



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE

AULA:

## Jogos baseados em plataforma (parte 1)

### Programação Multimídia

**Alba Lopes, Profa.**

<http://docentes.ifrn.edu.br/albalopes>  
[alba.lopes@ifrn.edu.br](mailto:alba.lopes@ifrn.edu.br)

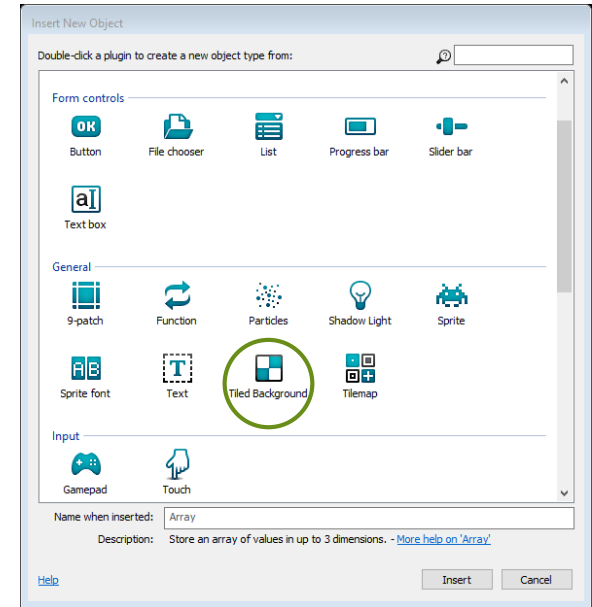
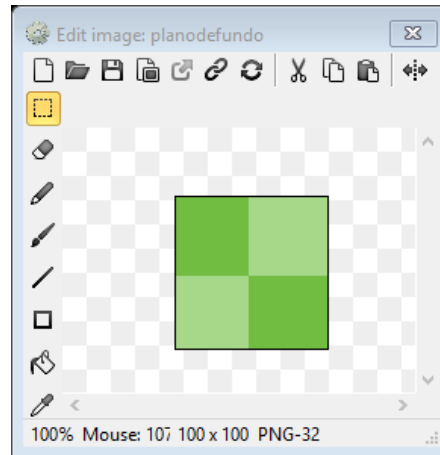
# Objetos essenciais para jogos Plataforma

- ▶ Um jogo de plataforma consiste de dois objetos essenciais:
  - ▶ Sólidos (solids)
  - ▶ Plataformas (platforms)
- ▶ Os **Solids** foram os pisos e as paredes que impedem os movimentos do jogador
- ▶ As **Platforms** são objetos que realizam as ações de pular, correr, lutar, etc..
- ▶ Ao pressionar as setas para esquerda e direita, os objetos do tipo Platform se movem, ao pressionar a seta para cima, o objeto pula
- ▶ Quando não há um objeto do tipo sólido, os objetos sofrem ação da gravidade e caem.



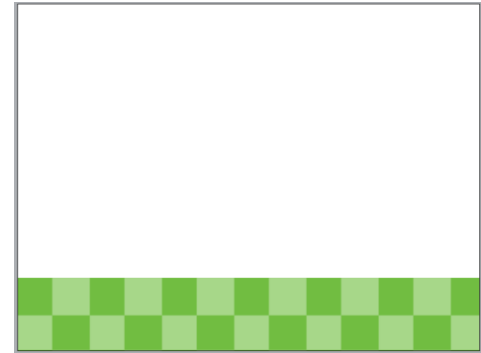
# Criando um jogo formato Plataforma

- ▶ Crie um novo projeto
- ▶ Adicione um novo objeto do tipo TiledBackground
- ▶ Utilizando as os elementos do pacote da aula, abra a figura de desenho do Construct, crie um plano de fundo e feche o editor de imagem



# Criando um objeto Solid

- ▶ Perceba que, ao redimensionar o plano de fundo, o padrão é replicado
- ▶ Adicione um comportamento ao plano de fundo do tipo **Solid**



The screenshot shows the Axure RP software interface. On the left is the Properties panel for the object 'fazenda\_bg'. On the right is the Behaviors panel for 'fazenda\_bg: Behaviors'. The main canvas shows a green and white checkerboard pattern at the bottom of a white area.

