

Minigames para o desenvolvimento do pensamento computacional

Luciana P. de Araújo Kohler, Dalton Solano dos Reis, Mauricio Capobianco Lopes, Gabriel Jorge Utyama de Carvalho, Umberto Oliveira de Araújo Neto Leonetti, Leonardo Linhares Sila, Felipe Augusto de Carvalho de Araujo, Artur Ricardo Bizon

¹Departamento de Sistemas e Computação (DSC)
Universidade Regional de Blumenau (FURB)
Blumenau – SC – Brasil.

{Lpa, dalton, mclopes, gjucarvalho, uleonetti}@furb.br

{leonardolinhaires, felaraujo}@furb.br, arbizon@inf.ufpr.br

Abstract. *Considering the technology evolution for education, the Computational Thinking (CT) as a curriculum component in the Basic Education, as the growing use of games by children and youngs, this paper presents a minigame collection that can be used to teach some knowledge areas that have one of the pillars of CT in a ludic way. For the game design it was used the SSM Framework that defines that all game have three layers, System, Story and Mental Model. After the development, it was applied an pre-test with some students with purpose to validate the initial solution for in other time do an another test with a class of Basic Education.*

Keywords— computational thinking, SSM framework, education

Resumo. *Considerando a evolução das tecnologias para o ensino, a inclusão do Pensamento Computacional (PC) como um componente curricular da Educação Básica, bem como o crescente uso de jogos por crianças e jovens, este trabalho apresenta uma coleção de minigames voltado ao ensino de diversas áreas do saber que contém algum dos pilares do PC tornando o aprendizado lúdico. Para o game design dos minigames utilizou-se o framework SSM que define que todo o jogo possui três camadas, Sistema, História e Modelo Mental. Após o desenvolvimento dos três minigames, aplicou-se um pré-teste com alguns alunos com o objetivo de validar a solução inicial concebida, para que posteriormente sejam realizados testes com turmas da educação básica.*

Palavras-chave— pensamento computacional, framework SSM, educação

1. Introdução

A evolução dos meios de comunicação e o frequente desenvolvimento de novas técnicas de ensino e aprendizagem, intensificaram a necessidade de estreitar a relação entre educação e tecnologia de modo a favorecer a relação aluno-docente [Prestes et al. 2019][Gil e Toscano 2019]. Coincidentemente a isso, surge como alternativa de ensino a utilização de jogos eletrônicos, no formato de *minigames*, para facilitar a compreensão de alunos, sobretudo nas duas fases principais do Ensino Fundamental. O

aproveitamento desse recurso tecnológico é capaz de proporcionar aos alunos uma experiência mais leve e produtiva no ambiente escolar, servindo como suporte de ensino aos professores, podendo ser empregado interdisciplinarmente [Gil e Toscano 2019].

A utilização inadequada de jogos digitais nas escolas, acabam se tornando meios de educação limitantes, atrapalhando mais do que ajudando. Essa limitação deve-se a diversos fatores, sendo os principais: falta de recursos nas instituições de ensino, falta de orientação e formação de professores sobre como eles podem utilizar tais ferramentas para aprimorar a aprendizagem dos seus alunos, bem como a falta de prática e, por último, a falta de acesso à escola [de Oliveira et al. 2019]. [Birmingham e Davies 2001] enfatizam a importância da figura do professor como um ator decisivo na correta aplicação das ferramentas disponíveis de ensino, em outras palavras, a figura do professor potencializa a efetividade das ferramentas de ensino. Em contrapartida, os jogos devem ser adequados ao assunto e à faixa etária dos alunos para os quais se pretende aplicá-los, uma vez que um jogo com dificuldade excessiva ou que contém diversos elementos desnecessários podem causar confusão, descumprindo seu papel como recurso educacional [Kirriemuir e Mcfarlane 2004].

