

Guia 1

P.O.M.! - Programação Orientada a Mechas!: Jogo Educativo para Reforço dos Conceitos de Programação Orientada a Objetos.

Leonardo A. Barros, Pedro H. R. S. Bomfim, Vitor E. Magalhães, Felipe D. Santiago, Henrique B. Diaz, André L. Brazil, Pâmela K. M. Gomes

Curso Superior de Tecnologia em Jogos Digitais
IFRJ - Instituto Federal de Educação, Ciência e Tecnologia do Estado do Rio de Janeiro
Eng. Paulo de Frontin, Brasil

leonardo.alves.barros.1@gmail.com, rosembackpedrob@gmail.com, vitor.ervilha02@gmail.com,
fipilito321@gmail.com, henriquebandeiradiaz@gmail.com, andre.brazil@iftj.edu.br, pamela.gomes@iftj.edu.br

Resumo— Este trabalho surge a partir da averiguação de uma dificuldade dos estudantes do Curso Superior de Tecnologia em Jogos Digitais com a assimilação e dos conceitos da Orientação a Objetos na programação. Por este paradigma ser abordado em diversos currículos da área de computação e, em especial, ser frequentemente aplicado e, portanto, extremamente necessário em áreas emergentes como a indústria de jogos, nós decidimos tentar abordá-lo através das metodologias dos Jogos Educativos. Com o objetivo de transmitir os conceitos do paradigma de forma lúdica e acessível, projetamos o jogo P.O.M.! (Programação Orientada à Mechas!) que emerge o jogador em uma narrativa de uma jovem “trainee” que vai entrar em um mundo de combate de mechas (robôs gigantes pilotados) sendo guiada por uma Mecânica Experiente que é sua treinadora. Por meio desta narrativa, buscamos alcançar, com o uso de diversas metáforas e paralelismos com a POO, uma maior relação de estudantes com os pilares da Abstração, Polimorfismo, Herança e Encapsulamento que o paradigma prega.

Palavras-chave: jogos educativos; aprendizagem imersiva; narrativa; programação orientada a objetos; produção de jogos;

1. INTRODUÇÃO

X

X X

Dentro dos contextos de desenvolvimento de softwares, o paradigma de orientação de objetos é amplamente abordado pelos currículos da área da computação. Contudo, esta aprendizagem se apresenta como um desafio, visto o contraste entre os conceitos de orientação a objetos em relação à programação procedural, que é ensinada primeiro aos estudantes (FIGUEIREDO, 2015). Assim sendo, o presente trabalho visa auxiliar os estudantes de graduação, de cursos da área de computação e jogos digitais, a entender e fixar os conceitos da Programação Orientada à Objetos (POO) através de uma abordagem lúdica, utilizando um jogo digital como ferramenta para tal.

Além disso, também foi identificado, através de uma pesquisa quantitativa, uma dificuldade geral entre os estudantes do Curso de Jogos Digitais do IFRJ, foco da nossa pesquisa, na fixação dos conceitos de orientação a objetos, sendo 60% dos respondentes afirmaram ter pouca ou nenhuma familiaridade com os conceitos de POO, 35% com nível de familiaridade média e apenas 5% com certo domínio dos mesmos.

Portanto, o presente trabalho, tem como público-alvo estudantes de graduação que possuam, em suas grades curriculares, conteúdos referentes à Programação Orientada a Objetos. Dessa forma, este estudo considerou como amostra os estudantes do Curso Superior de Tecnologia em Jogos Digitais (CSTJD) do IFRJ, em especial, os matriculados na disciplina de “Programação de Jogos 2D” e “Jogos 3D com Engines. Estes foram escolhidos por serem matérias introdutórias de programação de um curso da área da computação, que tem como objetivo formar profissionais na área de Jogos Digitais com enfoque à programação, possuindo matérias que abordam os conceitos de POO.

Dessa forma, no intuito de apresentar este conteúdo de difícil absorção para os alunos através de uma abordagem lúdica, para auxiliar no entendimento de forma complementar às dinâmicas de aula, este trabalho pretende desenvolver um jogo sério, de caráter educativo, para ser apresentar e desenvolver os conceitos básicos da Programação Orientada a Objetos. Logo, o jogo abstrai os fundamentos da POO, classes, funções, atributos, métodos e seus 4 Pilares, abstração, encapsulamento, herança e polimorfismo através de um jogo de montagem e luta de mechas. Assim sendo, o jogador será apresentado aos conteúdos abordados através de diálogos com personagens do jogo e, posteriormente, colocará seus conhecimentos recém adquiridos à prova. Para isso, o jogador terá que configurar e customizar um mecha, utilizando-se de recursos e componentes associados aos principais conceitos da POO, para derrotar inimigos em combate, com as configurações que o próprio jogador implementou. Portanto, a aprendizagem é favorecida de forma lúdica e dinâmica através do fluxo de jogo, demonstrando ao jogador uma abstração dos conceitos de POO, que o este está sendo exposto, na prática. Logo, esta abordagem visa manter os jogadores engajados na experiência sem perder o caráter educativo do jogo (CARVALHO, p.21, 2018):

Os jogos conseguem manter a concentração dos jogadores em uma mesma atividade durante horas. No cotidiano, o trabalho quase sempre é associado a algo difícil, que requer muito esforço, entediante, uma obrigação. Mas nos jogos, todo o trabalho se torna prazeroso, pois os objetivos a serem cumpridos gera recompensas e qualificações, em geral exigindo concentração, dedicação e inteligência. (CARVALHO, p.21, 2018)

Deste modo, o presente trabalho foi estruturado para seguir a seguinte metodologia, inicialmente destaca-se o processo de desenvolvimento de um jogo educativo na área de computação. Posteriormente, foram realizadas pesquisas sobre o tema e trabalhos relacionados que abordam a mesma temática com o uso de jogo digital como ferramenta de ensino, assim como jogos que possuem mecânicas que podem contribuir para o protótipo. Em seguida, o jogo foi testado para avaliar sua eficácia e especificidades. Por fim, foram apresentados os resultados acompanhados de uma breve conclusão acerca dos testes realizados.