**Properties Panel:**

- Object type properties
  - Name: fazenda\_bg
  - Plugin: Tiled Background
  - UID: 0
  - Global: No
- Common
  - Layer: PlanoDeFundo
  - Angle: 0
  - Opacity: 100
  - Position: 0, 379.011
  - Size: 639.063, 101
- Instance variables
  - Add / edit: [Instance variables](#)
- Behaviors
  - Solid
    - Initial state: Enabled
    - Add / edit: [Behaviors](#)
- Effects
  - Blend mode: Normal
  - Add / edit: [Effects](#)
- Container
  - No container: [Create](#)
- Properties
  - Image: [Edit](#)
  - Initial visibility: Visible
  - Hotspot: Top-left
- More information: [Help](#)

**Behaviors Panel:**

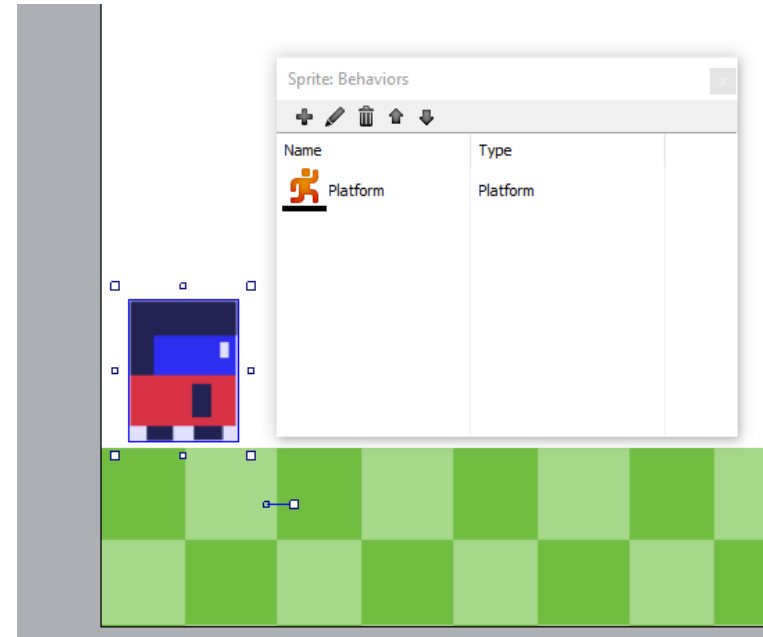
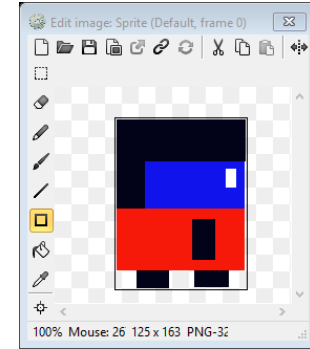
Name	Type
Solid	Solid

**Add / edit:** Click to add, change or remove behaviors.



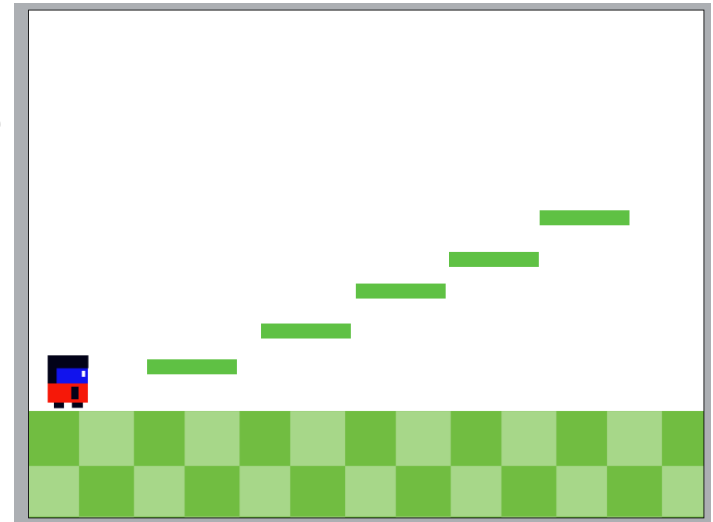
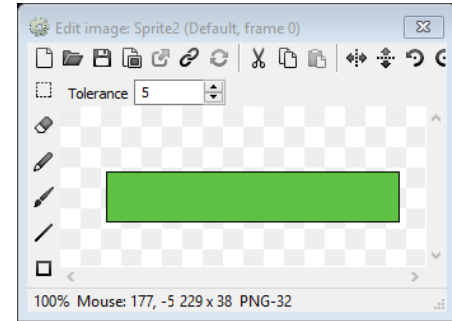
## Criando o objeto do tipo Platform

