

EAD
UNISANTA

PROJETO INTEGRADOR IV
Me. Claudio Ferreira de Carvalho

**GUIA DA
DISCIPLINA**

1. CRIANDO E MOVIMENTANDO UM JOGADOR

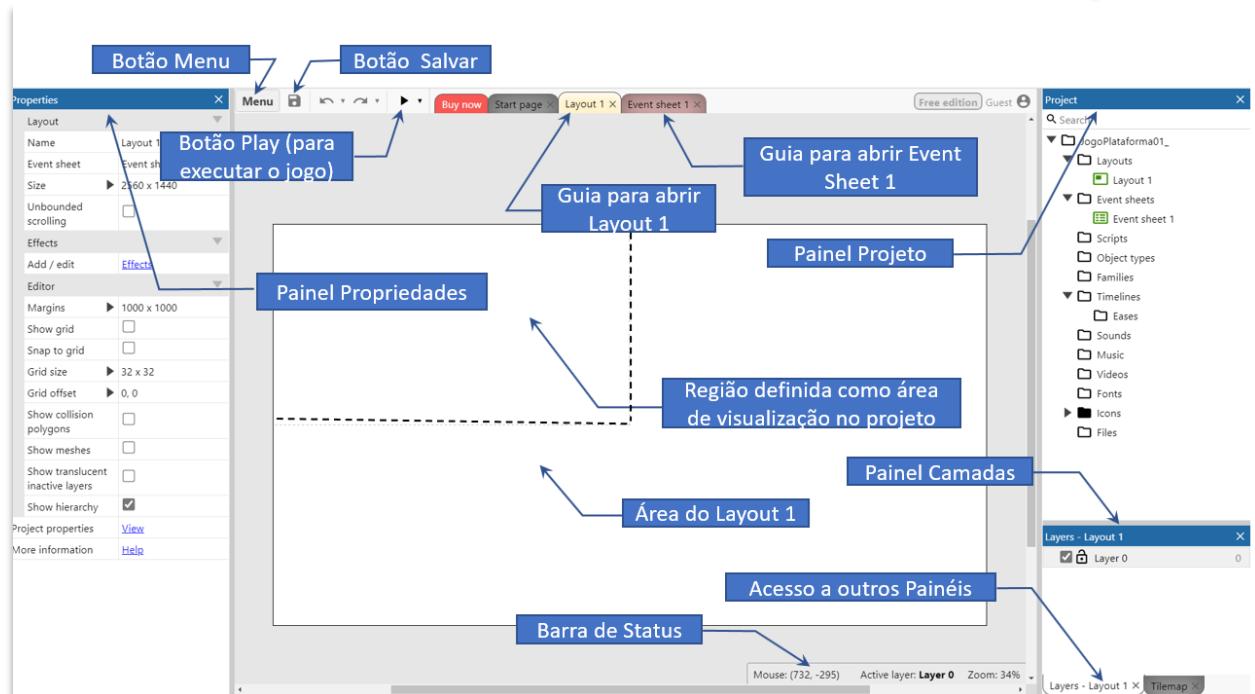
Apresentação:

Nesta aula serão discutidos os elementos básicos do Construct 3, tais como: Layout, onde serão colocados os elementos que farão parte jogo; o Event Sheets (Folha de Eventos), onde serão colocadas as instruções (a programação com eventos e ações) que o desenvolvedor deseja para executar o jogo; modos de gravação; criação e nomenclatura de Camadas; configuração de uma camada chamada de “GUI” para elementos fixos na tela e modo de nomear componentes. Serão também apresentados os elementos básicos da Área de Trabalho, as funções dos painéis e as possibilidades de distribuição destes na tela. Serão inseridos um jogador e um elemento básico para conter este jogador na tela, assim como, discutido ações para a movimentação básica do jogador.

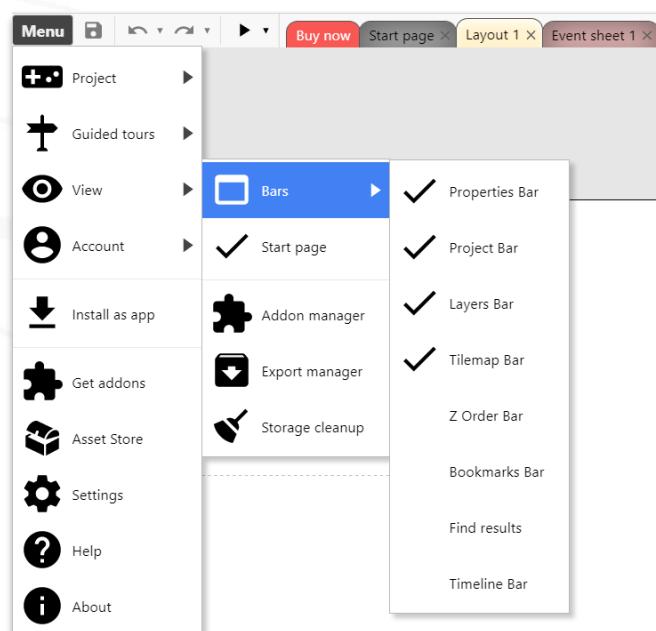
1.1. Desenvolvimento

1. Criar na Área de trabalho (ou em outro local desejado), uma pasta de nome **Construct**
2. Abrir o Construct 3 através do endereço: <https://editor.construct.net/>
3. Se necessário passar o idioma do editor para inglês,
 - a. Menu
 - b. Configurações
 - c. English (US)
 - d. Fechar
4. Fazer login com seu usuário
 - a. Guest
 - b. Log in
 - c. Preencher Username e Password
 - d. Login
5. Clicar no botão **New**
6. No quadro “New Project” preencher:
 - a. Name: **JogoPlataforma**
 - b. Em Choose preset: **SD Landscape 16:9**
 - c. Viewport size: **1280 x 720**
 - d. Orientations: **Landscape**
 - e. Start with: **Event Sheet**

- f. Clicar em **Create**
7. Gravar o projeto utilizando: **Menu; Project; Save as; Save as a single file.**
 8. Escolher a pasta **Construct** que deve estar na área de trabalho de seu computador
 9. Nomear como **JogoPlataforma01** (Em cada uma das aulas deste curso será criado um arquivo, com o nome de JogoPlataforma seguido do número da aula).
 10. Utilizando as barras de rolamento e o Scroll do Mouse com a tecla Control açãoada, ajustar a área de trabalho para que fique conforme figura a seguir:



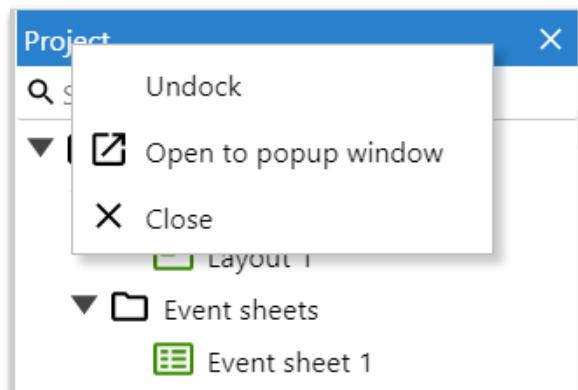
11. Clicar em **Menu** e verificar as Barras disponíveis e as que estão visíveis.



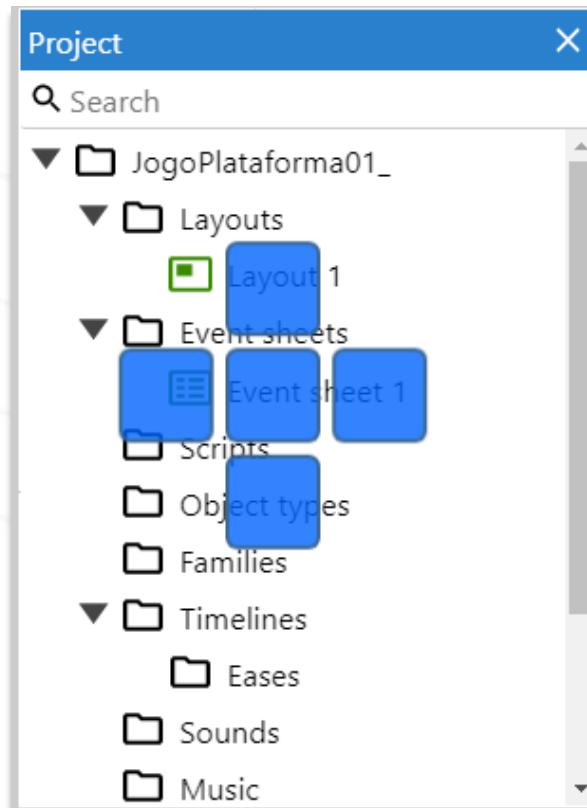
Obs: Esta opção é importante para recolocar barras que acidentalmente tenham sido fechadas.

Obs: É importante notar que algumas barras não estão disponíveis na versão free.

1. Caso deseje é possível escolher as posições dos Paineis, assim como esconder painéis. Para fazer um painel flutuar e ficar livre na tela, basta clicar com o botão direito do Mouse e escolher “Undock”.



2. O painel desacoplado flutua pela tela e pode ser colocado em diversas posições de acordo com as setas que aparecem indicando as quatro direções possíveis em diversos locais da área de Trabalho



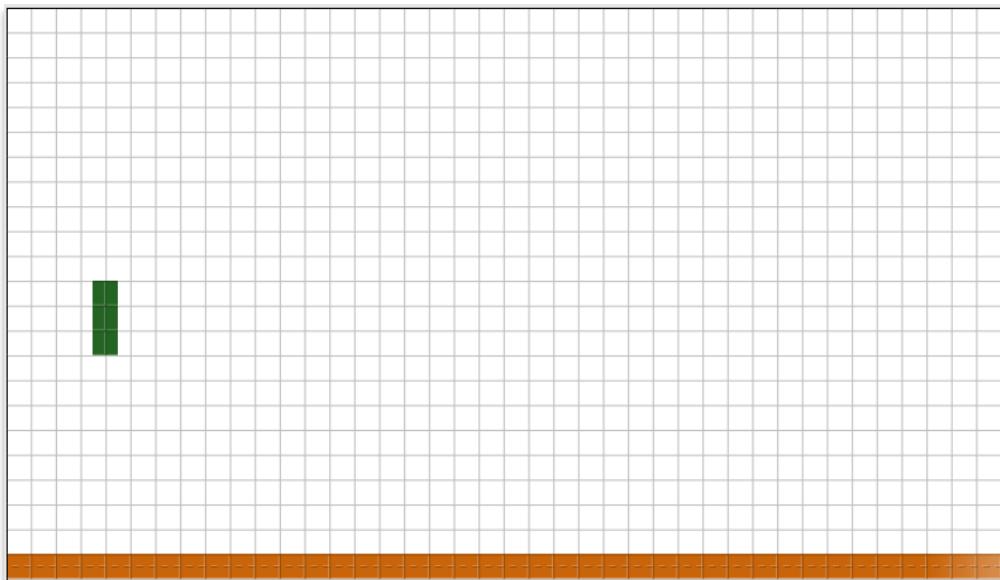
3. Para facilitar o posicionamento dos objetos na tela é aconselhável que as linhas de grade no layout sejam mostradas (chamadas de malha) assim como a opção “Ajustar a grade” (chamado de Alinhar à malha) sejam ativadas conforme a seguir:
 - a. No Painel Project clicar em **Layout 1**.
 - b. Alterar as dimensões do **layout em Size**, para 1920 x 720. (este é o tamanho da área de trabalho).
 - c. Com o **Layout 1** em evidência passar para o **Painel Properties** e marcar **Show grid** e **Snap to grid**.
 - d. Manter **Grid size** em 32 x 32
4. Completar algumas propriedades do projeto conforme a seguir:
 - a. No Painel **Project** marcar “JogoPlataforma”
 - b. Com o **JogoPlataforma** marcado passar para o **Painel Properties** e alterar propriedades confirme a seguir:
 - i. Version: Manter 1.0.0.0.
 - ii. Description: Jogo de Plataforma - Curso Desenvolvimento de Jogos.
 - iii. ID: br.eti.meusite.jogoplataforma (se possuir um domínio pode colocar o seu).
 - iv. Author: (nome do aluno).
 - v. Email: (colocar seu email)
 - vi. WebSite: <https://meusite.eti.br> (se possuir um domínio pode colocar o seu)
5. Passar para o Painel **Project** marcar Layout 1 e renomear para **lay_1**.
 - a. Clicar com o botão direito do Mouse em Layout 1 e no menu que surgir escolher “**Rename**”.
6. Ainda no Painel **Project** marcar Event Sheet 1 e renomear para **eve_1**.
 - a. Clicar com o botão direito do Mouse em Event Sheet 1 e no menu que surgir escolher “**Rename**”.
7. Passar para o Painel **Layers**, criar mais uma camada e renomear conforme a seguir:
 - a. Clicar com o botão direito dentro do Painel e escolher a opção “**Insert layer above**”
 - b. Renomear a layer 0 para **GUI** onde serão colocados os elementos que irão informar os dados do jogo, tais como: mensagens pontuação, número de vidas, bônus etc.

- c. Renomear a layer 1 para **Elementos** onde ficarão os demais elementos do jogo.
8. Passar para a layer GUI e no Painel Propriedades configurar:
 - a. Scale rate: **0%**
 - b. Parallax: **0% x 0%**

Obs: Para colocar os dados em Parallax é possível clicar na seta que existe ao lado do nome Parallax e colocar os valores em X e Y

Obs: estas configurações farão com que as informações colocadas nesta camada não sofram deslocamento permanecendo na tela, mesmo que a camera acompanhe o jogador.
9. Passar para a **Layer Elementos** e colocar um objeto obedecendo a sequência
 - a. Clicar com o **botão direito** na área de trabalho.
 - b. Escolher **Insert new object**
 - c. Type: Sprite.
 - d. Name: **spr_Jogador**
 - e. Clicar o Botão Insert.
 - f. Clicar sobre a área de trabalho.
10. Configurar o Sprite Jogador conforme a seguir:
 - a. Resize: (Dimensionamento): **32 x 96 pixels**.
 - b. Fill: (cor do sprite):(Red: 35; Green: 100; Blue: 35) Alfa: 255 (indica a transparência).
 - c. Edit the image points: (Ponto de referência da imagem):
 - i. Clicar com o Botão direto em Origin
 - ii. No menu que surgir escolher “Quick Assign”.
 - iii. No novo menu escolher “Bottom”.
 - d. Fechar o “Animations Editor” (Editor de Animações) e notar que o objeto spr_Jogador surge na lay_1.
11. Configurar o Sprite spr_Jogador para que ele obtenha o comportamento de plataforma conforme a seguir:
 - a. **Selecionar** o Sprite spr_Jogador
 - b. No Painel Proprieties, clicar em **Add / edit Behaviors**.
 - i. No quadro de diálogos clicar em **Add new behavior**
 - ii. No quadro que surgir escolher a opção: **Plataform**
 - iii. Clicar no botão “**Add**”
12. Ainda na **Layer Elementos** colocar um novo objeto:

- a. Type: Sprite.
 - b. Name: **spr_Chao**
 - c. Formato: Retângulo (ajustar na figura).
 - d. Cor: Marrom: (**Red: 200; Green: 100; Blue: 10**) Alfa: 255
 - e. Image point: Top left (0 x 0 pixels).
 - f. Behavior: Solid
13. Posicionar os objetos conforme figura.

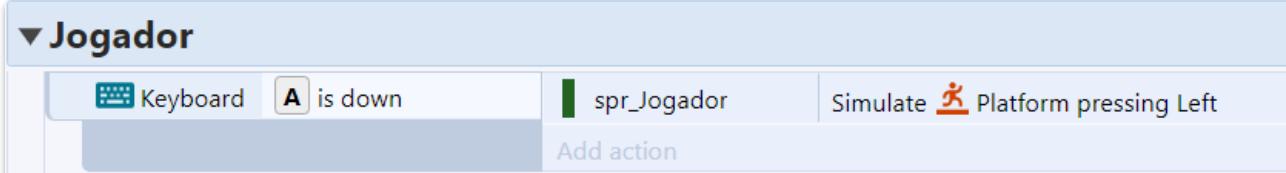


Obs: Somente área do jogo (dentro dos riscos pontilhados)

14. Testar o jogo clicando no botão Play e notar que o Jogador pode ser movimentado com as setas do teclado.
 15. Embora o Jogador possa ser movimentado com as setas do teclado, é muito comum em jogos utilizar também as teclas A, D, W para movimentar o jogador. Para que isto seja possível, será colocada uma programação na Event Sheet **eve_1**. Como esta programação será no teclado, o elemento **Teclado** precisará ser incluído como **Entrada** a partir de:
 - a. Clicar com o Botão direito => Insert new object => Input => Keyboard.
- Obs: Notar que o Teclado foi inserido no Painel Projeto
16. Passar para a eve_1
 17. As instruções de programação do Construct são executadas com a metáfora de “**Eventos**” que ao acontecerem disparam “**Ações**”. Tanto os “**Eventos**” como as “**Ações**” devem ser atribuídas a elementos disponíveis no projeto (tais como os sprites, teclado, o próprio sistema etc.). Então, primeiramente devem ser

escolhidos o(s) **Evento(s)** e em seguida a(s) “**Açõe(s)**” que se espera quando estes eventos acontecerem.

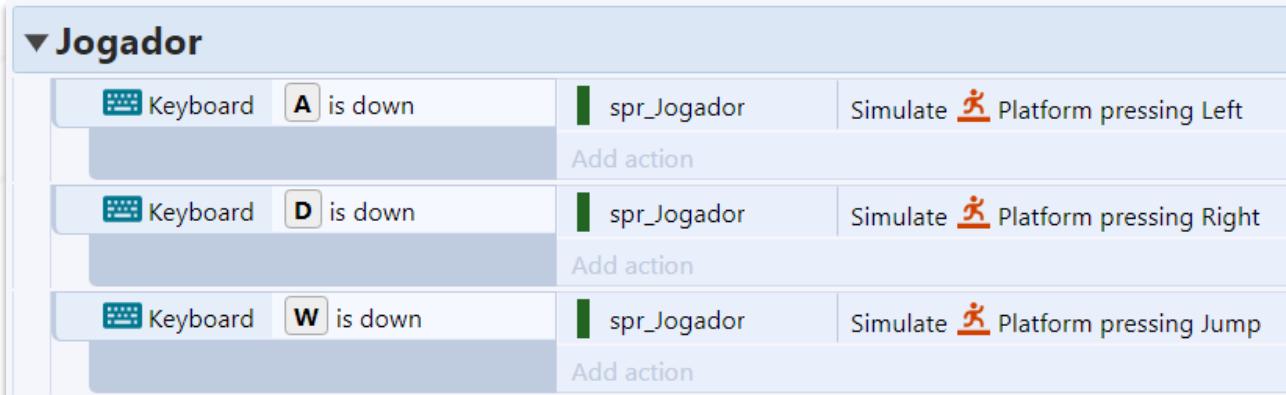
18. Neste material serão apresentadas sempre a sequência de instruções para o Evento e em seguida a ação ou ações que se deseja. Cada uma das instruções desta sequência será separada pelo símbolo =>. Dependendo do necessário, mais de um evento pode ser executado para uma ação e várias ações podem ser requeridas para o mesmo evento.
19. Para facilitar ações futuras pode ser interessante agrupar eventos e ações e criar nomes para estes grupos, neste jogo será inicialmente criado um grupo para colocar os eventos do jogador com o nome Jogador:
 - a. Clicar o Botão direito (em eve_1) => Add group
 - i. Name: Jogador
20. No grupo Jogador, criar um sub-evento para **ao clicar a tecla A (evento), mover o Jogador para a esquerda (ação)**:
 - a. Clicar com o Botão direito no grupo e escolher => Add event to “Jogador”
 - i. **Evento:** Keyboard => Key is down => Tecla A
 - b. Clicar em Add action.
 - i. **Ação:** spr_Jogador => Simulate control => Left
21. Após a execução, a Folha de eventos ficará conforme figura



```

when green flag clicked
    [Keyboard v] [A v] [is down v]
        [spr_Jogador v]
            [Simulate v Platform pressing Left v]
    end
end
  
```

22. Obedecendo a mesma sequência de instruções do item anterior, criar os demais sub-eventos para mover o Jogador para a direita e para ele pular. A folha de eventos ficará conforme figura:

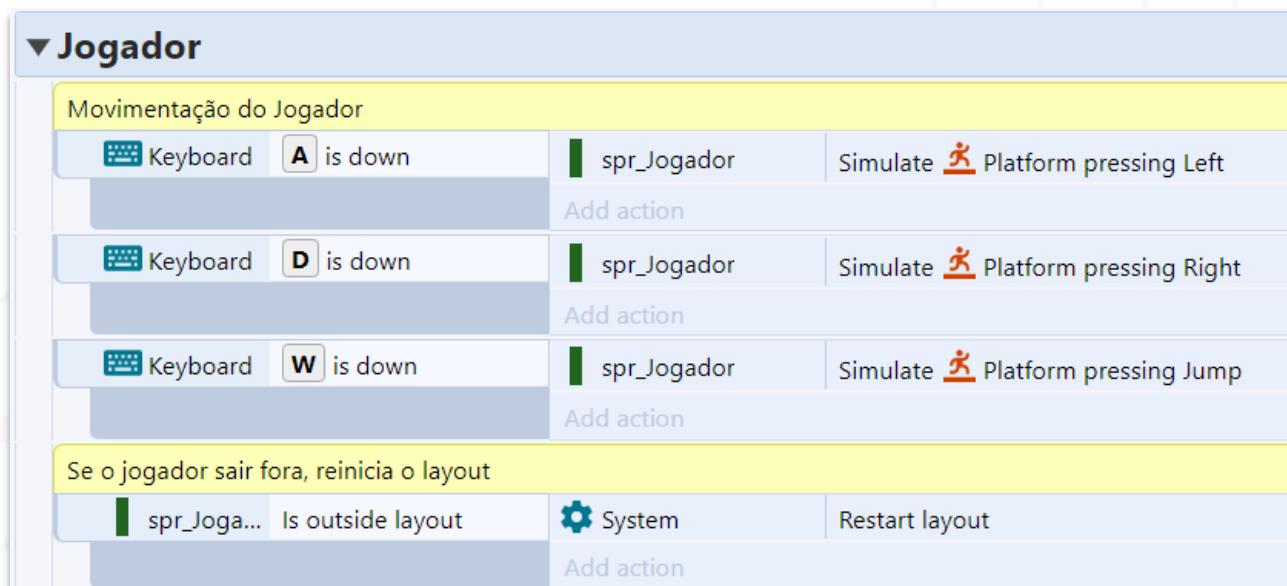


```

when green flag clicked
    [Keyboard v] [A v] [is down v]
        [spr_Jogador v]
            [Simulate v Platform pressing Left v]
    end
    [Keyboard v] [D v] [is down v]
        [spr_Jogador v]
            [Simulate v Platform pressing Right v]
    end
    [Keyboard v] [W v] [is down v]
        [spr_Jogador v]
            [Simulate v Platform pressing Jump v]
    end
end
  
```

Atenção que: Os eventos de Teclado devem ficar no grupo Jogador. Caso eles não estejam, arraste as instruções para as colocar corretamente.

23. Testar o jogo e verificar que a movimentação tanto pode ser feita com as setas como com as letras A, D, W.
24. Caso o jogador saia do layout o jogo é encerrado e para estas configurações iniciais isto não é bom. Para resolver este problema será acrescentado ao jogador o seguinte evento:
 - a. Evento: spr_Jogador => Is outside layout.
 - b. Ação: System => Restart layout
25. Com objetivo de facilitar o entendimento dos eventos é sempre interessante colocar comentários. Para criar comentários basta:
 - a. Clicar com o botão direito no Grupo Jogador => Add => Add comment
 - i. Antes da movimentação do jogador, colocar o comentário “Movimentação do Jogador”.
 - ii. Antes da instrução que reinicia o layout colocar o comentário “Se o jogador sair fora, reinicia o layout”
26. Após todas as implementações, os eventos ficarão conforme figura.



27. Salvar o projeto.
28. Fechar o Construct.

1.2. Exercício Aula 01

1. Tente refazer todo o jogo anterior, pintando o jogador de outra cor e modificando os “Pontos de referência das Imagens”. Observe a variação do posicionamento dos elementos à medida que você altera os “Pontos de referência das Imagens”.
2. Em seguida utilize a opção “Edite the collision polygon” do “Editor de Animação” do sprit spr_Jogador e modifique o polígono, de maneira que o jogador pare na metade do spr_Chao.
3. Observe também que nas condições acima, o jogador fica atras do chão, altere esta posição utilizando no “Menu suspenso do jogador” a opção “Z Order”
4. Grave este novo jogo utilizando a opção **Save as a single file** com o nome **“ExercicioJogoPlataforma01.c3p”** (notar que o C3p é a extensão colocada automaticamente pelo Construct).