2. TRABALHOS RELACIONADOS

Para o desenvolvimento do jogo, foram consultados artigos publicados sobre as definições dos conceitos e definições de POO, juntamente com artigos e jogos que abordam temas a respeito do uso de jogos digitais como ferramentas educativas de conceitos de programação. Além da pesquisa acerca dos conteúdos, também foi realizada uma busca por estilos de jogos que poderiam comportar as matérias a serem abordadas dentro do jogo. Desta feita, escolheu-se customização e luta de boxe como gêneros, com elementos de narrativa como principais características para o jogo. Paralelamente, também foram conduzidas pesquisas de jogos similares, que abordam o mesmo conteúdo de formas distintas.

2.1. Pesquisa sobre os fundamentos da POO

2.1.1 Conceitos básicos

Criada em 1967 (DAHL; NYGAARD, 1967), POO, é um paradigma de programação cujo núcleo é o conceito de objetos que contenham dados, propriedades e procedimentos que possam ser reaproveitados em múltiplas instâncias. Dessa forma, seus principais objetivos incluem a reutilização de código e modularização da escrita, uma vez que os códigos são reutilizados através dos objetos (DAHL; NYGAARD, 1967). Assim sendo, por conta da eficiência de reutilização de códigos, este paradigma foi adotado por quase todos os cursos da área de computação (FIGUEIREDO et al. 2015, apud Beck and Cunningham, 1989). Contudo, alunos enfrentam dificuldades ao serem expostos à esses conceitos, como aponta Holland, Griffiths e Woodman:

Conceitos de objetos são frequentemente ensinados, especialmente nas primeiras lições, com uma grande quantidade de demonstrações práticas durante palestras, e com muita ajuda especializada disponível para trabalhos de laboratório. Isso não é porque conceitos de objetos são intrinsecamente difíceis, mas porque o assunto oferece muitas oportunidades, especialmente nos estágios iniciais, para estudantes desenvolverem equívocos, que podem ser difíceis de corrigir posteriormente. Tais equívocos podem agir como barreiras pelas quais todo o ensino posterior do assunto possa ser inadvertidamente filtrado e distorcido. (HOLLAND, GRIFFITHS, WOODMAN, 1997, p.131, tradução nossa).

Entende-se como conceitos básicos os elementos fundamentais para a construção de um objeto que pode ser alterado e reutilizado de forma prática. Sendo eles: classes, objetos (instância de uma classe), atributos e métodos (DAHL; NYGAARD, 1967).

Segundo LEITE (2006, p. 245) “o conceito de classe, que pode ser definida, informalmente, como *‘um conjunto de elementos que possuem as mesmas características’*”. Objetos, são a definição fundamental da tecnologia da orientação a objetos (LEITE, 2006). Estes podem ser entendidos como abstrações de algo real, no qual serão guardadas informações relevantes ao mesmo, para o programa, sendo estes seus atributos, e interações através de operações. Assim como é definida por COAD e YOURDON (1991, p. 69): “Objeto é uma abstração de alguma coisa em um domínio de problemas, exprimindo as capacidades de um sistema de manter informações sobre ela, interagir com ela, ou ambos; um encapsulamento de valores de atributos e de seus serviços exclusivos”.

2.1.2 Os 4 Pilares

Para o presente projeto foram escolhidos os 4 Pilares da POO, sendo eles: Abstração, Encapsulamento, Herança e Polimorfismo. Estes foram selecionados pela sua relação com conceitos fundamentais e pela sua presença em aulas lecionadas dentro do curso onde a pesquisa foi aplicada. Dessa forma, outras definições de pilares de POO, que acrescentam ou diminuem os 4 que já são abordados dentro do Curso Superior de Tecnologia em Jogos Digitais do IFRJ, foram identificados como conteúdos que fogem ao escopo do trabalho e, então, descartados.

A abstração, é o pilar que consiste em modelar características do mundo real, descartando todas as características que não são implementação para o problema em questão, de forma a simplificar a implementação enquanto se mantém fiel à funcionalidade requerida. Assim como exemplifica LEITE (2006, p. 246):

Por exemplo, se o propósito é controlar as vendas de passagens de uma companhia aérea, *aeronave*, *passageiro* e *cidade*, seriam os assuntos que poderiam ser tratados a partir de uma classe denominada 'Vôo'. (...) Um vôo em particular seria um objeto que, entre outras coisas, teria a propriedade *Destino* para ser manipulada, por exemplo, pelo método *Voar()*. Desse modo, *cidade*, *passageiro* e *aeronave* seriam inseridos no contexto da abstração feita inicialmente.

Outro pilar é o Encapsulamento, este que consiste em permitir que um objeto se exponha o mínimo possível para outro objeto, como forma de prevenir acessos indesejáveis ao objeto, através dessa restrição de acesso (LEITE, 2006). Dessa forma, é possível modelar os atributos de uma classe conforme sua finalidade. Por exemplo, caso seja uma variável que precise ser acessada por outro objeto em ordem de realizar determinada função, como, ao apertar o controle de uma TV, este verifica se uma TV está ligada ou não e, dependendo deste retorno, realiza a ação de ligá-la ou desligá-la. Contudo, as informações que fazem a TV ligar e outros atributos relacionados estão protegidos e não podem ser acessados pelo usuário através do controle (LEITE, 2006).