- ▶ Insira um novo objeto do tipo Sprite para ser o personagem do jogo.
- ▶ À esse objeto, adicione o comportamento de Platform.
- ▶ Execute o seu projeto e veja que seu personagem é capaz de andar para ambos os lados e pular.



# Criando o objeto do tipo Platform

- ▶ Vamos criar mais alguns objetos do tipo Solid, para serem plataformas do nosso jogo.
- ▶ Adicione mais uma Sprite no jogo e desenhe as plataformas desejadas.
- ▶ Adicione o comportamento de Solid também a esse objeto.
- ▶ Mantenha a tecla **Ctrl** pressionada e replique esse objeto pela tela.
- ▶ Execute o seu jogo e veja que seu personagem consegue escalar pelos objetos.



# Aumentando o cenário do jogo

- ▶ Agora, vamos expandir o nosso cenário.
- ▶ Defina o Layout do jogo para 1640x480

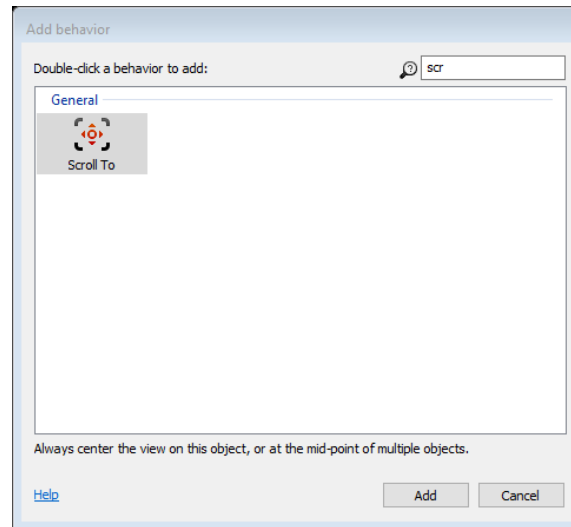


- ▶ Expanda também o plano de fundo.



# Fazendo com que a tela acompanhe o personagem

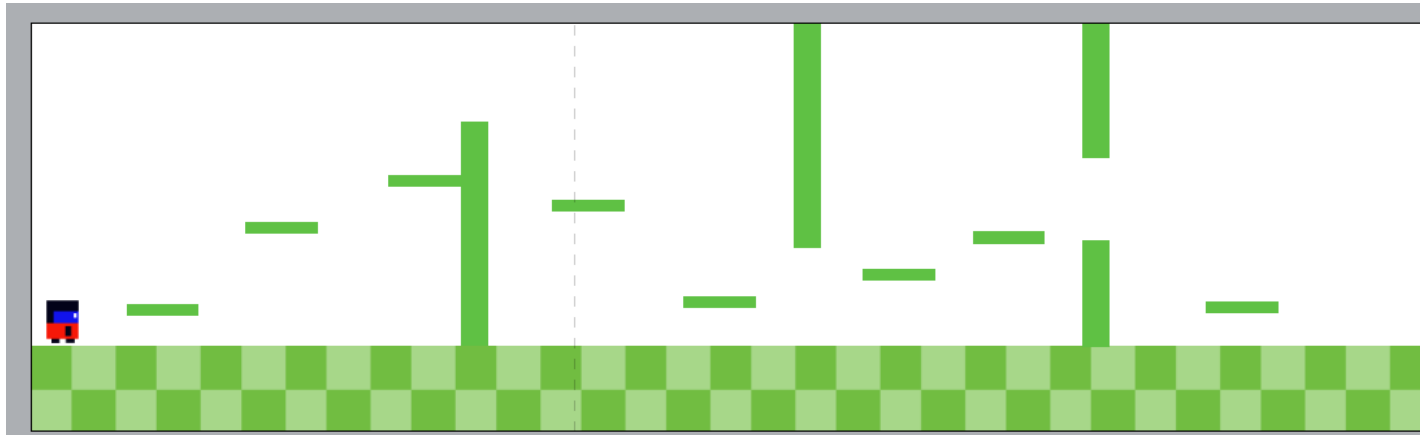
- ▶ Perceba que seu objeto consegue andar pelo cenário, mas a tela não o acompanha.
- ▶ Para que isso aconteça, adicione ao seu personagem o comportamento de **Scroll To**