2. PLATAFORMAS

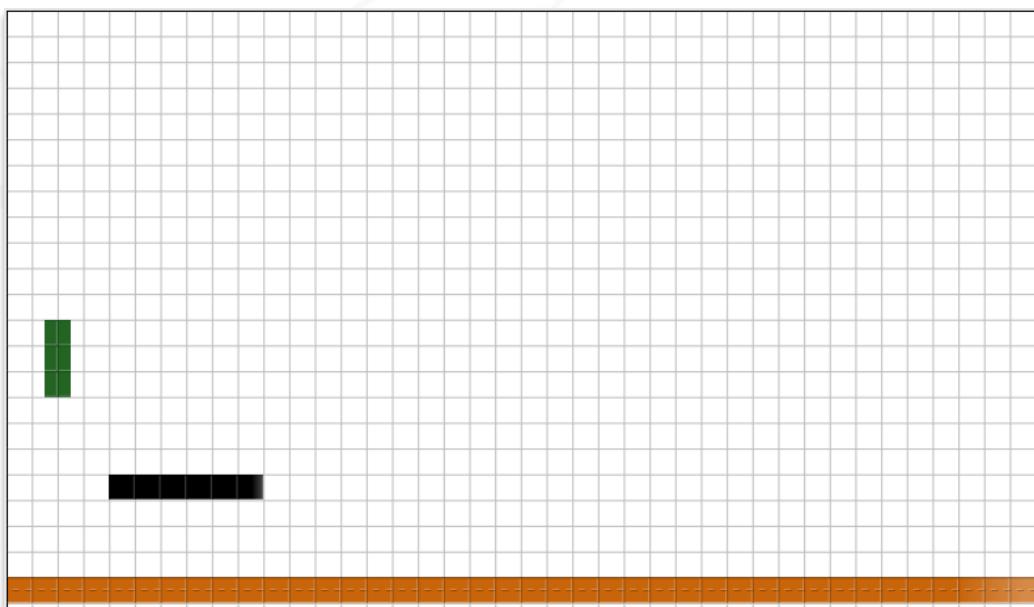
Apresentação:

Nesta aula serão inseridas no jogo criado na aula anterior plataformas sólidas e do tipo vasada. Serão também apresentados formatos de movimentos das plataformas.

2.1. Desenvolvimento

1. Abrir o Construct 3 através do endereço: <https://editor.construct.net/>
2. Se necessário passar o idioma do editor para inglês,
 - a. Menu
 - b. Configurações
 - c. English (US)
 - d. Fechar
3. Fazer login com seu usuário
 - a. Guest
 - b. Log in
 - c. Preencher Username e Password
 - d. Login
4. Abrir o projeto **JogoPlataforma01**
5. Gravar o projeto utilizando: **Menu; Project; Save as; Save as a single file**, com o nome **JogoPlataforma02**
6. Passar para **lay_1**
7. Passar para a **Layer Elementos** e colocar um objeto
 - a. Type: Sprite
 - b. Name: **spr_Plataforma**
 - c. Formato: Retângulo (ajustar na figura).
 - d. Cor: Preto. (**Red: 0; Green: 0; Blue: 0**) Alfa: 255
 - e. Image point: Top left (0 x 0 pixels).
 - f. Behavior: Solid

8. Colocar a plataforma conforme figura.

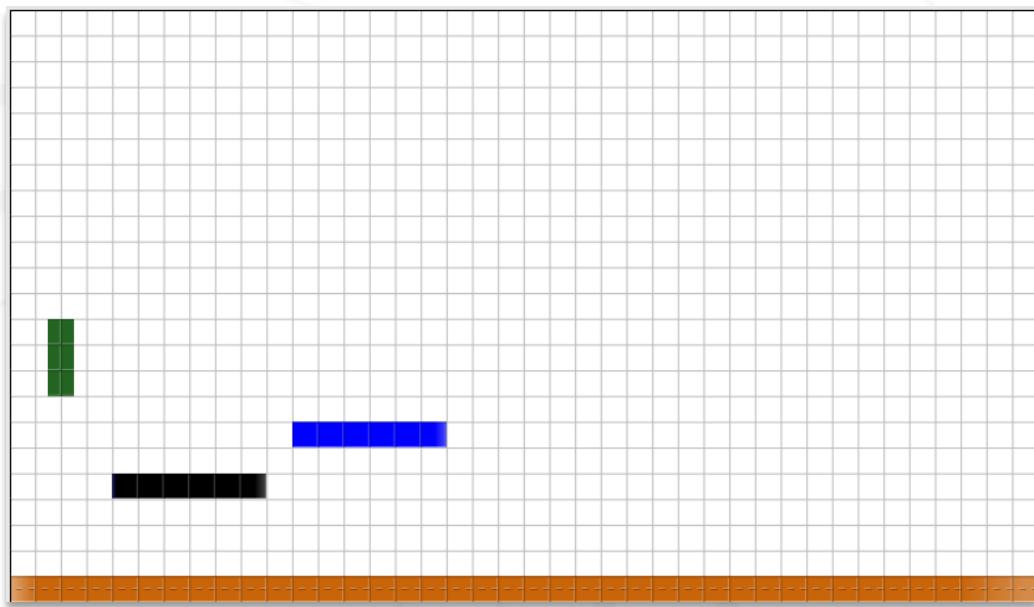


9. Testar o jogo e observar que o novo Sprite funciona como uma plataforma.

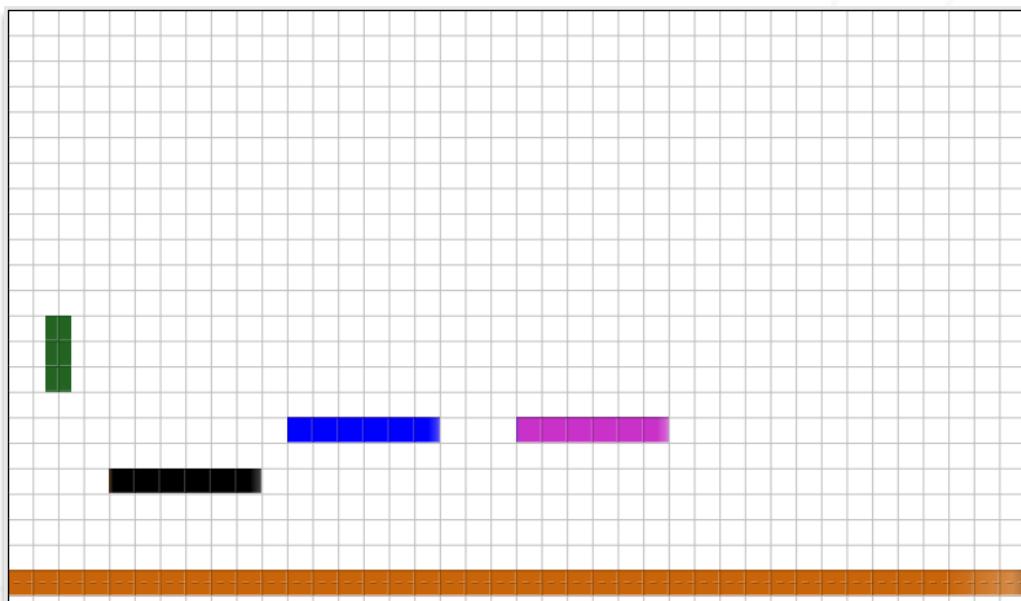
10. Ainda na **Layer Elementos** colocar outro objeto

- a. Type: Sprite
- b. Name: **spr_Plataforma_Mov_Hor**
- c. Formato: Retângulo (ajustar na figura).
- d. Cor: Azul: (**Red: 0; Green: 0; Blue: 255**) Alfa: 255
- e. Image point: Top left (0 x 0 pixels).
- f. Behavior: Solid
- g. Behavior: Sine

11. Colocar a plataforma conforme figura,

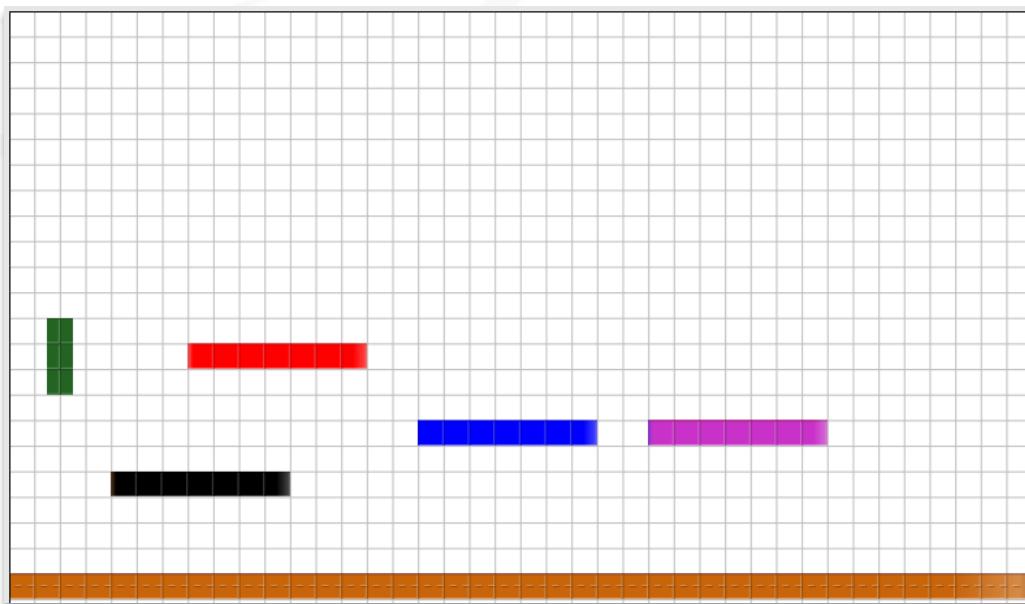


12. Testar o jogo e observar que o novo Sprite funciona como uma plataforma com movimento horizontal.
13. Ainda na **Layer Elementos** colocar outro Sprite
 - a. Type: Sprite
 - b. Name: **spr_Plataforma_Mov_Ver**
 - c. Formato: Retângulo (ajustar na figura).
 - d. Cor: Rosa. (**Reed: 200; Green: 100; Blue: 200**) Alfa: 255
 - e. Image point: Top left (0 x 0 pixels).
 - f. Behavior: Solid
 - g. Behavior: Sine
 - h. Movement: Vertical.
14. Colocar a plataforma conforme figura.



15. Testar o jogo e observar que o novo Sprite funciona como plataforma, porém com movimento vertical.
16. Ainda na **Layer Elementos** colocar outro Sprite
 - a. Type: Sprite
 - b. Name: **spr_Plataforma_Vasada**
 - c. Formato: Retângulo (ajustar na figura).
 - d. Cor: Vermelha. (**Red: 255; Green: 0; Blue: 0**) Alfa: 255
 - e. Image point: Top left (0 x 0 pixels).
 - f. Behavior: Jumpthru
 - g. Behavior: Sine

17. Redistribuir as plataformas conforme figura (notar que as dimensões foram alteradas).



18. Testar o jogo e observar que o novo Sprite funciona como uma plataforma quando um sprite cai sobre ela, mas pode ser atravessada de baixo para cima.
19. Caso o jogador não consiga alcançar a parte superior da plataforma ative o “Double-jump” no jogador em
- Behavior Plataform
 - Double-jump
20. Salvar o projeto.
21. Fechar o Construct.

2.2. Exercício Aula 02

- Crie um jogo com layout semelhante ao desenvolvido, mas altere tanto os tipos de plataformas como as movimentações. Faça várias experiências e procure explorar os diferentes tipos de movimentos disponíveis para plataformas
- Grave este novo jogo utilizando a opção **Save as a single file** com o nome **“ExercicioJogoPlataforma02.c3p”** (notar que o c3p é a extensão colocada automaticamente pelo Construct).

3. CRIANDO UM OBJETIVO PARA O JOGO

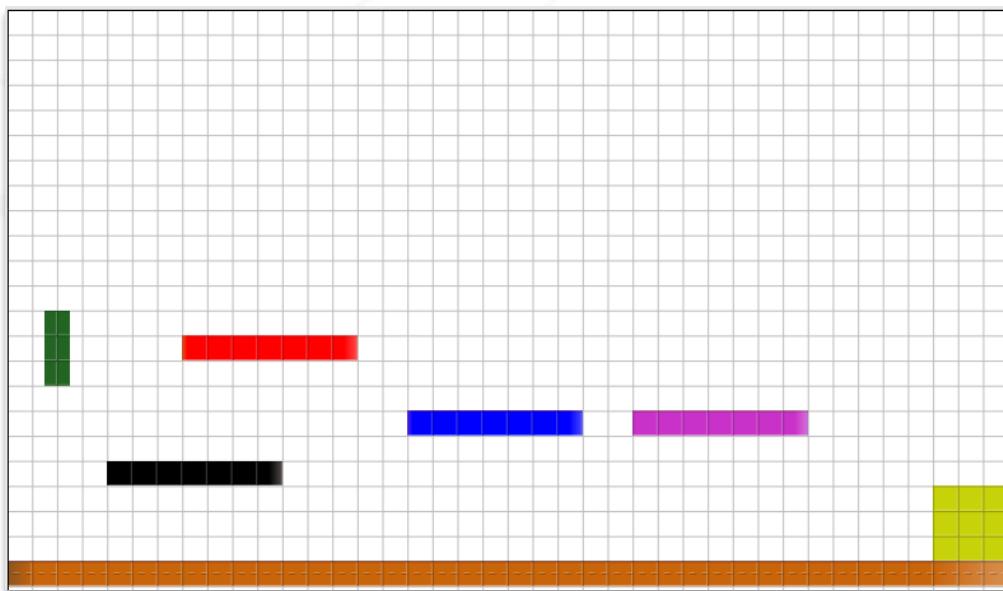
Apresentação:

Nesta aula será criado um objetivo para o jogo, pois, simplesmente movimentar o jogador pela tela, não apresenta a principal característica de um jogo que é existir um desafio para ser cumprido pelo jogador. Para o momento, o objetivo será simplesmente chagar ao final da tela (canto direto) e encontrar um objeto, que futuramente será substituído por uma figura.

3.1. Desenvolvimento

1. Abrir o Construct 3 através do endereço: <https://editor.construct.net/>
2. Se necessário passar o idioma do editor para inglês,
 - a. Menu
 - b. Configurações
 - c. English (US)
 - d. Fechar
3. Fazer login com seu usuário
 - a. Guest
 - b. Log in
 - c. Preencher Username e Password
 - d. Login
4. Abrir o projeto **JogoPlataforma02**
5. Gravar o projeto utilizando: **Menu; Project; Save as; Save as a single file**, com o nome **JogoPlataforma03**
6. Passar para **lay_1**
7. Passar para a **Layer Elementos** e colocar um objeto.
8. Para vencer o jogo, o jogador deverá atingir um objeto que será colocado no canto do layout.
 - a. Typo: Sprite
 - b. Name: **spr_Objetivo**
 - c. Formato: Retângulo (ajustar na figura).
 - d. Cor: Amarelo. (**Red: 200, Green: 210, Blue: 10**); (Alfa 255).
 - e. Image point: Top left (0 x 0 pixels).
 - f. Behavior: Solid

9. O jogo ficará com a tela conforme figura.



10. Testar o jogo e observar que utilizando as plataformas o jogador poderá chegar no objeto que é o Sprite amarelo.
11. Entretanto, chegar no objeto amarelo está muito fácil. As ações a partir de agora visam dificultar a chegada do jogador no ponto amarelo.
12. Executar as seguintes alterações:
- Aumentar a altura da plataforma vasada.
 - Diminuir o tamanho do chão de maneira que, caso o jogador caia entre as plataformas o jogo seja reiniciado.
 - Duplicar o chão (utilizando Ctrl + C e Ctrl + V) para que exista um pequeno pedaço abaixo do objeto. Isto pode ser feito também arrastando, a partir do Painel Project o Sprite spr_Chao para dentro da área do Jogo.



Importante

Observar, no Painel Project, que não foi criado um novo objeto. É exatamente isso que se deseja, pois, este será o mesmo objeto (chão) igual ao anterior colocado duas vezes no jogo. Caso se desejasse um novo objeto com as mesmas características iniciais deste objeto, mas com nome diferente, deveria ser utilizado o item “Clone” que pode ser obtido a partir do menu suspenso obtido ao clicar com o botão direito do mouse no objeto.

13. Após as alterações a tela do jogo deverá ficar conforme figura



14. Testar o jogo e verificar que quando o jogador cair entre as plataformas o jogo é encerrado.
15. Salvar o projeto.
16. Fechar o Construct 3.

Exercício

Abrir o aplicativo **ExercicioJogoPlataforma03** e adicionar o movimento com o toque. Como sugestão você talvez possa mudar as cores dos botões ou mesmo escolher outros botões utilizando imagens da Internet. Uma sugestão de site com outros elementos gratuitos pode ser <https://www.kenney.nl/assets> Outra sugestão de site para obter imagens e sons é o <https://www.gameart2d.com/freebies.html>

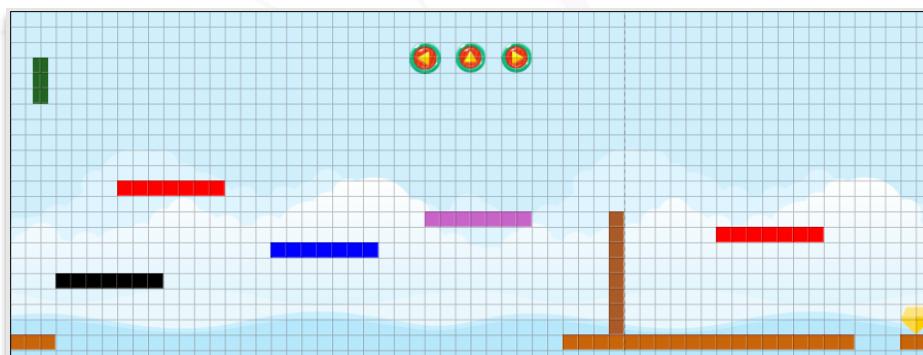
Nomear o jogo como **ExercicioJogoPlataforma04**

- 1) Criar um segundo layout
 - a. Inserir com Add layouts.
 - b. Adicionar também uma “Add event sheet”
 - c. Renomear o Layout para **lay_2**
 - d. Renomear Folha de Eventos para **eve_2**
- 2) No lay_2 alterar o layout Tamanho (Size) para 1920 x 720
- 3) Criar as mesmas camadas do lay_1 (GUI, Elementos).

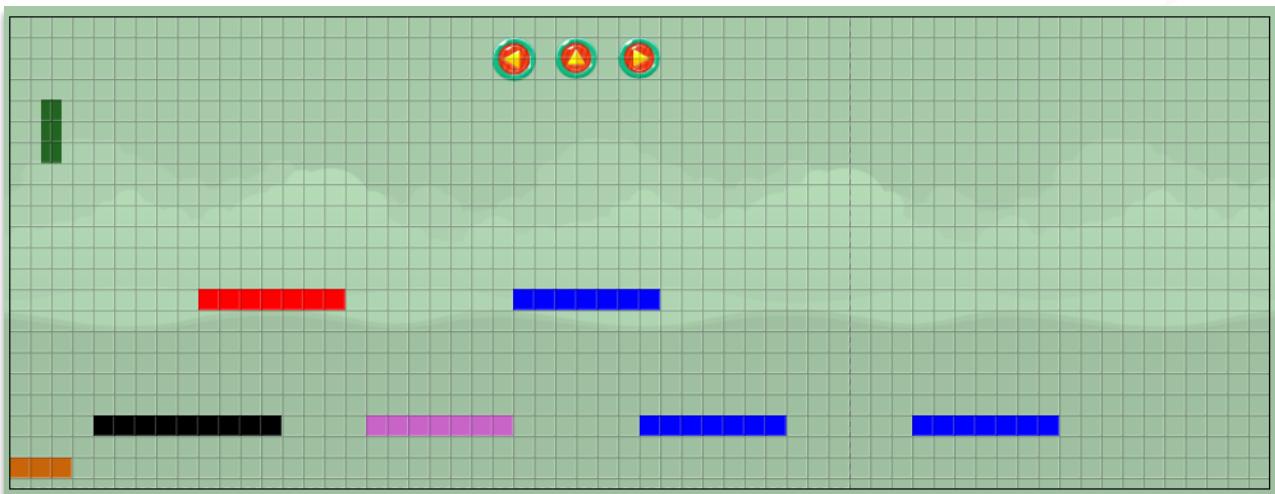
- 4) Dar um duplo clique na Folha de Eventos eve_2 para que ele também apareça na Barra de Layouts e Eventos.
- 5) A barra de Layouts e eventos ficará conforme figura abaixo:



- 6) Inserir, no layout lay_1 camada Elementos, utilizando Tiled Background a imagem bak_lay1 e redimensioná-la para que fique um pouco maior que os layouts. Em seguida colocar o background atrás das imagens.
 - a. Passar para o layout lay_1
 - b. Passar para a camada Elementos
 - c. Botão direito do Mouse => “Insert new object” => Tiled Background
 - d. Load images from file
 - e. Escolher bak_lay1
 - f. Redirecionar (maior que o lay_1)
 - g. Clicar com o botão direito em bak_lay1
 - h. Escolher => Z order
 - i. Escolher => Send to bottom of layer
- 7) Inserir no lay_1 os Botões para utilização como Touch conforme na parte teórica:
 - a. Colocar os 3 botõessm na layer GUI do Layout lay_1 nomeando-os de btn_Esquerda, btn_Direita e btn_Saltar
- 8) Desativar a opção “Defout controls” no jogador.
- 9) Inserir a partir do lay_1 o plugin Touch para que ele possa ser utilizado na programação dos Event Sheets.
- 10) Passar para o eve_1 e colocar a programação para que além do jogador ser movimentado com as teclas A, D e W ela também seja movimentada com as setas do teclado e com o Touch do Mouse nos botões.
- 11) O layout lay_1 ficará semelhante à imagem seguir:



- 12) Adicionar no layout 2 (lay_2) camada “Elementos”, algumas das plataformas já existentes em “Object Types”, alterando dimensões e posicionamentos. (Observar que não são novas plataformas e estes elementos não devem ser duplicados nem clonados). Coloque também os botões de movimentação.
- 13) Coloque o background bak_lay2 no layout 2 da mesma maneira que foi colocado o bak_1 no lay_1
- 14) O layout lay_1 ficará semelhante à imagem seguir:



- 15) Passar para o eve_1 e inserir o evento que fará com que o jogador ao colidir com o spr_Ojetivo passe para a segunda fase (layout 2 => lay_2):
 - a. Inserir no grupo jogador o comentário “Se o jogador atingir o objetivo, passa para a segunda fase”
 - b. Inserir o evento: spr_Jogador => on collision with =>spr_Ojetivo
 - c. Adicionar a ação => System => Go to layout (by name) => “lay_2”
- 16) Testar o jogo e verificar que, embora ao atingir o spr_Ojetivo o jogador vá para o Lautoyt lay_2 ele não pode ser movimentado pelas teclas o mesmo com Touch nos botões. Isso está acontecendo porque o Layout 2 que tem nome lay_2 está seguindo as instruções do eve_2 e as instruções de movimentação do jogador estavam no eve_1. Para fazer com que as instruções do eve_1 sejam utilizadas no eve_2 basta no eve_2 adicionar “Include event Sheet” e no quadro de mensagens que surgir escolher eve_1.

17) O Event Sheet eve_1 ficará conforme figura:

▼ Jogador

Movimentação do Jogador			
Keyboard	A is down	spr_Joga...	Simulate Platform pressing Left
Touch	Is touching btn_Esquerda	Add action	
Keyboard	D is down	spr_Joga...	Simulate Platform pressing Right
		Add action	
Keyboard	W is down	spr_Joga...	Simulate Platform pressing Jump
		Add action	
Se o Jogador sair fora, reinicia o layout			
spr_Joga...	Is outside layout	System	Restart layout
		Add action	
→ spr_Joga...	On collision with spr_Objetivo	System	Go to layout "lay_2"
		Add action	

18) Clone o sprite objetivo, de maneira que ele possa, no segundo layout receber uma nova imagem, que será a do botão com uma taça de vencedor. Mantenha o nome sugerido “spr_Objetivo2”. Coloque este elemento (spr_Objetivo2) no lay_2.