Dessa forma, o presente trabalho tem como objetivo desenvolver uma coleção de *minigames* digitais voltados para a aprendizagem com foco em promover a utilização do pensamento lógico e computacional nos alunos, tornando o contato com a tecnologia algo aprazível e educativo. Essa coleção conta com três jogos, sendo que cada um tem um propósito diferenciado.

O artigo segue dividido da seguinte forma. A seção 2 descreve conceitos utilizados para a elaboração deste trabalho, como a Aprendizagem Baseada em Jogos Digitais e o Pensamento Computacional. A seção 3 relata como ocorreu o desenvolvimento dos *minigames* apresentados neste trabalho. A seção 4 descreve cada um dos *minigames* relacionando-os ao temas da fundamentação teórica e da metodologia. Ao final desta seção é relatado um pré-teste executado com um pequeno grupo de alunos. Por fim, na seção 5 são apresentadas as discussões finais deste trabalho.

2. Fundamentação Teórica

No que tange à relação ensino-aprendizagem, são várias as considerações a serem feitas. Para [Belhot 1995], a aprendizagem é uma consequência do ensino. O ensino, por sua vez, não deve ser unilateral; deve ser abrangente e acolhedor, adaptando-se à realidade do aluno e suas dificuldades, buscando a inclusão e a promoção de conhecimento. Neste campo, é fundamental citar a adoção da tecnologia como técnica de ensino e aprendizagem [de Oliveira et al. 2019][Prensky 2008]. Por conta disso, atualmente, a existência de laboratórios de informática e computadores disponíveis já é realidade em algumas escolas, sendo utilizados com o intuito de facilitar a compreensão e tornar a experiência escolar mais simples [de Castro Santos 2001].

Uma das abordagens possíveis para o ensino e aprendizagem é a adoção de jogos que promovem a compreensão e o raciocínio lógico, principalmente para o público infantil. Com o desenvolvimento de jogos de computador, essa relação pode ir além: o emprego de jogos digitais como maneira de ensino e obtenção de conhecimento. Para [Fernandes 2010], a utilização de jogos educacionais como método de aprendizagem pode ser útil para que, além de ter um aprendizado mais prazeroso por parte dos alu-

nos, a tecnologia sirva de auxílio para o processo de ensino exercido pelo professor. Além disso, vale destacar que essa utilização também se torna interessante por conta da sua interdisciplinaridade, uma vez que os jogos podem ser desenvolvidos e aplicados para qualquer área do conhecimento, desde que haja o estudo e planejamento prévio [Fernandes 2010][Prensky 2008].

Contudo, mesmo diante deste contexto evolutivo, o acesso à tecnologia e, consequentemente, o contato com jogos eletrônicos pode ser uma realidade distante para alguns indivíduos [de Oliveira et al. 2019]. Essa falta de acessibilidade possui diversos motivos. Em alguns casos, as escolas não dispõem da infraestrutura necessária para propiciar aos alunos o acesso a este tipo de tecnologia; em outros casos, há a disponibilização das máquinas, porém não há a formação necessária de professores para que o ensino de informática possa ser repassado aos alunos e, em casos mais extremos, há a disposição de máquinas e a formação necessária dos docentes, mas o acesso à escola não é algo existente na vida do aluno [de Oliveira et al. 2019].

Outra limitação existente no uso de jogos como instrumento de aprendizado é o caráter e o nível de dificuldade existentes. Mesmo que haja recursos e a devida capacitação dos professores, a obtenção de conhecimento não será factível se o jogo tiver alto nível de dificuldade. Ao contrário, pode prejudicar o desenvolvimento do aluno e, inclusive, criar pré concepções equivocadas sobre a tecnologia. Neste sentido, torna-se válida a criação de *minigames*; jogos consideravelmente curtos, que geralmente compreendem fases breves com intuito bem definido e nível baixo de dificuldade [Prensky 2010]. Por esses fatores, os *minigames* possuem alta aplicabilidade como recurso para o ensino.