## Fazendo com que a tela acompanhe o personagem

- ▶ Adicione mais elementos do tipo Solid ao seu jogo, com o objetivo de agregar dificuldade. Rearranje também as plataforma. Veja o exemplo.



# Aumentando o cenário do jogo

- ▶ Agora, vamos expandir o nosso cenário.
- ▶ Defina o Layout do jogo para 1640x480

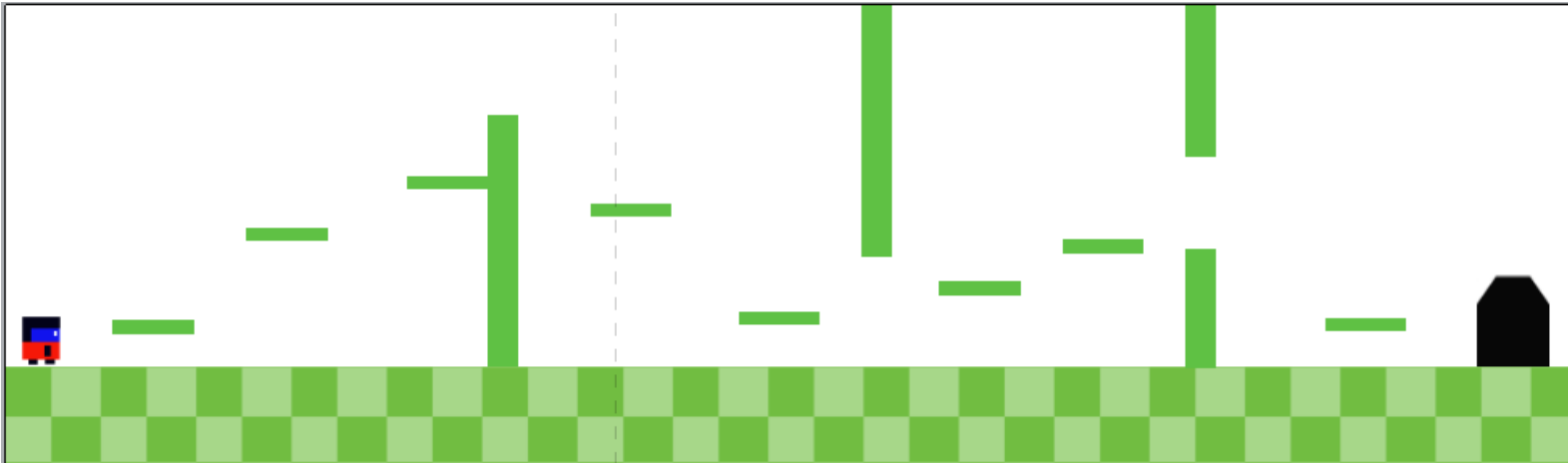


- ▶ Expanda também o plano de fundo.