Já a Herança, se caracteriza pela implementação de uma classe dentro de outra. Dessa forma, a classe que herda da outra receberá todos seus atributos, métodos e funções. Este pilar é utilizado para a reutilização de código em objetos que, apesar de não corresponder totalmente às definições de uma classe, por exigir mais ou menos informações ou funções, este pode usar essa classe como um molde para fazer outras implementações de acordo com os requisitos propostos (LEITE, 2006).

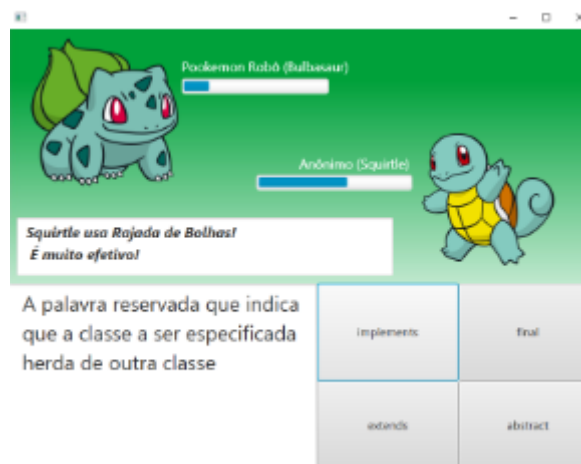
Por fim, o Polimorfismo, frequentemente utilizado juntamente da herança, é a implementação de novas instruções dentro de uma função já existente. Dessa forma, o programador consegue reaproveitar métodos já implementados e adaptá-los conforme é necessário para o novo problema em questão. Por exemplo, para calcular a área de um retângulo e de um trapézio, que possuem formas distintas, logo as fórmulas empregadas para este cálculo também são diferentes, contudo, é possível utilizar de uma atividade em comum, que seria "calcular área". Dessa forma, esta pode ser transformada em uma função *CalcularÁrea()*, que irá ser chamado e alterado pelo programador, para satisfazer os requisitos apresentados, sendo esta uma aplicação do polimorfismo (LEITE, 2006).

2.3. Jogos relacionados a Aprendizagem de Programação Orientada a Objetos

Dada a relevância do tema e dificuldade de aprendizado dos alunos, outros projetos de jogos sérios já foram desenvolvidos sobre o tema. Dessa forma, essa seção se propõe a analisar os principais resultados a fim de identificar pontos positivos, negativos e interessantes.

Primeiramente, temos POOkémon (GONÇALVES, MORAIS, SCHUINDT, BATISTA, 2018), um jogo que reimagina Pokémon Red (Game Freak, 1996) como um quiz. Neste jogo, o jogador possui um Pokémon e lhe é apresentada uma pergunta sobre POO na tela (Figura 1) e, ao invés de possuir uma lista de ataques, possui uma lista com possíveis respostas. Portanto, caso o jogador acerte a pergunta, o Pokémon inimigo recebe dano. Caso contrário, o Pokémon do jogador é danificado. Além disso, através dos arquivos do jogo, professores podem alterar as perguntas presentes no jogo. Logo, foram identificados os seguintes pontos positivos: Abordagem educativa através de uma obra popular; Reimaginação de um sistema de combate com quiz. Pontos negativos: Utilização de uma Propriedade Intelectual que não é de autoria dos desenvolvedores; Jogabilidade que reforça a memorização, não absorção do conteúdo. Pontos interessantes: Perguntas do jogo podem ser editadas pelo professor.

Figura 1 - Interface do Jogo POOkemon



Fonte: POOkémon: um jogo sobre programação orientada a objetos, 2018.

2.4. Jogos Utilizados como Referências de Jogabilidade

Figura 2 - Capturas de Tela do Jogo Digital Wolfstride



Fonte: POOkémon: Gameplay Wolfstride - Youtube.

Wolfstride (OTTA IMON STUDIO, 2021) é um jogo de customização de mecha e combate de turno. Este foi a principal referência estética do projeto, visto seu estilo artístico carismático e marcante. Além disso, foi referência para o fluxo de jogo, uma vez que este baseia-se em customizar o mecha, utilizando de peças novas com atributos e habilidades distintas. Além do mais, possui um momento de combate, onde a estratégia do jogador ao montar o mecha é posta à prova ao enfrentar mechas com suas próprias estratégias e vantagens. Alguns aspectos da interface visual do jogo desenvolvido foram baseados na interface apresentada deste jogo (Figura 2 - direita).

Figura 3 - Capturas de Tela do Jogo Digital “Punch-Out!!” (2009)



Fonte: Jogo Punch-Out!! para o Nintendo Wii, 2009.

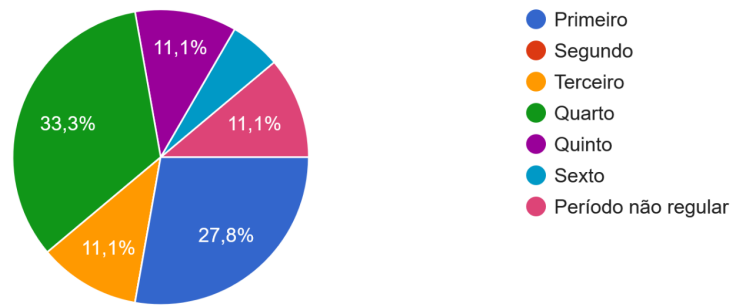
Punch-Out!! (Next Level Games, 2009) de Wii é um jogo de luta, especificamente boxe, onde o jogador deve vencer diversos inimigos ao participar de torneios. Cada inimigo possui mecânicas e comportamentos próprios e, apesar do mesmo não ser verdade para o jogador, os controles e jogabilidade simples permitem que o mesmo se adapte às estratégias de seus oponentes através da própria movimentação e aplicação de golpes do personagem principal. Isto é, seguindo um ritmo determinado pelo comportamento do particular de cada inimigo. Este jogo foi a principal referência para o combate deste projeto, visto sua complexidade com mecânicas de implementação simples, intuitivo e divertido de se jogar.

