

**Relatório sobre a execução da avaliação Percurso Cognitivo sobre
Ambientes de desenvolvimento
Dezembro/2017**

Daniel Bueno Domingueti

1. INTRODUÇÃO

O método do Percurso Cognitivo consiste na ação do avaliador se colocar no lugar do usuário ao realizar as tarefas pré-definidas, tentando abstrair as suas possíveis ações e assim identificar falhas de interação na interface [1]. Ele tem como objetivo avaliar o quanto a interface facilita a exploração e o aprendizado do software.

O método consiste nas seguintes etapas: (i) preparação, (ii) coleta de dados e interpretação, (iii) consolidação dos resultados e (iv) elaboração de um relatório consolidando os problemas encontrados e sugestões de correção dos mesmos.

Para este trabalho, foi realizado a avaliação do Percurso Cognitivo sobre as plataformas de desenvolvimento MIT App Inventor, GDevelop, e Unity utilizando a metodologia de avaliação descrita por [2].

2. PREPARAÇÃO

Foram definidas cinco tarefas para todas as plataformas. A primeira consiste em criar um Hello World, com a execução mínima de um projeto vazio, e tem como intuito introduzir o ambiente de desenvolvimento e o método de criação de projetos. A segunda consiste em programar uma animação de um personagem pois é um recurso essencial no desenvolvimento de jogos. A seguir, é realizada a alteração uma funcionalidade de um jogo pronto – ou seja, adicionar um placar ao jogo “Jokenpo” criado para este trabalho – para que o usuário tenha contato com o ambiente de programação específico da plataforma. Após, é feito a correção de um erro de sintaxe – considerando o estilo de programação dos ambientes em análise – para mensurar a qualidade das mensagens de erro – e por fim realizar o deploy de um projeto, ou seja, gerar o executável de um projeto.

Para a avaliação das tarefas acima, foi estabelecido que o perfil de usuário para os ambientes App Inventor e GDevelop eram de pessoas que estão dando o primeiro passo no desenvolvimento de aplicações, enquanto que na Unity foi estabelecido um usuário com conhecimento prévio sobre

Orientação a Objetos devido a complexidade da plataforma (ou seja, tem um conhecimento mínimo sobre desenvolvimento de aplicações e um pouco de experiência com a utilização de IDEs).

A especificação das tarefas considerou o nível de complexidade que os perfis de usuários estabelecidos poderiam cumprir, logo não demonstra o modo mais efetivo ou com melhor desempenho de se realizar as tarefas estabelecidas, e sim o modo mais simples ou adequado de serem cumpridas. Para a tarefa 2 (programar uma animação de um personagem), foi considerado apenas o processo de se criar uma animação em 2D, tendo em vista que apenas o ambiente Unity permite o desenvolvimento com objetos tridimensionais.

3. COLETA DE DADOS E INTERPRETAÇÃO

Para a coleta e interpretação dos dados, as tarefas foram realizadas seguindo os passos pré-definidos, e para cada tarefa busca-se responder três questionamentos definido por [2]: (a) o usuário irá saber o que fazer?; (b) O usuário saberá como fazer a tarefa?; (c) O usuário tem como saber se a ação realizada foi correta ou não, ou seja, ele tem “*feedback*” de suas ações?

A descrição das tarefas executadas para cada ambiente em específico é disposta nas próximas subseções.

3.1 APP INVENTOR

A seguir, são definidas as tarefas devidamente estabelecidas para a realização na plataforma Mit App Inventor.

3.1.1 Tarefa 1: Executar um Hello World, com a execução mínima de um projeto vazio

3.1.1.1 Criar um novo projeto

3.1.1.1.1 Clicar em "Iniciar novo projeto ..."

3.1.1.1.2 Digitar o nome “Hello World” e clicar em "OK”

3.1.1.2 Na aba "Palette" no lado esquerdo da tela, procurar a o item "Label" presente na subaba "User Interface".

3.1.1.3 Clicar e arrastar o componente "Label" para a tela.

3.1.1.4 No lado esquerdo da tela, nas propriedades do componente recém-criado, alterar o campo "Text" para "Minha primeira aplicação".

