

A tool for the introduction of programming and computational thinking with motivation using Virtual Reality

Herminio Paucar Curasma
Dept. Informática
Pontifícia Universidade Católica do Rio
de Janeiro
Rio de Janeiro, Brasil
herminiopaucar@gmail.com

Alberto Barbosa Raposo
Dept. Informática
Pontifícia Universidade Católica do Rio
de Janeiro
Rio de Janeiro, Brasil
abraposo@tecgraf.puc-rio.br

Abstract—The present work proposes the creation of a Virtual Reality tool that allows students to learn basic concepts of programming and computational thinking with the purpose that they enjoy the technology and feel motivated to learn more. The tool is a Visual Programming Language; the algorithms are formed by block-assembly, thereby solving one of the students' main problems, which are "syntax errors". In addition, the tool brings with it a set of level-ordered challenges aimed at teaching students basic principles of programming and logic (sequential programming, repetitive and conditional control structure), where at each level the student will learn the different concepts and behaviors of computational thinking. For the evaluations with the users we counted on the participation of 18 students between 12 and 15 years old coming from two public institutions of Rio de Janeiro. In these evaluations it was also considered to measure the sensation of immersion through Telepresence, Social Presence and Usability.

Keywords— *Computational thinking, virtual reality, programming in education, introduction to programming, programming and games, programming and virtual reality.*

I. INTRODUÇÃO

No nosso cotidiano ouvimos com frequência falar da importância das Tecnologias de Informação e Comunicações (TICs) pelos diversos atores sociais. Alguns estudiosos afirmam que “*nos primeiros decênios do séc. XXI, mais de 80% dos seres Humanos terão acesso ao ciberespaço e se servirão dele quotidianamente*” e que “*as atividades de pesquisa, aprendizagem e de lazer serão virtuais ou comandadas pela economia virtual*” [1]. A educação não pode ficar alheia a essas evidências.

A par do investimento em tecnologia nas escolas, tem sido proporcionada também formação básica em TIC para os professores. “A integração efetiva das TIC na educação tem de ir para além de simplesmente melhorar a eficiência ou acelerar as práticas atuais. As TIC ainda não tiveram um impacto significativo nas abordagens de aprendizagem e de ensino; estão de fato ainda subaproveitadas” [2]. Constatase, no dia-a-dia das escolas, que alguns profissionais da educação são defensores teóricos da integração das TIC nos currículos, todavia as suas práticas educativas andam muitas vezes defasadas de uma real integração das TIC no processo de ensino-aprendizagem.

O desenvolvimento da tecnologia informática tem causado um enorme impacto na educação, especialmente quando se trata da tecnologia da Realidade Virtual (RV), que tem um grande potencial para alterar ideias e métodos de educação. Essa tecnologia tem como base a visualização,

interação homem-máquina e simulação. Uma das características principais da RV é a imersão no mundo virtual, que tem dois ingredientes-chave de experiências inesquecíveis como “*eu estava lá*” e “*eu senti*” [3]. Um estudo feito por [4] onde eles ministram uma aula usando dois ambientes: um de Realidade Virtual e o outro, um laboratório tradicional. Nessa aula eles ensinaram conceitos de engenharia, e os autores concluíram que RV é uma experiência de aprendizagem mais memorável do que demonstrações baseadas em laboratório.

Uma das dificuldades que as pessoas têm quando começam a aprender a programação são os erros de sintaxe. Estudos feitos por [5,6] mostram alguns problemas relevantes no processo de aprendizagem e relacionam a sintaxe das linguagens de programação como um desses problemas. Com a descrição deste problema, *este trabalho propõe uma ferramenta* de realidade virtual que desenvolve os algoritmos com a montagem dos blocos (Drag and drop) e não com a escrita de código fonte, o que permitirá que o aluno se concentre na lógica de programação e no funcionamento das estruturas destinadas ao desenvolvimento dos algoritmos (laços de repetição, estrutura de decisão, entrada/saída de dados, etc.), deixando de lado os erros de sintaxe que poderiam atrapalhar sua aprendizagem. Além disso, o estudante usará o pensamento computacional para a solução dos 10 desafios que a ferramenta forneça.

II. TRABALHOS RELACIONADOS

O trabalho baseou-se nas diferentes ferramentas que existem, que foram categorizadas dependendo da tecnologia que usam: 2D, 3D e Realidade Virtual. As ferramentas 2D oferecem uma interface de saída 2D. Elas são conhecidas por usar imagens que representam os blocos e têm como finalidade executar animações, videogames, movimentações de robôs, etc. Algumas delas trabalham com componentes externos, como robôs. Dentre as ferramentas mais conhecidas nessa categoria, temos *Scratch*, uma linguagem de programação visual projetada pelo MIT-Media Lab. O software ajuda os mais jovens a aprender a pensar criativamente, raciocinar sistematicamente, e trabalhar colaborativamente através do desenho de suas próprias histórias interativas, jogos e animações. Eles usam a montagem de blocos para geração do código. No estudo feito por [7], desenvolveu-se um curso que tinha como objetivo principal o treinamento dos professores, sobre como eles deveriam incorporar o pensamento computacional em suas aulas. Os autores se focaram no desenvolvimento

de jogos usando Scratch. Posteriormente, eles testaram mediante algumas avaliações que consideram: número de scripts, número de blocos, número de variáveis e nível de complexidade do algoritmo. [8] observaram que os usuários que usam Scratch online permitem que outros usuários possam aprender e realizar algumas modificações na aplicação.

Temos a ferramenta MIT App Inventor, que é uma ferramenta de introdução à programação e à criação de aplicações mobile. O App Inventor, da mesma forma que o Scratch, usa a técnica drag and drop para construção do script de código. No estudo feito por [9], eles descrevem a estrutura de um curso que foi chamado MobileDev. Através do curso eles usam a curiosidade dos estudantes para desenvolver aplicações móveis e introduzir com essas aplicações os conceitos de pensamento computacional.