3. PROPOSTA e IMPLEMENTAÇÃO

3.1. Pesquisa Preliminar (incluir)

O desenvolvimento do projeto teve início com a realização de uma pesquisa com os alunos da graduação de jogos digitais do IFRJ Campus Eng. Paulo de Frontin, com o objetivo de compreender as dificuldades de aprendizado na área de programação. Mais em específico, a área de Programação Orientada a Objeto (POO), matéria que é abordada, principalmente no primeiro e segundo períodos da graduação. Dessa forma, o formulário foi aplicado buscando identificar, dentre os respondentes, qual o período, principal área de atuação dentro do desenvolvimento de jogos digitais e avaliar a familiaridade destes com os conceitos básicos de POO e seus 4 Pilares. Seguem as imagens de X à Y que demonstram os resultados.X

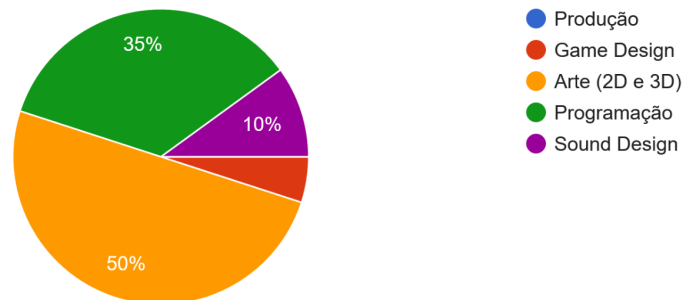
Figura 4 - Resultados da Pergunta 1 - Período do Curso



Fonte: Produção Própria.

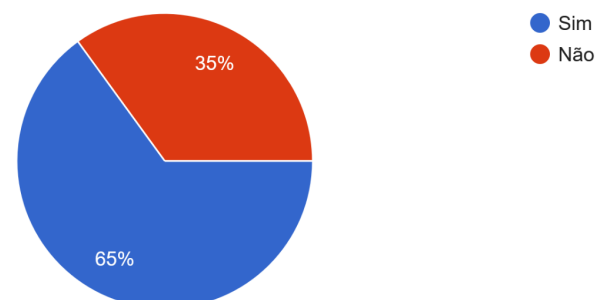
Os resultados da Pergunta 1 podem ser visualizados a partir da Figura 4. A partir deles é possível observar que a maioria dos participantes da pesquisa pertencia ao primeiro período (33%) ou quarto período (27,8%), ainda havendo outros estudantes dos demais períodos do curso, em número menor. Isto indica que houve uma ampla participação de diversos períodos do curso nesta pesquisa, trazendo feedbacks tanto dos calouros (33%) quanto dos demais estudantes ao longo de todo o curso.

Figura 5 - Resultados da Pergunta 1 - Período do Curso



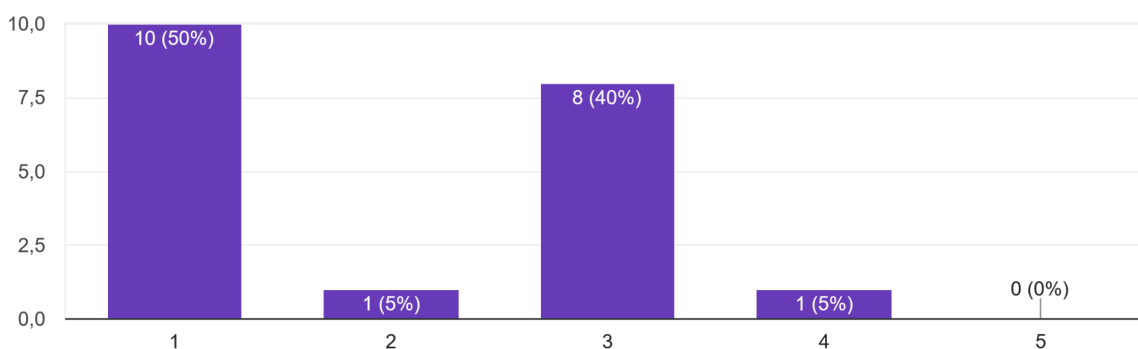
Fonte: Produção Própria.

Figura 6 - Resultados da Pergunta 2 - Pergunta sobre a familiaridade com a POO



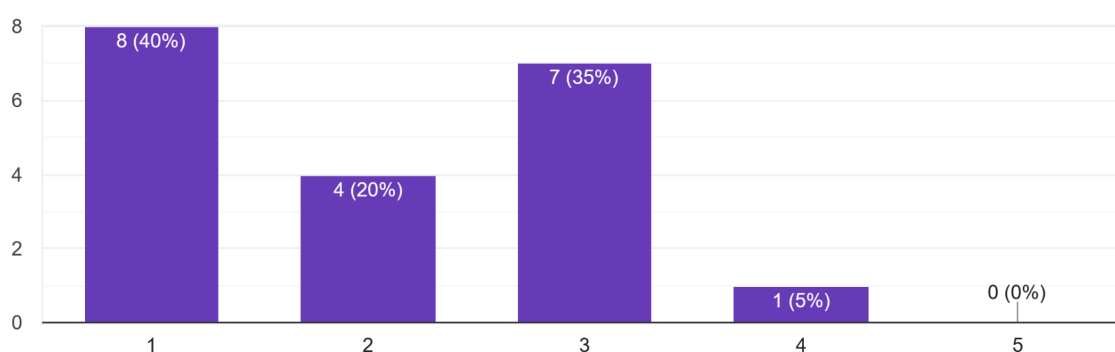
Fonte: Produção Própria

Figura 7 - Resultados da Pergunta Como você avalia seu conhecimento de POO?