3.1.2 Tarefa 2: Programar uma animação

3.1.2.1 Carregar os arquivos necessários

3.1.2.1.1 Na aba de mídia (parte inferior direita), clicar em "Upload File ..."

3.1.2.1.2 Clicar em "Escolher Arquivo"

3.1.2.1.3 Selecionar o arquivo "indio1.png" disponibilizado e clicar em "OK"

3.1.2.1.4 Repetir os passos 4.1, 4.2 e 4.3 para o restante dos arquivos (indio2.png a indio9.png)

3.1.2.2 Adicionar e configurar um Canvas

3.1.2.2.1 Na aba esquerda ("Palette"), procurar e abrir a subaba Drawing and Animation

3.1.2.2.2 Arrastar o componente "Canvas" para dentro da tela do smartphone exibido no centro da tela

3.1.2.2.3 Na aba "Properties" do Canvas criado, alterar os campos Height e Width para "Fill Parent"

3.1.2.3 Na aba "Drawing and Animation" arrastar um componente "ImageSprite" para o centro da tela

3.1.2.4 Adicionar e configurar um "Clock" (relógio)

3.1.2.4.1 Na aba "Sensors", arrastar e soltar na tela do smartphone um componente "Clock"

3.1.2.4.2 Na aba "Properties" do componente, alterar o campo "Time Interval" para 60

3.1.2.5 Clicar no botão "Blocks" presente no canto superior direito da tela

3.1.2.6 Adicionar o bloco de controle do relógio

3.1.2.6.1 Na aba esquerda ("Blocks") procurar e clicar no elemento "Clock"

3.1.2.6.2 Arrastar para a tela o bloco "When Clock1.Timer do"

3.1.2.7 Adicionar um bloco "If ... Else"

- 3.1.2.7.1 Procurar e clicar no item "Control" na aba esquerda
- 3.1.2.7.2 Arrastar e soltar o primeiro bloco exibido ("If then")
- 3.1.2.8 Realizar a primeira troca de imagens
 - 3.1.2.8.1 Na componente "Text", adicionar o bloco "compare texts ... < ..." ao bloco "if then" criado anteriormente
 - 3.1.2.8.2 Alterar o critério de comparação para "=" (ao invés de "<")
 - 3.1.2.8.3 Clicar no componente "ImageSprite1" presente na aba "Blocks"
 - 3.1.2.8.4 Procurar e arrastar para o primeiro espaço do bloco "compare texts" o bloco "ImageSprite.Picture"
 - 3.1.2.8.5 No componente "Text", adicionar o primeiro bloco (texto vazio) ao último bloco de comparação do bloco criado no passo 8.1 (depois do sinal de igual)
 - 3.1.2.8.6 Alterar o texto do bloco criado anteriormente para "*indio1.png*"
 - 3.1.2.8.7 No componente "ImageSprite1", procurar e arrastar para o encaixe "then" do bloco "If" o bloco "Set ImageSprite1.Picture to"
 - 3.1.2.8.8 Repetir os passos 8.5 e 8.6 para o bloco anteriormente criado, trocando seu nome para "*indio2.png*"
- 3.1.2.9 Adicionar mais uma comparação ao bloco
 - 3.1.2.9.1 Clicar na ferramenta azul no canto do bloco "If ... Then"
 - 3.1.2.9.2 Arrastar o bloco "else if" para dentro do bloco "if ... then" exibido
- 3.1.2.10 Realizar a segunda troca de imagens
 - 3.1.2.10.1 Repetir os passos 8.1 a 8.5 para o bloco recém adicionado ao bloco principal do *If ... Then*
 - 3.1.2.10.2 No componente "Text", adicionar o primeiro bloco (texto vazio) ao último bloco de comparação de textos
 - 3.1.2.10.3 Alterar o texto do bloco criado anteriormente para "*indio2.png*"
 - 3.1.2.10.4 Repetir os passos 8.7 e 8.8, trocando o nome para "*indio3.png*"
- 3.1.2.11 Realizar as demais trocas de imagens
 - 3.1.2.11.1 Repetir o passo 9 para adicionar mais uma comparação à medida que necessário
 - 3.1.2.11.2 Repetir o passo 10 para as imagens seguintes:

Texto do bloco de comparação de textos depois do igual	Texto do bloco "Set ImageSprite1.Picture to"
indio3.png	indio4.png
indio4.png	indio5.png
indio5.png	indio6.png
indio6.png	indio7.png
indio7.png	indio8.png
indio8.png	indio9.png

Tabela 1: Descrição dos itens a serem assimilados nos blocos de texto

3.1.2.12 Para o último bloco de troca de imagens, a imagem final deverá retornar para a inicial, para isso alterar os textos dos blocos para:

3.1.2.12.1 Texto do bloco de comparação de textos depois do igual:

indio9.png

3.1.2.12.2 Texto do bloco "Set ImageSprite1.Picture to": indio1.png

3.1.3 Tarefa 3: Alterar uma funcionalidade de um jogo pronto – adicionar um placar ao jogo “Jokenpo”

3.1.3.1 Na tela inicial dos projetos, selecionar o projeto "Jokenpo"

3.1.3.2 Alterar a tela para a tela do jogo

3.1.3.2.1 Na aba superior (verde) clicar no botão "Screen1"

3.1.3.2.2 Clicar no campo “Screen_Jogo”

3.1.3.3 Clicar no botão "Blocks" no lado direito superior

3.1.3.4 Criar uma variável global chamada "numVitorias" e inicializá-la com valor zero

3.1.3.4.1 Na aba esquerda (Blocks), encontrar o item "Variables" e clicar nele

3.1.3.4.2 Arrastar para a tela aonde se encontram os demais blocos o primeiro bloco ("initialize global name to")

3.1.3.4.3 Clicar no campo "name" e escrever "pontuacao"

3.1.3.4.4 Na aba esquerda, encontrar o item Math e clicar nele

3.1.3.4.5 Arrastar o primeiro bloco (número zero) e encaixar no bloco da variável criado anteriormente

- 3.1.3.5 Repetir o processo anterior e criar uma outra variável chamada “numDerrotas” e inicializá-la com o valor zero
- 3.1.3.6 Contar o número de vitórias
 - 3.1.3.6.1 Localizar a função "jogadaAdversario"
 - 3.1.3.6.2 Na aba esquerda, localizar no item "Variables" o bloco " set ... to ..." e arrastar para a tela
 - 3.1.3.6.3 Inserir o bloco logo abaixo na primeira comparação (if get resultado = 1)
 - 3.1.3.6.4 Clicar no campo vazio do bloco criado anteriormente e alterar para a variável "global numVitorias"
 - 3.1.3.6.5 Na aba esquerda, localizar no item "Math" o bloco de soma (... + ...), arrastar e encaixar no bloco da variável criado
 - 3.1.3.6.6 Na aba esquerda, localizar no item "Variables" o bloco " get ... ", arrastar e encaixar no primeiro campo do bloco de soma
 - 3.1.3.6.7 Repetir os passos 4.3 e 4.4
 - 3.1.3.6.8 Alterar o texto do bloco numérico para 1
- 3.1.3.7 Contar o número de derrotas
 - 3.1.3.7.1 Ainda no procedimento “jogadaAdversario”, localizar o “else” do bloco IF.
 - 3.1.3.7.2 Repetir as etapas 6.2 a 6.8, alterando o nome da variável do bloco “Get” para “global numDerrotas”.
- 3.1.3.8 Criar o procedimento exhibeResultado para escrever o resultado na tela
 - 3.1.3.8.1 Na aba esquerda, encontrar o item "Procedures" e clicar nele
 - 3.1.3.8.2 Encontrar e arrastar o bloco "to procedure do" (segundo exibido na lista) para a tela
 - 3.1.3.8.3 Clicar no campo “procedure” do bloco criado anteriormente para ExibeResultado
 - 3.1.3.8.4 Na aba esquerda, encontrar o item "Canvas1" e clicar nele
 - 3.1.3.8.5 Encontrar e arrastar para dentro do bloco "ExibeResultado" o bloco "Canvas1.DrawText text ... x ... y ..."
 - 3.1.3.8.6 Encaixar o bloco abaixo do bloco que altera a imagem do adversário (Set Sprite Jogada_Adversario.Picture to ...)

- 3.1.3.8.7 Encontrar na aba esquerda o item "Text" e clicar nele
- 3.1.3.8.8 Arrastar o bloco "join" para a entrada "text" do bloco "...DrawText" criado anteriormente
- 3.1.3.8.9 Clicar na ferramenta azul no bloco "join" e arrastar mais um campo de string ao bloco join
- 3.1.3.8.10 Na aba "Variables", colocar no primeiro encaixe do bloco "Join" o bloco "get" e alterar seu valor para "global numVitorias"
- 3.1.3.8.11 Na aba "Text", colocar no segundo encaixe do bloco join o bloco de texto vazio (primeiro bloco) e alterar seu valor para " x "
- 3.1.3.8.12 No item "Variables" do menu esquerdo, selecionar o bloco get e arrastar para o segundo encaixe do join e alterar seu valor para "global numDerrotas"
- 3.1.3.8.13 Escrever a pontuação na tela
 - 3.1.3.8.13.1 Alterar a posicao X do texto a ser desenhado
 - 3.1.3.8.13.1.1 Na item Math, encontrar e arrastar para o encaixe "X" o bloco de divisão
 - 3.1.3.8.13.1.2 No item "Canvas1", encontrar o bloco "Canvas1.Width" e encaixá-lo no primeiro operando do bloco de divisão
 - 3.1.3.8.13.1.3 Inserir um bloco numérico (passos 4.4 e 4.5) para o segundo operando do bloco de divisão
 - 3.1.3.8.13.1.4 Alterar o valor do bloco criado anteriormente para 2
 - 3.1.3.8.13.2 Alterar a posicao Y do texto a ser desenhado
 - 3.1.3.8.13.2.1 Adicionar um bloco numérico vazio (passos 4.4 e 4.5) para o encaixe Y e alterar seu valor para 15
- 3.1.3.9 Na aba Procedures, encontrar o bloco "call exibeResultado" e encaixá-lo ao fim do procedimento "jogadaAdversario"
- 3.1.3.10 Repetir o passo anterior, encaixando o bloco ao fim do procedimento "reiniciaJogo"

3.1.4 Tarefa 4: Corrigir um erro de sintaxe

- 3.1.4.1 Na área dos blocos, clicar no botão "Show Warnings" na parte inferior da tela
- 3.1.4.2 Localizar o erro exibido (X vermelho)
- 3.1.4.3 Remover o bloco com erro
- 3.1.4.4 Adicionar um bloco de texto com o texto "Pedra" no lugar do bloco removido

3.1.5 Tarefa 5: Realizar o deploy de um projeto

No caso da plataforma App Inventor, existem duas possibilidades: gerar o APK e salvar no computador localmente ou gerar um QR Code para instalar a aplicação.

- 3.1.6 Salvar o APK da aplicação no computador
 - 3.1.6.1 Clicar em "Build" no menu superior
 - 3.1.6.2 Clicar em "App(save .apk to my computer)"
- 3.1.7 Gerar o QR Code
 - 3.1.7.1 Clicar em "Build" no menu superior
 - 3.1.7.2 Clicar em "App (provide QR code for .apk)"

3.2 GDEVELOP

A seguir, são definidas as tarefas devidamente estabelecidas para a realização na plataforma GDevelop.