Outra categoria são as ferramentas 3D. Essas ferramentas oferecem uma interface de saída 3D e usam imagens 2D para a montagem de blocos para desenhar o código fonte, mas a execução do código é feita em três dimensões, podendo-se gerar animações, histórias ou videogames. A ferramenta mais conhecida é *Alice*, uma ferramenta desenvolvida pela Carnegie Mellon University para criação e animação de mundos tridimensionais, definida para ser fácil de programar e modificar. A programação em Alice é feita seguindo o paradigma Orientado a Objetos. No estudo feito por [10] os pesquisadores falam sobre a importância de criar aplicações onde os participantes se sintam representados pelos atores da aplicação, isto é, que o participante se sinta imerso dentro da aplicação. Para isso, eles desenvolveram uma aplicação onde os estudantes dão movimentos de dança a um boneco. Os estudantes podem gerar um script de código no qual dão alguns movimentos ao boneco: as mãos, as pernas e a cabeça. Os pesquisadores notaram que os estudantes se sentiram motivados pela tecnologia e gostaram de gerar todas as combinações de movimento. Com isto, os pesquisadores ensinaram aos estudantes conceitos do pensamento computacional como: algoritmos, funções, abstração, paralelismo e debugs.

A terceira categoria são as ferramentas de Realidade Virtual. Dentre essas ferramentas, podemos citar a VEnVi, ela foi construída pelo grupo de pesquisa da universidade CLEMSON. “A VEnVi (*Virtual Environment Interactions*) é uma aplicação na qual os alunos aprendem conceitos de ciência da computação através do processo de coreografia de movimento para um personagem virtual usando uma interface divertida e intuitiva” [3].

Um estudo feito por [3], no qual participaram 54 estudantes, teve como objeto a avaliação dessa ferramenta. Nesse estudo, os participantes tinham que criar uma performance de uma dança para um avatar virtual, isto montando os blocos que formam um script de código. Dos 54 participantes, só 16 participaram em uma experiência imersiva virtual, onde o participante era representado por um avatar (masculino ou feminino) e onde eles observavam o boneco dançar em função do script de código feito. Os pesquisadores verificaram que os participantes gostaram da ferramenta e ela os motivou a aprender mais sobre os conceitos de programação e pensamento computacional.

III. PROJETO DA FERRAMENTA

O objetivo do presente trabalho é desenvolver uma ferramenta que permita ensinar de uma forma introdutória a programação e o pensamento computacional aos estudantes (crianças e jovens), e que elas participem de uma forma divertida e lúdica. A ferramenta é um Compilador Visual de Programação que funciona com a montagem de blocos. Cada bloco apresenta o aluno a uma determinada instrução de código, que pode ser: funções pré-definidas, estruturas de controle repetitivas e condicionais.

Ela apresentará 10 desafios, estes desafios foram obtidos do Blockly Games de Google, consiste em um jogo de labirinto onde o objetivo é chegar de um ponto a outro, evitando alguns obstáculos e fazendo uso das funções que oferece a ferramenta, Estes desafios servirão para que o participante aprenda em cada nível um novo conceito e uma abordagem. Se usa a tecnologia de imersão que provê o Oculus Rift e o Microsoft Kinect para visualizar a execução do Script que dá solução para cada desafio.

Para o desenvolvimento da ferramenta definiu-se primeiro os operadores de produção pré-definidos, usando a Gramática de Disposição de Imagens (GDI). Segundo [11] as produções correspondem a operadores de composição de imagens e os atributos dos símbolos gramaticais apresentam a informação espacial para um elemento de imagem. Cada símbolo gramatical (bloco simples e bloco composto) possui 2 atributos (*male connector* e *female connector*), a sobreposição destes atributos indicam que blocos são adjacentes.

Os *Blocos Simples* estão representados por um retângulo. Este tipo de bloco não possui outros blocos em seu interior, e para nossa ferramenta ele apresentará as funções predefinidas. Na Figura 1(a) mostra-se o bloco simples “avançar”. Os *Blocos Compostos* contêm outros sub-blocos em seu interior. Para nossa ferramenta ele apresentará as estruturas de controle condicionais (if, if-else) e de repetição (while). A Figura 1(b) mostra os blocos compostos “Repetir até” e “se ou se não”.

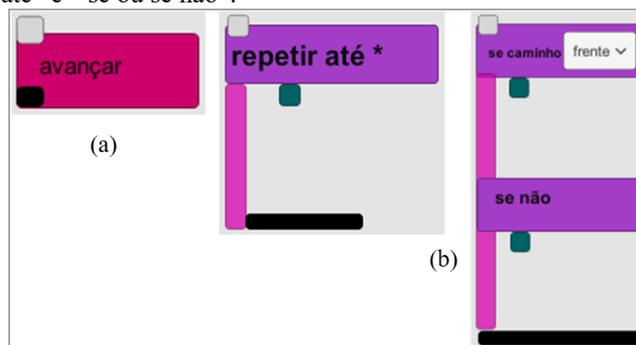


Fig. 1. Exemplo de um bloco simples (a) e compostos (b)

Os dois tipos de símbolo gramatical (bloco simples e composto) têm 2 atributos próprios: “*Male connector*” e “*Female connector*”, que podem ser vistos na Figura 2. O bloco composto além de ter os dois atributos (*male* e *female connector*) tem 1 ou 2 atributos internos que são “*Compound connector*”.