Fonte: Produção Própria.

Figura 8 - Resultados da Pergunta Qual seu nível de familiaridade com os conceitos dos 4 pilares da POO?



Fonte: Produção Própria.

Assim sendo, foi identificado que, apesar da POO ser reconhecida, pela maioria dos respondentes (65%), 60% indicam ter familiaridade abaixo da média com seus conceitos e 35% na média, sobrando apenas 5% acima da média.

X

3.2. Proposta do Jogo

O P.O.M.! (Programação Orientada a Mechas) é um jogo de estratégia, customização e de luta. O personagem do jogador é a Trainee Promissora, uma jovem determinada que entra em um torneio de treinamento, cujo prêmio é a vaga de Operador de Mecha Júnior na empresa Working Orcs.

A dinâmica do jogo consiste em construir robôs utilizando conceitos de POO para enfrentar adversários e progredir no enredo. A cada desafio, o jogador precisará aplicar de forma mais eficiente os princípios da POO para superar dificuldades cada vez maiores.

A narrativa do jogo é inspirada no conceito de *mechas*, robôs gigantes amplamente presentes na mídia japonesa, como na franquia multimídia *Mobile Suit Gundam* (Sunrise, 1979), e também em jogos digitais, como *Armored Core* (FromSoftware, 1997). A influência dos mechas também pode ser vista em jogos brasileiros, como *Wolfstride* (OTA IMON Studios, 2021) e *Chroma Squad* (Behold Studios, 2015), devido a influência de séries japonesas *tokusatsu* exibidas no Brasil.

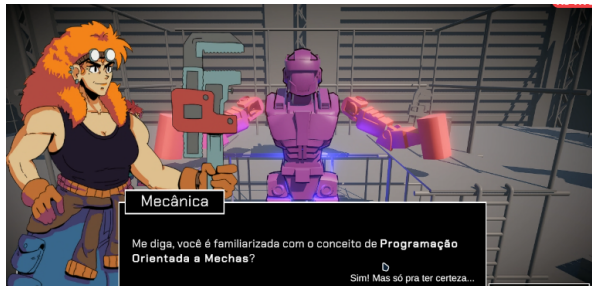


Figura 9 - Diálogo com a mentora Robô

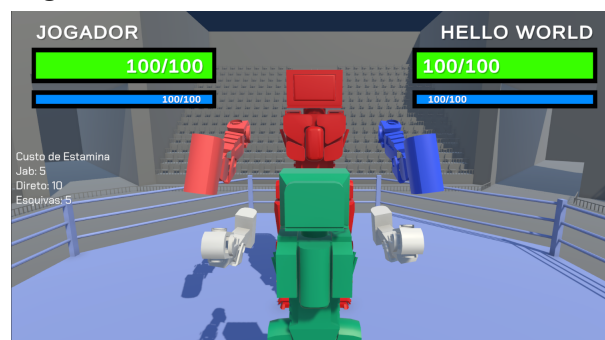


Figura 10 - Interface de Customização do

X O jogo utiliza os pilares da orientação a objetos (Abstração, Herança, Polimorfismo e Encapsulamento) como elementos centrais das mecânicas. Conceitos de programação, como classes, objetos e atributos, também são fundamentais para a experiência.

Dessa forma, o fluxo de jogo é dividido entre 3 momentos: diálogo com a mentora (Figura 9), customização e combate (representado, respectivamente, pelas Figuras 9, 10 e 11). No momento de diálogos com a Mecânica Experiente, a mentora do jogador, ela irá explicar como o próprio jogo funciona e os conceitos de POO. O momento de customização é onde o jogador irá aplicar os conceitos de POO ao configurar sua primeira classe e mecha (Figura 10). Já no combate, os jogadores enfrentarão outros mechas carismáticos, que irão desafiar as habilidades do jogador. Esta parte do jogo foi adicionada com o intuito de dar um objetivo a toda customização do robô e, além disso, prover um momento de relaxamento e entretenimento, onde os conteúdos de POO não estão sendo cobrados. Dessa forma, o jogo possui um fluxo pensado para apresentar novos conteúdos, incentivar o jogador a pô-los em prática e divertí-los e relaxá-los, preparando-os para repetir o ciclo, evitando que o este se torne massante.

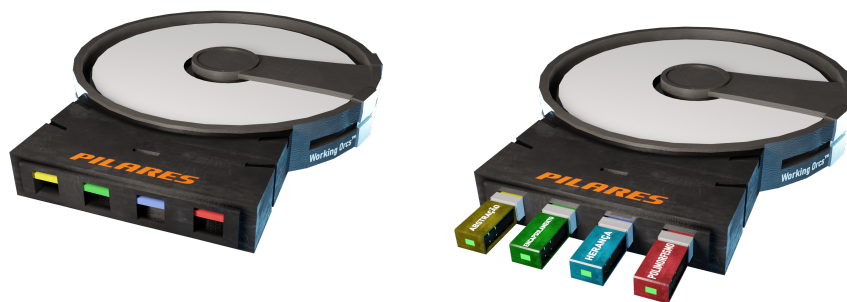
Figura 11 - Interface de Combate do “Mechas P.O.M.!”