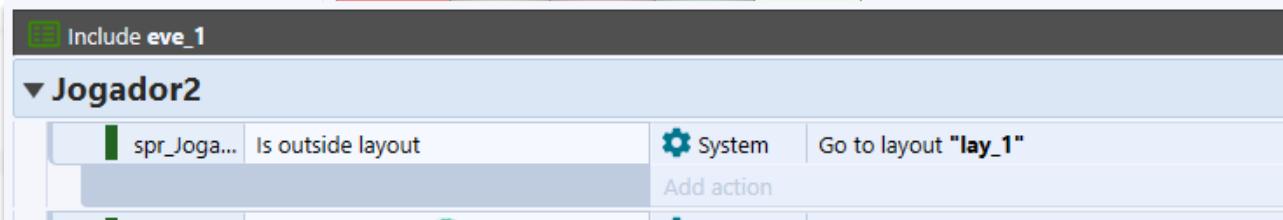
19) Após estas ações o layout 2 (lay_2) deverá ficar semelhante à figura abaixo.



20) Crie no Event Sheet 2 (eve_2) um grupo Jogador2 e insira o evento que fará o jogador voltar ao lay_1 se ele sair fora do layout no lay_2.

- Inserir o evento: spr_Jogador => is outside layout
- Adicionar a ação => System => Go to layout (by name) => “lay_1”

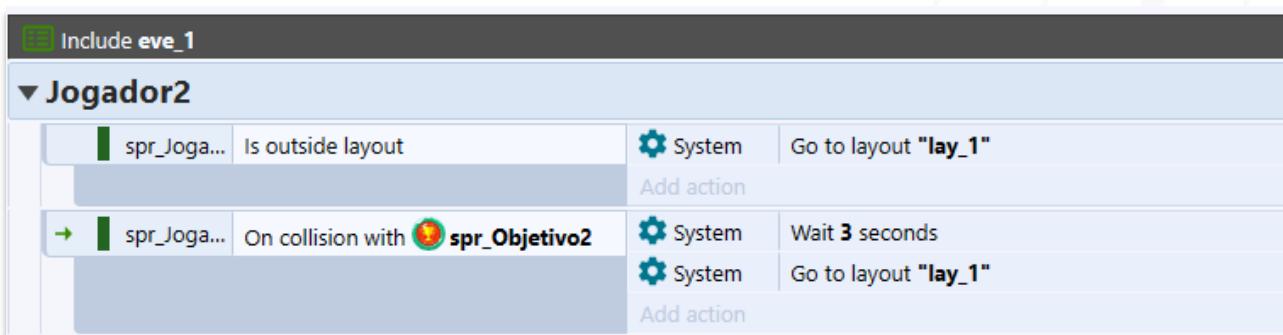
21) O Event Sheet eve_2 ficará conforme figura abaixo



22) Coloque no Event Sheet eve_2 uma instrução para quanto o spr_Jogador colidir com o spr_Ojetivo2, o sistema aguarde 3 segundos (ação de sistema) e reinicie o jogo.

- Inserir o evento: spr_Jogador => on collision with => spr_objetivo2
- Adicionar a ação => System => wait 3 seconds
- Adicionar a ação => System => Go to "lay_1"

23) O Event Sheet eve_2 ficará conforme figura abaixo



3.2. Exercício Aula 03

Abrir o Jogo “ExercicioJogoPlataforma02.c3p” que você criou na aula anterior, ou se desejar criar um novo jogo e alterar o layout, duplicando algumas plataformas de maneira que a obter mais dificuldade para a chegada do jogador ao objetivo. Estender os objetos para que a largura total fique maior que a largura da tela (1280 pixels).

Para dificultar podem ser colocadas algumas barreiras fixas e podem também ser experimentados outros movimentos das plataformas ou mesmo outras funções ao invés do senoidal, tal como o quadrado, que diminui o tamanho da plataforma podendo causar problemas para o jogador que está sobre a plataforma. Altere também o movimento de uma plataforma para Avançar/Retroceder ou mesmo Opacidade.



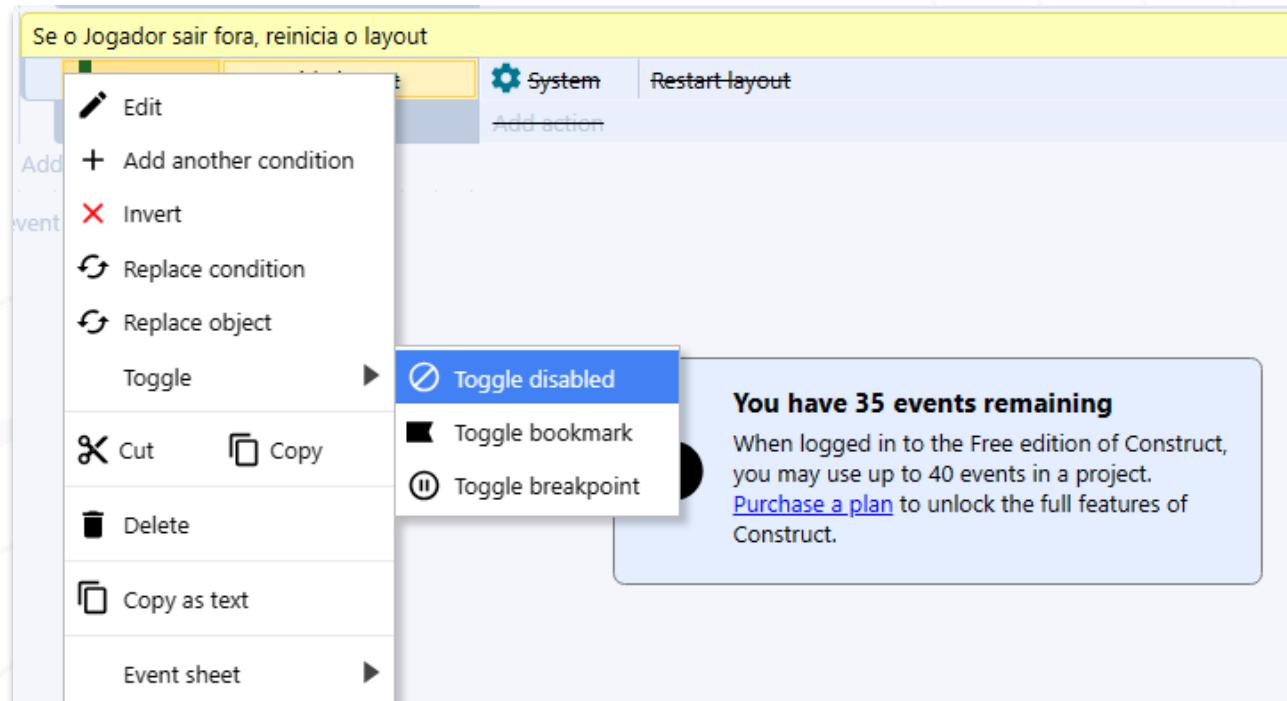
Importante

Para que seja possível ver o jogador mesmo quando ele ultrapassar a largura da tela, a ele (jogador), deve ser acrescentado o Comportamento “Scroll To”. Este comportamento faz com que a camera acompanhe o jogador.

Testar o jogo e verificar o funcionamento do jogador andando por todo o layout.

Com o jogador utilizando todo o layout, retirar a instrução “Se o jogador sair fora, reinicia o layout”, para apresentar mais dois comportamentos.

A eliminação de uma linha de “evento / ação”, pode ser feita apagando a linha (o que nem sempre é uma boa ideia) ou “desativando a linha”. Para desativar a linha, o que os programadores normalmente chamam de **“Comentar o comando”** no Construct é feito com a instrução “Toggle disabled”, que pode ser acionada clicando com o botão direito no início da instrução e escolhendo “Toggle disabled” em “Toggle”, conforme a seguir



Após desativar a instrução “Se o jogador sair fora, reinicia o layout” teste os dois comportamentos a seguir:

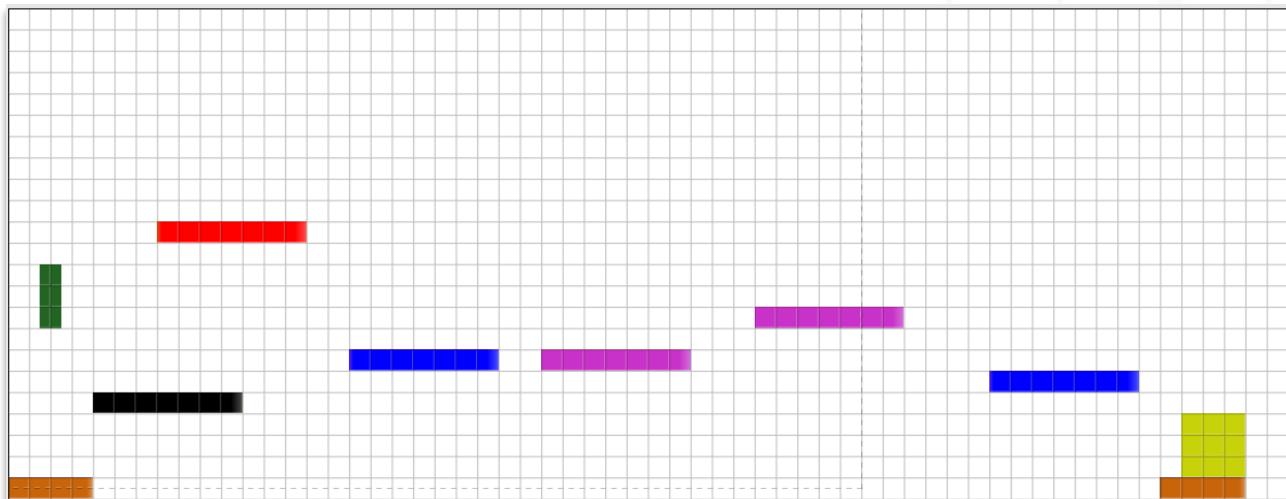
Para que o jogador ao chegar ao final do layout volte para o início do layout, pode-se adicionar o comportamento “Warp”

Para que o jogador não saia do layout pode-se adicionar o comportamento “BoundToLayout”. Observe que esta instrução não deixa o jogador sair do layout, porém se ele cair onde não existe o chão, embora o jogador fique na tela o jogo será encerrado.

Atenção que para utilizar estes dois comportamentos sugeridos anteriormente é preciso retirar a instrução “Se o jogador sair fora, reinicia o layout”

Nomear o jogo como **ExercicioJogoPlataforma03**

A figura a seguir é um exemplo de layout.



4. MOVIMENTAÇÃO COM TOQUES

Apresentação:

Até o momento, a movimentação do Jogador está sendo feita de duas maneiras através das teclas A, D, W, (instruções criadas em aula anterior no **eve_1**) e pelas setas do teclado (que foi configurado automaticamente quando o “comportamento de plataforma” foi adicionado ao jogador). Nesta aula será acrescentada movimentação através de Toques em elementos da tela. A movimentação pelas setas do teclado será desabilitada e programada em **eve_1**. Este procedimento será feito pois, no futuro, animações no personagem ou mesmo outras ações, serão executadas na medida que as setas forem pressionadas, então, toda programação adicional que for colocada nas teclas A, D, W deverá também acontecer para as teclas de setas do teclado. Do ponto de vista de programação é importante salientar que, para realizar estas funções será utilizada a cláusula “OU”, que executará ações caso um dos eventos sejam acionados.

4.1. Desenvolvimento

1. Baixar e descompactar a pasta “**Aula04_Elementos**” que está disponível.
2. Colocar esta pasta descompactada, dentro da pasta Construct que está na área de trabalho.
3. Abrir o Construct 3 através do endereço: <https://editor.construct.net/>
4. Se necessário passar o idioma do editor para inglês,
 - a. Menu
 - b. Configurações
 - c. English (US)
 - d. Fechar
5. Fazer login com seu usuário
 - a. Guest
 - b. Log in
 - c. Preencher Username e Password
 - d. Login
6. Abrir o projeto **JogoPlataforma03** (que deve estar na pasta Construct da área de trabalho).
7. Gravar o projeto utilizando: **Menu; Project; Save as; Save as a single file**, com o nome, **JogoPlataforma04**

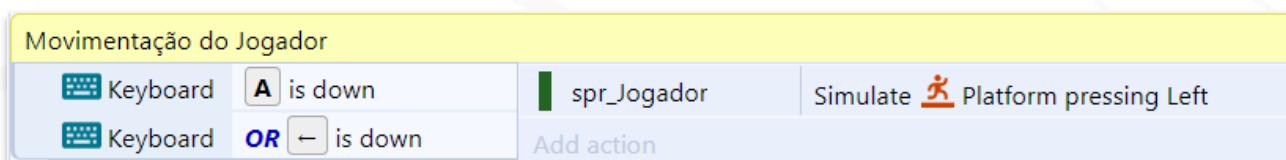
8. Passar para **lay_1**
9. No Painel Project, marcar o Jogador (spr_Jogador) e no Painel Properties escolher o Grupo “Behavior” e desmarcar “Default controls”.
10. Testar o jogo e verificar que o Jogador não se movimenta mais com as setas do teclado, somente com as teclas A, D, W.

Obs: Embora a movimentação com as setas tenha sido retirada do padrão, nos próximos itens, esta movimentação será recolocada via programação. Isto está sendo implementado, para que, no futuro, quando novas ações como as animações dos personagens forem adicionadas deseja-se que estas funcionem tanto com as teclas A, D, W como com as setas do teclado e também com a movimentação “Touch” que será inserida nesta aula.

11. Passar para **eve_1** e adicionar o evento de pressionar a tecla “seta esquerda”, para movimentar o jogador, da seguinte maneira:
 - a. Clicar com o botão direito do Mouse abaixo do evento “Keyboard A is down”.
 - b. No menu que surgir escolher “Add”.
 - c. No novo menu escolher “Add another condition”.
 - d. Inserir o evento: Keyboard => Key is down => Seta para esquerda.
 - e. Clicar no início das duas instruções do teclado, de maneira que elas fiquem marcadas.
 - f. Clicar com o botão direito e no Menu escolher “Make OR block”.

Obs: Notar que aparecerá a instrução “OR” o que indica que o jogador andará para esquerda caso seja tecida a letra “A” ou a tecla “Seta para esquerda”.

12. A Folha de eventos ficará conforme figura:



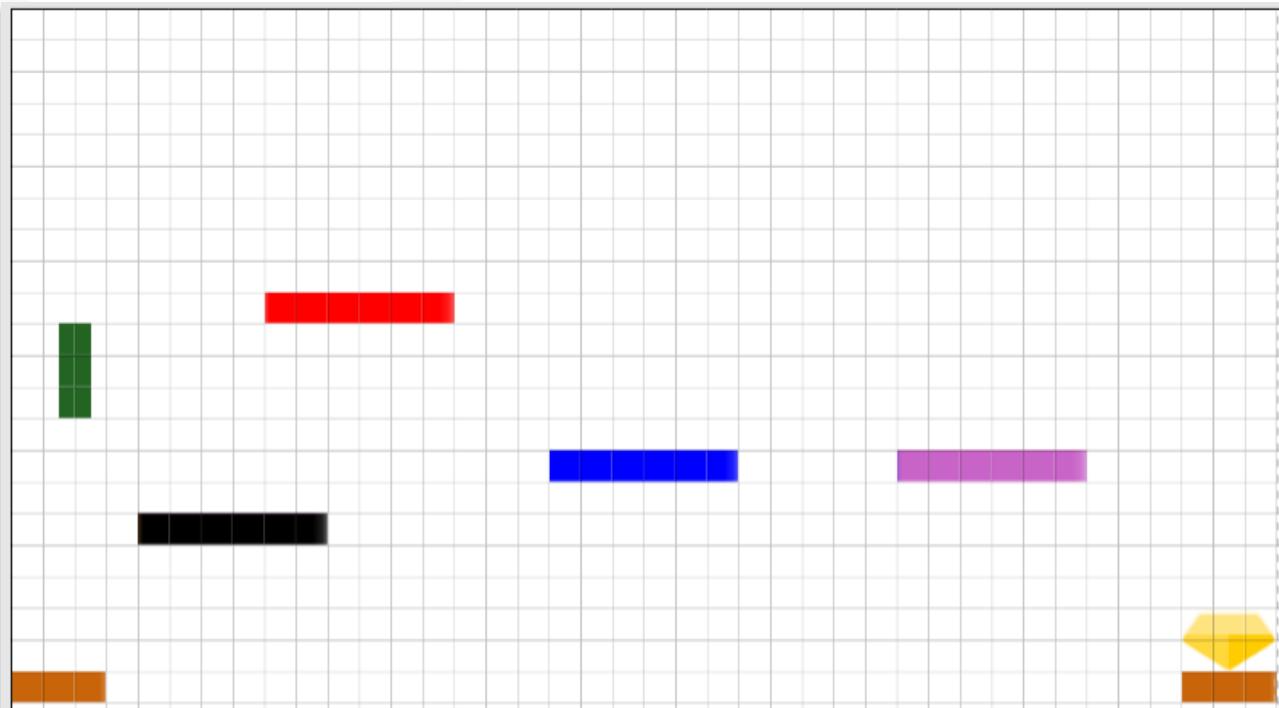
13. Testar o jogo e verificar que somente as letras A, D, W e a seta esquerda estão funcionando, as teclas seta para direta e seta para cima ainda não estão funcionando.
14. Adicionar a tecla “seta direita” para movimentar o jogador para a direita conforme a seguir:
 - a. Clicar com o botão direito do Mouse abaixo da instrução “Keyboard D is down”.

- b. No menu que surgir escolher “Add”.
 - c. No novo menu escolher “Add another condition”.
 - d. Inserir o evento: Keyboard => Key is down => Seta para direita
 - e. Clicar no início das duas instruções do teclado, de maneira que elas fiquem marcadas.
 - f. Clicar com o botão direto e no Menu escolher “Make OR block”.
15. Adicionar a tecla “seta para cima” para fazer o jogador saltar conforme a seguir:
- a. Clicar com o botão direito do Mouse abaixo da instrução “Keyboard W is down”.
 - b. No menu que surgir escolher “Add”.
 - c. No novo menu escolher “Add another condition”.
 - d. Inserir o evento: Keyboard => Key is down => Seta para cima
 - e. Clicar no início das duas instruções do teclado, de maneira que elas fiquem marcadas.
 - f. Clicar com o botão direto e no Menu escolher “Make OR block”.
16. Após encerrar estas configurações a Folha de eventos **eve_1** ficará conforme a seguir:

Movimentação do Jogador		
Keyboard A is down	spr_Jogador	Simulate Platform pressing Left
Keyboard OR ← is down	Add action	
Keyboard D is down	spr_Jogador	Simulate Platform pressing Right
Keyboard OR → is down	Add action	
Keyboard W is down	spr_Jogador	Simulate Platform pressing Jump
Keyboard OR ↑ is down	Add action	

17. Testar o jogo e verificar que agora o jogador poderá ser movimentado tanto pelas letras A, D, W com pelas teclas “para a esquerda”, “para a direita” e “para cima” do teclado.
18. Adicionar ao spr_Objetivo, a figura do diamante conforme a seguir:
- a. Dar um duplo clique em spr_Objetivo.
 - b. Carregar utilizando a opção “Load image from files” a imagem “Diamante.png”, que foi fornecida em “Elementos” no material desta aula.
 - c. Clicar em “Crop transparent edges” para que a figura fique delimitada retirando as partes brancas.

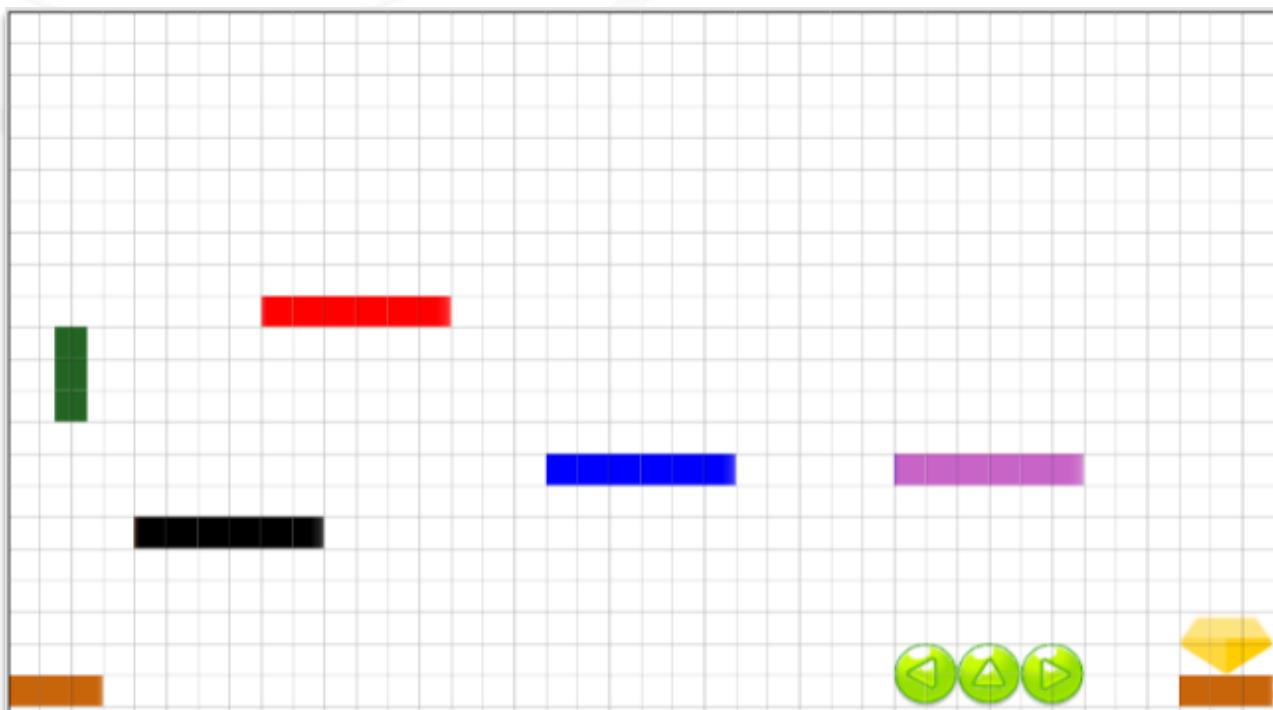
- d. Passar para “Edit the image ponits” e em “Origin” escolher a partir de “Quick Assign”: “Top left” (0 x 0 pixels).
 - e. Fechar o “Editor de Animações” e redimensionar o diamante na figura
19. O layout ficará conforme figura abaixo:



20. Passar para **lay_1**
21. Passar para a Layer GUI.
22. Inserir um Sprite para colocar o botão que irá mover o Jogador para a esquerda.
- a. Tipo: Sprite.
 - b. Nome: **btn_Esquerda**
 - c. Carregar utilizando a opção “Load image from files” a imagem “buttons.png”, que foi fornecida em “Elementos” no material desta aula.
 - d. Movimentar o conjunto de botões até encontrar o botão verde para a esquerda e recortar este botão conforme a seguir:
 - i. Escolher a opção “Rectangle select”.
 - ii. Selecionar a figura do botão desejada.
 - iii. Clicar na ferramenta “Crop”.
 - iv. Clicar novamente em “Drop” para que a figura fique bem delimitada.
 - v. Passar para “Edit the image ponits” e em “Origin” escolher a partir de “Quick Assign”: “Top left” (0 x 0 pixels).
 - vi. Redimensionar a figura utilizando o botão “Resize” para (64 x 64).
23. Inserir um Sprite para colocar o botão que irá fazer o Jogador saltar.

- a. Tipo: Sprite.
 - b. Nome: **btn_Saltar**
 - c. Carregar utilizando a opção “Load image from files” a imagem “buttons.png”, que foi fornecida em “Elementos” no material desta aula.
 - d. Movimentar o conjunto de botões até encontrar o botão verde para cima e recortar este botão conforme a seguir:
 - i. Escolher a opção “Rectangle select”.
 - ii. Selecionar a figura do botão desejada.
 - iii. Clicar na ferramenta “Crop”.
 - iv. Clicar novamente em “Drop” para que a figura fique bem delimitada.
 - v. Passar para “Edit the image ponits” e em “Origin” escolher a partir de “Quick Assign”: “Top left” (0 x 0 pixels).
 - vi. Redimensionar a figura utilizando o botão “Resize” para (64 x 64).
24. Inserir um Sprite para colocar o botão que irá mover o Jogador para a direita.
- a. Tipo: Sprite.
 - b. Nome: **btn_Direita**
 - c. Carregar utilizando a opção “Load image from files” a imagem “buttons.png”, que foi fornecida em “Elementos” no material desta aula.
 - d. Movimentar o conjunto de botões até encontrar o botão verde para a direita e recortar este botão conforme a seguir:
 - i. Escolher a opção “Rectangle select”.
 - ii. Selecionar a figura do botão desejada.
 - iii. Clicar na ferramenta “Crop”.
 - iv. Clicar novamente em “Drop” para que a figura fique bem delimitada.
 - v. Passar para “Edit the image ponits” e em “Origin” escolher a partir de “Quick Assign”: “Top left” (0 x 0 pixels).
 - vi. Redimensionar a figura utilizando o botão “Resize” para (64 x 64).