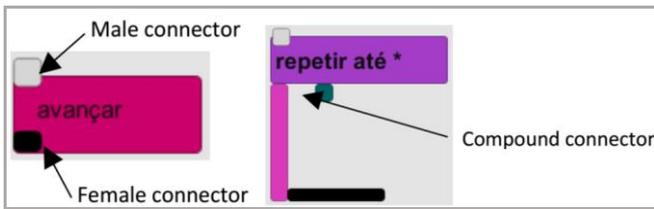


Fig. 2. Blocos com seus atributos “male Connector”, “female connector” e “Compound connector”

Para criar as produções da ferramenta, se criaram operadores de produção predefinidos. O *Operador Next* define que o bloco C segue depois do bloco B (Figura 3). Se declara formalmente o operador next: $A \rightarrow next\{C,B\}$, onde as coordenadas de posição do “male connector” do Bloco C é a mesma que a posição do “female connector” do Bloco B: $C_{maleconnector}(X,Y) == B_{femaleconnector}(X,Y)$. O *Operador Contain* define que o bloco B contém o bloco C (Figura 3). Se declara formalmente o operador contain: $A \rightarrow contain\{B,C\}$, onde as coordenadas de posição do “male connector” do bloco C é a mesma que a posição do “compound connector” de B: $B_{compoundconnector}(X,Y) == C_{maleconnector}(X,Y)$

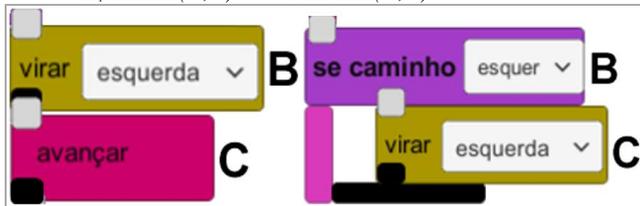


Fig. 3. Operadores de produção next e contain

A. Regras de Produção

Com a produção acima explicada, pode-se começar a criação das regras de produção da ferramenta. Na Figura 4 mostram-se as regras de produção que definem de maneira formal a linguagem aceita pela ferramenta. Os elementos terminais das produções estão escritos em minúscula e os elementos não terminais estão escritos em maiúscula.

1. $START \rightarrow next(STATEMENT, startBlock)$
2. $STATEMENT \rightarrow next(STATEMENT, BLOCK)|BLOCK$
3. $BLOCK \rightarrow turnLeftBlock|turnRightBlock|forwardBlock$
4. $BLOCK \rightarrow REPEAT|IF|IF_ELSE$
5. $REPEAT \rightarrow contain(repeatBlock, STATEMENT)$
6. $IF \rightarrow contain(ifBlock, STATEMENT)$
7. $IF_ELSE \rightarrow next(ELSE, IF)$
8. $ELSE \rightarrow contain(elseBlock, STATEMENT)$

Fig. 4. Produção gerada a partir das regras definidas anteriormente

Na Figura 5(a) mostra-se um script de código que foi elaborado na ferramenta (Figura 6), este algoritmo faz uso dos blocos simples *avançar* e *virar* e dos compostos: *repetir até* e *se caminho*. Na Figura 5(b) mostra-se a árvore de análise sintática que foi gerada a partir do Script de código.

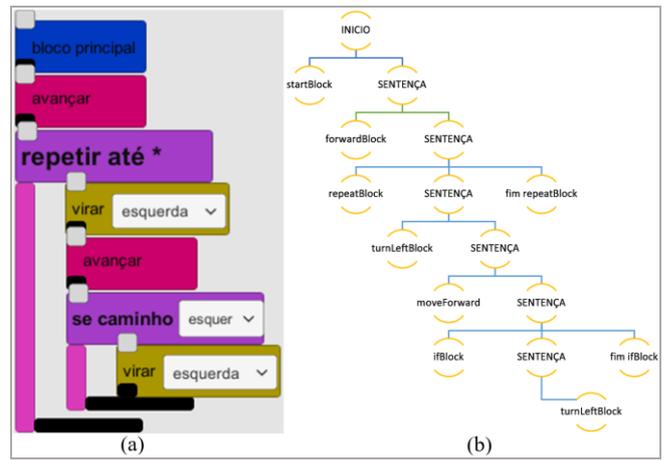


Fig. 5. Script de código gerado pelo compilador visual (a) e sua árvore de análise sintática (b).

Na figura 6, mostra-se a janela principal com todas as opções de programação.

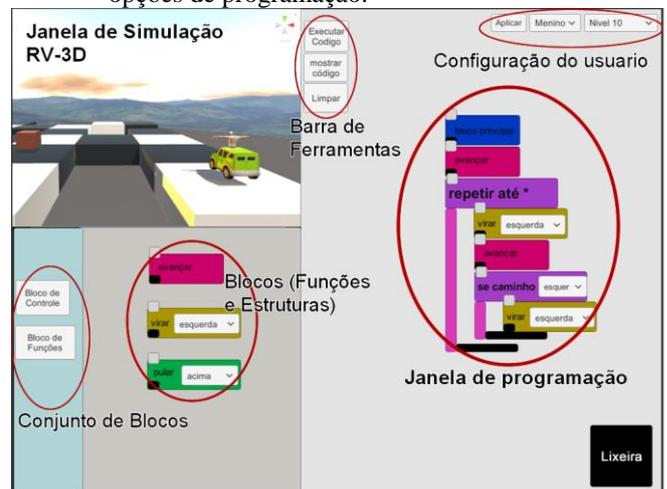


Fig. 6. Janela principal da Ferramenta

Na Figura 7 mostra-se o código gerado pela ferramenta de programação a partir do código visual da Figura 5. Este script será compilado e executado pelo Unity 3D (C#)

```

void animationMain() {
    moveForward();
    while (notDone()) {
        turnLeft();
        moveForward();
        if (isPathLeft()) {
            turnLeft();
        }
    }
}

```

Fig. 7. Script em linguagem C# gerado pela árvore de análise sintática.

IV. AVALIAÇÃO DA FERRAMENTA

A avaliação realizada teve como objetivo verificar se a ferramenta desenvolvida proporciona um ambiente no qual os estudantes se sentissem motivados a aprender mais sobre a programação. Além disso, pretendeu-se verificar o impacto do uso da tecnologia adotada, avaliando se a sensação de imersão ajuda para que os estudantes estejam motivados a cumprir todos os desafios que se propõe.

