamebrasil.rds.land/2022-painel-gratuito-pgb22>. Acessado em: junho de 2022.
- Griffiths, M. (2002). The educational benefits of videogames. Disponível em: http://irep.ntu.ac.uk/id/eprint/15272/1/187769_5405%20Griffiths%20Publisher.pdf. Acessado em: junho de 2022.
- Hipp, D. (2000). About sqlite. Disponível em: <https://www.sqlite.org/about.html>. Acessado em: junho de 2022.

- IPCC (2021). The evidence is clear: the time for action is now. we can halve emissions by 2030. Disponível em: <https://www.ipcc.ch/2022/04/04/ipcc-ar6-wgiii-pressrelease/>. Acessado em: julho de 2022.
- Krichane, S. (2018). When seeing is playing: The history of the videogame camera. Disponível em: <http://gamestudies.org/2102/articles/krichane>. Acessado em: junho de 2022.
- Martins, C. H. B., de Carvalho, P. G. M., Barcellos, F. C., e Moreira, G. G. (2015). Da rio-92 à rio+ 20: avanços e retrocessos da agenda 21 no brasil. *Indicadores Econômicos FEE*, 42(3):97–108.
- OMOCAT (2020). About omori. Disponível em: <https://www.omori-game.com/en/about>. Acessado em: junho de 2022.
- Pablo Bes, Amanda de Sena Fornarolli Pereira, I. G. P. M. P. C. L. R. M. (2019). *Metodologias para aprendizagem ativa*. Retirado de <https://app.minhabiblioteca.com.br/#/books/9788595029330/>. Acessado em: novembro de 2022.
- Pacete, L. G. (2022). 2022 promissor: mercado de games ultrapassará u\$ 200 bi até 2023. Disponível em: <https://forbes.com.br/forbes-tech/2022/01/com-2022-decisivo-mercado-de-games-ultrapassara-us-200-bi-ate-2023/>. Acessado em: junho de 2022.
- Raquel, A. e Tarouco, L. (2008). O uso de jogos educacionais do tipo rpg na educação. *RENOTE*, 6:11.
- Rimi, T. G. (2020). A exposição de temáticas ético-morais nos videogames: uma análise sobre o processo de tomada de decisão e sua implementação em undertale. Disponível em: <https://ariel.pucsp.br/jspui/handle/handle/23742>. Acessado em: junho de 2022.
- Rimi, T. G. (2021). How camera placement affects gameplay in video games. Disponível em: <https://dblp.org/rec/journals/corr/abs-2109-03750.html>. Acessado em: junho de 2022.
- Rogers, S. (2013). *Level up Um Guia Para o Design de Grandes Jogos*. Blucher, 1th edition.
- Schildgen, J. (2014). Sql island: An adventure game to learn the database language sql. Disponível em: https://www.researchgate.net/publication/267094856_SQL_Island_An_Adventure_Game_to_Learn_the_Database_Language_SQL, Acessdo em junho de 2022.
- Schneider, M. F. (2015). Scrum'ed: um jogo de rpg para ensinar scrum. Disponível em: http://www.gqs.ufsc.br/files/2020/02/TCCfinal_SCRUMed.pdf. Acessado em: junho de 2022.
- Seth Poulsen, Liia Butler, A. A. G. L. H. (2020). Insights from student solutions to sql homework problems. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/3341525.3387391>. Acessado em: junho de 2022.
- SQUARE ENIX (1990). History of final fantasy. Disponível em: <https://na.finalfantasy.com/histories>. Acessado em: junho de 2022.
- StackOverflow (2021). Developer survey. Disponível em: <https://insights.stackoverflow.com/survey/2021>. Acessado em: junho de 2022.
- Suits, B. (2014). *The Grasshoppers*. Broadview Press Ltd, 3th edition.
- Susana Cristina dos Reis, A. F. G. (2015). A produção de jogos sérios interdisciplinares na universidade: novos desafios e possibilidades para o ensino da linguagem. Disponível

- em: <https://periodicos.unb.br/index.php/horizontesla/article/download/1414/1247/2451>. Acessado em: junho de 2022.
- Susi, F., Johannesson, M., e Backlund, P. (2007). Serious games – an overview. Disponível em: <https://www.diva-portal.org/smash/get/diva2:2416/FULLTEXT01.pdf>, Acessado em 2022.
- Technica, A. (2019). How mind control saved oddworld: Abe's oddysee. Disponível em: <https://www.youtube.com/watch?v=Y7f0YtzWBG4>. Acessado em: junho de 2022.
- Trista Patterson, S. B. (2019). Playing for the planet - how video games can deliver for people and the enviroment. Disponível em: https://gridarendal-website-live.s3.amazonaws.com/production/documents/:s_document/506/original/gamingpub2019.pdf?1569241220. Acessado em: junho de 2022.
- Unity Documentation (2022). Unity scripting reference. Disponível em: <https://docs.unity3d.com/ScriptReference/>. Acessado em: junho de 2022.
- Whitton, N. (2010). *Learning with Digital Games - A Practical Guide to Engaging Students in Higher Education*. RoutLedge.