2.1. Aprendizagem Baseada em Jogos Digitais

Segundo [Prensky 2010], com a Aprendizagem Baseada em Jogos Digitais, do inglês *Digital Game Based Learning* (DGBL), é possível combinar computador e videogame com qualquer conteúdo educacional, alcançando resultados até melhores do que o método tradicional de ensino. Para [Chang e Hwang 2019], os DGBLs incorporam conteúdo educacional ou design de aprendizagem em jogos digitais. Esses jogos podem envolver os alunos em atividades simples ou complexas, como auxiliar nas habilidades de resolução de problemas [Deubel 2006].

Para o jogo se enquadrar em um DGBL ele deve permitir que o jogador pratique as habilidades e receba *feedbacks* indicando os acertos e erros, para que assim ele possa aprender sobre determinado tema enquanto joga, isto é, aprendizagem por descoberta [Prensky 2010][Chang e Hwang 2019]. Nesses jogos há um grande nível de engajamento por parte do jogador, sendo este um usuário ativo.

2.2. Pensamento computacional

O termo Pensamento Computacional (PC), introduzido em [Wing 2006], é um método para ensinar a solucionar problemas, conceber sistemas e compreender o comportamento humano inspirando-se em conceitos da Ciência da Computação ou, como colocado por [de Paula et al. 2014], “é uma maneira específica de se pensar e de analisar uma situação ou um artefato, sendo independente do uso da tecnologia”.

Nos últimos anos, a Sociedade Brasileira de Computação (SBC), junto com o Centro de Inovação para a Educação Brasileira (CIEB), vem estudando formas de introduzir

o pensamento computacional ainda na matriz curricular da educação básica. Para isso, o PC foi dividido em quatro pilares, sendo eles: reconhecimento de padrões, decomposição, algoritmos e abstração. O reconhecimento de padrões auxilia na identificação de aspectos comuns em um processo. Como exemplo, pode-se citar o diagnóstico de uma doença baseada nos sintomas presentes no paciente ou a identificação de frutas baseada em suas cores. A decomposição refere-se à divisão de um problema em partes menores para que assim seja mais fácil resolvê-lo. Os algoritmos representam uma sequência de passos que deve ser executada para alcançar a solução do problema. Por fim, a abstração se refere à filtragem da informação permitindo que seja trabalhado somente com os elementos essenciais para a resolução do problema. Um exemplo da abstração é um desenho de um carro que possui os elementos básicos como quatro rodas e a sua carcaça. Não é necessário desenhar as portas, janelas, trincos, entre outros para identificar que o desenho representa um carro.

3. Metodologia

Para o game design dos minigames foi utilizado o *framework* SSM difundido por [Grip 2017] que define que todos os jogos são divididos em três dimensões, sendo elas: Sistema, História e Modelo Mental (*System, Story, Mental Model*). Segundo [Grip 2017], o SSM separa a compreensão do jogador do jogo do seu comportamento real no jogo, sendo fácil definir e discutir como será a história do jogo em relação a *gameplay*, fornecendo várias vias de ataques e formas para resolvê-las.

A dimensão Sistema diz respeito ao espaço que fica todo o código-fonte existente e no qual ocorre a simulação. É nesta dimensão que as coisas de fato acontecem, concentrando-se somente na funcionalidade. Essa dimensão é formada pelas camadas “mecânicas” e “dinâmicas de forma abstrata”, sem levar em conta como estas são interpretadas pelos jogadores.

A dimensão História contextualiza as coisas que acontecem no espaço do sistema. Neste caso, a história pode ser densa na qual é apresentada por meio de *cutscenes* sem interação por um longo período de tempo ou curtas e sem história bem definida, como o exemplo do jogo Tetris. Essa dimensão é formada pelas camadas *Bottom Layer* que define todos os elementos que o jogador pode ver, ouvir ou que alcance seus órgãos sensoriais; e *Layer Above* que retrata o drama da história, gerado pela *Bottom Layer*.