25. Após a colocação dos 3 elementos, a tela do jogo deverá ser arrumada conforme figura:



26. Inserir um novo objeto do tipo Input que será o **Touch**.

- Botão direito do Mouse => “Insert new object” => Toque

27. Passar para **eve_1** e adicionar o evento de “Is touching object” para o botão **btn_Esquerda** da seguinte maneira:

- Clicar com o botão direito do Mouse no início da instrução “Keyboard A is down”.
- No menu que surgir escolher “Add”.
- No novo menu escolher “Add another condiction”.
- Inserir o evento: Touch => “Is touching Object” => **btn_Esquerda**

28. A Folha de Evetos ficará conforme figura:



```

Movimentação do Jogador
Keyboard [A] is down
Keyboard [OR ←] is down
OR [Is touching btn_Esquerda v]
  spr_Jogador
    Simulate [Platform pressing Left v]

```

29. Adicionar o evento de “Is touching object” para o botão **btn_Direita** da seguinte maneira:

- Clicar com o botão direito do Mouse no início da instrução “Keyboard D is down”.

- b. No menu que surgir escolher “Add”.
 - c. No novo menu escolher “Add another condiction”.
 - d. Inserir o evento: Touch => “Is touching Object” => => btn_Direita.
30. Adicionar o evento de “Is touching object” para o botão btn_Saltar da seguinte maneira:
- a. Clicar com o botão direito do Mouse no início da instrução “Keyboard W is down”...
 - b. No menu que surgir escolher “Add”.
 - c. No novo menu escolher “Add another condiction”.
 - d. Inserir o evento: Touch => “Is touching Object” => => btn_Saltar.
31. Após encerrar as três opções do toque a Folha de Eventos ficará conforme figura.

Movimentação do Jogador			
Keyboard	A is down	spr_Jogador	Simulate Platform pressing Left
Keyboard	OR ← is down	Add action	
Touch	OR Is touching btn_Esquerda		
Keyboard	D is down	spr_Jogador	Simulate Platform pressing Right
Keyboard	OR → is down	Add action	
Touch	OR Is touching btn_Direita		
Keyboard	W is down	spr_Jogador	Simulate Platform pressing Jump
Keyboard	OR ↑ is down	Add action	
Touch	OR Is touching btn_Saltar		

- 32. Testar o jogo e verificar que os toques nos botões movimentam o jogador.
- 33. Salvar o projeto.
- 34. Fechar o Construct 3.

4.2. Exercício

Abrir o aplicativo **ExercicioJogoPlataforma03** e adicionar o movimento com o toque. Como sugestão você talvez possa mudar as cores dos botões ou mesmo escolher outros botões utilizando imagens da Internet. Uma sugestão de site com outros elementos gratuitos pode ser <https://www.kenney.nl/assets> Outra sugestão de site para obter imagens e sons é o <https://www.gameart2d.com/freebies.html>

Nomear o jogo como **ExercicioJogoPlataforma04**

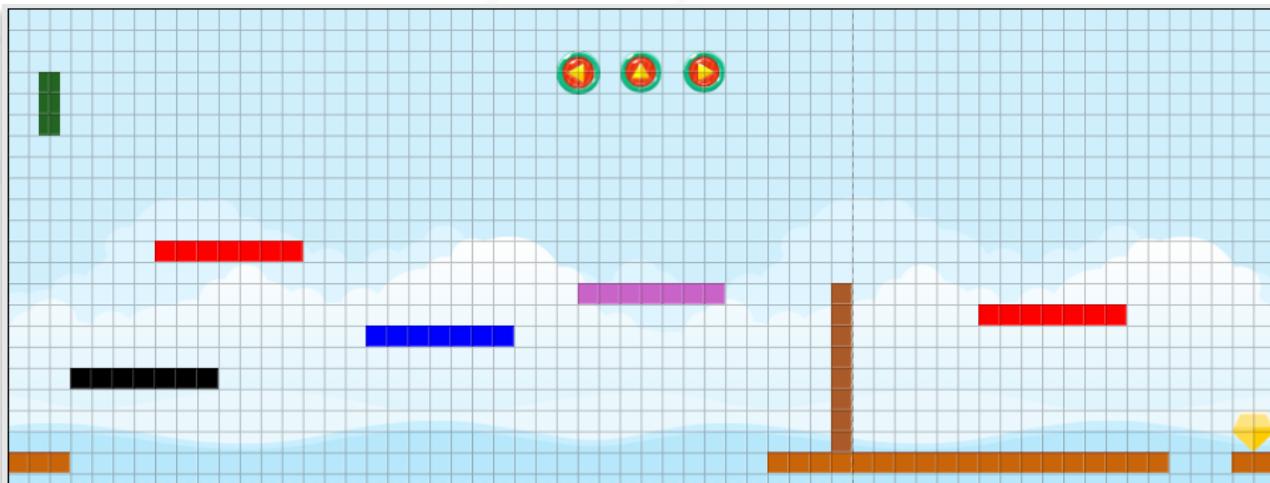
- 1) Criar um segundo layout

- a. Inserir com Add layouts.
 - b. Adicionar também uma “Add event sheet”
 - c. Renomear o Layout para **lay_2**
 - d. Renomear Folha de Eventos para **eve_2**
- 2) No lay_2 alterar o layout Tamanho (Size) para 1920 x 720
- 3) Criar as mesmas camadas do lay_1 (GUI, Elementos).
- 4) Dar um duplo clique na Folha de Eventos eve_2 para que ele também apareça na Barra de Layouts e Eventos.
- 5) A barra de Layouts e eventos ficará conforme figura abaixo:



- 6) Inserir, no layout lay_1 camada Elementos, utilizando Tiled Background a imagem bak_lay1 e redimensioná-la para que fique um pouco maior que os layouts. Em seguida colocar o background atrás das imagens.
- a. Passar para o layout lay_1
 - b. Passar para a camada Elementos
 - c. Botão direito do Mouse => “Insert new object” => Tiled Background
 - d. Load images from file
 - e. Escolher bak_lay1
 - f. Redirecionar (maior que o lay_1)
 - g. Clicar com o botão direito em bak_lay1
 - h. Escolher => Z order
 - i. Escolher => Send to bottom of layer
- 7) Inserir no lay_1 os Botões para utilização como Touch conforme na parte teórica:
- a. Colocar os 3 botõesm na layer GUI do Layout lay_1 nomeando-os de btn_Esquerda, btn_Direita e btn_Saltar
- 8) Desativar a opção “Defout controls” no jogador.
- 9) Inserir a partir do lay_1 o plugin Touch para que ele possa ser utilizado na programação dos Event Sheets.
- 10) Passar para o eve_1 e colocar a programação para que além do jogador ser movimentado com as teclas A, D e W ela também seja movimentada com as setas do teclado e com o Touch do Mouse nos botões.

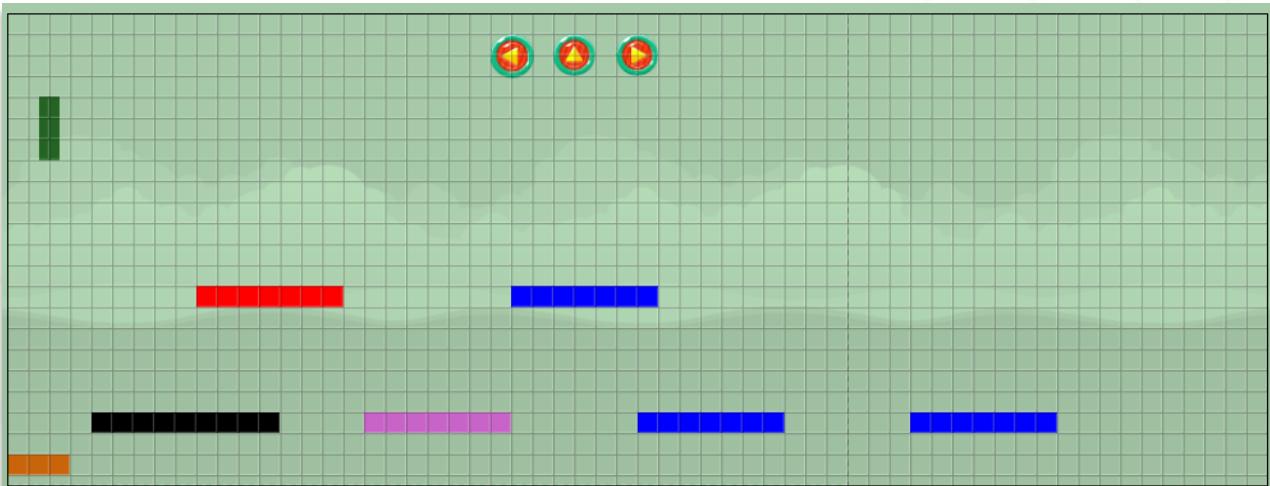
11) O layout lay_1 ficará semelhante à imagem seguir:



12) Adicionar no layout 2 (lay_2) camada “Elementos”, algumas das plataformas já existentes em “Object Types”, alterando dimensões e posicionamentos. (Observar que não são novas plataformas e estes elementos não devem ser duplicados nem clonados). Coloque também os botões de movimentação.

13) Coloque o background bak_lay2 no layout 2 da mesma maneira que foi colocado o bak_1 no lay_1

14) O layout lay_1 ficará semelhante à imagem seguir:



15) Passar para o eve_1 e inserir o evento que fará com que o jogador ao colidir com o spr_Ojetivo passe para a segunda fase (layout 2 => lay_2):

- Inserir no grupo jogador o comentário “Se o jogador atingir o objetivo, passa para a segunda fase”
- Inserir o evento: spr_Jogador => on collision with =>spr_Ojetivo
- Adicionar a ação => System => Go to layout (by name) => “lay_2”

16) Testar o jogo e verificar que, embora ao atingir o spr_Ojetivo o jogador vá para o Lautoy lay_2 ele não pode ser movimentado pelas teclas ou mesmo com Touch nos botões. Isso está acontecendo porque o Layout 2 que tem nome lay_2 está seguindo as instruções do eve_2 e as instruções de movimentação do jogador estavam no eve_1. Para fazer com que as instruções do eve_1 sejam utilizadas no eve_2 basta no eve_2 adicionar “Include event Sheet” e no quadro de mensagens que surgir escolher eve_1.

17) O Event Sheet eve_1 ficará conforme figura:

▼ Jogador

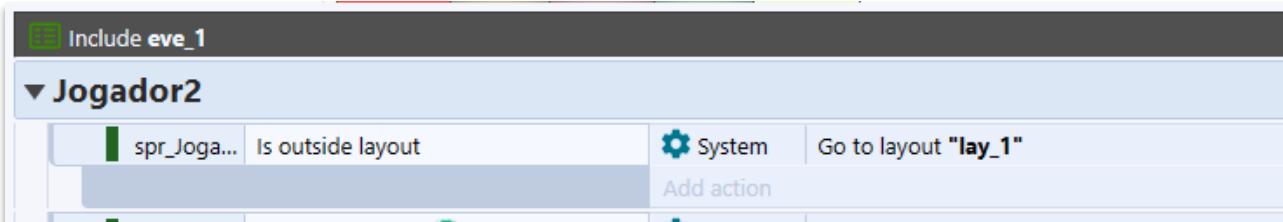
Movimentação do Jogador			
Keyboard	A is down	spr_Joga...	Simulate 🏃 Platform pressing Left
Touch	Is touching ⚡ btn_Esquerda	Add action	
Keyboard	D is down	spr_Joga...	Simulate 🏃 Platform pressing Right
Keyboard	W is down	spr_Joga...	Simulate 🏃 Platform pressing Jump
		Add action	
Se o Jogador sair fora, reinicia o layout			
spr_Joga...	Is outside layout	System	Restart layout
		Add action	
→ spr_Joga...	On collision with 🏆 spr_Ojetivo	System	Go to layout "lay_2"
		Add action	

18) Clone o sprite objetivo, de maneira que ele possa, no segundo layout receber uma nova imagem, que será a do botão com uma taça de vencedor. Mantenha o nome sugerido “spr_Ojetivo2”. Coloque este elemento (spr_Ojetivo2) no lay_2.

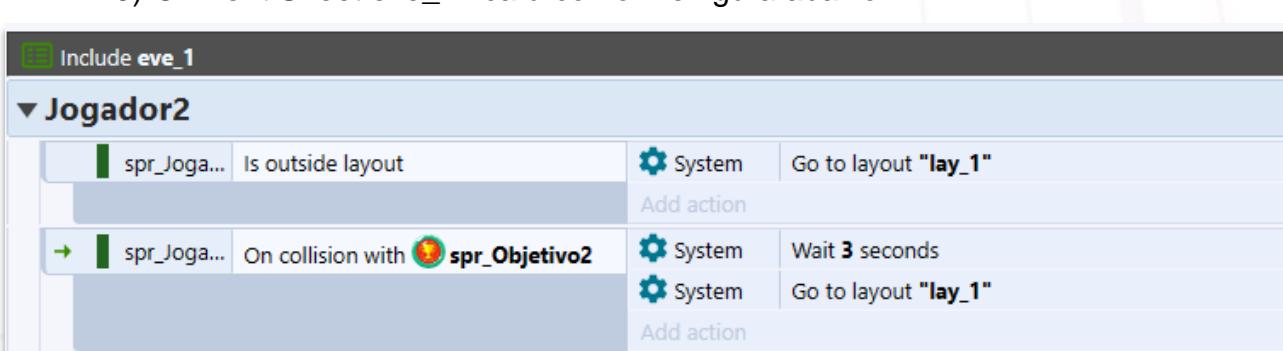
19) Após estas ações o layout 2 (lay_2) deverá ficar semelhante à figura abaixo.



- 20) Crie no Event Sheet 2 (eve_2) um grupo Jogador2 e insira o evento que fará o jogador voltar ao lay_1 se ele sair fora do layout no lay_2.
- Inserir o evento: spr_Jogador => is outside layout
 - Adicionar a ação => System => Go to layout (by name) => "lay_1"
- 21) O Event Sheet eve_2 ficará conforme figura abaixo



- 22) Coloque no Event Sheet eve_2 uma instrução para quanto o spr_Jogador colidir com o spr_Objetivo2, o sistema aguarde 3 segundos (ação de sistema) e reinicie o jogo.
- Inserir o evento: spr_Jogador => on collision with => spr_objetivo2
 - Adicionar a ação => System => wait 3 seconds
 - Adicionar a ação => System => Go to "lay_1"
- 23) O Event Sheet eve_2 ficará conforme figura abaixo



5. ADICIONANDO VIDAS – VARIÁVEIS LOCAIS

Apresentação:

Incrementando os conceitos de objetivo do jogo da aula anterior, é necessário que se estabeleçam limites para que o jogador consiga chegar com sucesso ao objetivo do jogo, visto que, se sempre que ele sair fora do Layout ele obtiver nova chance, sem dúvida ele acabará conseguindo vencer o jogo. Uma técnica comum para exercer estas limitações é determinar o número de vidas que o jogador pode possuir e cada vez que ele errar, uma vida será perdida.

O número de vidas será determinado por uma variável adicionada ao jogador. Este procedimento, de colocar variáveis em **elementos do jogo** recebe a denominação de Variável Local, (chamadas de Variáveis de Instância – Instance Variables) pois a variável está atrelada ao jogador. Existem também Variáveis Globais, que são atribuídas às Folhas de Evento, mas estas serão discutidas em próximas aulas

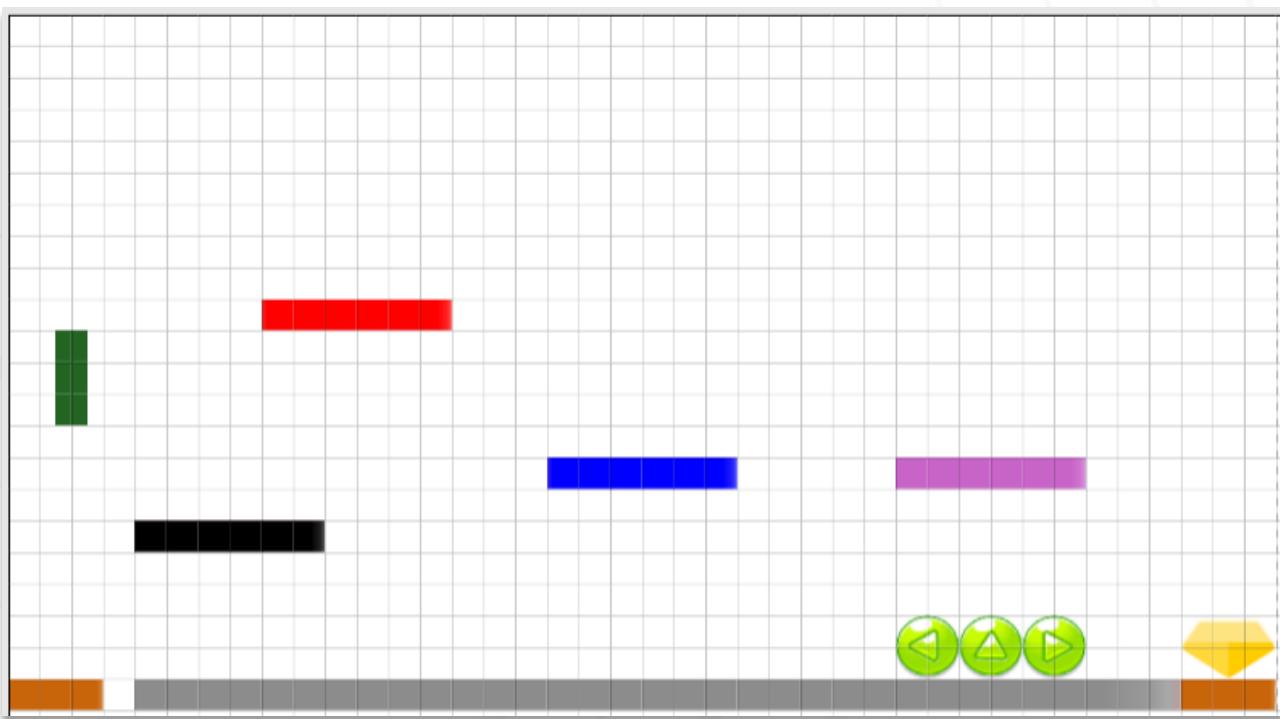
Nesta aula será também apresentada a criação e utilização de funções, ferramentas muito úteis para reduzir o número de instruções em programação. O conceito de função, consiste em: caso existam ações que se repitam em diversos eventos; escrever estas ações uma só vez e agrupá-las em uma “Função”, e toda a vez que estas ações precisarem ser realizadas, ao invés de as escrever em cada um dos eventos, simplesmente chamar a “Função”.

Obs: outras limitações, assim como brindes serão acrescentados nas próximas aulas.

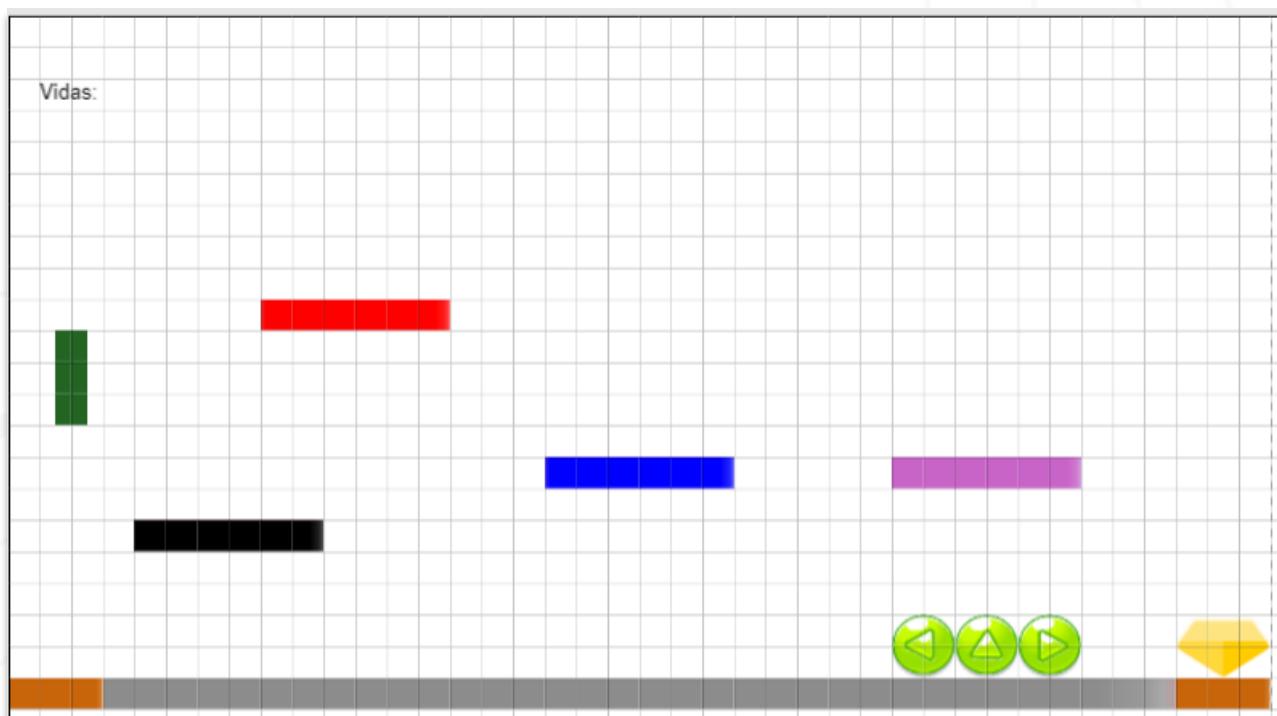
5.1. Desenvolvimento

1. Abrir o Construct 3 através do endereço: <https://editor.construct.net/>
2. Se necessário passar o idioma do editor para inglês,
 - a. Menu
 - b. Configurações
 - c. English (US)
 - d. Fechar
3. Fazer login com seu usuário

- a. Guest
 - b. Log in
 - c. Preencher Username e Password
 - d. Login
4. Abrir o projeto **JogoPlataforma04** (que deve estar na pasta Construct da área de trabalho).
 5. Gravar o projeto utilizando: **Menu; Project; Save as; Save as a single file**, com o nome, **JogoPlataforma05**
 6. Passar para **lay_1**
 7. Passar para a **Camada Elementos** e colocar um objeto que indicará que o jogador caiu a atingiu a parte inferior do layout.
 - a. Tipo: Sprite
 - b. Nome: **spr_Morte**
 - c. Formato: Retângulo (ajustar na figura).
 - d. Cor: Cinza. (Vermelho = 140, Verde = 140, Azul = 140); (Alfa 255).
 - e. Ajustar o image point utilizando a ferramenta “Edit the image point” e em Origin escolher a partir de “Quick Assing”: “Top Left” (0 x 0 pixels).
 - f. Comportamento: Solid.
 8. Reposicionar as plataformas e o Jogador para que fiquem conforme figura.