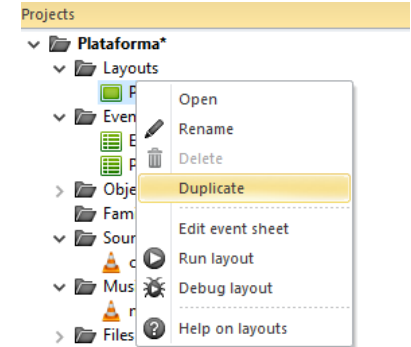
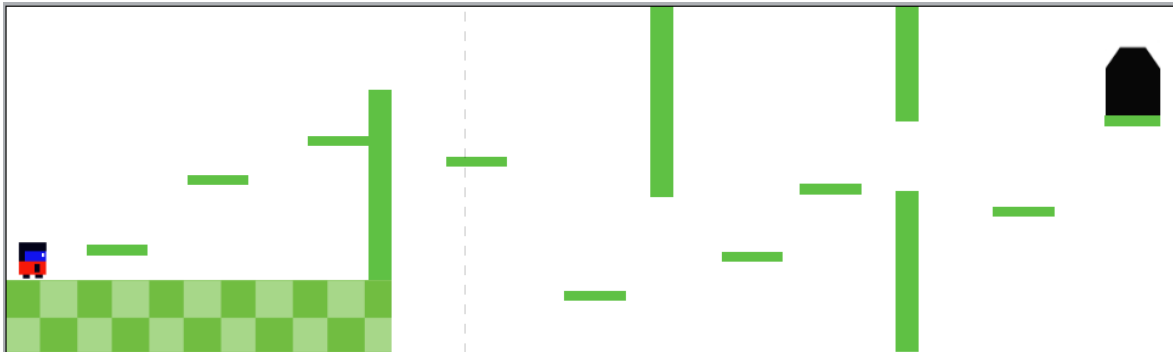
## Encerrando a fase

- ▶ Adicione um elemento para indicar o fim do fase, quando o personagem atingí-lo. Nesse caso, o personagem deve entrar por uma porta.



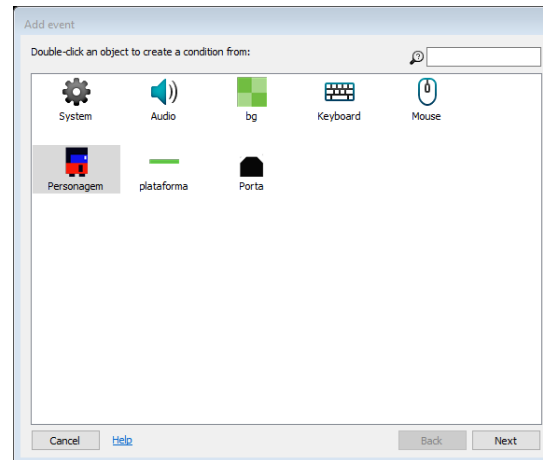
## Criando uma nova fase

- ▶ Para criar uma nova tela, devemos adicionar um novo layout ao nosso projeto.
- ▶ Como nossa fase será parecida com a primeira, vamos duplicar o Layout existente.
- ▶ Para isso, clique com o botão direito sobre o layout, e selecione a opção Duplicate. Renomei os Layouts para Fase 1 e Fase 2.
- ▶ Modifique o layout, aumentando a dificuldade, caracterizando a nova fase.



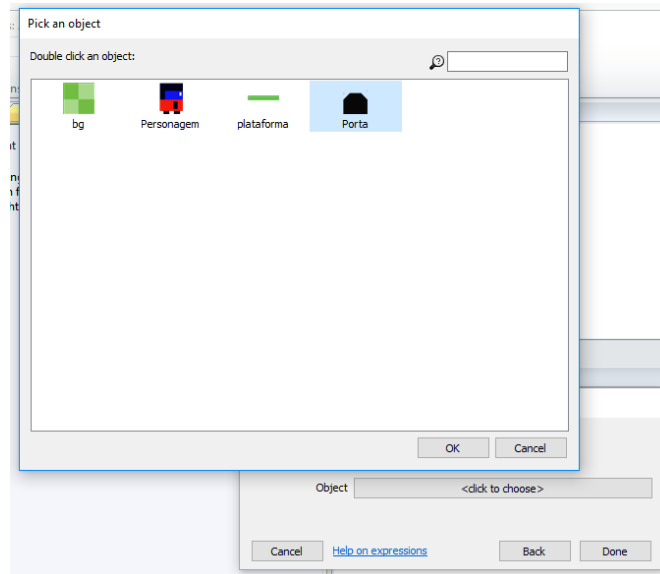
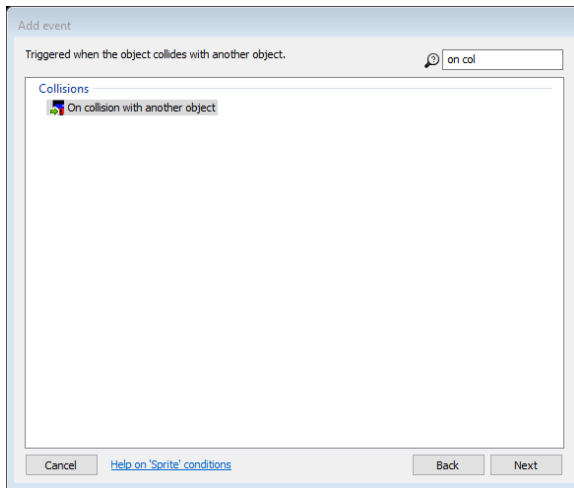
# Adicionando eventos para mudança de fase