Por fim, a dimensão Modelo Mental une as duas dimensões anteriores trazendo a experiência pessoal do jogador no jogo. Ela deriva do que acontece nas mecânicas do jogo, porém tem seu próprio espaço separado. Quando ocorre a vivência entre o espaço do sistema e o da história, tem-se o modelo mental formalizado. Às vezes, este espaço nem sempre contém coisas que vem do sistema ou da história, pode conter coisas que o jogador inventa de sua própria imaginação, sendo que o jogo pode ser projetado para que isso ocorra ou não. Essa dimensão é formada pelas camadas: *Affordance* que diz respeito às funcionalidades que o jogador atribui ao perceber os objetos, como por exemplo uma porta se abrindo; *Schema* que representa como o jogador pensa que deveria se comportar em várias situações e como essas situações deveriam acontecer.

Como limitações deste *framework* tem-se que fica em aberto para que o jogador conclua a respeito de seus sentimentos e percepções em relação ao jogo, sem necessariamente ser algo previamente estruturado ou pensado. Além disso, não são definidas as

questões visuais ou gráficas do jogo.

A Unity foi utilizada como ferramenta para o desenvolvimento dos *minigames*, utilizando a linguagem C. Ela é uma *Engine* gráfica utilizada para o desenvolvimento de jogos, tanto 2D quanto 3D. Quando comparada a outras plataformas de desenvolvimento de jogos, a Unity se destaca tanto por sua versatilidade e praticidade, quanto por ter características que o fazem ser simples. Além disso, com a Unity é possível compilar o mesmo código-fonte para diversas plataformas, como para computadores com os sistemas operacionais Windows, Linux, MacOS, para dispositivos móveis Android e iOS, para consoles como Xbox One, Nintendo Switch e Playstation, além de ser possível compilar para web com o WebGL.

Ainda, para cada *minigame* desenvolvido elaborou-se o diagrama de estrutural de como será o projeto dentro da Unity. Nesse diagrama definiu-se: os *scripts*, podendo ser do tipo *singleton*, interface ou *scriptable object*; os *GameObjects*, podendo ser *prefabs* ou não; os *components*; e seus relacionamentos. Cada diagrama estrutural foi aprovado previamente pelo analista da equipe de desenvolvimento para que então seguisse para a produção. Já o layout dos *minigames* foram previamente elaborados pela equipe de designers e aprovados pelos diversos integrantes da equipe (desenvolvedores, líderes e analistas), para que então o processo de desenvolvimento do *minigame* fosse iniciado.

Para validar os *minigames* desenvolvidos, foi realizado um pré-teste em relação a usabilidade e jogabilidade com um pequeno grupo de quatro estudantes da faixa etária alvo. Após este pré-teste e as correções a serem realizadas, pretende-se desenvolver um teste maior com professores e alunos de uma turma de ensino fundamental.

4. Resultados alcançados

A partir de pesquisas previamente realizadas e da fundamentação teórica, concluiu-se que o incentivo ao pensamento computacional deveria ser mais atrativo, uma vez que se trata de um público majoritariamente infantil. Com isso, houve o questionamento: como desenvolver um jogo que, além de divertido e didático, possa induzir o pensamento computacional em crianças e adolescentes de maneira fluida e natural?

Iniciou-se então o desenvolvimento de diversos *minigames* com o propósito de ensinar algum dos pilares do Pensamento Computacional, além de introduzir algum conhecimento geral para atrair professores e alunos. Assim, o pensamento computacional foi abordado de maneira indireta e disfarçada por meio de jogos que também fazem ponte com outras áreas de conhecimento. Identificação de animais, quebra cabeça e jogos de combinação, em ambientes com design atrativos, contribuem para o aprendizado e interesse do público alvo.

Para o desenvolvimento do jogo, optou-se por utilizar o modo 2D disponibilizado pela plataforma e, até o presente momento, os jogos foram compilados para Windows, Android e WebGL.