9. Com este posicionamento, pode acontecer do jogador não conseguir alcançar a plataforma vasada, se isto acontecer, basta acessar o jogador e marcar (caso ainda não esteja marcada) a opção “Double-jump”:
10. Testar o jogo e observar que caso o jogador erre as plataformas ele tocará o objeto spr_Morte.
11. Criar uma variável local no jogador utilizando o item “Instance Variables” do Painel Properties.
 - a. Nome: **var_Vidas**
 - b. Tipo: Número
 - c. Valor inicial: 3
 - d. Descrição: Número de vidas do jogador.
12. Criar um objeto do tipo Texto, na **Camada GUI**, para manter atualizado na tela o número de vidas do jogador
 - a. Tipo: Text
 - b. Nome: **txt_Vidas**
 - c. Texto: Vidas:
 - d. Fonte: Arial (16)
 - e. Posição: conforme Layout



13. Passar para **eve_1**

14. Criar um comentário “Vidas do Jogador” e colocar as instruções a seguir abaixo deste comentário.

15. Criar uma instrução para manter o sistema atualizando o texto do elemento txt_Texto constantemente:

Obs: Os eventos e ações inseridos em uma Folha de Eventos são executadas sequencialmente desde o primeiro até o último continuamente. Ao chegar no final ele reinicia no primeiro e fica repetindo continuamente essas execuções. Este procedimento, no ambiente computacional, é normalmente chamado de “loop”. Cada uma destas voltas (loops), é **referenciada pelo Construct por “tick”**. Então, iremos utilizar a instrução “**Every tick**”, para atualizar o número de vidas no texto que está na tela do jogo da seguinte maneira.

a. Evento: System => Every tick

b. Ação: Texto => txt_Vidas => Set text: “Vidas: “ & spr_Jogador.var_vidas

16. Criar uma instrução para cada vez que o Jogador encostar no objeto Morte o sistema subtraia 1 ponto da variável var_vidas.

a. Evento: spr_Jogador => On collision with another object => Objeto spr_Morte

b. Ação: spr_Jogador => Subtract from => Instance variable: var_Vidas; Value 1

17. Criar uma instrução para caso a variável vidas chegue a zero o sistema seja Reiniciado:

a. Evento: spr_Jogador => => Compare instance variable => Instance variable: var_vidas; Comparison: = Equal to: Value 0.

b. Ação: System=> Restart layout

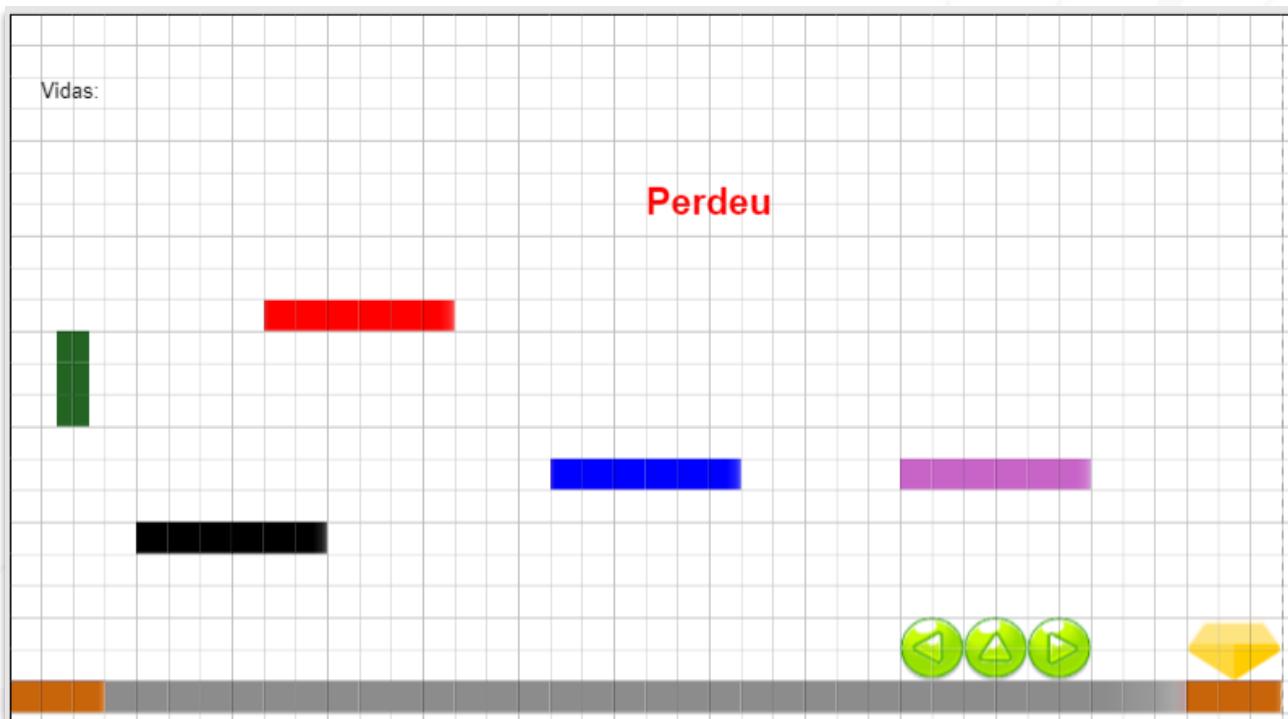
18. Os eventos ficarão conforme figura:

Vidas do Jogador		
System	Every tick	txt_Vidas Set text to "Vidas: " & spr_Jogador.var_Vidas Add action
→ spr_Joga...	On collision with spr_Morte	spr_Jogador Subtract 1 from var_Vidas Add action
spr_Joga...	var_Vidas = 0	System Restart layout Add action

19. Testar o jogo e verificar que cada vez que o Jogador cair no objeto spr_Morte, o número de vidas será diminuído de 1 e quando a variável var_vidas chegar a zero o jogo será automaticamente recomeçado.

20. Passar para **lay_1**

21. Criar na **Layer GUI**, uma mensagem com o texto “Perdeu”, para ser colocado na tela sempre que o número de vidas chegar a zero. Para mostrar esta mensagem, primeiramente criar o objeto Text conforme a seguir:
- Tipo: Texto
 - Nome: **txt_Perdeu**
 - Texto: Perdeu
 - Fonte: Arial (28)
 - Bold: Marcado
 - Color: Vermelha (255,0,0)
 - Horizontal Alignment: Center
 - Vertical Alignment: Center
 - Initially visible: desmarcar
 - Posição: Conforme Layout



22. Passar para **eve_1**

23. Criar dentro do evento spr_jogador => var_vidas = 0, as seguintes ações adicionais, **antes do System => Reiniciar layout**:

- Uma ação para colocar na tela a mensagem “Perdeu”:
 - Ação: Txt_Perdeu => Set visible => Visible
- Uma ação para solicitar que o sistema aguarde um tempo para que o jogador possa ver a mensagem.

i. Ação: System => Wait => Seconds 1,0.

24. O evento ficará conforme figura:

spr_Jogador	var_Vidas = 0	txt_Perdeu	Set visibility Visible
		System	Wait 1.0 seconds
		System	Restart layout
Add action			

25. Como o jogador pode perder o jogo também saindo fora dos limites da tela pelas laterais, a sequência de comandos do item anterior deverá ser colocada no evento “Está fora do layout”, ficando a Folha de eventos conforme a seguir:

Se o jogador sair fora, reinicia o layout			
spr_Jogador	Is outside layout	txt_Perdeu	Set visibility Visible
		System	Wait 1.0 seconds
		System	Restart layout
Add action			
Vidas do Jogador			
System	Every tick	txt_Vidas	Set text to "Vidas: " & spr_Jogador.var_Vidas
			Add action
→ spr_Jogador	On collision with spr_Morte	spr_Jogador	Subtract 1 from var_Vidas
			Add action
spr_Jogador	var_Vidas = 0	txt_Perdeu	Set visibility Visible
		System	Wait 1.0 seconds
		System	Restart layout
			Add action

Obs: Isto pode ser executado com um Ctrl C + Ctrl V porém, após copiar as instruções, para executar o Ctrl V, o mouse deve apontar o fim do evento

26. Testar o jogo.

Como pode ser observado, as instruções para mostrar o texto txt_Perdeu, aguardar 1 segundo e reiniciar o sistema estão repetidas em dois eventos. Este é um caso típico para a utilização de **funções** (Functions), que permitem criar grupos de instruções que podem ser chamadas através de comandos em diversos locais do programa. A utilização de funções é muito utilizada em linguagens de programação. Será criada uma função com o nome de “**Jogador_Perdeu**” e nesta serão colocadas as ações que devem ser executadas caso o jogador perca por sair fora da tela (spr_Jogador => is outside

layout) ou perca por colidir 3 vezes com spr_Morte (spr_Jogador => var_Vidas = 0).

27. Passar para eve_1

28. Criar um comentário “Jogador Perdeu”

29. Criar uma função com o nome “Jogador_Perdeu” obedecendo o seguinte procedimento:

a. Clicar com o botão direito em “Add event to Jogador” e o menu suspenso que surgir escolher Add function.

b. No quadro de diálogos Add function preencher:

i. Name: => “Jogador_Perdeu”

ii. Description: Coloca mensagem e reinicia o layout

iii. Category: (não necessária)

iv. Return type: None

c. Ações: Copiar as ações do evento spr_Jogador = Is outside layout

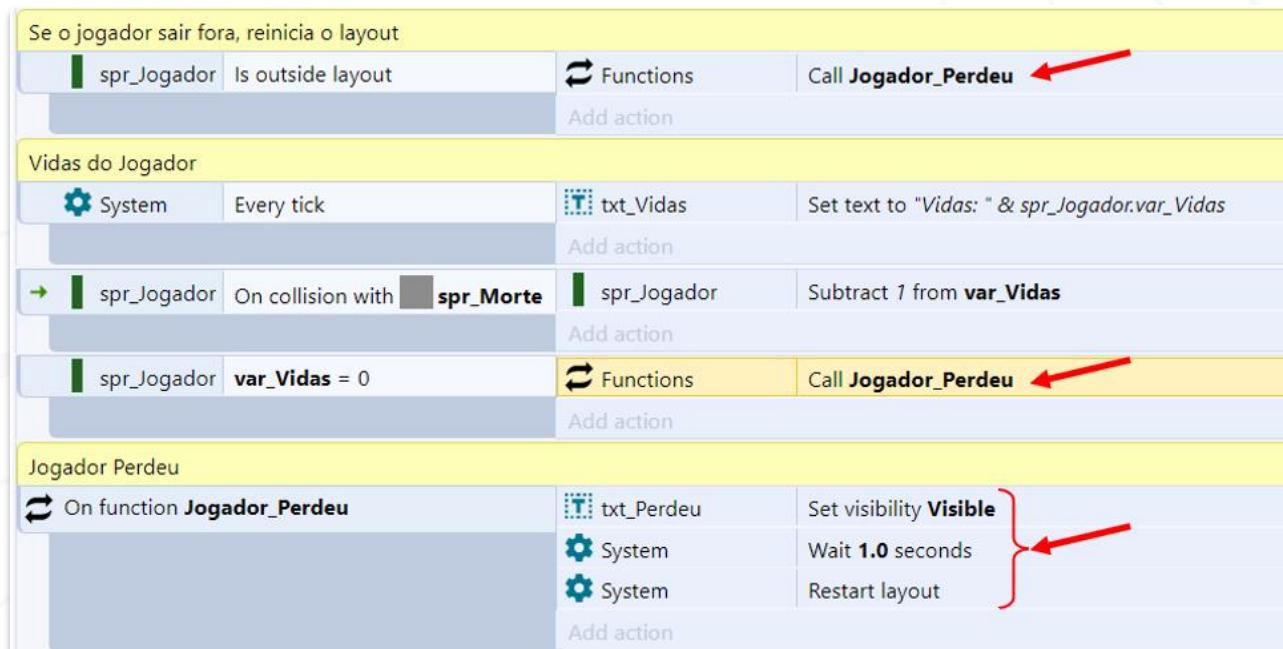
30. No evento: Spr_Jogador => Is outside layout

a. Ação: Functions => Jogador_Perdeu

31. No Spr_Jogador => var_Vidas = 0

a. Ação: Function => Jogador_Perdeu

32. A Folha de Eventos ficará conforme figura:



33. Testar o jogo e observar que continua funcionando como estava, porém, com a utilização da função Jogador_Perdeu, a Folha de Eventos ficou mais enxuta, o que é sempre aconselhável em técnicas de programação.

34. Salvar o projeto.
35. Fechar o Construct.

5.2. Exercício

Acrescentar ao exercício da aula anterior, **ExercicioJogoPlataforma04**, a contagem de vidas e uma Barra de Progresso (Progress bar) para mostrar o número de vidas.

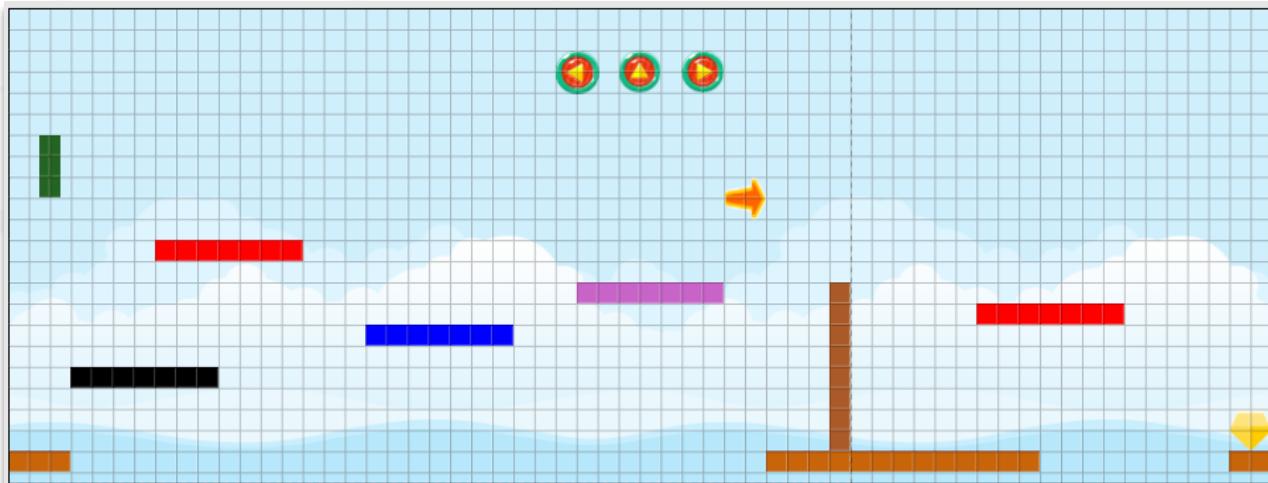
Manter as duas fases sugeridas na aula anterior. Colocar o número de vidas inicial no jogador em 3.

Colocar também um elemento para ativar o Behavior “Scroll To” no jogador de maneira a que seja possível ver os objetivos.

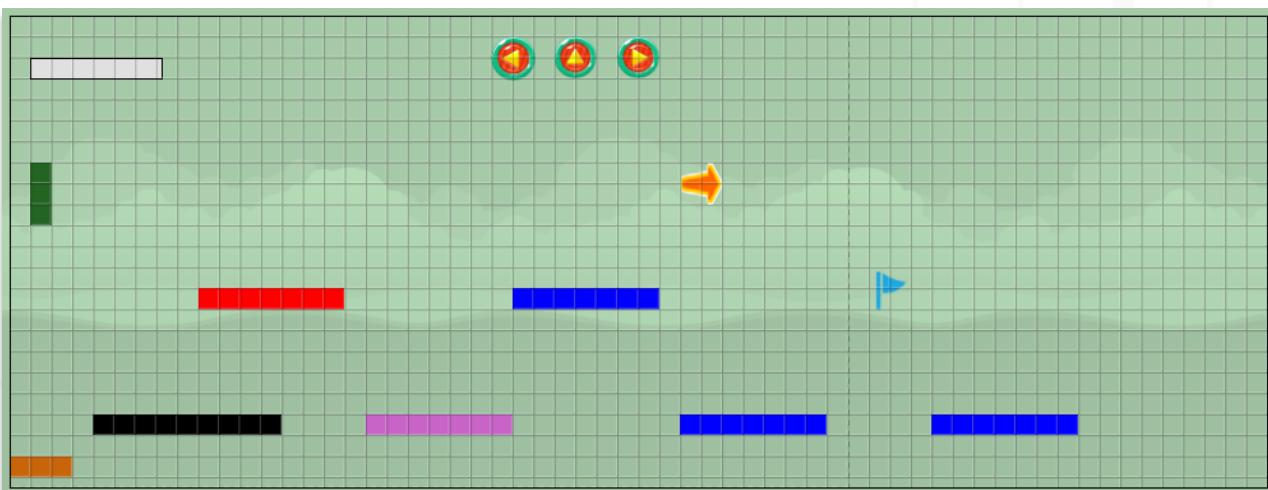
Colocar uma animação no spr_Objetivo2 para mostrar que ele foi atingido.

Nomear o jogo como **ExercicioJogoPlataforma05**

- 1) Passar para o Layout lay_1, selecionar o Jogador e desmarcar a opção “Enabled” no Behavior “Scroll To”, de maneira que o jogador seja obrigado a atingir um elemento, (que será uma seta) que será adicionada ao Layout para poder chegar ao spr_Objetivo.
- 2) Passar para o Layout lay_2, selecionar o Jogador e desmarcar a opção “Enabled” no Behavior “Scroll To”, de maneira que o jogador seja obrigado a atingir um elemento, (que será uma seta) que será adicionada ao Layout para poder chegar ao spr_Objetivo2.
- 3) Passar para Elementos em lay_1 e adicionar um sprite spr_Setas (com a figura de Setas que está na pasta “Aula05 Elementos) para ativar o Behavior “ScrollTo”, do spr_Jogador, de maneira que o jogador possa chegar ao Objetivo para mudar de Fase (Layout). (Colocar o spr_Setas, próximo ao final da área do Projeto)



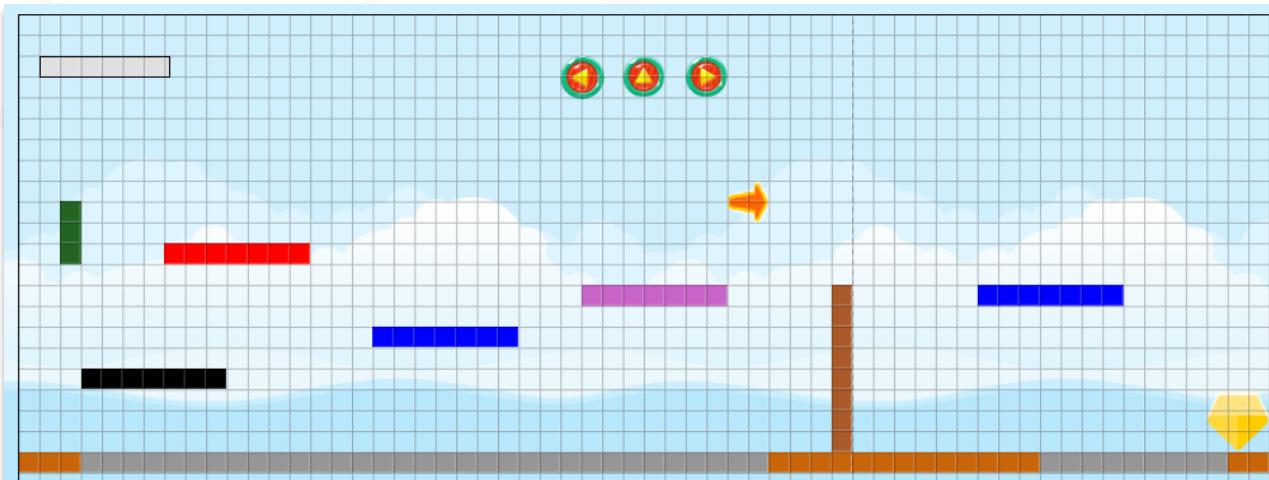
- 4) Passar para Elementos em lay_2 e adicionar um sprite spr_Setas para ativar o Behavior “ScrollTo”, do spr_Jogador, de maneira que o jogador possa chegar ao Objetivo2 para encerrar o jogo. (Colocar o spr_Setas, próximo ao final da área do Projeto)
- 5) Ainda no Layout lay_2, substitua a imagem do spr_Objetivo 2 pela figura da bandeira que pode ser encontrado na pasta Aula05_Elementos



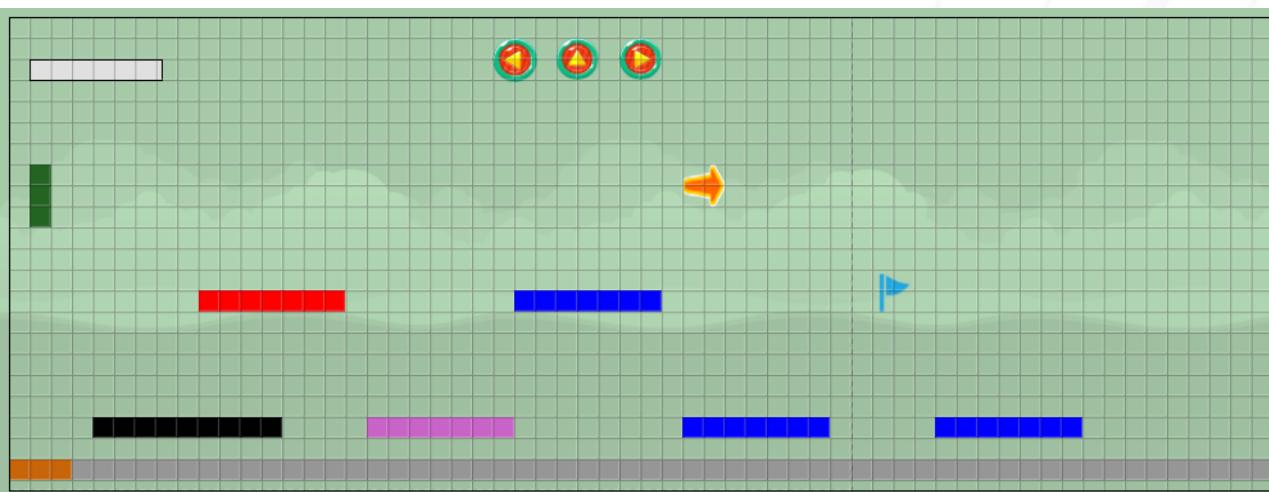
- 6) Passar para o eve_1 e escrever a programação para quando o jogador colidir com a seta, esta desaparecer e o Behavior “ScrollTo” no Jogador seja habilitado

Se o Jogador sair fora, reinicia o layout			
spr_Jogador	Is outside layout	System	Restart layout
Add action			
Se o jogador colidir Habilitar ScrollTo e esconder a seta			
→ spr_Jogador	On collision with spr_Setas	spr_Jogador	Set ScrollTo enabled
		spr_Setas	Set visibility Invisible

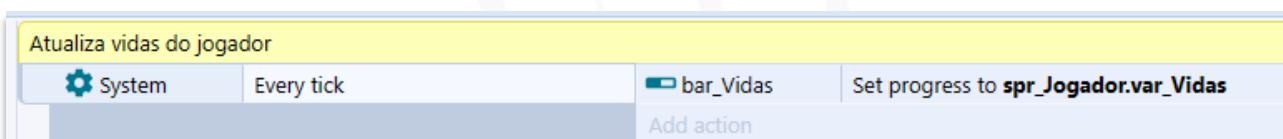
- 7) Passar para a Layer “Elementos” em lay_1 e adicionar um sprite spr_Morte e colocá-lo na parte inferior com o Behavior “Solid”.



- 8) Adicionar o mesmo spr_Morte na Layer “Elementos” do Layout lay_2



- 9) Adicionar ao jogador uma variável de instância com o nome var_Vidas e atribuir um valor inicial 3.
- 10) Adicionar um objeto do tipo “Progress bar” com o nome bar_Vidas na layer GUI dos dois Layouts. Este objeto irá indicar dinamicamente o número de vidas do jogador. Colocar (em Maximum) o número máximo de vidas = 3 (nos jogadores dos dois layouts)
- 11) Adicionar ao Event Sheet eve_1 uma instrução para a cada ciclo ele atualizar a bar_Vidas, com o número de vidas do jogador.



```

[{"id": "e1", "label": "Atualiza vidas do jogador", "type": "event_sheet", "x": 100, "y": 100, "w": 800, "h": 150}, {"id": "t1", "label": "System", "type": "trigger", "x": 100, "y": 100, "w": 150, "h": 50}, {"id": "a1", "label": "Set progress to spr_Jogador.var_Vidas", "type": "action", "x": 300, "y": 100, "w": 400, "h": 50}, {"id": "a2", "label": "Add action", "type": "button", "x": 500, "y": 150, "w": 200, "h": 50}]

```

- 12) Adicionar ao Event Sheet eve_1 uma instrução para retirar uma vida do jogador, cada vez que ele colidir com o spr_Morte

Se o jogador atingir o spr_Morte retirar uma vida do jogador

→ [spr_Jogador]	On collision with [spr_Morte]	[spr_Jogador]	Subtract 1 from var_Vidas
Add action			

- 13) Adicionar no início do Event Sheet eve_1 uma instrução para manter atualizada a barra de progresso

▼ Jogador

Atualiza vidas do jogador

System	Every tick	[bar_Vidas]	Set progress to spr_Jogador.var_Vidas
Add action			

- 14) Adicionar ao Event Sheet eve_1 uma instrução para reiniciar o jogo caso o jogador perca as 3 vidas disponíveis.

Caso o jogador perca todas as vidas var_Vidas = 0, reinicia o layout

[spr_Jogador]	var_Vidas ≤ 0	System	Restart layout
Add action			

- 15) Passar para o Lyout lay_1 e adicionar um objeto do tipo “Partículas” de nome par_Particulas e alteara as configurações deste objeto conforme a seguir
- Imagen Explosão que está na pasta Aula05Elementos.
 - Spray cone 360
 - Deixar o objeto fora da tela pois ele será criado quando for necessário
- 16) Passar o ponto de referência do spr_Ojetivo para o centro, de maneira que a explosão do par_Particulas se inicie no centro.
- 17) Alterar no Event Sheet eve_1 as instruções para quando o jogador atingir o spr_Ojetivo, para criar um par_Partículas e aguardar 3 segundos para passar para o layout lay_2

Se o Jogador atingir Objetivo, vá para o Layout lay_2

→ [spr_Jogador]	On collision with [spr_Ojetivo]	System	Create object [par_Particles] on layer 0 at (spr_Ojetivo.X, spr_Ojetivo.Y), create hierarchy: False, template: ""
Add action			
System	Wait 3 seconds	🕒	
System	Go to layout "lay_2"	Add...	