Os participantes da avaliação foram alunos de ensino fundamental da escola *Darcy Vargas* e do *Instituto Rogerio Steinberg (IRS)* no Rio de Janeiro, com idades compreendidas entre os 11 e 15 anos de idade, e o nível de ensino fundamental foi do 6º até o 9º ano. O processo de avaliação com os usuários foi realizado em três etapas: um pré-entrevista, teste e pós-entrevista. Essas avaliações foram acompanhadas do professor da matéria de computação.

Pré-Entrevista: esta etapa consistiu em uma série de perguntas para conhecer o nível do participante sobre os conceitos da ciência da computação, conhecer a experiência no uso de aparelhos de Realidade Virtual, assim como, a quantidade de tempo que usa o computador e com que fim.

Testes: nesta etapa juntaram-se dois participantes por computador. Estes participantes teriam que resolver o desafio número 1 para conhecer as características da ferramenta e poder familiarizar-se com ela. O trabalhar junto com outro colega ajuda à interação entre eles e ajuda muito no rápido entendimento da programação. Além disso, se põe em prática uma característica da metodologia de programação “*extreme programming - Pair Programming*” de uma forma muito básica. Para a formação de equipes levou-se em conta: a afinidade, e em segundo lugar a idade (participantes cuja idade tenham a diferença mínima devem estar juntos). Cada desafio era resolvido em uma sessão de 10 minutos no máximo; ao final da sessão os alunos teriam que testar seu código fazendo uso dos dispositivos de realidade virtual (Oculus Rift e Microsoft Kinect). Para isto, considerou-se um tempo de 15 min. como máximo para cada sessão. Ao terminar de testar todas as equipes, o pesquisador explicava o algoritmo ótimo que deu solução ao desafio, e também os conceitos e práticas do *pensamento computacional* que se utilizou para resolvê-lo. Com isto asseguramos-nos que os estudantes entendiam claramente como deveriam resolver os desafios, e ficariam mais habilitados para resolver o desafio seguinte.

Pós-Entrevista: após a realização dos testes, foi feita novamente uma entrevista com os estudantes. Nesta entrevista foram consideradas perguntas para conhecer o impacto que a ferramenta causou e conhecer como foi sua primeira reação com a realidade virtual. Além disso, a entrevista também visava conhecer sua opinião sobre a ferramenta virtual (coisas que gostaram e que não gostaram), e que coisas poderiam melhorar, remover ou adicionar. Também foram consideradas perguntas próprias para avaliar a *imersão no mundo virtual*, tendo como finalidade medir a sensação de *Telepresença*, que refere-se à sensação de estar presente no ambiente virtual, isto é, medir esse senso de “estar lá”. *Presença Social*, refere-se a quanto um corpo se sente imerso e representado no ambiente virtual e com habilidades para se conectar com outros elementos do ambiente e *Usabilidade, satisfação e entusiasmo*, refere-se a conhecer a aceitação da ferramenta de programação e da experiência de imersão no mundo virtual.

V. RESULTADOS

A. Resultados dos questionários

Os alunos responderam dois questionários: o questionário pré-teste e o questionário pós-teste. Alguns critérios da avaliação (perguntas e opções de resposta) foram tomadas do trabalho de [3]. O primeiro questionário

nos permitiu entender sobre a experiência no uso dos dispositivos eletrônicos (computadores e dispositivos de RV), além do conhecimento dos participantes sobre os conceitos básicos da programação.

A seguir faremos uma comparação entre os participantes das duas escolas levando em consideração as perguntas do questionário pré-teste. Para isso começaremos com a pergunta. Com *que dispositivo (s) você interage? Frequência de utilização do computador? e Com que fim usa o computador?*. Como resultado temos o seguinte: 92% (n = 11) dos participantes da escola Darcy Vargas e 67% (n = 4) dos participantes do IRS usam o computador pelo menos uma vez ao dia, o que demonstra que eles interagem com o computador muito frequentemente. Para a pergunta *Com que fim usa o computador?* se considerou uma múltipla escolha das alternativas. Verificou-se que as duas atividades que realizam em sua maioria são *Jogo de entretenimento* e *Internet*. Além de alguns participantes do IRS expressarem que eles usavam o computador para “*Fazer trabalhos escolares e assistir Aulas de robótica e codificação*”.

Com essa pergunta reforçamos nossa ideia de que os estudantes estão muito familiarizados com computadores e, além disso, alguns deles já têm uma experiência em relação à programação.

Para conhecer alguma experiência no uso de dispositivos de Realidade Virtual, se fez as perguntas: *Que dispositivos de Realidade Virtual você utilizou? Qual (s) tipo (s) de aplicação de RV (3D) você utiliza? e Qual é sua experiência utilizando aplicações de realidade virtual?*, nessas perguntas o participante pode fazer uma múltipla escolha das alternativas: a maioria de participantes 83% (n=10) e 67% (n=4) respectivamente nunca tiveram a oportunidade de usar os dispositivos de realidade virtual, e os participantes que usaram estes dispositivos como o PlayStation VR (8% e 33%) expressaram que o usaram para videogames, e além disso consideraram sua experiência no uso de aplicações de RV como pouca.

Dessa pergunta destacamos que a maioria dos participantes não conhece os dispositivos de RV.

A seguir descrevemos os resultados das 2 perguntas: *O que é programar? e O que é um algoritmo?*, que permitirá saber se eles conhecem os conceitos básicos da programação e se já tiveram alguma experiência na programação e elaboração de algoritmos. Essas perguntas foram realizadas no pré-teste e repetidas no pós-teste. Na Figura 8 pode-se ver os estudantes respondendo o questionário antes de começar a avaliação no IRS.