4.1. Minigame 1: encontre o animal na Amazônia

A partir das características de animais da Amazônia com o desenhos desses animais, o jogo “encontre o animal na Amazônia” foi pensado com o objetivo de proporcionar a interdisciplinaridade entre a área da computação e a fauna e flora brasileira, fazendo com

que os usuários possam obter conhecimento acerca dos animais da Amazônia e, também, sobre o habitat de cada um deles.

Conforme especificado na metodologia, o *framework* SSM foi aplicado para a concepção do jogo, sendo neste caso definido cada um dos itens conforme segue:

- Sistema: como mecânicas do jogo tem-se que o jogador deve encontrar um animal presente no cenário por meio de uma dica que lhe é dada, para isso, ele possui um certo tempo. Como dinâmica, caso ele acerte dentro deste tempo, ele receberá pontos. Caso erre ou acerte, uma nova dica e um novo cenário com outros animais aparecerão.
- História: a história deste jogo é praticamente ausente, sendo que o jogador poderá imaginá-la por meio do cenário visual que deverá apresentar diversos animais da floresta amazônica em um espaço harmonioso.
- Modelo Mental: o jogador deve perceber que assim que acertar um animal com base na dica, o cenário será alterado para novos animais com nova dica. Assim, espera-se que ele procure o novo animal para receber mais pontos. Quanto mais animais acertar dentro do tempo, mais pontos receberá. Para auxiliar o jogador em relação a sua assertividade, um determinado som é executado quando o jogador acerta o animal e outro é executado quando o jogador erra.

Como cenário do jogo, o jogador é introduzido a um espaço amplo e colorido, baseado na Floresta Amazônica como pode-se observar na 1. A mecânica do jogo consiste em o jogador selecionar um entre os animais escondidos no cenário, baseado em uma dica que recebe a respeito do animal. Na 1 esta dica localiza-se no canto superior esquerdo. Ao clicar no animal correto, o jogador recebe pontos (localizado no canto superior direito) e, em seguida, uma nova dica e um novo cenário com outros animais é redesenhado na interface. Assim, o aluno é instigado a usar o seu pensamento lógico e computacional para que seja possível relacionar as dicas com os respectivos animais. Além disso, desenvolveu-se no jogo um sistema de pontuação com tempo estimado para completar o jogo. Esse sistema consiste na contagem de pontos a cada acerto do aluno, servindo como motivação para que encontre o maior número de animais no menor período de tempo. Seguindo os princípios de jogos DGBLs, sempre que o jogador acertar o animal ele recebe um *feedback* visual e auditivo, pois o cenário é renderizado e recebe pontos. Caso o jogador erre o animal, ele perde pontos e precisa tentar novamente identificar o animal correto, trabalhando assim a aprendizagem por tentativa.



Figura 1. Cenário do jogo da Amazônia

Em relação ao Pensamento Computacional, este jogo trabalha os pilares da abstração e o reconhecimento de padrões. A abstração é contemplada nos desenhos dos animais

que mostram suas características de forma minimalista. Já o reconhecimento de padrões está relacionado à interpretação da dica que se referencia a algum dos animais presentes no cenário.

4.2. Minigame 2: a pesca das cores

O jogo a pesca das cores foi concebido com o objetivo de ensinar como uma determinada cor se forma. Conforme especificado na metodologia, o *framework* SSM foi aplicado para a concepção do jogo, sendo neste caso definido cada um dos itens conforme segue:

- Sistema: como mecânicas do jogo tem-se que o jogador deve pescar dois peixes com coloração distinta para formar a cor do terceiro peixe que aparecerá na tela. Para realizar a pesca, o jogador irá interagir com a vara de pescar por meio do mouse, sendo utilizada a coordenação motora fina. Como dinâmica do jogo, caso o jogador acerte as cores receberá pontos. Sempre que pescar dois peixes corretamente, um novo peixe aparecerá para que a pesca seja iniciada novamente. Isso acontecerá até o tempo do jogo esgotar.
- História: a história deste jogo é praticamente ausente, sendo que o jogador poderá imaginá-la por meio do cenário visual existente.
- Modelo Mental: o jogador deve perceber que ao pescar os peixes que compõem a terceira cor, um novo problema de coloração será apresentado. Também deverá perceber como o mouse ou o toque interage com a vara de pesca, para que possa pescar peixes que se encontram mais ao fundo ou mais a superfície.