Fonte: Produção Própria

Para configurar seus mechas, os jogadores irão construir sua primeira classe, que é representada, dentro do jogo, pelo disco da Figura 11. Esta pode ser nomeada pelo jogador e irá receber, conforme o progresso do jogador, pen drives, cada um contendo um dos 4 Pilares da Programação Orientada a Objetos (Figura 12). Dessa forma, o jogador irá configurar esta classe e inseri-la em seu mecha, que são a representação de objetos dentro do jogo.

Figura 12 - Interface sem (esquerda) e com os Pen-drives (direita) correspondentes aos 4 pilares da Orientação a Objetos



Fonte: Produção Própria.

Ao começar o jogo, o jogador pode configurar esta classe, jogadores podem alterar as funções e atributos que o mecha possui. Além de poder trocar suas peças de braços e chassi, que possuem atributos próprios. Como demonstra a imagem X, esta é a adaptação do jogo dos conceitos básicos de classes e objetos.

A abstração, é o pilar que define, para uma classe, quais aspectos são relevantes e quais não são, além de comprimir as informações não relevantes. Assim sendo, em P.O.M.! os jogadores irão receber pseudocódigos que podem realizar ações e estas devem ser agrupadas por eles em funções que eles mesmos nomeiam. Além, claro, de excluir códigos e funções que não são úteis para o contexto de combate do jogo. Um exemplo: para compor a função `SocoRapido()`, o jogador precisa dos códigos que identificam o input do jogador, preparam o ataque, realizam o ataque e atualizam o estado do mecha. Dessa forma, ao agrupar estes pseudo códigos dentro de uma única função, não é relevante como esta funciona em todas as suas etapas, mas sim que, quando o input é pressionado, o mecha realizará um ataque rápido.

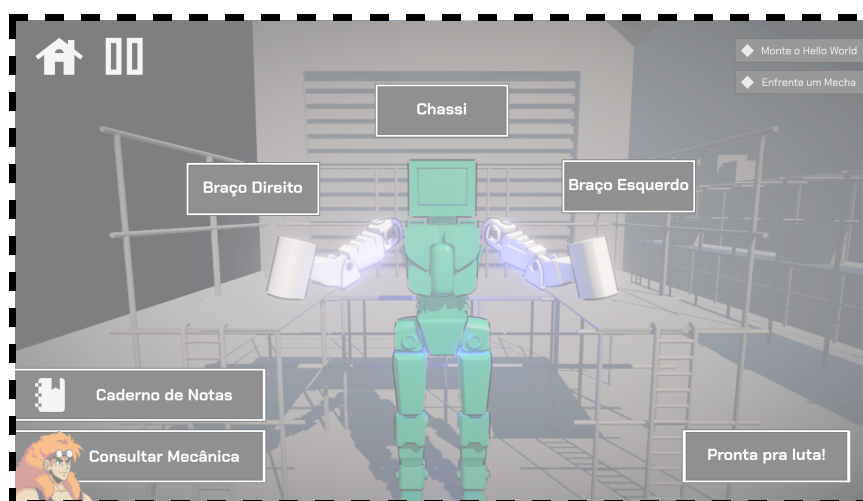
O pilar do encapsulamento se baseia em proteger os atributos da classe, para que estes não sejam acessados ou alterados de forma incorreta por funções que precisam acessá-las. Dessa forma, o jogador possui espaços extras de funções que servirão para proteger suas variáveis e serem utilizados por outras funções que necessitem do acesso a estas.

Na orientação a objetos, entende-se herança pela capacidade de uma classe herdar todo o conteúdo de outra classe. Esta é útil pois, uma vez que uma classe é um molde de objetos que possuem as mesmas características, com ela é possível importar este molde e adicionar novos moldes para representar objetos que possuem características além destes

moldes. Assim sendo, em P.O.M.! os jogadores podem fazer sua classe herdar de outras classes, que irão dar novas funcionalidades para seu mecha.

O conceito de polimorfismo é aplicado juntamente da herança, uma vez que este trata-se de alterar atributos ou funções que são herdadas de outras classes para realizar novas ações, alterar valores ou retirar ações. Dessa forma, no jogo os jogadores podem alterar funções herdadas adicionando novas funcionalidades, como dano extra ou outros efeitos.

Figura 13 - Interface de Customização do “P.O.M.!”



Fonte: Produção Própria.

4. TESTES e RESULTADOS

Após o desenvolvimento de uma versão estável do jogo, foram realizados testes com este protótipo para verificar a qualidade do jogo em jogabilidade, facilidade de uso e como ferramenta de aprendizado.

O grupo com o qual realizamos estes primeiros testes foi relacionado com o público alvo da nossa pesquisa. Buscamos alunos que estavam cursando as disciplinas de 2º e 3º Trabalho de Conclusão de Período, pois são alunos que, normalmente, cursaram ou estão a cursar também a matéria de Programação para Jogos 3D com Engines que aborda os conceitos da POO.

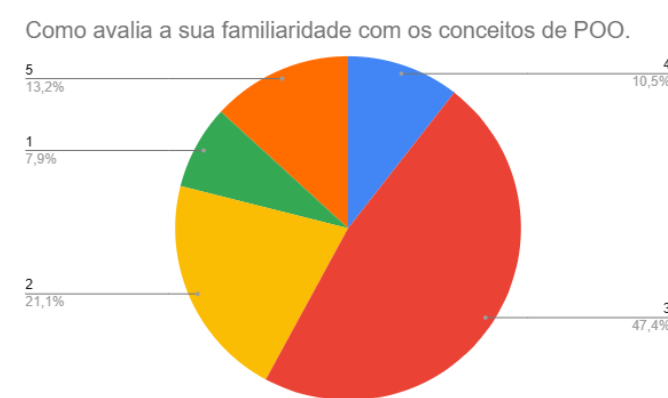
Nossa metodologia de testes foi baseada em 4 etapas: **1.** Explicação para o grupo de testes sobre o jogo e o projeto (realizado por meio de uma videoconferência), **2.** O preenchimento de um formulário online pré-jogo, **3.** A jogatina de aproximadamente 20 minutos do protótipo do jogo e, por último **4.** O preenchimento de um formulário pós-jogo.