- 18) Adicionar no início do Event Sheet eve_2 uma instrução caso o jogador saia fora do layout volte para o Layout lay_1

▼ Jogador2

Caso o jogador saia do Layout, volte para o layout 1

spr_Joga... Is outside layout

System

Go to layout "lay_1"

Add action

- 19) Adicionar ao spr_Objetivo2 o Behavior “Pin”

- 20) Adiciona ao Event Sheet eve_2 a instrução para caso o jogador atinja o spr_Objetivo2, este objetivo fique grudado ao Jogador.

Caso o jogador atinja o objetivo ele captura o objetivo e o jogo volta para o layout lay_1

→ spr_Joga... On collision with spr_Objetivo2

System Pin Pin to spr_Jogador (X: True, Y: True, angle: True, width: Absolute, height: Absolute, Z: True)

System Wait 3 seconds

System Go to layout "lay_1"

Add action Add...

6. ADICIONANDO PREMIAÇÕES – VARIÁVEIS GLOBAIS

Apresentação:

Conforme citado na aula anterior, além do objetivo, que é conseguir chegar ao objeto spr_Objetivo, um jogo deve ter premiações e inimigos. Nesta aula serão adicionadas premiações (brindes), ou seja, durante o percurso existirão moedas, que caso o jogador as capture, irão adicionar vidas ao Jogador. Cada moeda terá um valor de 100 pontos e a cada 300 pontos o jogador deverá receber mais uma vida.

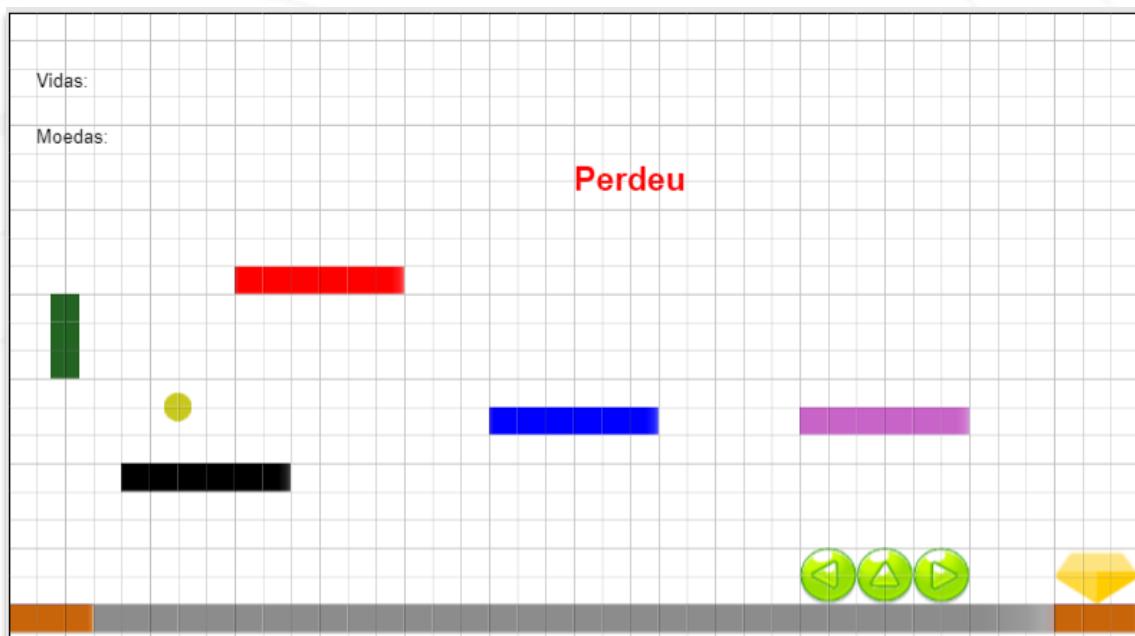
O número de vidas da aula anterior foi declarado como uma Variável do jogador. Este tipo de variável é considerado como uma Variável Local (instance variable), ou seja, está atrelada ao jogador. Ela tem algumas características como ser reiniciada cada vez que o jogador inicia o jogo.

O valor das moedas, que será utilizado nesta aula, será declarado como uma Variável Global (também chamada de Variável de Sistema – Global Variable) que tem outras propriedades, dentre elas a de não ser reiniciada automaticamente a cada mudança ou reinício de uma fase.

6.1. Desenvolvimento

1. Abrir o Construct 3 através do endereço: <https://editor.construct.net/>
2. Se necessário passar o idioma do editor para inglês,
 - a. Menu
 - b. Configurações
 - c. English (US)
 - d. Fechar
3. Fazer login com seu usuário
 - a. Guest
 - b. Log in
 - c. Preencher Username e Password
 - d. Login
4. Abrir o projeto **JogoPlataforma05** (que deve estar na pasta Construct da área de trabalho).

5. Gravar o projeto utilizando: **Menu; Project; Save as; Save as a single file**, com o nome **JogoPlataforma06**
6. Passar para **lay_1**
7. Passar para a **layer Elementos** e colocar um objeto que será base para criação de moedas.
 - a. Tipo: Sprite
 - b. Nome: **spr_Moeda**
 - c. Formato: Círculo (obtido pela ferramenta Brush com Size = 200 e Hardness 1).
 - d. Cor: Amarela. (Red: 200, Green: 200, Blue: 30); (Alfa 200).
 - e. Utilizar a ferramenta Crop para ajustar os limites da figura.
 - f. Image point origin: Middle
 - g. Resize 32 x 32 pixels
 - h. Posição, conforme figura
8. Passar para a layer GUI e colocar um texto para mostrar a quantidade de moedas que o jogador possui.
 - a. Tipo: Text
 - b. Nome: **txt_Moedas**
 - c. Texto: Moedas:
 - d. Fonte: Arial (16)
 - e. Color: Preta (0, 0, 0)
 - f. Posição, conforme figura
9. Posicionar a moeda e o texto conforme figura abaixo.



10. Passar para eve_1

11. Criar uma Variável Global, ou seja, uma Variável de Sistema, para acumular a quantidade de moedas capturadas pelo jogador:

- Botão direito do Mouse => Add Global Variable
- Name: var_moedas
- Type: Number
- Initial Value: 0
- Description: Pontos referentes às moedas capturadas.

Obs: Notar que a variável e seu valor inicial surge no Event Sheet no início da programação. Isto é feito automaticamente, até porque, declarar variáveis antes de iniciar a codificação é uma boa prática em programação.

12. Colocar um comentário “Gerenciamento de Moedas”.

13. Criar uma instrução para cada vez que o jogador capturar uma moeda sejam acrescentadas duas ações: À Variável var_moedas sejam acrescentados: 100 pontos e o Sprite spr_Moeda será destruído:

- Evento: spr_Jogador => On collision with another object => spr_Moeda
- Ação: System => Add to => Variable: var_moedas; Value: 100
- Ação: spr_Moeda => Destroy

14. Acrescentar no evento Every Tick a ação para mostrar a quantidade de pontos referentes às moedas capturadas.

- Ação: Texto => txt_Moedas => Set Text to: “Moedas: “ & var_moedas

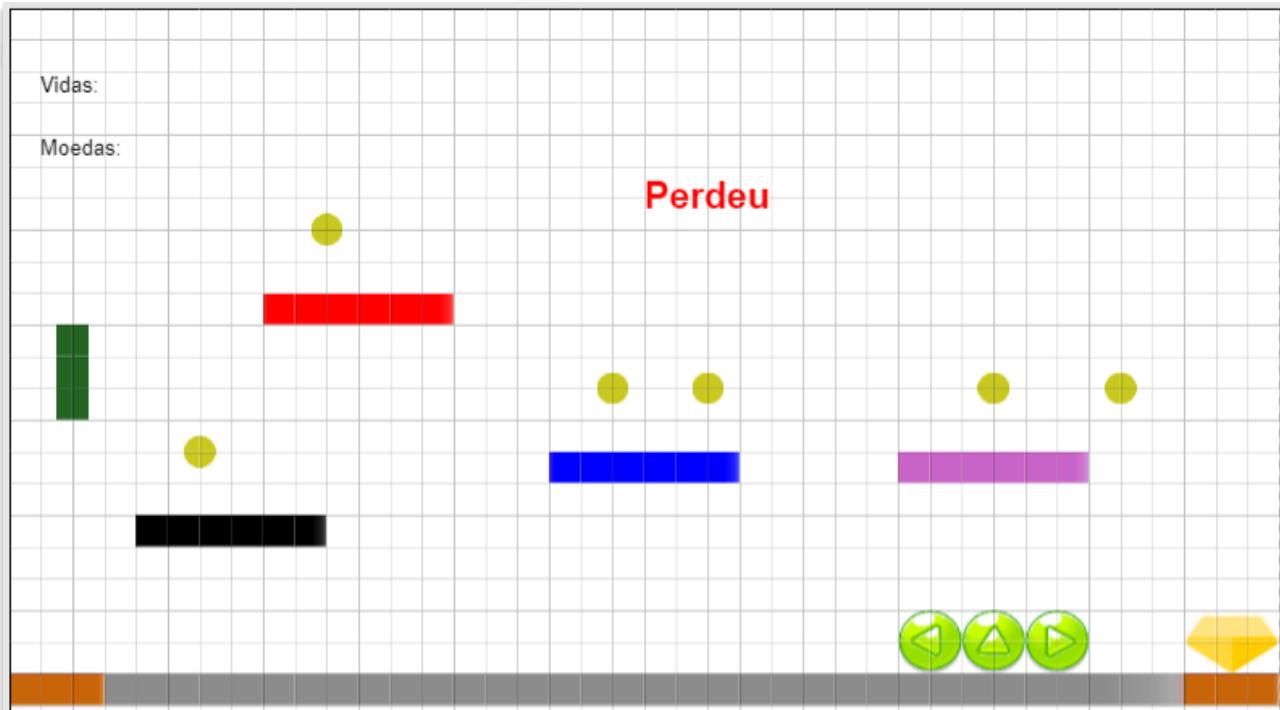
15. Os eventos ficarão conforme figura:



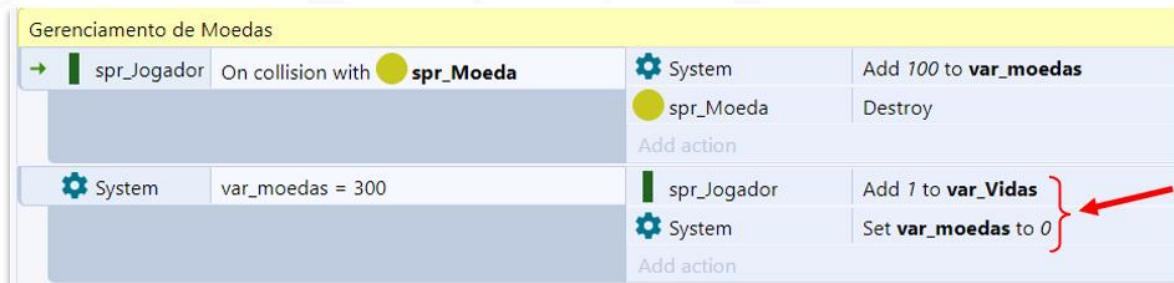
The image shows the Scratch Event Sheet with three scripts:

- Vidas do Jogador** (Player Lives):
 - System: Every tick
 - txt_Vidas: Set text to "Vidas: " & spr_Jogador.var_Vidas
 - txt_Noedas: Set text to "Moedas: " & var_moedas
 - spr_Jogador: On collision with spr_Morte
 - spr_Jogador: Subtract 1 from var_Vidas
 - spr_Jogador: var_Vidas = 0
 - Functions: Call Jogador_Perdeu
- Jogador Perdeu** (Player Lost):
 - On function Jogador_Perdeu
 - txt_Perdeu: Set visibility Visible
 - System: Wait 1.0 seconds
 - System: Restart layout
- Gerenciamento de Moedas** (Coin Management):
 - spr_Jogador: On collision with spr_Moeda
 - System: Add 100 to var_moedas
 - spr_Moeda: Destroy

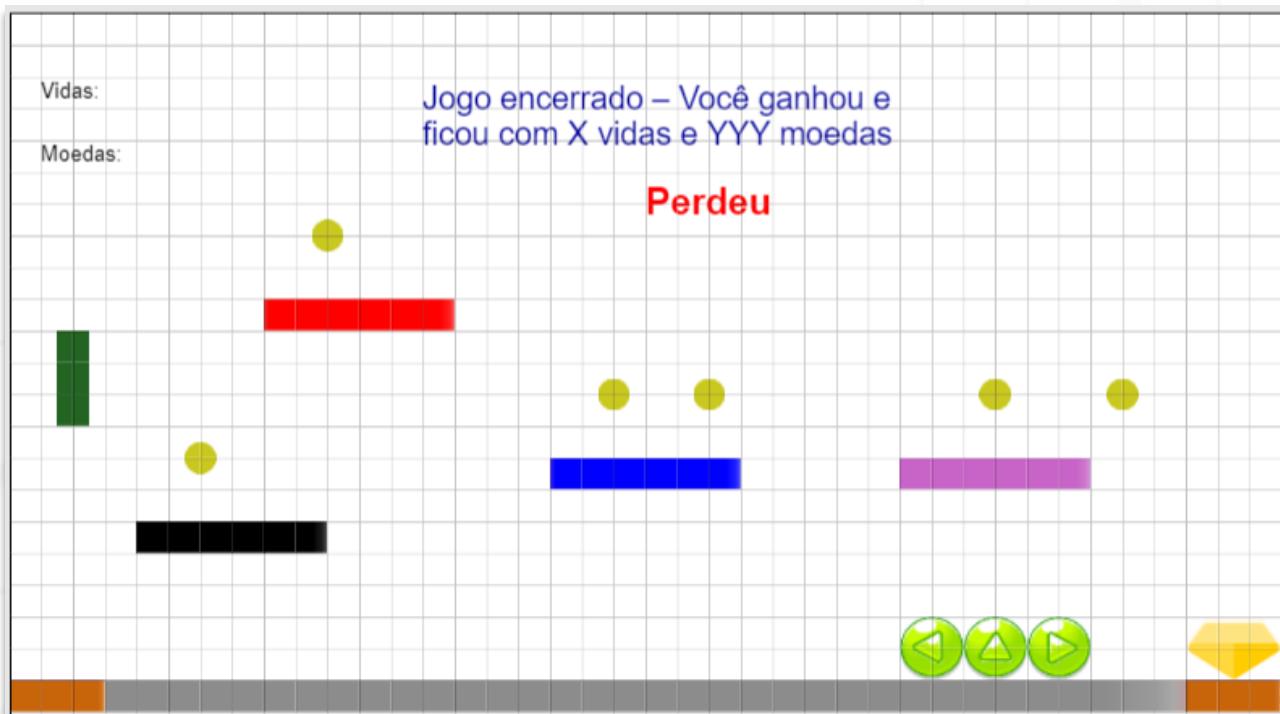
16. Passar para lay_1
17. Passar para a camada Elementos
18. Duplicar moedas de maneira que a tela fique conforme figura.



19. Testar o jogo e verificar a contagem de moedas e a contagem de vidas. Verificar também quando o jogo reinicia, o número de vidas (variável local) é reiniciado (em 3), mas o número de moedas (variável global) não reinicia
20. Passar para **eve_1**
21. Criar uma instrução, para cada vez que o Jogador obtiver 300 pontos, a quantidade de vidas seja aumentada de 1 e a quantidade de moedas volte a zero,
 - a. Evento: System => Compare two values => First value: var_moedas; Comparison = Equal to; Second value 300
 - b. Ação: spr_Jogador => Add to => Instance variable: var_vidas; Value 1
 - c. Ação: System => Set value => Variable: var_moedas; Value: 0
22. Os eventos ficarão conforme figura:



23. Testar o jogo e verificar que sempre que o jogador obtiver 300 moedas a quantidade de vidas do jogador será aumentada em uma unidade e a quantidade de moedas disponíveis voltará a zero.
24. Acrescentar ao jogo as condições para que, caso o jogador chegue ao spr_Ojetivo o jogo seja encerrado e o número de vidas e o valor de moedas mostrado.
25. Passar par lay_1
26. Passar para a layer GUI e colocar um texto para mostrar a mensagem “Jogo encerrado – Você ganhou e ficou com X vidas e YYY moedas”.
 - a. Tipo: Text
 - b. Nome: **txt_Encerrado**
 - c. Texto: Jogo encerrado – Você ganhou e ficou com X vidas e YYY moedas
 - d. Fonte: Arial (24)
 - e. Color: Azul (0, 0, 153)
 - f. Inicial visibility: desmarcado
 - g. Posição, conforme figura
27. A tela ficará conforme figuras



28. Passar para **eve_1**
29. Criar um grupo “Encerrando_Jogo”

30. Criar dentro do grupo “Encerrando_Jogo”, uma instrução para que cada vez que o Jogador colidir com spr_Ojetivo o jogo seja encerrado e mostrado na tela o conteúdo do texto txt_Encerrado.
- Evento: spr_Jogador => On collision with another object => spr_Ojetivo
 - Ação: txt_Encerrado => Set Text: "Jogo encerrado – Você ganhou e ficou com " & spr_Jogador.var_vidas & " vidas e " & var_moedas & " moedas"
 - Ação: txt_Encerrado => Set Visible => Visible

31. Os eventos ficarão conforme figura:

▼ Encerrando_Jogo			
→	spr_Jogador	On collision with	spr_Ojetivo
		txt_Encerrado	Set text to "Jogo encerrado - Você ganhou e ficou com " & spr_Jogador.var_vidas & " vidas e " & var_moedas
Add action			

32. Testar o jogo e notar que, embora a mensagem tenha falado que o jogo foi encerrado, o jogador ainda pode ser movimentado, dando continuidade ao jogo. Para realmente encerrar o jogo, ainda dentro do evento spr_Jogador => On collision with another object => spr_Ojetivo, colocar:

- Ação: System => Set group active => Group name => “Jogador” => State: Deactivated

33. Os eventos ficarão conforme figura:

▼ Encerrando_Jogo			
→	spr_Jogador	On collision with	spr_Ojetivo
		txt_Encerrado	Set text to "Jogo encerrado - Você ganhou e ficou com " & spr_Jogador.var_vidas & " vidas e " & var_moedas
		txt_Encerrado	Set visibility Visible
		System	Set group "Jogador" Deactivated
Add action			

34. Testar o jogo e observar que agora o jogador não pode mais ser movimentado quando o jogo encerrar.
35. Para reiniciar o Jogo, será colocado um texto com uma mensagem avisando que para reiniciar o jogo basta digitar R. Este sprite também poderá ser tocado para reiniciar o jogo. Lembrar que antes de iniciar o jogo é preciso ativar o grupo “Jogador”.
36. Passar para lay_1
37. Passar para a **layer GUI** e colocar um texto com a mensagem para que o jogador **digite a tecla R ou toque sobre a mensagem** para reiniciar o jogo.

- a. Tipo: Text
- b. Nome: **txt_Reiniciar**
- c. Text: Digite R ou toque esta mensagem para reiniciar.
- d. Font: Arial 16
- e. Horizontal Alignment: Center
- f. Vertical Alignment: Center
- g. Initial visibility: Invisible
- h. Posição, conforme figura



38. Passar para eve_1

39. Acrescentar ao evento spr_Jogador => On collision with another object => spr_Objetivo, a ação para mostrar o texto txt_Reiniciar.

- a. Ação: txt_Reiniciar => Set visible => Visible

40. Criar um grupo “Reiniciando_Jogo”

41. Acrescentar dois eventos (um de teclado e outro de Touch) para reiniciar o jogo, restartar as variáveis locais e ativar o grupo Jogador.

- a. Evento: Keyboard => On key pressed => R
OR (utilizando add another condition)
- b. Evento: Touch => Is Touching object => txt_Reiniciar
- c. Ação: System => Reset Global variables => to default
- d. Ação: System => Set Group active => Group name “Jogador” State: Activated

e. Ação: System => Restart layout

42. Os eventos ficarão conforme figura:



▼ Encerrando_Jogo

→ [spr_Jogador] On collision with [spr_Objetivo]	[txt_Encerrado]	Set text to "Jogo encerrado - Você ganhou e ficou com " & spr_Jogador.var_Vidas & " vidas e " & var_moedas & " moedas"
	[txt_Encerrado]	Set visibility Visible
	[System]	Set group "Jogador" Deactivated
	[txt_Reinic平ar]	Set visibility Visible
	Add action	
		Add...

Add event to 'Encerrando_Jogo'

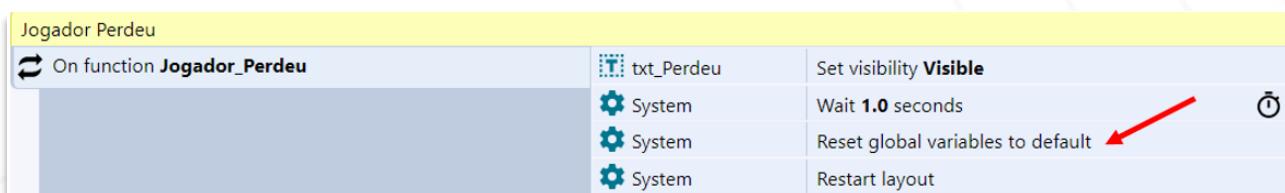
▼ Reiniciando_Jogo

→ [Keyboard] On [R] pressed	[System]	Reset global variables to default
→ [Touch] OR Is touching [txt_Reinic平ar]	[System]	Set group "Jogador" Activated
	[System]	Restart layout
	Add action	
		Add...

43. Testar o jogo e observar que o Jogo pode ser reiniciado com a tecla R ou mesmo com um toque no texto.

44. Como existe a possibilidade do jogador sair do layout ou o número de vidas chegar a zero é importante que na função Jogador_Perdeu, acrescentar a instrução Reset Global variables" conforme a seguir

- Evento: Jogador_Perdeu
- Ação: Reset global variables to default



Jogador Perdeu

On function Jogador_Perdeu	[txt_Perdeu]	Set visibility Visible
	[System]	Wait 1.0 seconds
	[System]	Reset global variables to default (arrow pointing here)
	[System]	Restart layout

45. Salvar o projeto.

46. Fechar o Construct 3.

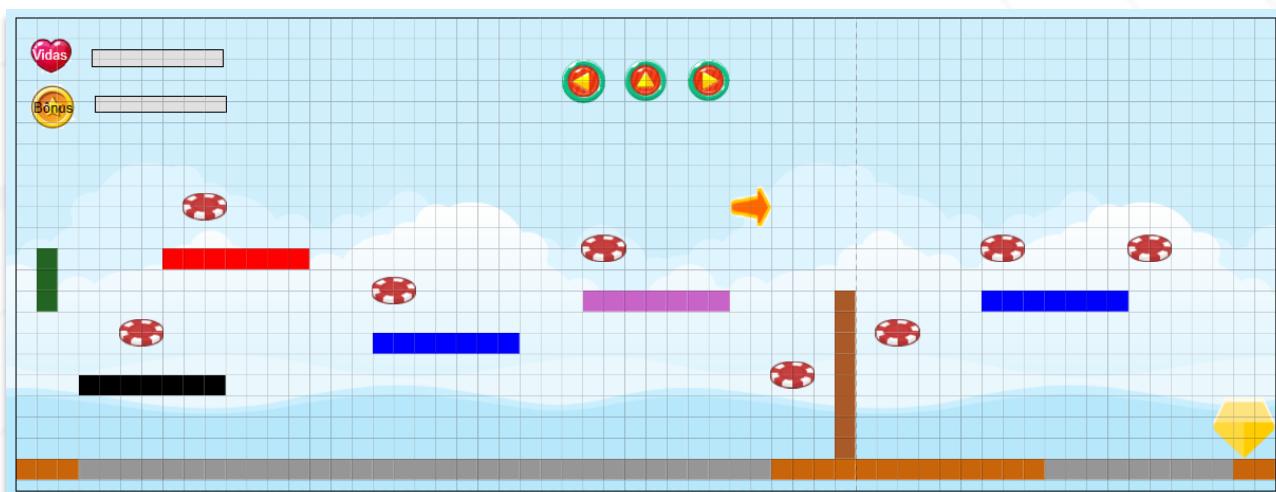
6.2. Exercício

Acrescentar ao exercício da aula anterior, **ExercicioJogoPlataforma05** os objetos e a programação discutida na parte teórica desta aula colocando todas as funcionalidades que foram explicadas no tanto no conteúdo teórico original como no Adicional desta aula 06.