Assim, como mecânica de jogo tem-se a pesca de dois peixes de cores distintas que, combinadas, formam uma terceira cor. Previamente, ao iniciar o jogo, o jogador visualiza a cor do peixe que será formado por meio da pescaria, por exemplo, um peixe verde como o exemplo da 2 que o peixe é representado no canto superior direito. Nesse caso, vários peixes de cores distintas passarão na tela do jogador e ele terá que pescar os peixes com as colorações amarela e azul para que, assim, forme a cor verde conforme indicado no início da partida. Cada vez que o jogador conseguir pescar os peixes que correspondem às cores que formarão a terceira cor, um novo peixe aparece para que seja coletado outros peixes e assim sucessivamente até acabar o tempo da partida. Quanto mais peixes o jogador conseguir coletar, mais pontos ele irá receber, localizado no canto superior esquerdo da 2. Em relação aos *feedbacks* fornecidos para o jogador, seguindo os conceitos do DGBL, além da pontuação, o jogador visualiza o peixe que pescou corretamente no sombreado presente no canto superior direito da tela. Assim, ele sabe qual cor já está correta e quantas estão faltando. Neste jogo, caso o jogador não souber as cores para formar a cor resultante, também poderá aprender com base na tentativa, promovendo assim um novo conhecimento.

Em relação ao Pensamento Computacional, este jogo trabalha o pilar da decomposição, de modo que o jogador deve identificar quais são as cores que compõem a cor resultante para que assim pesque os referidos peixes. Além disso, o jogo relaciona a computação com a disciplina de artes que proporciona aos alunos o estudo das cores.

4.3. Minigame 3: odontogame

O terceiro e último *minigame* foi denominado Odontogame. Este jogo faz o relacionamento da área da Odontologia, tendo como objetivo principal o ensino em saúde bucal

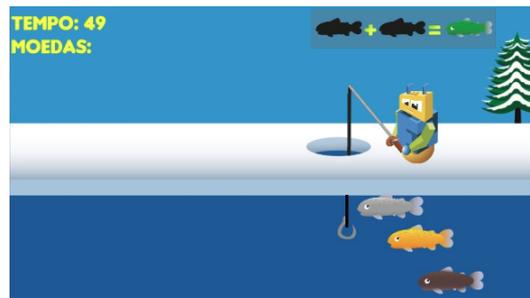


Figura 2. Cenário do jogo da Pesca

vinculado com o pensamento computacional. Conforme especificado na metodologia, o *framework* SSM foi aplicado para a concepção do jogo, sendo neste caso definido cada um dos itens conforme segue:

- Sistema: como mecânicas do jogo tem-se que o jogador deve atirar insígnias que representam símbolos da higiene bucal, como pasta de dente, fio dental e flúor, para combater os *mutans* que são cáries que danificam os dentes. Ao mesmo tempo, os *mutans* tentam eliminar o jogador também atingindo-o com tiros. Como dinâmica, quando o jogador atingir o mutan com as insígnias corretas, este é eliminado da cena. Além disso, existe uma quantidade de *mutans* que devem ser eliminados e uma quantidade de vidas disponíveis para o jogador. Se as vidas dele se esgotarem a fase é reiniciada.
- História: a história é contada antes de iniciar o jogo em si. Ela conta sobre a origem dos *mutans*, o que eles representam, o que são as cáries e como elas devem ser combatidas para que não atinjam os dentes.
- Modelo Mental: o jogador deve perceber que ao eliminar os *mutans* recebe pontos e ao ser eliminado perde vidas.