Na primeira etapa, membros da pesquisa e do time de desenvolvimento compartilharam com os *playtesters* as informações gerais do projeto, uma sinopse do jogo e uma explicação das etapas posteriores e como eles deveriam prosseguir.

Já na segunda etapa, todos os testadores preencheram um formulário anônimo no qual eles só se destacavam através de um código aleatório composto de um número de 4 dígitos +

uma letra (ex: 1234A), este que deveria ser o mesmo utilizado posteriormente no segundo formulário, para assim podermos traçar dados relacionais e ainda manter a anonimidade. Neste formulário pré-jogo havia perguntas sobre os temas do jogo e da pesquisa como “Como avalia a sua familiaridade com os conceitos de POO.” e “O que é uma Classe?” ou “O que é um Objeto?” que foram repetidas na última etapa. Abaixo na Figura 14, observe os dados da pergunta sobre familiaridade.

Figura 14 - Formulário de Testes - Pergunta sobre os Conceitos da POO



Fonte: Produção Própria.

Na terceira etapa os alunos jogavam o jogo enquanto, se possível, compartilhavam suas telas (Figura 15). Não foi feito registro dessas capturas de telas, mas serviu para que os desenvolvedores pudessem notar quaisquer comportamentos inesperados ou erros durante a jogatina que, talvez, não fosse apreendido pelos jogadores que ainda não conheciam o jogo e nem sabiam, intencionalmente, quais os comportamentos deveriam esperar.

Figura 15 - Múltiplas Transmissões de Tela Durante os Testes

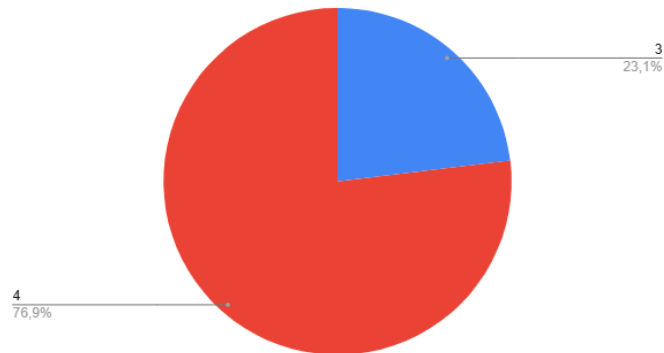


Fonte: Produção Própria.

Por fim, na quarta etapa deste nosso método, enviamos o link do segundo formulário a ser preenchido. Aqui, além das perguntas do primeiro formulário, pedimos opiniões, geralmente com respostas em uma escala quantitativa de 1 à 5, sobre o aprendizado do ciclo de jogabilidade (Figura 16), organização do conteúdo (Figura 17) e valor educacional (Figura 18). Além de um espaço para relatos de *bugs*, erros e sugestões de melhorias na experiência do “P.O.M.!” (Figura 19).

Figura 16 - Pergunta sobre o aprendizado do jogo.

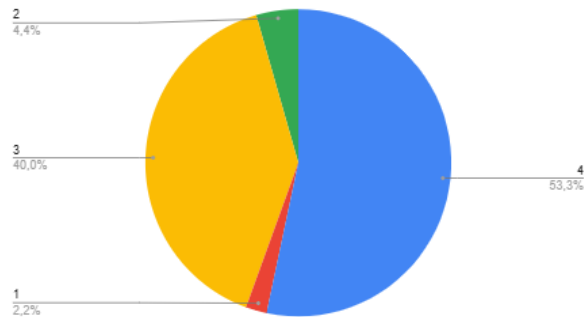
Aprender a jogar o jogo foi fácil?



Fonte: Produção Própria

Figura 17 - Pergunta sobre a organização do conteúdo.

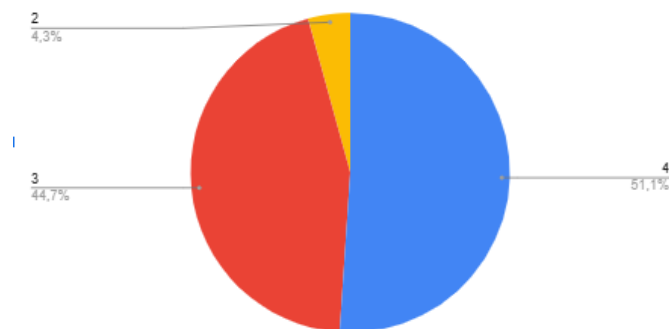
A organização do conteúdo ajudou?



Fonte: Produção Própria

Figura 18 - Pergunta sobre o valor de aprendizado.

Acredita que o jogo contribuiu para a sua aprendizagem na disciplina?



Fonte: Produção Própria

Figura 19 - Questões para inserir relatos da experiência com o jogo.

Você teve algum problema ou bug enquanto jogava?