Nomear o jogo como **ExercicioJogoPlataforma06**

1) Passar para o Layout lay_1

- Adicionar 8 sprites para que o jogador possa capturá-los e ganhar bônus. Escolher uma das imagens (Azul, Verde ou Vermelho) dentre os que estão na pasta Aula 06_Elementos. Chamar estes Sprites de spr_Pontos. Colocar o ponto de referência no centro.
- Desmarcar a opção “Snap to grid” no lay_1
- Adicionar uma Progress Bar (bar_Bonus) para fornecer dinamicamente o número de pontos.
- Colocar a imagem do coração (spr_Coracao) para ilustrar a Progress Bar bar_Vidas
- Colocar a imagem da estrela (spr_Bonus) para ilustrar a Progress Bar bar_Bonus
- Criar um objeto texto (txt_Vidas) para escrever sobre o Coração a palavra “Vidas”.
- Criar um objeto texto (txt_Bonus) para escrever sobre o Bônus a palavra “Bônus”.
- Cada um destes Bônus ao serem capturados devem dar ao jogador uma quantidade de 10 pontos através de uma variável global chamada var_bonus e sempre que o jogador atingir 40 pontos ele deverá ganhar uma vida.
- Alterar o número máximo da bar_Vidas para 10

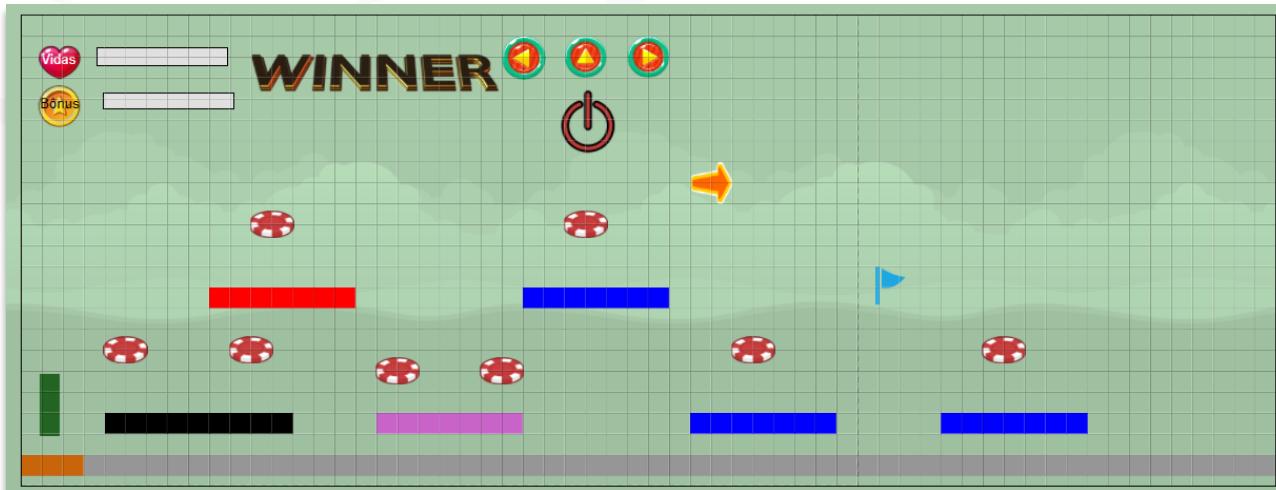


2) Elaborar a programação para:

- Atribuir os pontos sempre que o jogador capturar os Bônus.

- b. Destruir o Bônus capturado para que o jogador não o pegue novamente;
 - c. Adicionar uma vida sempre que a quantidade de Bônus atingir 40
 - d. Atualizar o número de Bônus do jogador a cada tick.
 - e. Garantir que sempre que o jogador perder todas as vidas ou sair do layout ele também perca todos os bônus e volte para o layout lay_1. Fazer isso utilizando uma função
- 3) Passar para o Layout lay_2
 - a. Adicionar 8 sprites para que o jogador possa capturá-los e ganhar bônus.
 - b. Desmarcar a opção “Snap to grid” no lay_2
 - c. Colocar na Progress Bar (bar_Vidas) o máximo de 10.
 - d. Adicionar a Progress Bar (bar_Bonus) para manter o fornecimento dinâmico o número de pontos. (Colocar na mesma posição do lay_1). Configurar máx 40
 - e. Colocar a imagem do coração (spr_Coracao) para ilustrar a Progress Bar bar_Vidas (colocar na mesma posição do lay_1).
 - f. Colocar a imagem da estrela (spr_Bonus) para ilustrar a Progress Bar bar_Bonus (colocar na mesma posição do lay_1).
 - g. Colocar o objeto texto (txt_Vidas) para escrever sobre o Coração a palavra “Vidas”.
 - h. Colocar o objeto texto (txt_Bonus) para escrever sobre o Bônus a palavra “Bônus”.
 - i. Garantir que também nesse Layout (lay_2), sempre que o jogador perder todas as vidas ou sair do layout ele também perca todos os bônus e volte para o layout lay_1. Fazer isso utilizando uma função
 - 4) Elaborar a programação para que
 - a. Quando o jogador passe do lay_1 para o lay_2 ele mantenha o número de vidas. Utilizar uma variável global var_PassaVidas.
 - 5) Passar para o Layout lay_2
 - a. Inserir a imagem Winner com o nome spr_Venceu (obtida na pasta Aula06_Elementos) na posição 1400,250

- b. Inserir a imagem Resetar com o nome spr_Reiniciar (obtida na pasta Aula06_Elementos)



- 6) Alterar a programação para que:
- Quando o jogador atingir o objetivo
 - As imagens Winner e Restart apareçam
 - O jogador pegue a bandeira e vá para a posição acima e no meio da plataforma spr_Pataforma_Mov_Hor.
 - Em seguida o jogador não possa mais mover-se
 - Quando o usuário clicar no Botão Reiniciar o jogo reinicie

7. INSERINDO IMAGENS

Apresentação:

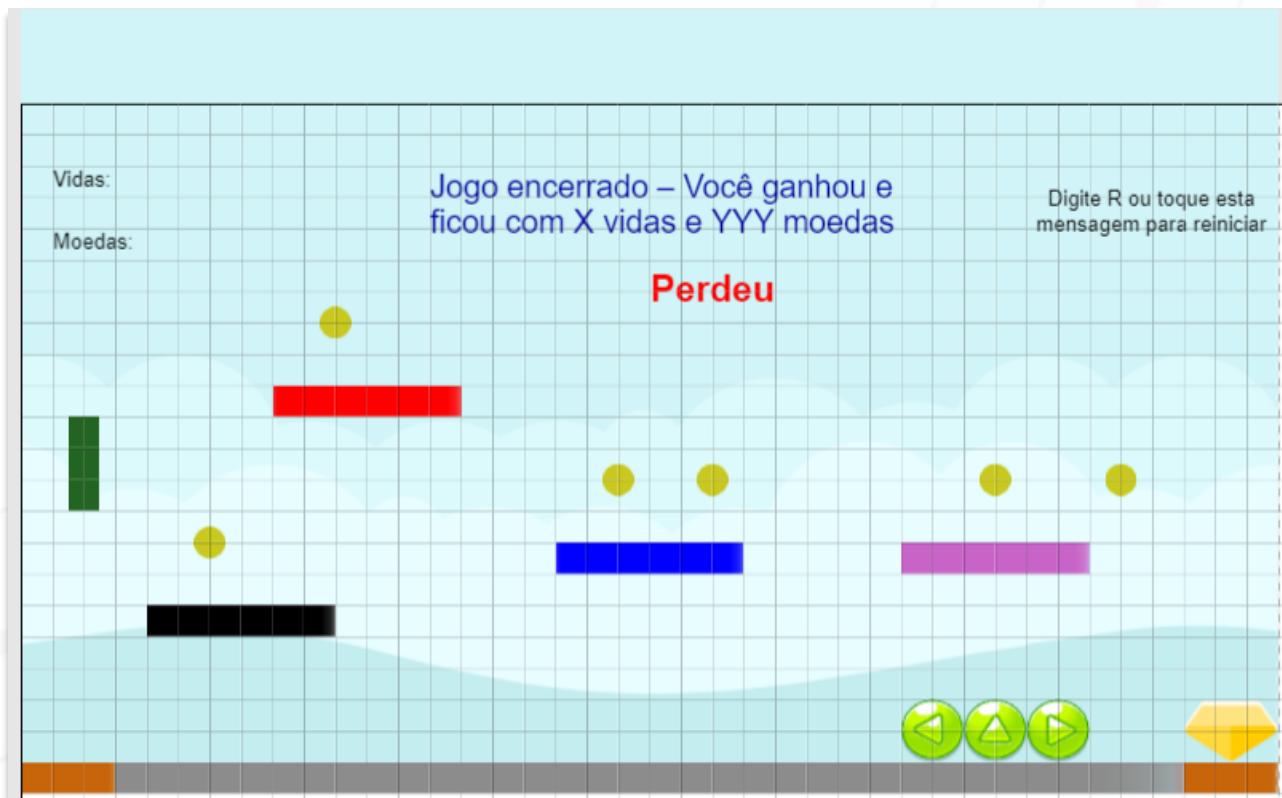
Nesta aula serão inseridas sobre os objetos do jogo imagens, para que o jogo fique mais bonito e agradável. Esta ação, normalmente é feita a partir de desenhos elaborados pela equipe multimídia. Para que algumas imagens se apresentem corretamente na tela do jogo será necessário a inserção de comportamentos e instruções, portanto, este não é um trabalho somente da equipe multimídia.

Caso o jogo tenha objetivo de ser comercializado, as Imagens podem e devem ser desenvolvidas por equipe própria. Caso se deseje, é possível comparar ou mesmo encomendar imagens, ou ainda, todo um cenário para o jogo. Para este curso serão utilizadas imagens gratuitas disponíveis na internet. Dentre os diversos sites existentes pode ser citado o <https://www.kenney.nl/assets> que possui uma grande quantidade de material, muitos gratuitos. Outra sugestão de site para obter imagens e sons é o <https://itch.io/game-assets>.

7.1. Desenvolvimento

1. Baixar e descompactar a pasta “**Aula07_Elementos**” que está no AVA.
2. Colocar o conteúdo desta pasta dentro da pasta Construct que está na área de trabalho.
3. Abrir o Construct 3 através do endereço: <https://editor.construct.net/>
4. Se necessário passar o idioma do editor para inglês,
 - a. Menu
 - b. Configurações
 - c. English (US)
 - d. Fechar
5. Fazer login com seu usuário
 - a. Guest
 - b. Log in
 - c. Preencher Username e Password
 - d. Login
6. Abrir o projeto **JogoPlataforma06** (que deve estar na pasta Construct da área de trabalho).
7. Gravar o projeto utilizando: **Menu; Project; Save as; Save as a single file**, com o nome **JogoPlataforma07**

8. Passar para **lay_1**
9. Passar para a **layer Elementos** e inserir um fundo.
 - a. Tipo: Tiled Backgrond
 - b. Nome: **bac_Fundo**
 - c. Imagem: Background.png encontrado na pasta “Aula07_Elementos” disponível nesta aula.
 - d. Clicar com o botão direito na imagem e escolher a opção “Z order”
 - i. Passar a posição para “Send to bottom of layer”
 - e. Posição: Conforme figura. Atenção, primeiro aumente e posicione a parte inferior próximo ao chão, em seguida reduza ou aumente a imagem utilizando a parte superior para que os efeitos fiquem nos locais desejados. Provavelmente a parte superior ficará maior que a visão da camera, entretanto, isto não afetará a qualidade do material visto que, o excedente, não será mostrado em tempo de execução.



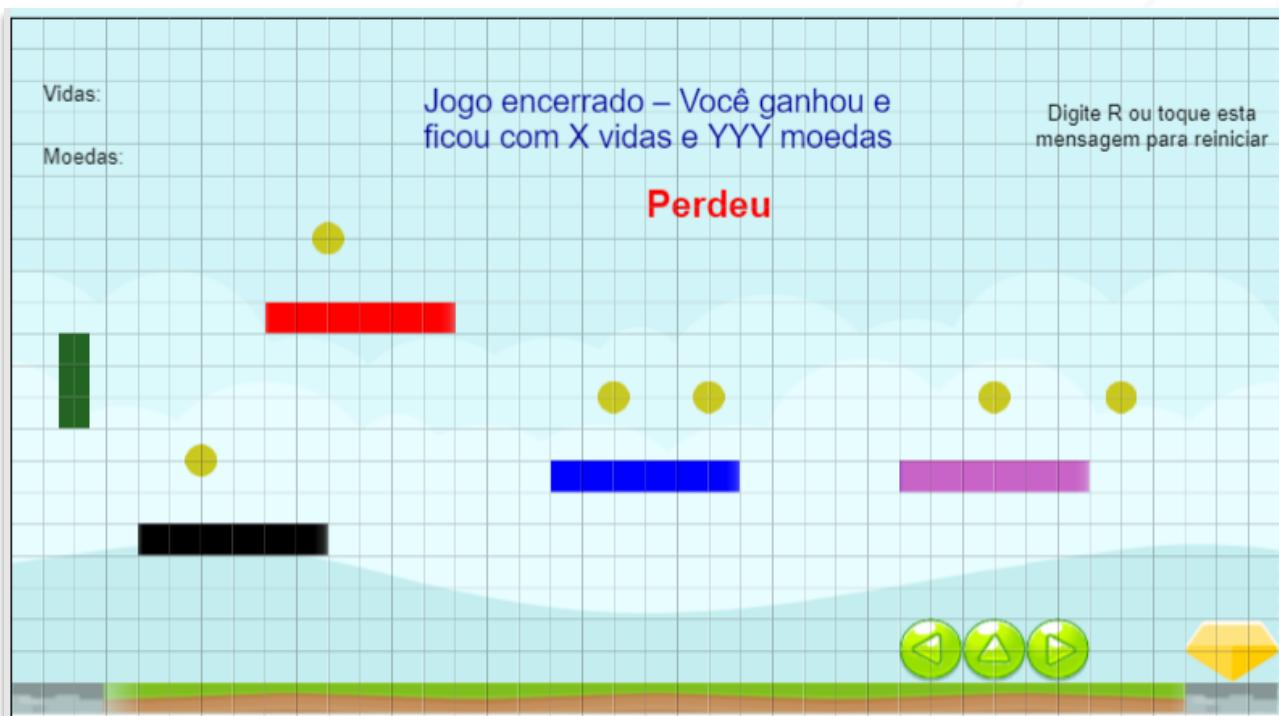
10. Abrir (com um duplo clique) o Sprite **spr_Chao**.
 - a. Alterar a imagem para Chao.png utilizando o botão “Load images from files”.
 - b. Utilizar a ferramenta Crop para ficar somente com a parte escura da imagem.
 - c. Posição: Conforme figura.

- d. Redimensionar as imagens (nos dois locais onde está a figura spr_Chao) para que fiquem na posição que estavam anteriormente.

11. Abrir (com um duplo clique) o Sprite **spr_Morte**

- Alterar a imagem para Morte_.png utilizando o botão “Load images from files”.
- Utilizar a ferramenta Crop para ficar somente com a parte escura da imagem.
- Posição: Conforme figura
- Redimensionar a imagem para que fique na posição que estava anteriormente.

12. Após a inserção do fundo e das imagens na parte inferior a tela ficará conforme figura abaixo:



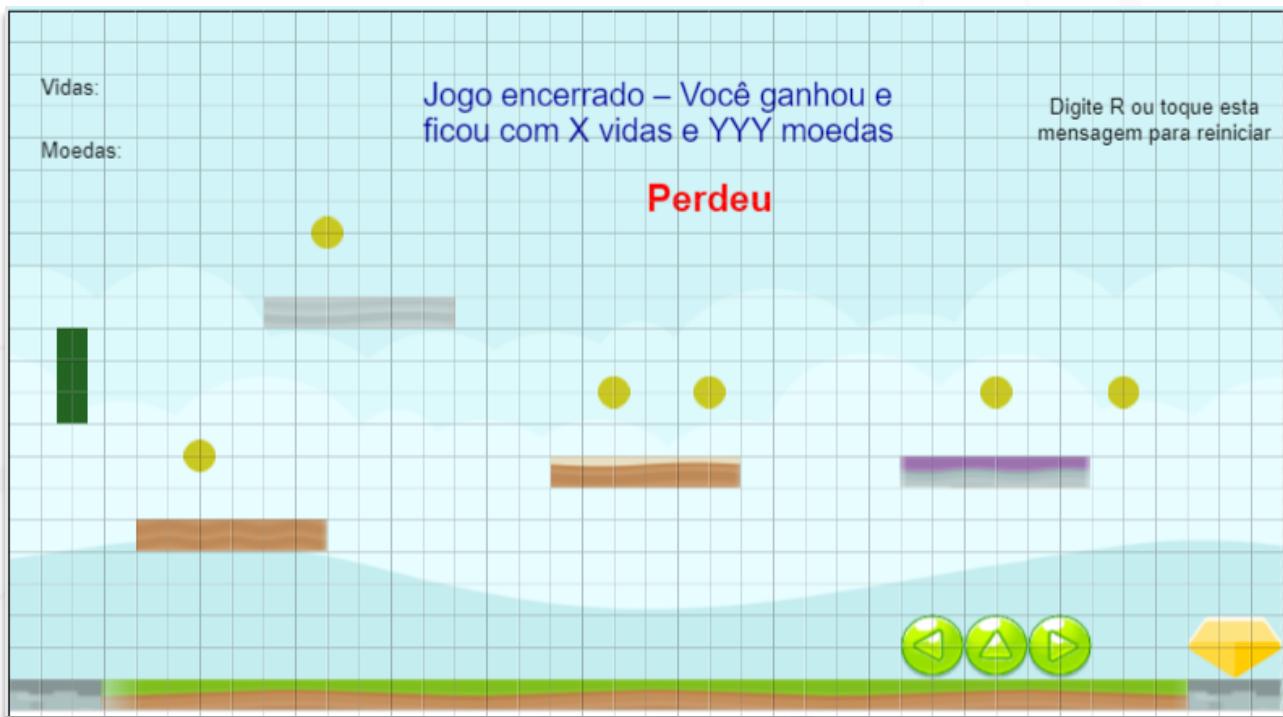
Obs: Se necessário ajuste a imagem de spr_Morte para ficar abaixo das imagens do chão e evitar que fique um pequeno vazio na união dos elementos. Clicando com o botão direito nas imagens spr_Chao, utilize o Z order para colocar estas imagens acima da spr_Morte, usando a instrução “Send to top of layer”.

13. Abrir (com um duplo clique) o Sprite **spr_Plataforma**

- Alterar a imagem para Plataforma.png utilizando o botão “Load images from files”.
- Redimensionar a imagem para que fique na posição que estava anteriormente.

14. Abrir (com um duplo clique) o Sprite **spr_Plataforma_Vasada**

- a. Alterar a imagem para Plataforma_Vasada.png utilizando o botão “Load images from files”.
 - b. Redimensionar a imagem para que fique na posição que estava anteriormente.
15. Abrir (com um duplo clique) o Sprite **spr_Plataforma_Mov_Hor**
- a. Alterar a imagem para Plataforma_Move_Hor.png utilizando o botão “Load images from files”.
 - b. Redimensionar a imagem para que fique na posição que estava anteriormente.
16. Abrir (com um duplo clique) o Sprite **spr_Plataforma_Mov_Ver**
- a. Alterar a imagem para Plataforma_Move_Ver.png utilizando o botão “Load images from files”.
 - b. Utilizar a ferramenta Crop para ficar somente com a parte escura da imagem.
 - c. Redimensionar a imagem para que fique na posição que estava anteriormente.
17. Após a inserção de todas as imagens desta aula, a tela ficará conforme figura abaixo:



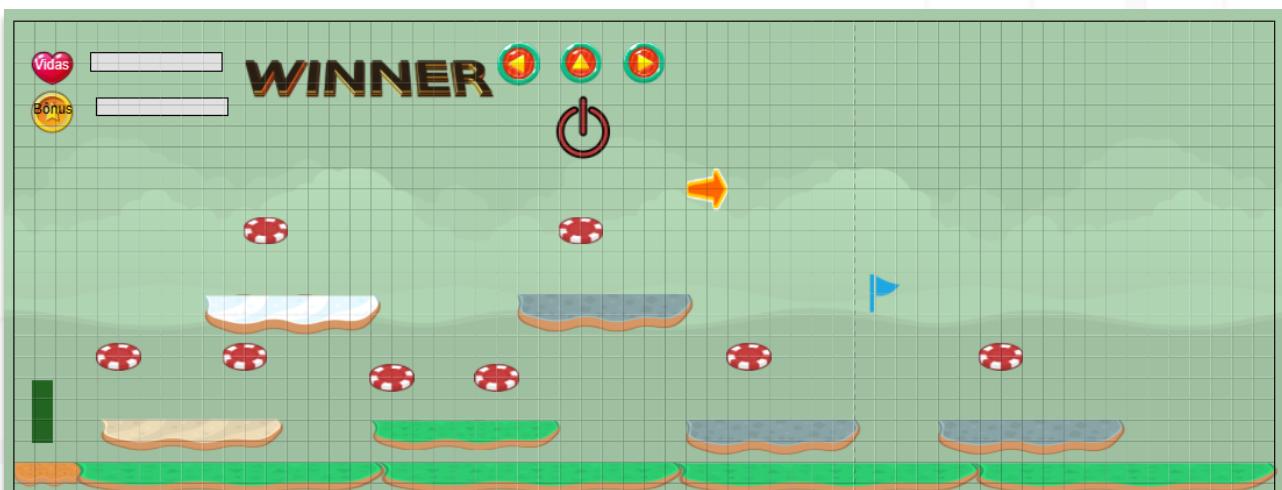
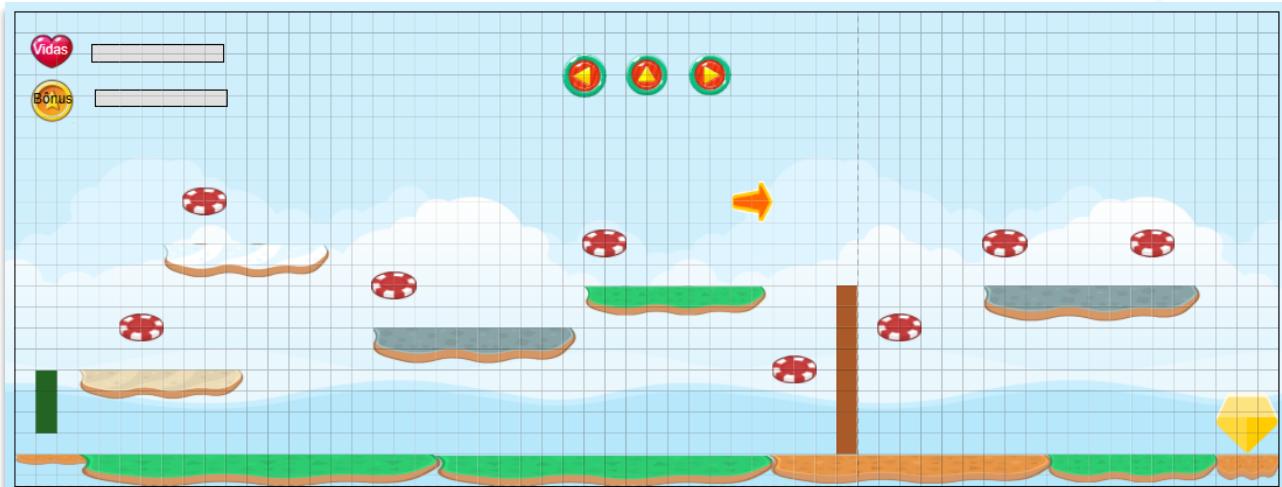
18. Testar o Jogo e observar.
19. Salvar o projeto.
20. Fechar o Construct 3.

7.2. Exercício

Adicionar ao exercício da aula anterior, imagens nas plataformas e os demais elementos do jogo.

Nomear o jogo como **ExercicioJogoPlataforma07**

Os Layouts ficarão conforme a seguir:



8. INSERINDO ANIMAÇÕES

Apresentação: Nesta aula serão inseridas imagens para o jogador e estas imagens serão sobrepostas, de maneira a produzirem animações. Serão executadas animações para o objetivo, que será substituído pela figura de uma bandeja, para as moedas e para o jogador, este com 4 animações: na posição inicial sobre a plataforma, jogador andando, jogador pulando e jogador caindo.

Em desenvolvimentos profissionais com objetivos financeiros, as imagens para estas animações também devem ser construídas por equipes multimídia. Para esta aula serão utilizadas imagens obtidas de <https://www.kenney.nl/assets>. Outra sugestão de site para obter imagens e sons é o <https://itch.io/game-assets>.