3.2.1 Tarefa 1: Executar um Hello World

- 3.2.1.1 Criar um novo projeto
 - 3.2.1.1.1 Clicar em "New" no menu superior
 - 3.2.1.1.2 Selecionar a plataforma "*Web (Web and Android Games)*"
 - 3.2.1.1.3 Selecionar "Empty Project"
 - 3.2.1.1.4 Clicar no botão representado por "..." ao fim do campo para inserir o caminho do projeto e escolha uma pasta
 - 3.2.1.1.5 Renomeie o arquivo para "Hello World" e clique em "Salvar"
 - 3.2.1.1.6 Clicar em "*Create a New Project*"

3.2.2 Tarefa 2: Programar uma animação

3.2.2.1 Importar os arquivos necessários

3.2.2.1.1 Na área esquerda, nomeada de “Project Manager”, clicar duas vezes em “Images”

3.2.2.1.2 Na aba superior, clicar em “Add an Image”

3.2.2.1.3 Navegar até a pasta onde estão as imagens, selecionar todas e pressionar “Abrir” (ou similar)

3.2.2.2 Na aba Project, dar dois cliques na cena “New Scene”, dentro de “Scenes” para abrir a cena.

3.2.2.3 Clicar com o mouse do lado direito na área cinza e clicar em “Add an Object”

3.2.2.4 Selecionar o primeiro item (Sprite) e clicar em OK.

3.2.2.5 Caso a janela possua apenas as informações do banco de imagens (Image bank's editor), clicar no quadrado branco do lado esquerdo da tela para minimizá-la ou clicar na área roxa e arrastar para outro canto na janela para que todas as outras áreas fiquem visíveis.

3.2.2.6 Arrastar a imagem “indio1.png” para a área “Images”, abaixo de “Animations”.

3.2.2.7 Repetir a etapa anterior para todas as outras imagens.

3.2.2.8 Alterar a velocidade de exibição da animação

3.2.2.8.1 Na área “Animations”, clicar no relógio presente

3.2.2.8.2 Na janela que apareceu, digitar o valor 0.1

3.2.2.8.3 Pressionar “OK”

3.2.2.8.4 Ainda na área “Animations”, clicar no botão em formato de seta circular (para que a animação fique em loop)

3.2.2.9 Clicar na aba “Scene” presente no menu superior

3.2.2.10 Pressionar o botão “Preview” para ver a animação ocorrer.

3.2.3 Tarefa 3: Alterar um jogo pronto (inserir contagem de pontos)

3.2.3.1 Na hierarquia do projeto, clicar duas vezes na cena “Jogo” para poder editar

3.2.3.2 Criar e configurar um objeto do tipo texto

- 3.2.3.2.1 Dentro da tela do jogo, clicar com o botão direito do mouse e selecionar “Insert a New Object”
- 3.2.3.2.2 Selecionar “Text” na janela que foi aberta e pressionar “OK”
- 3.2.3.2.3 Na caixa de texto, digite “0 x 0”
- 3.2.3.2.4 Clique no número “20” e altere o tamanho da fonte para “14”
- 3.2.3.2.5 Clique em “Color...” e altere a cor da fonte para preto
- 3.2.3.2.6 Pressionar OK
- 3.2.3.2.7 Na aba esquerda (Objects Editor), clicar com o botão direito do mouse no item “New Object” e clicar em Rename
- 3.2.3.2.8 Apagar o texto presente, digitar “Pontuação” e pressionar Enter
- 3.2.3.2.9 Clicar com o botão do mouse direito novamente sobre o item “Pontuação” e clicar em “Other Properties”
- 3.2.3.2.10 Alterar o valor do campo X para 390, o valor do campo Y para 55 e pressionar OK
- 3.2.3.3 Adicionar a variável numVitorias para contabilizar o número de vitórias
 - 3.2.3.3.1 Na aba “Project Manager”, clicar com o botão direito do mouse na cena “Jogo” e selecionar “Modify Initial Variables”
 - 3.2.3.3.2 Clicar no “+” na parte de cima da nova janela
 - 3.2.3.3.3 Escrever o nome “numVitorias” e pressionar “OK”
- 3.2.3.4 Adicionar a variável numDerrotas para contabilizar o número de derrotas
 - 3.2.3.4.1 Repetir a etapa anterior, nomeando a variável de numDerrotas
 - 3.2.3.4.2 Pressionar “OK” na janela em que foi feita a criação das variáveis
- 3.2.3.5 Alterar para a aba “Events”
- 3.2.3.6 Baixar a tela até o último evento cadastrado
- 3.2.3.7 Contabilizar o número de vitórias
 - 3.2.3.7.1 Com o mouse sobre a ação do evento identificado como número 4, clicar no botão “Add an Action”
 - 3.2.3.7.2 Selecionar a aba “Variables” e selecionar “Value of a variable”