14 respostas

Fonte: Produção Própria

5- CONCLUSÕES OU CONSIDERAÇÕES FINAIS

Neste estudo, investigamos a dificuldade de estudantes do Curso Superior de Tecnologia em Jogos Digitais na assimilação dos conceitos de Programação Orientada à Objetos (POO). E por meio do desenvolvimento do jogo P.O.M.! (Programação Orientada à Mechas!), criamos uma abordagem única para facilitar a compreensão dos pilares da POO. Os testes realizados com o protótipo mostraram que, apesar de erros técnicos a serem corrigidos e um refinamento de certas abordagens para maior eficiência educativa do projeto, a proposta é cativante e gera interesse dos alunos das disciplinas de programação. O que demonstra a validade e importância de projetos de Jogos Educativos como este, abrindo espaço e sendo um caso exemplo para futuras investigações sobre a aplicação de desse tipo de jogo em diferentes contextos do ensino, em especial, em disciplinas e cursos que envolvam programação, computação ou jogos, já que seus alunos possuem uma relação com as mídias interativas que torna estes jogos uma interessante ferramenta de ensino.

REFERÊNCIAS BIBLIOGRÁFICAS

STROUSTRUP, Bjarne. **What is object-oriented programming? (1991 revised version)**. 1991. Disponível em: <https://www.stroustrup.com/whatis.pdf>. Acesso em: 18 nov. 2024.

WEGNER, Peter. **Concepts and paradigms of object-oriented programming**. ACM SIGPLAN OOPS Messenger, v. 1, n. 1, p. 7–87, 1 ago. 1990. Disponível em <https://doi.org/10.1145/382192.383004> . Acesso em: 18 nov. 2024.

EPIC GAMES. **Unreal Engine Terminology**. Disponível em: <https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-terminology#object>. Acesso em: 18 nov. 2024.

UNITY TECHNOLOGIES. **Unity - Manual: Object-oriented development**. Disponível em: <https://docs.unity3d.com/6000.0/Documentation/Manual/object-oriented-development.html>. Acesso em: 18 nov. 2024.

Applying object-oriented principles in Godot. Disponível em: https://docs.godotengine.org/en/stable/tutorials/best_practices/what_are_godot_classes.html. Acesso em: 18 nov. 2024.

FIGUEIREDO, K. d. S., RIBEIRO, J. M., Souza, R., ANGELO, V.R. Uma abordagem gamificada para o ensino de programação orientada a objetos. WEI - XXIII Workshop sobre educação em Computação, 2015.

JUVRUD, Joshua. et al. **Game or watch: The effect of interactivity on arousal and engagement in video game media**. IEEE Transactions on Games, p. 1–1, 2021.

NAUL, Emily, e MIN, Liu. “**Why Story Matters: A Review of Narrative in Serious Games**”. Journal of Educational Computing Research, vol. 58, no 3, junho de 2020, p. 687–707. Disponível em: <https://doi.org/10.1177/0735633119859904>. Acesso em: 18 nov. 2024.

MOMBACH, Jaline G.; CASTRO, Bruno M. N. de; SANTOS, Eduardo S.; SANTOS, Matheus B. B. **POOkémon: um jogo sobre programação orientada a objetos**. 2018 Disponível em: <https://www.sbgames.org/sbgames2018/files/papers/EducacaoShort/188359.pdf>. Acesso em: 18 nov. 2024.

MOMBACH, Jaline G.; CASTRO, Bruno M. N. de; SANTOS, Eduardo S.; SANTOS, Matheus B. B. **POOkémon: um jogo sobre programação orientada a objetos**. *Revista Brasileira de Tecnologia e Ensino*, Brasília, 2018.

DA SILVA, Leuson. et al. **POOGame: Um Jogo Sério para o Ensino de Programação Orientada a Objetos**. WEI - 24º Workshop sobre Educação em Computação, 2016, p. 2333-2342. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/9677/9578>. Acesso em: 18 nov. 2024.

MENDES, Thiago G. **Jogos Digitais como Objetos de Aprendizagem: Apontamentos para uma Metodologia de Desenvolvimento**. X SBGames - Salvador - BA, p. 01-08, 2011. Disponível em:
<http://www.sbgames.org/sbgames2011/proceedings/sbgames/papers/art/full/92067.pdf>. Acesso em: 18 nov. 2024.

DÖRNER, R. et al. (EDS.). **Serious Games: Foundations, Concepts and Practice**. Cham: Springer International Publishing, 2016.

PARR, Kealan. **The Four Pillars of Object-Oriented Programming**. freeCodeCamp. 2020 Disponível em:
<https://www.freecodecamp.org/news/four-pillars-of-object-oriented-programming/> Acesso em 05 nov. 2024.

DAL MOLIN, Felipe. **As Cores do Game Design E se o design de jogos pudesse ser resumido em quatro cores primárias e suas misturas?**. UX Collective, 2023. Disponível em:
<https://brasil.uxdesign.cc/as-cores-do-game-design-56085af672ea> Acesso em: 12 nov. 2024.

LEITE, M. **Técnicas de programação: uma abordagem moderna**. Brasport, Rio de Janeiro, 2006; Disponível em:
<https://books.google.com.br/books?hl=pt-BR&lr=&id=H8A0lZtqaXsC&oi=fnd&pg=PA1&dq=LEITE,+M.+T%C3%A9cnicas+de+programa%C3%A7%C3%A3o:+uma+abordagem+moderna&ots=bGSHGqoBOG&sig=dBRCN2ruxkshypJrhj9Re4-mMGo#v=onepage&q=orienta%C3%A7%C3%A3o&f=false> Acesso em 4 fev 2025

CARVALHO, Gabriel Rios de. **A importância dos jogos digitais na educação**. 2018. 41f. Trabalho de Conclusão de Curso (Graduação em Curso de Tecnologia em Sistemas de Computação) - Universidade Federal Fluminense, Escola de Engenharia, Niterói, 2018.