8.1. Desenvolvimento

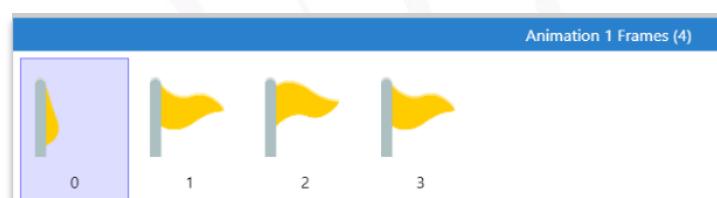
1. Baixar e descompactar a pasta “**Aula08_Elementos**” que está no AVA.
2. Colocar o conteúdo desta pasta dentro da pasta Construct que está na área de trabalho.
3. Abrir o Construct 3 através do endereço: <https://editor.construct.net/>
4. Se necessário passar o idioma do editor para inglês,
 - a. Menu
 - b. Configurações
 - c. English (US)
 - d. Fechar
5. Fazer login com seu usuário
 - a. Guest
 - b. Log in
 - c. Preencher Username e Password
 - d. Login
6. Abrir o projeto **JogoPlataforma07** (que deve estar na pasta Construct da área de trabalho).
7. Gravar o projeto utilizando: **Menu; Project; Save as; Save as a single file**, com o nome **JogoPlataforma08**
8. Passar para **lay_1**

9. Passar para a **layer Elementos** e inserir um conjunto de imagens para criar uma animação para substituir o spr_Ojetivo. Esta animação será um conjunto de bandeiras inseridas conforme a seguir:

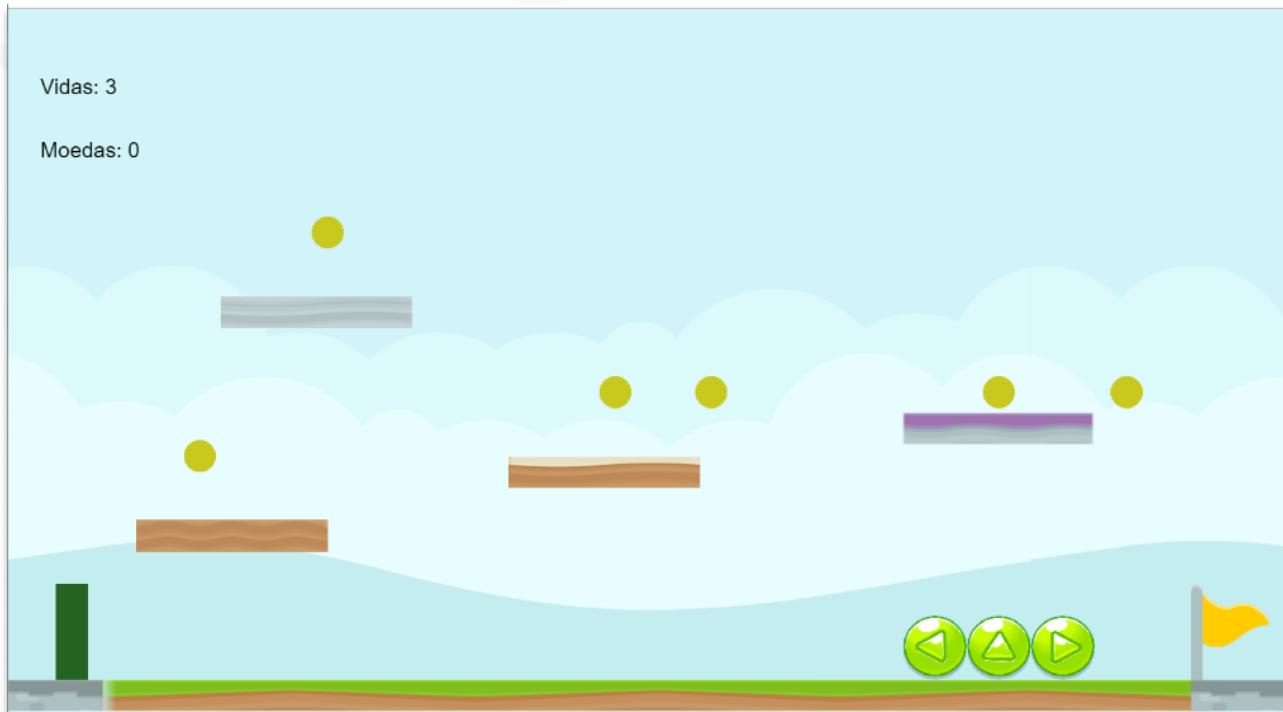
Obs: Para agilizar, estas imagens podem ser inseridas a partir da Barra “Animation frames” que pertence ao editor de imagens.

- a. Abrir com um duplo clique o objeto spr_Ojetivo
- b. Clicar com o botão direito no Painel “Animation frames” e escolher a opção “import frames” e em seguida “from files”
- c. Procurar na pasta “Aula08_Elementos” fornecida com o material desta aula, as Imagens: Bandeira1.png, Bandeira2.png e Bandeira3.png
- d. Estas imagens vão se posicionar em 1, 2 e 3.
- e. Apagar a imagem 0 que está com a figura antiga.
- f. Passar para a nova imagem 0 o ponto de origem para Top-left
- g. Aplicar em todas as imagens da animação a Origem Top-left utilizando a opção “Apply to whole animations” do menu suspenso da opção “Origin”.
- h. Redimensionar todas as imagens para Width 64 pixels e Height 64 pixels escolhendo uma das animações utilizando a ferramenta Resize. (Para aplicar em todas as imagens basta marcar o botão “Apply to whole animations”).
- i. Passar para o Painel “Animations” e marcar a animação que está com o nome de Animation 1
- j. Passar para o Painel “Animation Properties” e marcar o botão “loop” para a animação ficar em loop.
- k. Voltar ao Painel “Animations” clicar com o botão direito na Animation 1 e utilizar a opção “Preview” para testar a animação.
- l. Fechar o Preview e melhorar a animação duplicando e reposicionando a imagem 1 de maneira que a animação ficará com 4 imagens, conforme figura abaixo:

Obs: Para duplicar basta clicar com o botão direito na imagem e escolher “Duplicate”, e para reposicionar basta arrastar a imagem na barra “Animation frames”



- m. Fechar o “Animations Editor”
 - n. Passar para Lay_1 e reposicionar a imagem da bandeira.
10. Testar o jogo e verificar a animação conforme figura abaixo:



11. Manter-se na **layer Elementos** e inserir um conjunto de imagens para criar uma animação para substituir o spr_Moeda. Esta animação será formada por diversas imagens de moedas conforme a seguir:

Obs: Para agilizar, estas imagens podem ser inseridas a partir da Barra “Animation frames” que pertence ao editor de imagens, basta:

- a. Abrir com um duplo clique o objeto spr_Moedas
- b. Clicar com o botão direito no Painel “Animation frames” e escolher a opção “import frames” e em seguida “from files”
- c. Procurar na pasta “Aula08_Elementos” fornecida com o material desta aula, as Imagens: Moeda1.png, Moeda2.png e Moeda3.png (não pegar a moeda 4)
- d. Estas imagens vão se posicionar em 1, 2 e 3.
- e. Apagar a imagem 0
- f. Redimensionar todas as imagens para Width 32 pixels e Height 32 pixels escolhendo uma das animações utilizando a ferramenta Resize. (Para aplicar em todas as imagens basta marcar o botão “Apply to whole animations”).

- g. Passar para o Painel “Animations” e marcar a animação que está com o nome de Animation 1
- h. Passar para o Painel “Animation Properties” e marcar o botão “loop” para a animação ficar em loop.
- i. Voltar ao Painel “Animations” clicar com o botão direito na “Animation1” e utilizar a opção “Preview” para testar a animação.
- j. Notar que a animação está se movendo somente para a esquerda
- k. Fechar o Preview e melhorar a animação duplicando espelhado e reposicionando as imagens 1 e 2 de maneira que a animação ficará com 5 imagens, conforme figura abaixo:

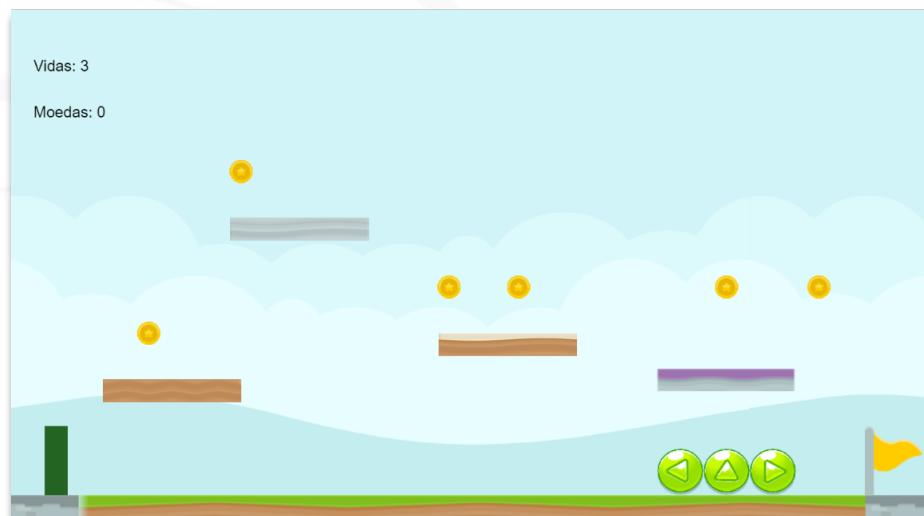
Obs: Para duplicar basta clicar com o botão direito na imagem e escolher “Duplicate”, para espelhar utilizar a ferramenta “Mirror” e para reposicionar basta arrastar a imagem na barra “Animation frames”

Obs: Atenção que para ver o espelhamento precisa clicar na imagem anterior.

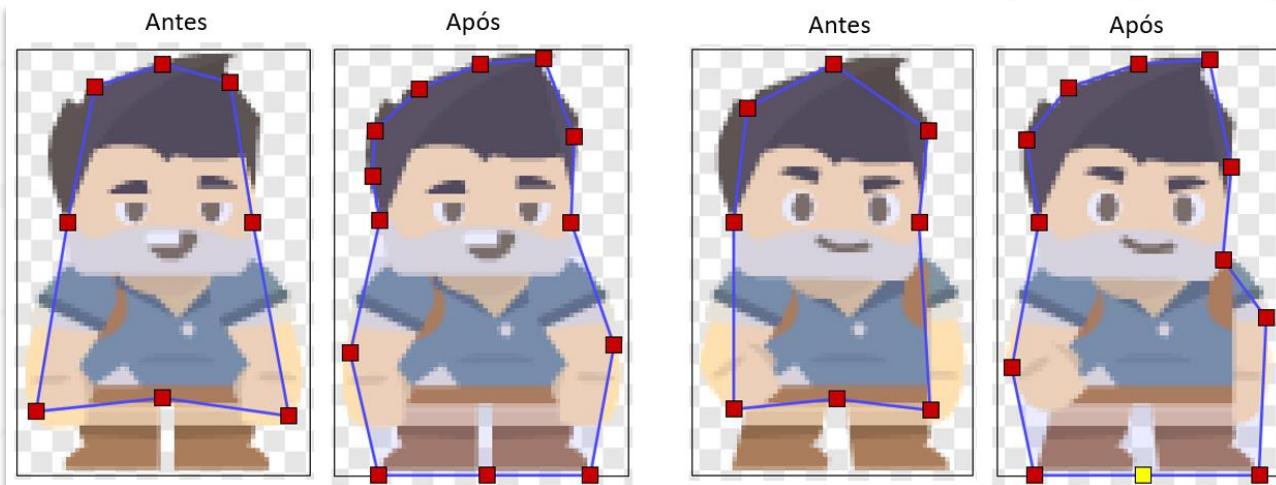


I. Fechar o “Animations Editor”

12. Testar o jogo e verificar a animação conforme figura abaixo:



- a. Abrir com um duplo clique o objeto spr_Jogador
- b. Clicar com o botão direito no Painel “Animation frames” e escolher a opção “import frames” e em seguida “from files”
- c. Procurar na pasta “Aula08_Elementos” fornecida com o material desta aula, as Imagens: Idle1.png e Idle2.png
- d. Apagar a imagem 0
- e. Utilizar a ferramenta crop para ajustar as duas imagens (**utilizar a tecla shift para cortar todas as imagens simultaneamente**)
- f. Passar na imagem 0 a Origem: Botton
- g. Com a imagem 0 marcada clicar com o botão direito em Origin e escolher “Apply to whole animation”. (o ponto de origem será marcado igual ao da imagem 0 em todas as imagens **desta animação**).
- h. No Painel “Animations”, renomear a animação para IDLE
- i. Com a animação marcada, passar para o painel “Animation Properties” e marcar o botão para a animação ficar em loop.
- j. Testar a animação, clicando com o botão direito em IDLE e escolhendo a opção Preview.
- k. Após observar, fechar o Preview e fechar o editor de imagens.
- l. Ajustar o Polígono de colisão para que o jogador fique com os pés exatamente sobre o spr_Chao seguindo as instruções abaixo
 - i. Ativar a ferramenta “Edit the collision polygon”
 - ii. Alterar o polígono de colisão para que fique conforme figura abaixo.



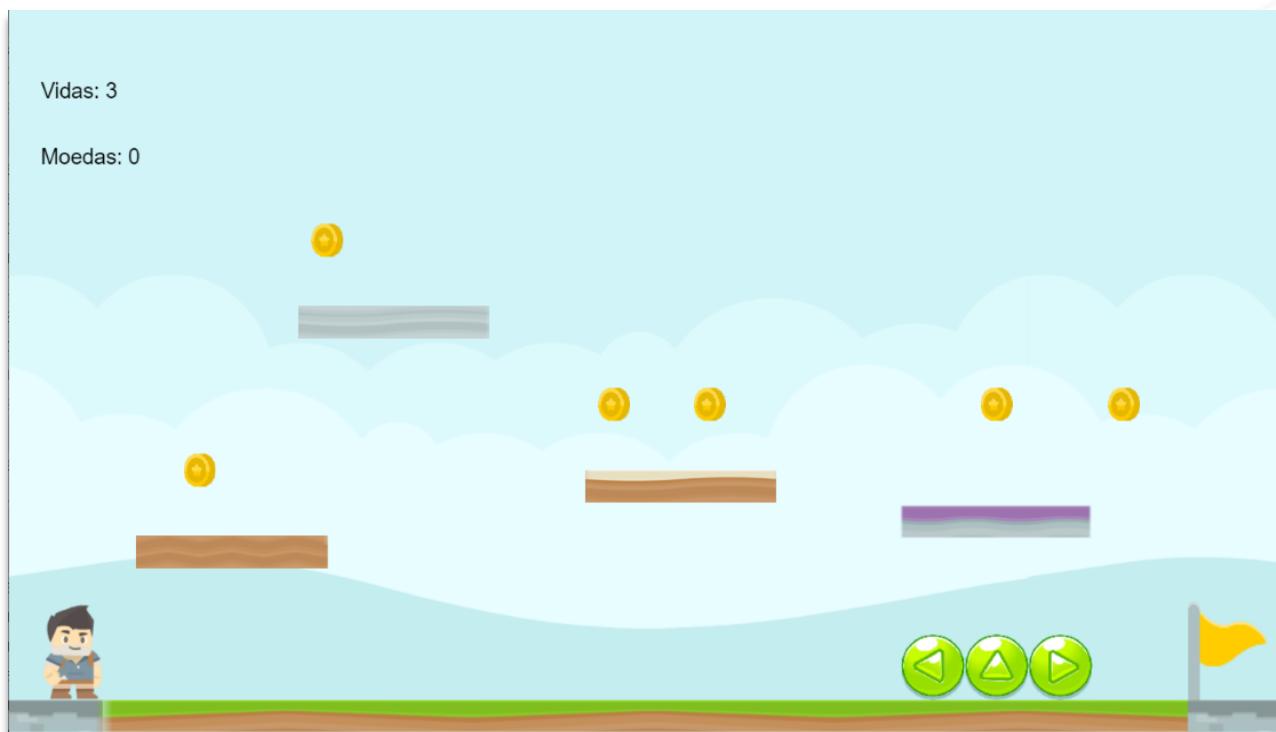
Obs: **Atenção, que todos os pontos interiores, nesta e nas demais animações devem estar coincidentes como “Image point” que foi**

colocado no Bottom e deverá ser colocado em todas as demais animações.

Obs: Para adicionar mais pontos no polígono, basta clicar com o botão direito em um ponto e escolher a opção Add Point

m. Fechar o “Animations Editor”

13. Testar o jogo e observar que o spr_Jogador já está executando a animação.



14. Editar o spr_Jogador e adicionar a animação para quando ele estiver andando (Walk)

- a. Clicar com o botão direito no quadro Animation e escolher Add Animation
- b. Nome: Walk
- c. Imagens: walk1, walk2 e walk3 (obtidas na pasta Aula08_Elementos desta aula).
- d. Apagar a imagem 0.
- e. Utilizar a ferramenta crop para eliminar as partes brancas das três imagens.
- f. Ponto de origem: Botton, na imagem 0
- g. Passar o ponto de origem para todas as imagens desta animação
- h. Ajustar o Polígono de colisão para que o jogador fique com os pés exatamente sobre o spr_Chao.
- i. Com a animação marcada, passar para a Barra Properties e marcar o Botão **Loop**

- j. Testar a animação
 - k. Fechar o “Animations Editor”
15. É possível ver esta animação no jogo, para isto basta, na Barra Properties de spr_Jogador, em “Properties” procurar o item “Inicial animation”, que deve estar em default, e alterar para walk.
16. Editar o spr_Jogador e adicionar a animação para quando ele estiver pulando (Jump)
- a. Clicar com o botão direito no quadro Animation e escolher Add Animation
 - b. Nome: Jump
 - c. Imagens: jump1, jump2, jump3 e jump4 (obtidas na pasta Aula08_Elementos desta aula).
 - d. Apagar a imagem 0
 - e. Utilizar a ferramenta crop para eliminar as partes brancas das quatro imagens.
 - f. Ponto de origem: Botton, na imagem 0
 - g. Ajustar o Polígono de colisão para que o jogador fique com os pés exatamente sobre o spr_Chao.
 - h. Passar o ponto de origem para todas as imagens desta animação
 - i. Com a animação marcada, passar para a Barra Properties marcar o botão **Loop**
 - j. Testar a animação
 - k. Fechar o “Animations Editor”
17. É possível ver esta animação no jogo, para isto basta, na Barra Properties de spr_Jogador, em “Properties” procurar o item “Inicial animation”, que deve estar em default, e alterar para Jump.
18. Editar o spr_Jogador e adicionar a animação para quando ele estiver caindo (Fall)
- a. Clicar com o botão direito no quadro Animation e escolher Add Animation
 - b. Nome: Fall
 - c. Imagens: fall1, fall2 e fall3 (obtidas na pasta Aula08_Elementos desta aula).
 - d. Apagar a imagem 0
 - e. Utilizar a ferramenta crop para eliminar as partes brancas nas três imagens.
 - f. Ponto de origem: Botton, na imagem 0
 - g. Passar o ponto de origem para todas as imagens desta animação.

- h. Ajustar o Polígono de colisão para que o jogador fique com os pés exatamente sobre o spr_Chao.
 - i. Com a animação marcada, passar para a Barra Properties e marcar o botão **Loop**
 - j. Testar a animação
19. É possível ver esta animação no jogo, para isto basta, na Barra Properties de spr_Jogador, em “Properties” procurar o item “Inicial animation”, que deve estar em default, e alterar para Fall.
20. Passar para **eve_1**
21. Criar um grupo e nomear como “Animacoes”
22. Colocar um comentário “Jogador andando”
23. No grupo animações colocar dois eventos na condição de **and** (os dois devem estar acontecendo simultaneamente) para executar a animação “Walk” se a sprite spr_Jogador estiver no chão e se movendo:
Obs: Para criar a condição **and**, deve-se, após criar a primeira condição (primeiro evento), clicar com o botão direito no início da condição (**como sub-evento do grupo Animações**) e no menu suspenso que surgir escolher **“add another condiction”**
 - a. Evento: spr_Jogador => Plataform is on floor
 - b. Evento: spr_Jogador => Plataform is moving
 - c. Ação: spr_Jogador => Set animation to “Walk” (play from beginning)
24. Colocar um comentário “Jogador parado”
25. No grupo animações colocar dois eventos na condição de **and** (os dois devem estar acontecendo simultaneamente) para executar a animação “IDLE” se a sprite spr_Jogador estiver no chão e não estiver se movendo:
 - a. Evento: spr_Jogador => Plataform is on floor
 - b. Evento: spr_Jogador => X Plataform is moving
Obs: O **X** é a negação do **is moving**. É obtido clicando com o botão direito do Mouse no início da instrução “Plataform is moving” e no menu suspenso que surgir escolher “Invert”.
 - c. Ação: spr_Jogador => Set animation to “IDLE” (play from beginning)
26. Colocar um comentário “Jogador pulando”
27. No grupo animações colocar o evento para executar a animação “Jump” se a sprite spr_Jogador estiver pulando:
 - a. Evento: spr_Jogador => Plataform is jumping

- b. Ação: spr_Jogador => Set animation to "Jump" (play from beginning)
28. Colocar um comentário "Jogador caindo"
29. No grupo animações colocar o evento para que quando o jogador estiver caindo "falling" seja executada a animação "Fall".
- Evento: spr_Jogador => Platform is on falling
 - Ação: spr_Jogador => Set animation to "Fall" (play from beginning).
30. Após a realização, os eventos ficarão conforme figura:

▼ Animacoes

Jogador andando			
spr_Jogador	Platform is on floor	spr_Jogador	Set animation to "Walk" (play from beginning)
spr_Jogador	Platform is moving	Add action	Add...
Jogador parado			
spr_Jogador	Platform is on floor	spr_Jogador	Set animation to "IDLE" (play from beginning)
spr_Jogador	X Platform is moving	Add action	Add...
Jogador pulando			
spr_Jogador	Platform is jumping	spr_Jogador	Set animation to "Jump" (play from beginning)
		Add action	Add...
Jogador caindo			
spr_Jogador	Platform is falling	spr_Jogador	Set animation to "Fall" (play from beginning)
		Add action	Add...

Obs: Para melhorar a animação fall é aconselhável passar a última imagem para a posição de imagem 0

31. Testar o Jogo e observar que agora conforme o movimento, parado, andando, pulando ou caindo as animações são diferentes, entretanto ainda existe um problema, quer o jogador ande para a esquerda, quer ande para a direita a animação é andando para a direita. Para resolver este problema, quando o jogador estiver andando para a esquerda sua posição será espelhada com a instrução Mirrored e quando estiver andando para a direita sua posição não será espelhada "Not Mirrored". Estas instruções serão colocadas na movimentação do jogador.
32. Ainda em **eve_1**
33. Passar para o grupo "Jogador"
34. Nos eventos que movimentam o jogador para a esquerda inserir seguinte ação:
- Ação: spr_Jogador => set Mirrored => Mirrored
35. Nos eventos que movimentam o jogador para a direita inserir seguinte ação:
- Ação: spr_Jogador => set Mirrored => Not Mirrored

36. Aproveitar a oportunidade e colocar 3 comentários na movimentação do jogador
“Jogador andando para a esquerda”; “Jogador andando para direita” e “Jogador pulando”
37. Após a realização, os eventos ficarão conforme figura:

▼ Jogador

Movimentação do Jogador			
Jogador andando para a esquerda			
Keyboard	A is down	spr_Jogador	Simulate  Platform pressing Left
Keyboard	OR  is down	spr_Jogador	Set Mirrored
Touch	OR Is touching  btn_Esquerda		Add action
Jogador andando para a direita			
Keyboard	D is down	spr_Jogador	Simulate  Platform pressing Right
Keyboard	OR  is down	spr_Jogador	Set Not mirrored
Touch	OR Is touching  btn_Direita		Add action
Jogador pulando			
Keyboard	W is down	spr_Jogador	Simulate  Platform pressing Jump
Keyboard	OR  is down	spr_Jogador	Add action
Touch	OR Is touching  btn_Saltar		

38. Salvar o projeto.
39. Fechar o Construct 3.

8.2. Exercício

Acrescentar ao exercício da aula anterior, os objetos e a programação discutida na parte teórica desta aula.

Alterar as imagens utilizando os elementos fornecidos na pasta Aula 07 Adicionais ou outros obtidas na Internet ou em um dos sites fornecidos no início desta aula.

Nomear o jogo como **ExercicioJogoPlataforma08**

- 1) Abrir o ExercicioJogoPlataforma07 e salvar com o nome ExercicioJogoPlataforma08
- 2) Acrescentar às imagens da premiação mais elementos de maneira que fiquem animados, alterando entre os dos modelos das 3 cores disponibilizadas (vermelho, azul e verde)

- 3) Acrescentar na seta uma segunda imagem (com fundo branco na pasta point Aula 07 Adicionais) para que a seta fique animada.
- 4) Adicionar 4 imagens para produzir 4 animações no jogador (Idle, Walk, Jump e Fall). Escolher dentre as imagens das pastas Player, Players ou Poses da pasta Aula 07 Adicionais)
- 5) Criar a programação para ativar as animações do jogador, dependendo da movimentação do jogador.