O jogo Odontogame trabalha com o ensinamento de métodos de prevenção contra cáries. Primeiramente, o aluno deve coletar insígnias que representam produtos de higiene bucal. Posteriormente, as insígnias são utilizadas para eliminar inimigos, representados no formato de pequenos mutantes que, neste caso, são as cáries capazes de danificar os dentes. Desta forma, é possível que o usuário aprenda sobre ensinamentos de saúde bucal, exercendo o raciocínio lógico em um jogo desafiador e divertido, dotado de cores e um cenário interessante para crianças e adolescentes.

No cenário é apresentada a quantidade de vidas e de *mutans* restantes. O jogador deve atirar insígnias para acertar os *mutans*. Os *mutans* também soltam tiros que podem atingir o robô e fazer com que este perca vidas. Para que o robô vença a partida ele deve eliminar todos os *mutans* do cenário. Em relação ao DGBL, o usuário recebe o *feedback* sempre que acertar um *mutan* ou que perder uma vida, sabendo assim o seu status em relação ao jogo.

Em relação ao Pensamento Computacional, este jogo trabalha o pilar do reconhecimento de padrões, uma vez que as insígnias que matam o *mutan* devem ser da sua cor correspondente. Também trabalha levemente a abstração, trazendo os conceitos da higiene bucal para o jogo interativo.

4.4. Design dos *minigames*

O design dos *minigames* foi projetado e desenvolvido tendo um intuito principal de conciliar entretenimento com aprendizagem. Desta forma, o desafio era criar um ambiente colorido e dinâmico, com as informações necessárias para que o aluno pudesse executar todas as fases do jogo com facilidade, porém sem sobrecarregar o jogo com muitos objetos e que poderiam atrapalhar o jogador.

Para a criação dos jogos, estudou-se as paletas de cores, as relações entre os objetos, a passagem de fases e até mesmo a fonte dos textos. No jogo Amazônia, os animais foram representados de acordo com desenhos infantis, buscando a fácil identificação por parte dos alunos. Além disso, buscou-se o uso de fontes que facilitassem a leitura das crianças, criando *sprites* com cores vibrantes. Ainda, é necessário destacar que todos os desenhos desenvolvidos foram totalmente criados pela equipe, visando personalidade e identidade aos jogos. Os desenhos desenvolvidos foram totalmente criados pela equipe, visando personalidade e identidade aos jogos.

4.5. Pré-testes

Os *minigames* desenvolvidos foram testados por um grupo de quatro alunos como um pré-teste, para que posteriormente sejam aplicados com uma turma maior de alunos. Os alunos participantes possuíam entre 11 e 12 anos de idade, jogam no mínimo uma vez por semana, principalmente na plataforma Playstation 4.

Para a realização deste teste, os alunos jogaram os *minigames* supervisionados por especialistas nos jogos e foram orientados quando necessário sobre sua utilização. Após jogarem, solicitou-se para que estes alunos respondessem um questionário de jogabilidade sobre cada um dos jogos. Em relação ao jogo de encontrar um animal na amazônia, todos os participantes gostaram das mecânicas de jogo e acharam muito fácil de compreender as regras do jogo. Como ponto positivo foi mencionado sobre a diversidade dos animais presentes e como ponto negativo que as perguntas ou se repetem ou são muito fáceis. Em relação ao jogo da pesca de cores, os participantes acharam difícil as regras do jogo, pois elas estavam confusas e tiveram que ser orientados a respeito de como jogá-lo. Contudo, após a compreensão, os participantes relataram ter achado as mecânicas envolvidas para a coleta dos peixes e a composição das cores muito boa. Em relação aos pontos positivos mencionaram que gostaram da combinação de cores, entretanto, como ponto negativo mencionaram pouca diversidade de cores no jogo. Em relação ao terceiro jogo, o Odontogame, os jogadores ficaram divididos sendo que dois acharam as regras fáceis e gostaram das mecânicas envolvidas e dois acharam as regras difíceis e não gostaram das mecânicas apresentadas. Como melhorias, todos sugeriram que os tiros poderiam ser mais rápidos e como pontos positivos foi identificado o propósito do jogo.