- 3.2.3.7.3 No campo “Variable” clicar na caixa ao lado, selecionar a variável numVitorias e pressionar “Choose” na janela das variáveis
- 3.2.3.7.4 Associar o valor de modificação a variável “numDerrotas”
 - 3.2.3.7.4.1 Clicar no botão representado pela letra “Sigma” no terceiro campo para preenchimento
 - 3.2.3.7.4.2 Na aba esquerda da nova janela (Other Functions), selecionar “Variables” e clicar duas vezes em “Scene Variables”
 - 3.2.3.7.4.3 Escolher a variável “numDerrotas”
 - 3.2.3.7.4.4 No campo acima, escrever “+ 1” ao fim da expressão e pressionar OK
- 3.2.3.7.5 Pressionar OK na janela principal
- 3.2.3.8 Contabilizar o número de derrotas
 - 3.2.3.8.1 Com o mouse sobre a ação do evento identificado como número 5, clicar no botão “Add an Action”
 - 3.2.3.8.2 Repetir as etapas 2.2.3.7.2 a 2.2.3.7.4 para a variável “numVitorias”
- 3.2.3.9 Exibir o placar na tela
 - 3.2.3.9.1 Encontrar o evento “Variable PodeJogar = 0” (numerado como 4 acima)
 - 3.2.3.9.2 Passar o mouse sobre as ações do evento e clicar em “+ A sub Event”
 - 3.2.3.9.3 Arrastar o evento recém-criado para abaixo de todos os eventos cadastrados. Ele deverá ficar numerado como 7
 - 3.2.3.9.4 Passar o mouse sobre a área de ações do novo evento (no momento demonstrado por “No Actions”) e clicar sobre “Add an action”
 - 3.2.3.9.5 Encontrar e clicar sobre o item “Text object” na hierarquia do lado esquerdo da nova tela exibida
 - 3.2.3.9.6 Clicar em “Modify the text”
 - 3.2.3.9.7 No campo “Object”, clicar na caixa ao lado e selecionar o objeto “Pontuação”

3.2.3.9.8 No terceiro campo, digitar: `VariableString(numVitorias) + " x "`
`+ VariableString(numDerrotas)`

3.2.4 Tarefa 4: Corrigir um erro

3.2.4.1 Procurar o evento 4, que tem como condição a variável `PodeJogar` ser igual a 1

3.2.4.2 No primeiro evento, (The cursor/touch is on Pedra) verificar a variável que é feita a atribuição do texto “Pedra”

3.2.4.3 Dar dois cliques na ação que está com a variável errada (`MinhaJogada`)

3.2.4.4 Clicar no botão “Var” ao lado do primeiro campo e selecionar a variável “`JogadaAtual`” e clicar em “Choose”

3.2.4.5 Pressionar “OK” na tela anterior

3.2.5 Tarefa 5: Realizar Deploy do projeto

3.2.5.1 Abrir o menu “File”

3.2.5.2 Clicar em “Export to the web”

3.2.5.3 Na aba abaixo de “Choose how to export the game”, abra e selecione o segundo item (Export to iOS and Android with Cordova (Intel XDK))

3.2.5.4 Pressionar Export

3.3 UNITY

A seguir, são definidas as tarefas devidamente estabelecidas para a realização na plataforma Unity. Foi considerado a utilização do editor de código Mono Develop para a correção do erro de sintaxe.

3.3.1 Tarefa 1: Executar um Hello World

3.3.1.1 Criar um projeto

3.3.1.1.1 Na tela inicial da Unity, clicar no botão “New” na parte superior da tela

- 3.3.1.1.2 Escrever o nome do projeto como "Hello World"
- 3.3.1.1.3 Selecionar (caso não esteja) a opção “3D” abaixo
- 3.3.1.1.4 Clicar em "Create Project"
- 3.3.1.2 Na aba Layout no canto superior direito, selecionar "2 by 3"
- 3.3.1.3 Criar um Cubo
 - 3.3.1.3.1 Clicar com o mouse do lado direito na aba "Hierarchy"
 - 3.3.1.3.2 Navegar até 3D Object e selecionar "Cube"

3.3.2 Tarefa 2: Programar uma animação

- 3.3.2.1 Criar o projeto
 - 3.3.2.1.1 Repetir o passo 1 da tarefa 1, alterando o nome do projeto para “Animação” e marcando a opção “2D” abaixo.
- 3.3.2.2 Importar os arquivos necessários
 - 3.3.2.2.1 Bater com o lado direito do mouse na pasta "Assets" na hierarquia, navegar até "Create" e clicar em "Folder"
 - 3.3.2.2.2 Escrever "Sprites" na pasta criada
 - 3.3.2.2.3 Refazer o passo 2.1 para as pastas de nome "Cenas" e "Animações"
 - 3.3.2.2.4 Abrir a pasta "Sprites" a partir do editor e arrastar de uma janela do sistema operacional para o editor as imagens necessárias
- 3.3.2.3 Salve a cena com o nome "Animacao" na pasta "Cenas"
 - 3.3.2.3.1 Pressione “Ctrl S” para salvar a cena
 - 3.3.2.3.2 Escolha o caminho e escreva o nome desejado
 - 3.3.2.3.3 Pressione OK na janela que foi aberta
- 3.3.2.4 Na pasta Sprites, selecionar todas as imagens dos índios e arrastar para a tela "Scene" no lado esquerdo
- 3.3.2.5 Salvar o arquivo criado na pasta animações com o nome "indio"
- 3.3.2.6 Alterar a velocidade de exibição da animação
 - 3.3.2.6.1 Com o índio selecionado na hierarquia, clicar no item "Window" no canto superior direito
 - 3.3.2.6.2 Clicar em "Animation"
 - 3.3.2.6.3 Alterar o campo "Samples" da janela que foi aberta para 10