Fig. 8. Participantes respondendo às perguntas do questionário pré-teste

Depois de fazer a pergunta *O que é programar?* obteve-se que um total de (n=6) alunos marcaram a opção *é reparar o computador*, depois (n=5) alunos marcaram *é usar as aplicações do computador word, facebook, twitter, email, etc.* (n=3) alunos marcaram a opção *não tenho ideia* e só (n=4) alunos marcaram a opção *é escrever instruções para que o computador o faça*, que é a mais acertada a respeito da atividade dos cientistas da computação. Dessa pergunta se deduz que só 22% de alunos tinham uma noção de que é programar. Posteriormente essa mesma pergunta foi feita, no final da avaliação, e obteve-se como resultado que (n=17), ou 94% dos participantes, escolheram a opção correta.

Na pergunta *O que é um algoritmo?* Temos como resultado que um total de (n=7) alunos marcaram a opção *uma parte do computador*, depois (n=6) alunos marcaram *não tenho ideia*, (n=2) alunos marcaram a opção *uma receita de cozinha* e só (n=3) alunos marcaram a opção *um conjunto de passos ordenados para resolver algo*. Dessa pergunta se deduz que só 16% de participantes tinham uma noção sobre o que é um algoritmo. Posteriormente essa mesma pergunta foi feita, no final da avaliação, e obteve-se como resultado que (n=18), isto é, 100% dos participantes, escolheram a opção correta.

B. Resultados dos testes.

Pode-se ver na Figura 9 a formação das duplas de participantes por computador. Essa forma de trabalhar em pares permitiu que eles compartilhassem o conhecimento e facilitou resolver a maior quantidade de desafios.



Fig. 9. Participantes do IRS (esquerda) e Darcy Vargas (direita) resolvendo os desafios em duplas.

Na figura 10 pode-se ver o histograma do desvio padrão que resultou dos tempos que levaram os participantes da escola D.V. para solucionar cada desafio. Além disso, pode-se notar a dificuldade que eles tiveram durante a solução dos desafios avançados n°(7, 8, e 9). Também que só 8 deles conseguiram resolver o desafio n° 8 e 10 deles conseguiram resolver o desafio n° 9, e nenhum dos participantes conseguiu resolver o desafio n° 10

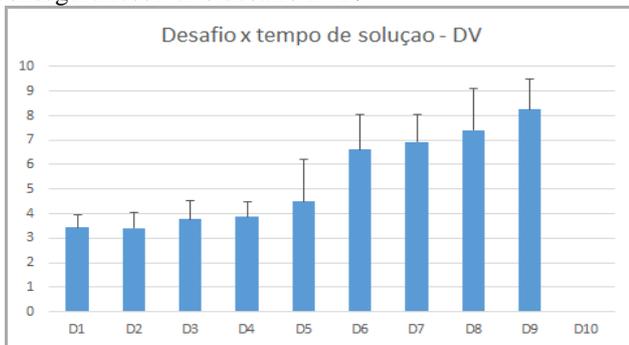


Fig. 10. Desvio padrão do tempo gasto pela escola D.V. para cada um dos 10 desafios da ferramenta

Sobre o pensamento computacional, pode-se explicar que os participantes usaram os conceitos e comportamentos de uma forma indireta, isto é, aplicando o processo de solução a todos os desafios. Este processo consiste em que os participantes primeiro analisaram o desafio, posteriormente propuseram uma solução usando os algoritmos de construção, e depois verificaram se a solução é funcional no ambiente virtual. Em caso de ter algum erro, teriam que encontrar o erro e resolvê-lo, e posteriormente propor alguma melhora do código fonte (caso seja possível otimizar o código levando em conta o tempo de chegada ao objetivo e a menor quantidade de blocos). A ferramenta fornece todas as funcionalidades que se necessita para a elaboração do código. Dentre essas funcionalidades encontra-se *depurar o código fonte*, isto é, ele permitiu encontrar o bloco de código que ocasionava o problema e facilitou sua correção.

Para a elaboração do código eles primeiro tinham que aplicar a lógica de programação, posteriormente para alguns desafios eles usaram estruturas de controle repetitivas, isto é, reutilizar alguns dos procedimentos repetidos como padrões. Por exemplo, no caso do desafio número 4, da Figura 11, a maioria dos participantes, cerca de 78% (n=14), desenvolveram o algoritmo com a primeira Figura 12(a), isto é, usando 16 blocos, os outros 4 alunos deram como solução a Figura 12(b), isto é, usando 5 blocos, e nenhum dos estudantes conseguiu resolver o desafio como apresenta a terceira Figura 12(c). Pode-se notar que a maioria de estudantes preferiu repetir o código fonte, mas depois que o guia mostrou para eles a terceira solução alguns dos estudantes ficaram assombrados sobre essa nova forma de solução.

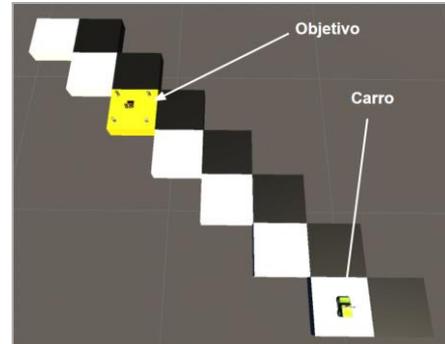


Fig. 11. Desafio número 4 da ferramenta virtual

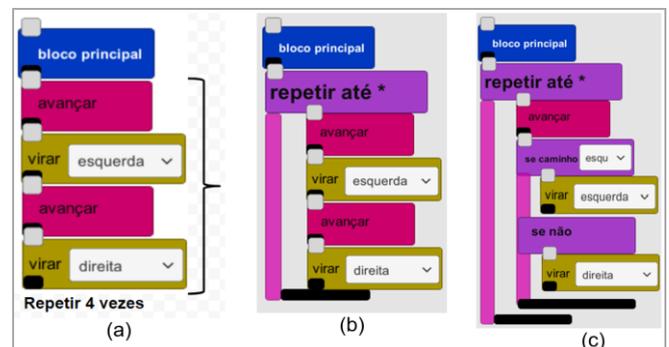


Fig. 12. Possíveis soluções para o desafio número 4

Pode-se ver na Figura 13 a quantidade de estudantes que conseguiram resolver os desafios.