5. Discussões

Conforme os objetivos definidos, foram desenvolvidos três *minigames* utilizando a Engine Unity com a linguagem de programação C#. Pode-se dizer que os *minigames* cumpriram com as expectativas iniciais, apresentando a ludicidade e didaticidade esperada. Além disso, a partir do pré-teste efetuado com um grupo reduzido de alunos, foi possível fazer com que os *minigames* apresentassem certa interdisciplinaridade, podendo ser aplicados como ferramenta de ensino-aprendizagem em escolas para realização de um teste efetivo.

Assim, observou-se, ao final do projeto, que os jogos elaborados são capazes de relacionar a lógica computacional com outros âmbitos de maneira simples e intuitiva, que também era um dos propósitos previamente estabelecidos.

Espera-se que seja possível dar continuidade ao trabalho, apresentando os jogos elaborados para crianças e adolescentes com o intuito de polir os jogos, adicionar fases e, se possível, desenvolver mais jogos para compor o projeto, tornando-o mais atrativo para o público alvo.

Referências

- Belhot, R. V. (1995). A informática no ensino. In UFPep, editor, *Anais do XXIII Congresso Brasileiro de Ensino de Engenharia*, pages 533–542.
- Birmingham, P. e Davies, C. (2001). Storyboarding shakespeare: learners' interactions with storyboard software in the process of understanding difficult literary texts. *Journal of Information Technology for Teacher Education*, 10(3):241–256.
- Chang, C. Y. e Hwang, G. J. (2019). Trends in digital game-based learning in the mobile era: a systematic review of journal publications from 2007 to 2016. *International Journal of Mobile Learning and Organisation*, 13(1):68.
- de Castro Santos, V. L. (2001). O uso do computador em sala de aula: a visão do aluno. Dissertação, Universidade Federal de Santa Catarina.
- de Oliveira, E. D., Fonseca, B. A., e Pavani³, G. S. (2019). Breve avaliação dos laboratórios de informática na educação básica: estudo de caso da rede municipal de jandaia do sul - pr. *Cadernos da Pedagogia*, 13(25):116–128.
- de Paula, B. H., Valente, J. A., e Burn, A. (2014). O uso de jogos digitais para o desenvolvimento do currículo para a educação computacional na Inglaterra. *Currículo sem Fronteiras*, 14(3):46–71.
- Deubel, P. (2006). Game on! *T.H.E. Journal*, 33(6).
- Fernandes, J. C. L. (2010). Educação digital: Utilização dos jogos de computador como ferramenta de auxílio à aprendizagem. *FaSCi-Tech*, 1(3):88–97.
- Gil, H. e Toscano, C. (2019). O contributo da aplicação digital kahoot! nas aprendizagens no 1.º ciclo do ensino básico. In *ESECB - Comunicações em encontros científicos e técnicos*, pages 385–394. Universidade do Minho.
- Grip, T. (2017). The SSM framework of game design. Gamasutra. Disponível em: <https://www.gamedeveloper.com/design/the-ssm-framework-of-game-design>.
- Kirriemuir, J. e Mcfarlane, A. (2004). Literature Review in Games and Learning. A NESTA Futurelab Research report - report 8.
- Premsky, M. (2008). *Digital game-based learning*. Paragon House, Saint Paul, MN.
- Premsky, M. (2010). *Não Me Atrapalhe, Mãe! Eu Estou Aprendendo*. Phorte.
- Prestes, L. P., Bos, A. S., de Castro, P., Zaro, M., e Pizzato, M. (2019). Aplicação e avaliação do moodle como tecnologia complementar na capacitação de professores. *Brazilian Journal of Development*, 5(4):2734–2746.
- Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3):33–35.