- ▶ Volte para o Layout da Fase 1.
- ▶ Vamos adicionar um evento para que, ao atingir a porta, o jogo passe para a fase seguinte.
- ▶ Vá na Planilha de Eventos (renomei-a para Fase 1, de forma a facilitar a organização).
- ▶ Adicione um novo evento, tendo como base o objeto do Personagem.



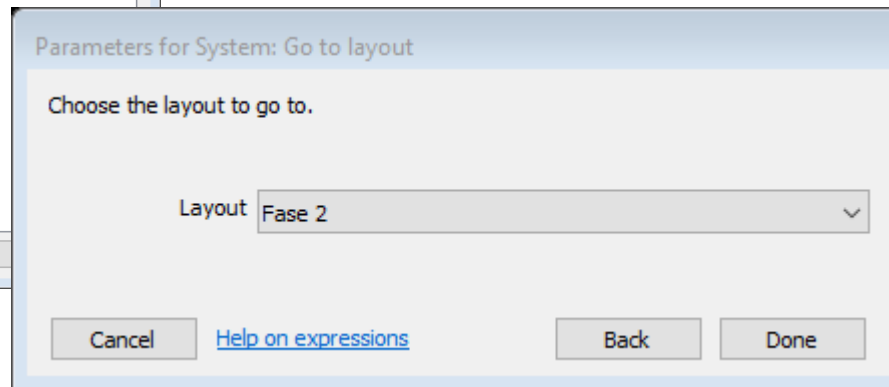
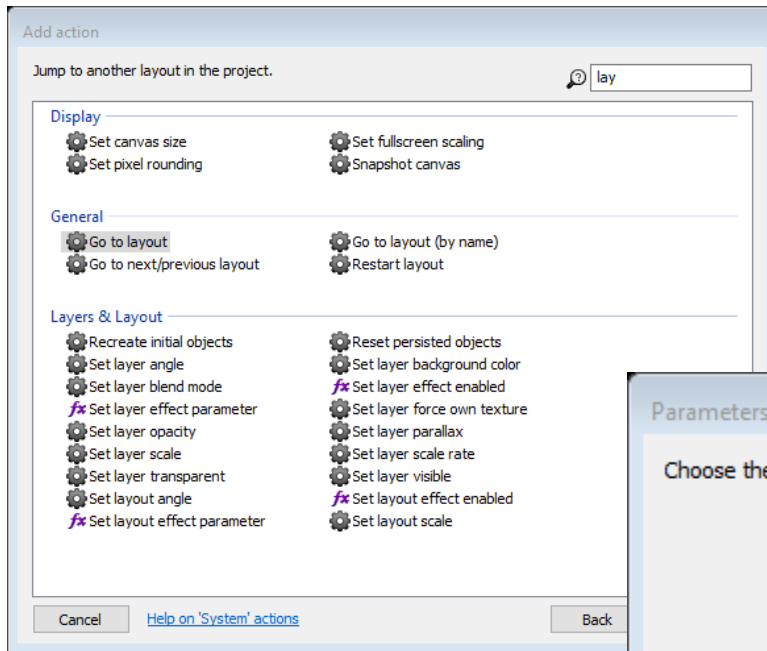
# Adicionando eventos para mudança de fase

- ▶ Selecione o evento “On collision with another object” e selecione o objeto da Porta (você pode utilizar a caixa de busca para facilitar a localização dos eventos).