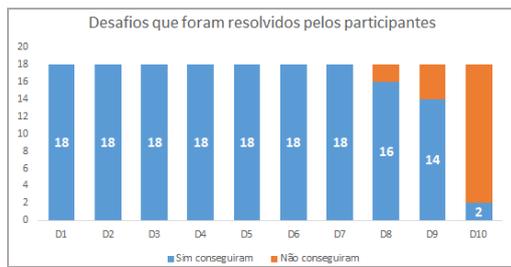


Fig. 13. Histograma que mostra a quantidade de desafios que foram e não foram resolvidos pelos participantes

Do total dos 10 desafios que foram propostos, 100% (n=18) dos participantes conseguiu resolver os primeiros 7 desafios. Os primeiros 3 foram resolvidos com ajuda de um guia; os outros 4 foram resolvidos por eles mesmos. Cerca de 89% (n=16) dos participantes conseguiram resolver o desafio número 8, enquanto 78% (n=14) deles conseguiram resolver o desafio número 9. Se observou que o principal problema que eles tiveram nestes desafios foi o uso da estrutura de controle condicional "if-else", apresentado pelo bloco composto "se, fazer alguma ação; senão, fazer outra ação". Cabe ressaltar que só dois participantes conseguiram resolver o desafio número 10 da Figura 14(a). Os outros participantes propuseram soluções que não conseguiam fazer com que o carro chegasse ao ponto final. Pode-se ver na Figura 14(b) uma solução proposta pelo par de estudantes que foram os únicos que resolveram o desafio, nessa proposta de solução eles descobriram que sua proposta fazia com que o carro percorresse todo os caminhos, isto é, não seguiu o caminho mais curto para chegar ao objetivo, mas eles se justificaram pela quantidade de blocos utilizados para a solução (7 blocos), já que isto é menor que a solução esperada (10 blocos) da Figura 14(c). Esta situação ajudou muito a discernir qual das duas soluções é a melhor, isto é, fazer uma escolha dentre a menor quantidade de blocos ou o menor caminho para encontrar o objetivo. Então eles começaram a discutir sobre a melhor solução e como resultado dessa discussão a maioria de estudantes cerca de 89% (n=16) deles considerou optar pela rota mais curta. A explicação que eles deram foi que a pessoa no carro precisa chegar no menor tempo possível a seu objetivo, e eles associaram este caso a suas atividades de transporte da vida diária.

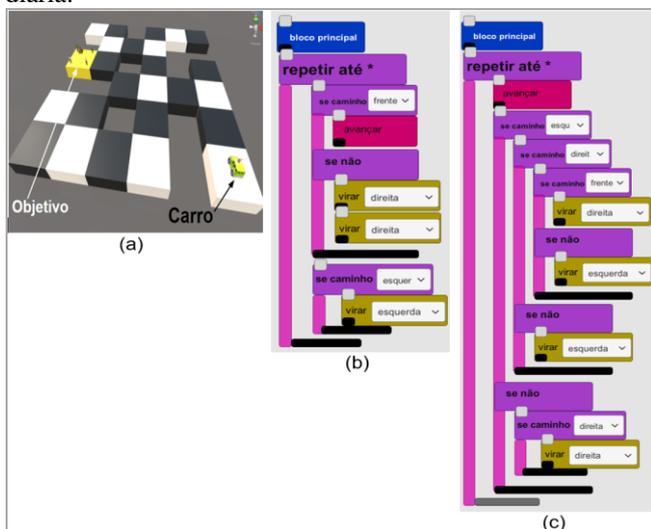


Fig. 14. Desafio nº10 e seus respectivos Scripts de solução.

C. Resultados da interação no mundo virtual

1) Resultados em Telepresença

A ferramenta de programação oferece uma experiência de imersão no ambiente 3D, isto depois que os participantes desenvolveram seus desafios e começam a avaliar o script de código que eles fizeram. Para medir este senso, se realizou as seguintes perguntas. "Você *sentiu que estava dentro e cercado pelo meio ambiente?*", e "*Como se sentiu vendo seu próprio corpo lá?*" As respostas de alguns participantes foram: "*Foi uma experiência muito legal :) ...*", "*Foi muito importante, me senti feliz por ter tido a experiência de me ver lá...*", "*Foi ótimo, senti uma altura tipo que ia cair e tinha que observar as pessoas que teve batendo palmas ...*" Na pergunta "Você sentiu que estava dentro e cercado pelo meio ambiente?", suas respostas foram: "*Sim!!!!*", "*não*", "*sim, era tudo muito real...*", "*sim, parecia que eu estava dentro do jogo...*"

Nem todas as respostas foram positivas para a experiência de imersão, cerca de 20% dos participantes fizeram comentários negativos, tais como: "*as pessoas no lugar não são muito reais ...*", "*A água não se mexe como numa lagoa real*", "*O carro está andando muito devagar...*"

Dessas respostas frente à avaliação de Telepresença, pode-se resumir que a ferramenta deu para os participantes novos sentimentos, como: emoção, alegria e realismo.

2) Resultados em presença social

A ferramenta permitiu ao participante escolher entre dois tipos de avatar (homem e mulher) que estarão montados acima do carro, além de um conjunto de pessoas que formavam a audiência. A ideia era fazer com que o participante se sinta ligado ao seu avatar, além disso esperávamos que o participante se sentisse motivado pela audiência e que pudesse interagir com os movimentos de seu braços e pernas. Para medir o sentimento de enlace social dos participantes com o personagem virtual, fizemos perguntas como: *Como sentiu-se quando era incentivado pelo público?*, *O personagem se sentiu real?* e *A pessoas que aparecem na ferramenta parecem reais?*

Na pergunta *Como sentiu-se quando era incentivado pelo público?*, O participante tinha a opção de escolher diferentes tipos de sentimentos, a maioria de participantes sentiram-se motivados e animados. Na pergunta "*O personagem se sentiu real?*" As respostas foram um total de 92% dos participantes escolheram (sim), e os outros escolheram (não). Na pergunta "*A pessoas que aparecem na ferramenta parecem reais?*" as respostas foram um total de 69% dos participantes escolheram (sim) e os outros (não).