3.3.2.6.4 Pressionar o botão representado por um círculo vermelho para parar a gravação

3.3.2.6.5 Fechar a janela

3.3.2.7 Pressionar o botão “Play” na parte superior do editor e ver a animação ocorrer.

3.3.3 Tarefa 3: Alterar um jogo pronto (inserir contagem de pontos)

3.3.3.1 Selecionar o projeto "Jokenpo" na janela inicial

3.3.3.2 Na pasta Cenas no editor, abrir a cena "Jogo"

3.3.3.3 Adicionar um componente "Text" e renomear

3.3.3.3.1 Clicar com o mouse direito no item "Canvas" na hierarquia

3.3.3.3.2 Navegar até UI

3.3.3.3.3 Clicar com o mouse do lado esquerdo no item "Text"

3.3.3.3.4 Clicar com o mouse direito no componente "Text" criado clicar em "Rename"

3.3.3.3.5 Digitar Text_Pontuacao

3.3.3.4 Configurar o componente Text_Pontuacao

3.3.3.4.1 Com o componente selecionado na hierarquia, apagar o texto automático e escrever "0 x 0"

3.3.3.4.2 No campo "Paragraph", selecionar as opções para centrar o texto horizontalmente e verticalmente

3.3.3.4.3 Ainda no inspector, abrir a sub-aba Anchors (no componente Rect Transform) e alterar seus valores para: Min (X; Y) = (0.5; 1) e Max (X; Y) = (0.5; 1) Ou: arrastar a cruz (X) exibida do centro da tela para a quina superior da tela (se mantendo no centro)

3.3.3.4.4 Alterar a Width no Rect Transform para 80

3.3.3.4.5 Alterar sua posição (Position) para (X = 0, Y = -40, Z = 0)

3.3.3.5 Atualizar o placar e exibir o resultado na tela

3.3.3.5.1 No editor, navegar até a pasta "Scripts" e abrir o script Engine

3.3.3.5.2 Criar as seguinte variáveis privadas no script (abaixo de onde algumas já se encontram declaradas): (i) private int numVitorias; e (ii) private int numDerrotas;

- 3.3.3.5.3 Criar a seguinte variável pública no script: (i) public Text pontuacao;
- 3.3.3.5.4 Editar o método "ExibeResultado() { ... } " para contabilizar as derrotas e vitórias e exibir o resultado na tela
 - 3.3.3.5.4.1 Adicionar logo abaixo da primeira comparação [if (ganhou == 0)] a seguinte linha de código (sem aspas):
“numDerrotas++;”
 - 3.3.3.5.4.2 Adicionar logo abaixo da primeira comparação [if (ganhou == 1)] a seguinte linha de código (sem aspas):
“numDerrotas++;” “numVitorias++;”
 - 3.3.3.5.4.3 Antes da atribuição "podeJogar = false", adicionar a seguinte linha de código: “pontuacao.text = numVitorias + " x " + numDerrotas;”
- 3.3.3.5.5 Salvar o Script Engine (Ctrl S)
- 3.3.3.5.6 No editor, selecionar o objeto Engine na hierarquia
- 3.3.3.5.7 Arrastar o objeto Text_Pontuacao para o campo "Pontuacao" no inspector

3.3.4 Tarefa 4: Corrigir um erro

- 3.3.4.1 Clicar na mensagem em vermelho que apareceu no canto inferior da tela
- 3.3.4.2 Clicar duas vezes na mensagem que apareceu na janela recém-aberta
- 3.3.4.3 Encontrar a palavra grafada erroneamente (identificada por um cor diferenciada ou sublinhada pelo editor de código em utilização)
- 3.3.4.4 Substituir a palavra errada por "jogadaAtual"
- 3.3.4.5 Salvar o Script e fechar o editor de código
- 3.3.4.6 Verificar se a mensagem de erro sumiu

3.3.5 Tarefa 5: Realizar deploy do projeto

- 3.3.5.1 Abrir o menu File e selecionar Build Settings
- 3.3.5.2 Escolhar a plataforma Android no lado esquerdo da janela aberta e pressionar "Switch Platform"

3.3.5.3 Configurar o pacote para gerar a Build

3.3.5.3.1 Pressionar o botão "Player Settings" na janela "Build Settings"

3.3.5.3.2 No Inspector, encontrar a aba "Others"

3.3.5.3.3 Alterar o campo "Package Name" para:
com.AvaliacaoIDE.Jokenpo

3.3.5.4 Configurar as cenas para serem construídas

3.3.5.4.1 Na janela Build Settings, clicar em Add Open Scenes

3.3.5.4.2 Abrir a cena "Menu" e repetir os passos 1 e 4.1 para a cena em questão

3.3.5.5 Alterar a ordem das cenas de forma que o menu fique antes da cena do jogo (basta arrastar os itens na caixa)

3.3.5.6 Pressionar o botão "Build" e escolher a pasta de destino

4. CONSOLIDAÇÃO DOS RESULTADOS

Apresenta-se nesta seção a consolidação dos resultados obtidos ao executar a avaliação Percurso Cognitivo sobre os três ambientes de desenvolvimento. Os resultados obtidos ao responder os três questionamentos propostos estão dispostos em tabelas, e a discussão dos resultados obtidos é feita logo abaixo para cada ambiente de desenvolvimento em específico. Ao final deste documento, é realizado uma discussão sobre os três ambientes de desenvolvimento.