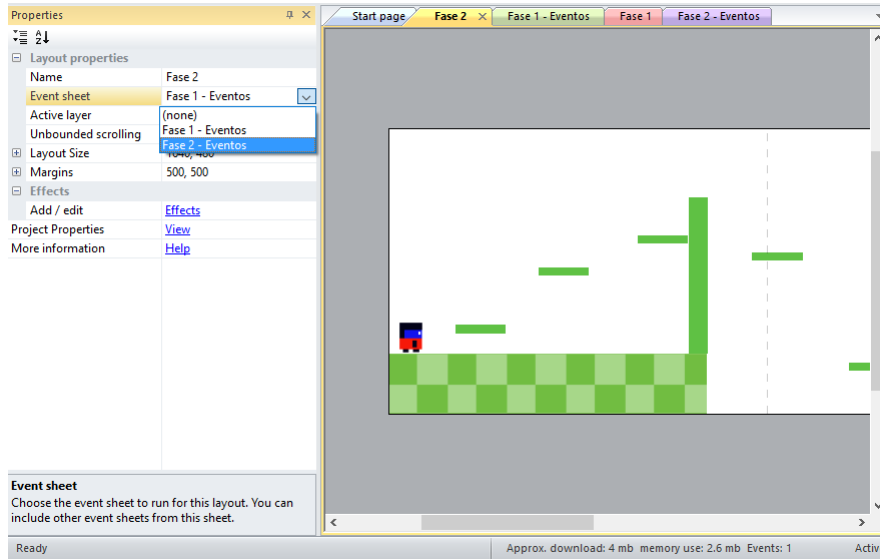
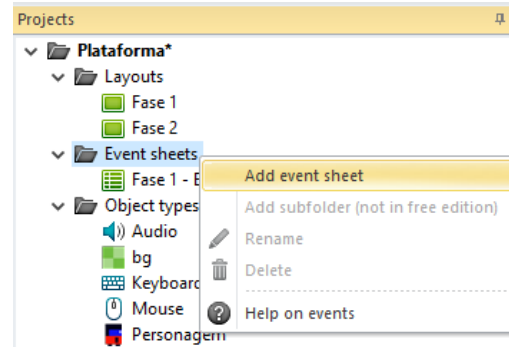
# Adicionando eventos para mudança de fase

- ▶ Na ação, selecione o elemento System
  - ▶ Na opção General, selecione a opção “Go to Layout”. E escolha o Layout Fase 2.



# Eventos para a Fase 2

- ▶ Crie uma nova planilha de eventos para a Fase 2 e nomeei-a como Fase 2- Eventos:
- ▶ Associe-a ao Layout da Fase 2





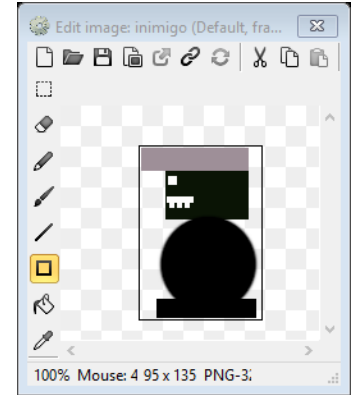
## Eventos para a Fase 2

- ▶ Vamos adicionar um evento na Planilha de Eventos da Fase 2 para que, quando o Personagem sair do Layout, a fase seja reiniciada:
  - ▶ Fase 2 - Eventos → Add event → Personagem → Is outside Layout
  - ▶ Add action → System → Go to Layout → Fase 2



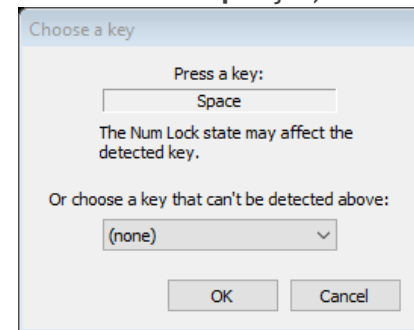
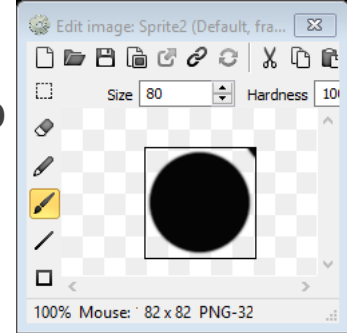
## Adicionando inimigos