3) Resultados em Usabilidade, Satisfação e Entusiasmo

A perguntas feitas com respeito à Usabilidade, Satisfação e Entusiasmo nos permitiu verificar se os participantes estiveram satisfeitos e entusiasmados com o uso da ferramenta, para isto se fizeram as perguntas: "*Quão provável que você use esse sistema?*" *O que você mudaria nesse sistema?* e *Você acha que essa experiência imersiva o ajudaria a aprender melhor?*". As respostas dos participantes ajudaram-nos no futuro a seguir com o desenvolvimento da ferramenta. As respostas para a pergunta "O que você mudaria nesse sistema?" São: "*nada*", "*Uns gráficos e Só...*", "*Iria fazer com que ele ficasse mais real...*", "*Eu colocaria uma coisa, mas*

espontânea...” E para a pergunta de “Quão provável é que você use esse sistema?” e “Você acha que essa experiência imersiva o ajudaria a aprender melhor?” Desenhou-se um histograma (Figura 15) sobre a escolha dos participantes entre as opções de “muito, pouco ou nada”

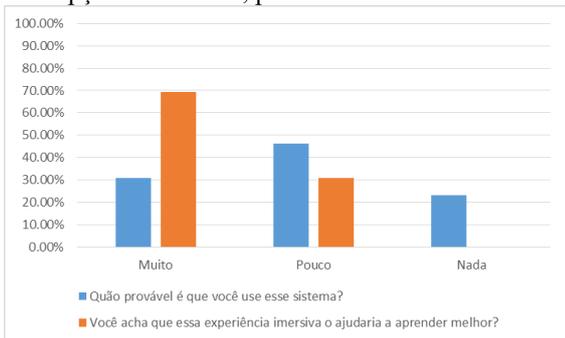


Fig. 15. Histograma que mostra o resultado dos participantes respeito à usabilidade da ferramenta.

D. Observações do comportamento dos participantes dentro da ferramenta virtual.

Na Figura 16 pode-se ver quando o carro começou a se mexer, eles apoiaram as mãos sobre suas pernas e alguns deles exclamaram “wouuu ...calma”.



Fig. 16. Participantes do IRS (esquerda) e D.V. (direita) com as mãos nas pernas.

Na Figura 17 pode se ver quando os participantes começaram a mexer os braços quando eles chegaram ao ponto final, isto deve-se a que neste ponto final estão as outras pessoas que formam a audiência. A audiência está fazendo sinais de motivação com seus braços e apoiando o participante para que ele chegue ao ponto final.



Fig. 17. Participantes dentro do ambiente virtual interagindo com os outros personagens

VI. DISCUSSÃO

A ferramenta de programação utilizou a técnica do *drag and drop de blocos* própria das linguagens visuais de programação, que é uma técnica muito intuitiva na elaboração do código visual. Usou 2 avatares para representar homens e mulheres. e também usou as cores para diferenciar cada bloco (função ou uma estrutura de controle). Além disso, há o conjunto de desafios que permite que o usuário use para cada nível uma determinada estrutura de programação (repetitivas e condicional).

A tecnologia de Realidade Virtual ajudou muito a que os estudantes ficassem ligados à ferramenta e conseguissem resolver todos os desafios.

Os participantes gostaram da ferramenta, devido à fácil geração de código através da montagem de blocos e à imersão no mundo virtual permitiu que eles comparassem essa sensação com os jogos tradicionais no computador. Alguns deles tinham o interesse em usar a ferramenta para ensinar a seus colegas de outras turmas e a seus familiares.

Eles também forneceram algumas sugestões de melhoria, por exemplo, propuseram um novo cenário onde o objeto virtual (carro) podia se mover, além de sugerir adicionar seus próprios desafios colocando mais obstáculos. Essas ações de propor melhorias e novas funcionalidades provocou que os participantes pensassem como cientistas da computação. A maioria dos participantes consideram que essa experiência imersiva os ajudaria a aprender melhor. Um participante comentou que queria levar a tecnologia de realidade virtual a outras matérias como biologia, anatomia e geografia.

Através das respostas da entrevista pré-teste e pós-teste e da análise do comportamento, descobrimos que a experiência de imersão no mundo virtual motivou que eles aprendessem um pouco da programação e sua introdução a ciências da computação. A análise também destacou a necessidade de gerar mais aplicações reais em outras matérias, isto é, que incluíam a experiência de imersão.

VII. CONCLUSÕES E TRABALHOS FUTUROS

A ferramenta permitiu ao participante a elaboração do código fonte de uma forma fácil. Isto devido à técnica de "Drag and drop", que consiste na montagem de blocos, de forma a mitigar os erros de sintaxe e permitir que os participantes concentrem-se mais na lógica de programação. Além disso a ferramenta possui a funcionalidade de apresentar para o participante o código fonte na linguagem de programação C#. Com isto, os alunos sabiam que os blocos eram só representações dos comandos que se usam para programar.

O processo de desenvolvimento consiste em analisar o desafio, propor uma solução, avaliar e depurar a solução, corrigir a proposta caso ela tenha erros e otimizar a solução. Este processo aplicado a cada desafio ajudou muito a que os participantes conhecessem de uma forma indireta os conceitos e comportamentos do pensamento computacional, e como este serviu para gerar a solução.