4.1 APP INVENTOR

Nesta seção, encontram-se os dados coletados a respeito da realização das tarefas definidas para o ambiente App Inventor segundo o perfil de usuário estabelecido anteriormente . Os dados estão dispostos na Tabela 2:

Tarefa	Ele irá saber o que fazer?	Ele irá saber como fazer?	Possui feedback de sua ação?
1	Sim	Sim	Sim
2	Sim	Sim	Não

3	Sim	Sim	Não
4	Sim	Sim	Sim
5	Sim	Sim	Sim

Tabela 2: Consolidação dos resultados da avaliação do ambiente App Inventor.

O ambiente permite a programação em termos de arrastar e encaixar blocos, o que facilita a interação e a programação das tarefas. Apenas são mostradas as interfaces necessárias para cada ambiente – definir o layout da aplicação ou fazer a programação dos blocos. As tarefas, no entanto, apesar de simples, se tornam complexas para serem descritas na forma de etapas, uma vez que é necessário que o usuário procure cada elemento de interface correspondente e faça a sua assimilação.

O ambiente peca na questão de dar um feedback para as ações do usuário nas tarefas dois e três – programar uma animação e alterar um jogo pronto – desde que o usuário necessita de ou realizar a build da aplicação ou conectar o telefone pela aplicação Android disponibilizada ou iniciar o emulador para testá-la e verificar se a mudança foi efetiva ou não.

No caso da tarefa quatro, pode-se observar que as mensagens de erro são claras e facilmente identificadas. A plataforma também permite que o usuário configure se elas irão ser exibidas ou não, o que pode levar a falhas desnecessárias caso o usuário iniciante não visualize ou desabilite o botão de forma errônea.

Por fim, a realização do deploy da aplicação se dá de forma simples e provê o feedback suficiente para o usuário compreender o processo.

4.2 GDEVELOP

Nesta seção, encontram-se os dados coletados a respeito da realização das tarefas definidas para o ambiente GDevelop segundo o perfil de usuário estabelecido anteriormente . Os dados estão dispostos na Tabela 3.

Tarefa	Ele irá saber o que fazer?	Ele irá saber como fazer?	Possui feedback de sua ação?
1	Sim	Sim	Sim
2	Sim	Não	Sim

3	Sim	Sim	Sim
4	Sim	Sim	Não
5	Sim	Não	Sim

Tabela 3: Consolidação dos resultados da avaliação do ambiente GDevelop.

O ambiente peca sobre a disposição dos elementos da interface, que apesar de tentar agrupar elementos de mesma categoria e com isso exibir apenas os elementos necessários, acaba falhando e não executando bem essa tarefa. Alguns menus, como a edição de variáveis da cena, não são trivialmente acessíveis, causando confusão por parte do usuário sobre como alcançar os menus necessários.

Para a criação do projeto, existe a possibilidade de se criar um projeto vazio ou algum modelo já estabelecido e oferecido pela plataforma, além de ser necessário configurar qual a plataforma alvo para exportação. Uma vez escolhido, estes parâmetros não podem ser alterados – ou seja, caso o usuário demarque a plataforma errada é necessário apagar e criar outro projeto.

Outra questão apresentada em relação a interface é a capacidade do usuário poder organizar todos os elementos da forma que desejar. Enquanto possibilita maior flexibilidade e customização para melhorar a experiência de uso, alguns elementos inicialmente são mal posicionados, requeirindo que o usuário reorganize a interface para que outras opções venham à tona. Isso é observado na tarefa 2, em que o menu responsável por editar as animações é preenchido por uma aba do editor de imagens, necessitando que o usuário arraste e reconfigure as interfaces para poder fazer a programação das animações. Uma vez organizado, o processo é simples (apenas arrastar as imagens e soltar na área informada), e permite que o usuário veja uma prévia sobre a animação que está sendo construída.

A plataforma peca ao dar um feedback adequado sobre os erros encontrados. Na tarefa realizada – fazer uma atribuição a uma variável não definida –, não é reportado pela plataforma nenhum erro de execução – apenas não executa a ação correspondente no erro. Outro problema é ao associar expressões: é exibido que existem erros de sintaxe quando é feito um procedimento errado, entretanto não é mostrado aonde o erro se encontra.

Por fim, o deploy para Android, apesar de ser uma tarefa simples, utiliza um recurso descontinuado, e apesar de existir um procedimento experimental para exportação nativa para Android o mesmo é complexo e exige a instalação de programas de terceiros.

4.3 UNITY

Nesta seção, encontram-se os dados coletados a respeito da realização das tarefas definidas para o ambiente Unity segundo o perfil de usuário estabelecido anteriormente. Os dados estão dispostos na tabela 4.