- ▶ Crie uma nova Sprite para representar seu inimigo no jogo.
- ▶ Adicione uma ação para que, quando o Personagem colidir com o inimigo, o Personagem seja destruído
  - ▶ Vá na Planilha de Eventos da Fase 2, e adicione um Evento
    - ▶ Add Event → Personagem → On collision with another object → inimigo
    - ▶ Add action → Personagem → Destroy.
  - ▶ Adicione outra ação para que o jogo volte para o início da fase após a destruição do personagem.
    - ▶ Add action → System → Go to Layout → Fase 2



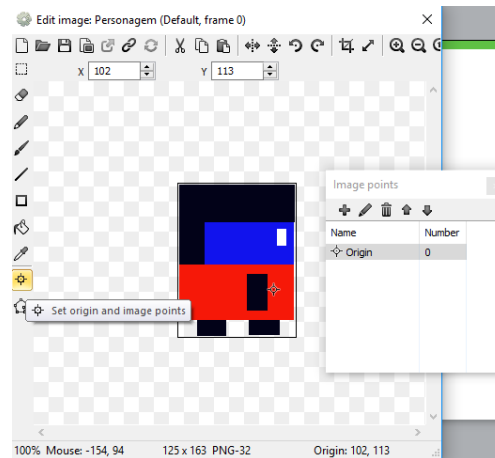
# Adicionando arma ao personagem

- ▶ O personagem terá uma “arma” para poder destruir o inimigo.
- ▶ Adicione uma nova Sprite para ser a “bala” a ser disparada pelo personagem.
- ▶ À bala, adicione o comportamento (behaviour) de Bullet.
- ▶ Adicione um objeto do tipo Keyboard ao seu jogo, caso ele não exista
- ▶ Na planilha de eventos, adicione o seguinte evento e ações:
  - ▶ Add Event → Keyboard → On key pressed → (e pressione a tecla espaço)
  - ▶ Add Action → System → Create object → bala → on Layer 1
  - ▶ Add Action → Personagem → Spawn another Object → bala



## Modificando o ponto de origem

- ▶ Perceba que agora, ao pressionar a tecla espaço, o personagem irá disparar balas.
- ▶ Se você achar estranha a posição de onde as balas saem e desejar posicioná-las para partir do centro do objeto, você deve modificar o ponto original do seu personagem.
- ▶ Para isso, abra a edição do seu personagem, selecione o componente Set origin and image points, e posicione-o no centro do seu personagem



## Destruindo o inimigo com a arma

- ▶ Adicione um evento para que, quando a bala colidir com o inimigo, o inimigo seja destruído:
  - ▶ Add event → bala → on collision with another object → inimigo
    - ▶ Add action → inimigo → destroy
    - ▶ Add action → bala → destroy



# Exercícios

1. Adicione uma ação ao jogo que, no caso da Fase 2 quando o usuário sair do Layout, a fase ser reiniciada
2. Crie um novo Layout para ser o menu do seu jogo, com um Sprite que, quando for clicado, abre o Layout da Fase 1
  - ▶ DICAS:
    - ▶ Para fazer com que seu menu seja o primeiro a ser aberto, na aba Projects, clique na pasta com o nome do seu Projeto. Em seguida, na janela de Propriedades, altere a propriedade First Layout para o layout correspondente ao seu menu.
    - ▶ Para que um determinado objeto seja clicado, adicione um objeto do tipo Mouse e utilize o evento On object clicked.
3. Crie mais duas fases para o seu jogo (contendo mais inimigos) e encadeie as fases.
4. Crie um Layout para indicar o fim do jogo, quando o usuário passar por todas as fases.
5. Inclua áudios no seu jogo para serem disparados quando um inimigo for eliminado; quando o personagem pisar nas plataformas; quando entrar pela porta, indicando o fim da fase.

## Referências

- ▶ <https://www.scirra.com/construct2>
- ▶ <https://www.scirra.com/tutorials/>