Observou-se que o conjunto de desafios permitiu que os participantes se sentissem atraídos pela ferramenta, ao mesmo tempo em que gerou competições entre eles, procurando sempre quem terminava antes. Além disso, o trabalho em duplas permitiu a comunicação e o intercâmbio

de conhecimento; essa ação permitiu a eles conhecer uma forma de trabalho comum entre os cientistas da computação, conhecida como "Pair-Programming". Alguns dos estudantes também começaram a propor outros tipos de soluções para os desafios. Isto permitiu entender que os desafios têm soluções ótimas e soluções não ótimas, onde os critérios para decidir que solução eleger poderiam ser a quantidade de blocos utilizados ou o tempo que se precisava para chegar ao objetivo.

O uso da realidade virtual permitiu a conexão entre o participante e a ferramenta, isto notou-se mediante as expressões deles quando viam seu corpo apresentado pelo avatar, que foi um dos principais fatores pelo qual eles ficavam ligados à ferramenta até cumprir todos os desafios. Na imersão do participante no mundo virtual destacou-se os 3 princípios básicos "Telepresença", "Presença Social" e a "Usabilidade"; a ferramenta de programação atendeu todos estes três princípios, confirmado posteriormente pelas opiniões no questionário do pós-teste.

O uso das avaliações divididas em pré-teste, teste e pós-teste permitiu conhecer o estado inicial dos participantes antes de começar com o teste. O questionário pós-teste ajudou a ter um feedback por parte deles, como por exemplo algumas sugestões com respeito ao desenho gráfico e os níveis de complexidade dos desafios, e conhecer também as opiniões que trazem sobre a realidade virtual e a programação.

O estudo feito em duas escolas diferentes do Rio permitiu conhecer as diferenças destes estudantes, isto é, os participantes da escola Darcy Vargas tiveram algumas dificuldades a mais que os participantes do IRS, isto porque os participantes do IRS são alunos que anteriormente já haviam trabalhado com programação em outras ferramentas visuais. Além disso, a diferença notou-se na quantidade de perguntas que faziam ao guia.

Como trabalhos futuros, seria interessante trabalhar com a integração de mais elementos da programação como funções, variáveis, operações matemáticas e entrada de dados, o que levaria à elaboração de desafios mais complexos que aprofundem mais o conhecimento da programação e o pensamento computacional. A implementação de mais componentes à ferramenta permitirá com que os participantes gerem seus próprios desafios.

Também será importante fazer com que o participante interaja mais no mundo virtual. Pode-se por exemplo usar uma técnica de gamificação, que consiste em que os participantes durante a imersão possam colecionar objetos com a movimentação de suas extremidades (braços e pernas) e estes objetos apresentem algum bônus, gerando com isto uma competição entre os participantes, o que ajudaria a mantê-los mais ligados e entusiasmados ao uso da ferramenta.

Podia-se usar a ferramenta para o ensino de alguma outra matéria de ciências, isto é, mediante a elaboração de desafios focados numa matéria específica podendo ser: matemática, física, biologia, entre outras.

Para uma futura avaliação, seria bom levar em consideração uma maior quantidade de participantes divididos por níveis de educação, isto para uma gerar um melhor resultado e conhecer melhor as opiniões deles, o que servirá para a elaboração de desafios adequados para um certo nível de ensino. Além disso, elaborar uma avaliação

com estes mesmos participantes usando ferramentas comuns, isto é, que não usem a realidade virtual.

AGRADECIMENTOS

À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), ao Tecgraf e à PUC-Rio pelo apoio financeiro, espaço e de recursos.

REFERÊNCIAS

- [1] Levy, P., & da Costa, C. I. (1993). "tecnologias da inteligência", As. Editora 34.
- [2] Van den Branden, K., Van Gorp, K., & Verhelst, M. (2009). "Tasks in Action: Task-Based Language Education from a Classroom-Based Perspective". Cambridge Scholars Publishing.
- [3] Parmar, D., Isaac, J., Babu, S. V., D'Souza, N., Leonard, A. E., Jorg, S., ... Daily, S. B. (2016). "Programming moves: Design and evaluation of applying embodied interaction in virtual environments to enhance computational thinking in middle school students". In 2016 IEEE Virtual Reality (VR). <https://doi.org/10.1109/vr.2016.7504696>
- [4] T. Nadan, V. Alexandrov, R. Jamieson, and K. Watson. "Is virtual reality a memorable experience in an educational context?" International Journal of Emerging Technologies in Learning (IJET), 6(1), 2011.
- [5] Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). "A study of the difficulties of novice programmers". In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE '05. <https://doi.org/10.1145/1067445.1067453>
- [6] Robins, A., Rountree, J., & Rountree, N. (2003). "Learning and teaching programming": a review and discussion. Computer Science Education, 13(2), 137-172. <http://dx.doi.org/10.1076/csed.13.2.137.14200>.
- [7] Boechler, P., Artym, C., Dejong, E., Carbonaro, M., & Stroulia, E. (2014). "Computational Thinking, Code Complexity, and Prior Experience in a Videogame-Building Assignment". In 2014 IEEE 14th International Conference on Advanced Learning Technologies. <https://doi.org/10.1109/icalt.2014.118>
- [8] Dasgupta, S., Hale, W., Monroy-Hernández, A., & Hill, B. M. (2016). "Remixing as a Pathway to Computational Thinking". In Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16. <https://doi.org/10.1145/2818048.2819984>
- [9] Fronza, I., El Ioini, N., & Corral, L. (2015). "Students Want to Create Apps". In Proceedings of the 16th Annual Conference on Information Technology Education - SIGITE '15. <https://doi.org/10.1145/2808006.2808033>
- [10] Daily, S. B., Leonard, A. E., Jörg, S., Babu, S., & Gundersen, K. (2014). "Dancing alice". In Proceedings of the 45th ACM technical symposium on Computer science education - SIGCSE '14. <https://doi.org/10.1145/2538862.2538917>
- [11] Golín, E. J., & Reiss, S. P. (1990). "The specification of visual language syntax". Journal of Visual Languages & Computing, 1(2), 141-157.