Tarefa	Ele irá saber o que fazer?	Ele irá saber como fazer?	Possui feedback de sua ação?
1	Sim	Sim	Sim
2	Sim	Não	Sim
3	Sim	Não	Sim
4	Sim	Sim	Sim
5	Sim	Não	Sim

Tabela 4: Consolidação dos resultados da avaliação do ambiente Unity.

Considerando o perfil de usuário estabelecido, ou seja, possui um conhecimento mínimo sobre programação e orientação a objetos, espera-se que o usuário saiba o que terá de ser feito em todas as etapas, desde que são procedimentos mais conhecidos e necessários para a construção de programas utilizando o paradigma orientado a objetos e ambientes de desenvolvimento envolvidos.

Entretanto, o usuário não saberá como realizar as tarefas dois, três e cinco, já que todos os recursos estão dispostos na plataforma em formato de menus, e assim não provê ao usuário uma sequência de ações que propicie o aprendizado [3]. Aliado a essa disposição, nota-se a a quantidade de características para realizar a modificação dos componentes na aba “Inspector”. Novos usuários podem ter dificuldades e/ou receio de alterar valores indevidamente.

A plataforma no entanto apresenta um bom feedback ao usuário, demonstrando todas as modificações realizadas nos componentes através da aba Inspector na aba “Scene”. As mensagens de erro são, em maior parte, claras considerando o perfil de usuário adotado. Entretanto, nota-se que usuários completamente novatos (perfil adotado para as demais plataformas) não conseguiriam compreender e sanar o problema apresentado.

Por fim, o deploy da aplicação é mais complexo quando selecionada a plataforma móvel Android: é necessário configurar para qual plataforma será feita a exportação, e no caso de aplicações Android é necessário configurar o pacote. O feedback dado em caso de falhas nem sempre é claro, e pode acarretar em falhas sucessivas até que o erro seja sanado.

5. CONSIDERAÇÕES FINAIS

Neste documento, foi feito o relatório sobre a realização do Percurso Cognitivo sobre cinco tarefas consideradas essenciais para o aprendizado e desenvolvimento nas plataformas App Inventor, GDevelop e Unity.

Notou-se que o desenvolvimento no App Inventor, apesar de ser de simples entendimento, requer que sejam feitos mais passos para a realização de tarefas em que a plataforma não oferece nenhum recurso específico – como foi observado ao lidar com a criação de uma animação. O ambiente possibilita opções de customização básicas tanto para os componentes abordados quanto para a organização de elementos na interface – permite apenas layouts pré-definidos, requerindo que sejam utilizados “elementos fantasmas” para a criação de interfaces mais complexas ou realizar o desenho diretamente no Canvas da aplicação (como realizado na tarefa 3). O estilo de programação oferecido e as imagens de erro claras permitem que as tarefas sejam realizadas de maneira eficaz e possibilita o aprendizado sobre conceitos de lógica de programação para pessoas que estão começando a compreender programação.

No caso do ambiente GDevelop, a programação via eventos aproxima o desenvolvimento do desenvolvimento nativo, ou seja, é possível notar que os eventos formam implicitamente a função de loop de jogo, mascarada no ambiente Unity e não existente (a não ser que seja feita a sua criação pelo usuário) no ambiente App Inventor. Os eventos e ações básicos utilizados para a realização das avaliações se assimilam a estruturas de seleção de dois caminhos presentes em linguagens de programação, e permite que seja feita a preparação para outras plataformas de desenvolvimento que requerem um menor nível de abstração. Foi observado também durante a avaliação que o ambiente possui uma vasta gama de eventos e assim permite que sejam criadas lógicas mais complexas. No entanto, a plataforma peca em lidar com o feedback ao usuário com relação a erros e na organização dos itens da interface influenciando negativamente o aprendizado e a experiência do usuário.

Por fim, no ambiente Unity, é necessário que o usuário tenha um conhecimento a respeito de programação e Orientação a Objetos para compreender e utilizar a plataforma adequadamente. Foi observado que a disposição dos elementos de interface para o acesso a suas funcionalidades não propicia um meio sólido para o aprendizado, necessitando que o

usuário busque por tutoriais a respeito de recursos específicos para seu entendimento. A plataforma permite que os objetos criados na aplicação sejam customizados o quanto necessários por uma aba no canto direito da tela, e a sua customização é refletida imediatamente na aba gráfica do editor, permitindo que o usuário tenha um feedback sólido sobre as suas ações na plataforma. Este recurso, aliado a programação via código, permite que o usuário crie aplicações mais complexas do que o permitido nos outros ambientes. Um dos problemas da plataforma são as mensagens de erro, que em forma de exceção pode confundir usuários novatos. Por fim, destaca-se que o procedimento para a exportação da aplicação para o ambiente Android é mais trabalhosa no ambiente Unity, necessitando de mais passos e configurações para sua consolidação.

6. REFERÊNCIAS

- [1] BARBOSA, S. D. J.; SILVA, B. S. *Interação Humano Computador*. 1ª Edição. Rio de Janeiro, 2010.
- [2] PREECE, J.; ROGERS, Y.; SHARP, H. *Interaction design: beyond human-computer interaction*. 1ª Edição, Nova York, 2002.
- [3] SHNEIDERMAN, Ben. *Promoting Universal Usability with Multi-Layer Interface Design*.