



Aprendendo a Desenvolver

Jogos Usando a engine GDevelop 5

Lição 1: Iniciando

- Será apresentado o software GDevelop 5.
- Começaremos a entender os conceitos básicos do GDevelop 5.
- Será apresentado alguns conceitos chaves para desenvolver jogos com GDevelop 5.
- Usaremos o GDevelop 5 para criar os nossos próprios jogos.





O que é Computational Thinking?

Alguma ideia?

O que poderia ser?



Computational Thinking é...

...Um conjunto de habilidades para resolução de problemas que permitem:

1. Rapidamente decompor os problemas em problemas menores que podem ser resolvidos com mais facilidade.
2. Remover os detalhes que não são necessários, para os problemas serem mais facilmente definidos e entendidos, ou seja, devemos abstrair.
3. Pensar (think) em **algoritmos** para criar soluções passo a passo para os problemas (**pensamento algorítmico - algorithmic thinking**)



O que é um algoritmo?

Apesar de não utilizarmos uma linguagem de programação específica para desenvolver jogos com GDevelop 5, será necessário pelo menos o conhecimento básico sobre os algoritmos.



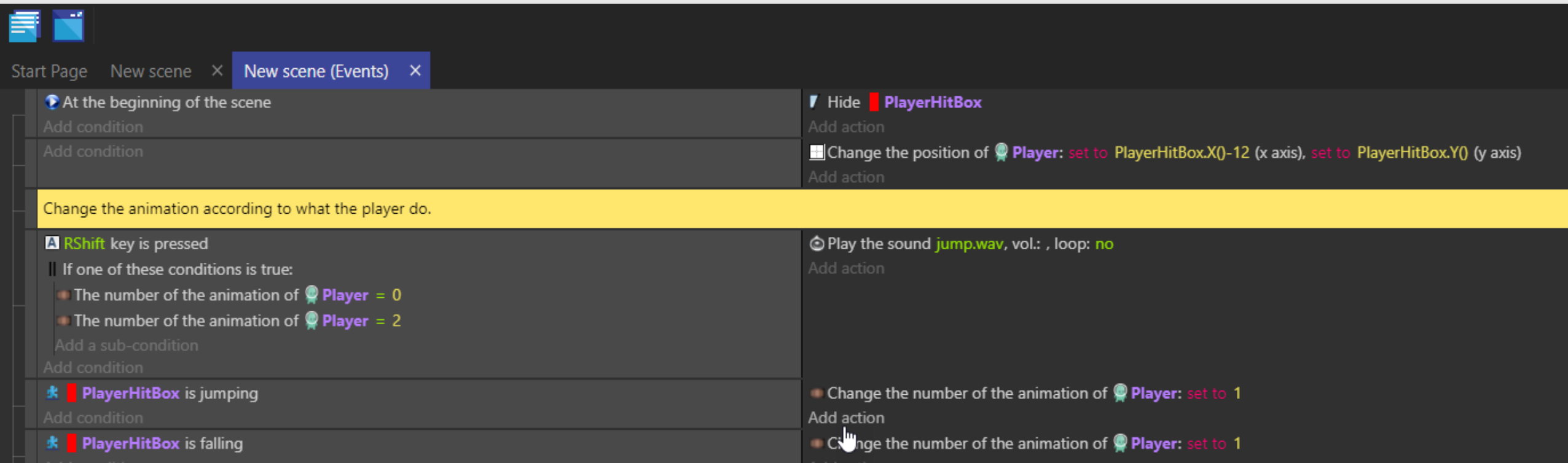
O que é programação?

Alguma ideia?

Programação é...

...o processo de escrever algoritmos que o computador possa entender e realizar uma determinada tarefa.

Nós usaremos blocos de eventos do GDevelop para desenvolver os jogos.





O que é o GDdevelop 5?

GDdevelop é uma engine open-source, cross-platform para criar jogos, desenvolvida para ser utilizada por qualquer um – sem precisar de possuir habilidades de programação.

O que o GDevelop 5 pode fazer?

- **GDevelop 5** é uma engine utilizada para desenvolver jogos 2D, sem utilizar linhas de código, entretanto, se preferir, é possível programar utilizando JavaScript.

Veja abaixo alguns jogos desenvolvidos com GDevelop 5.



<https://gdevelop-app.com/games-showcase>



Conceitos chaves no GDevelop 5

Vamos conhecer alguns deles...

Termos chaves do GDevelop 5

Um Sprite é...

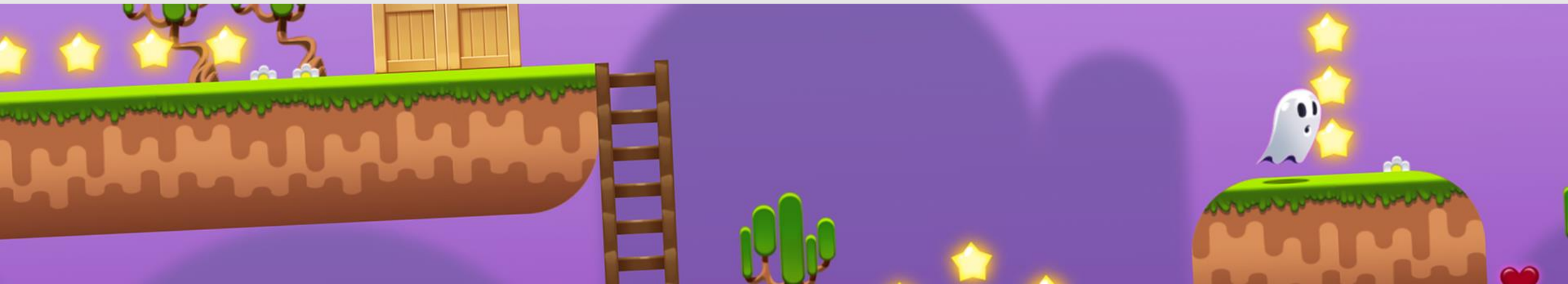
...um objeto que permite exibir uma imagem bidimensional que pode ser animada ou estática, podendo ser utilizada para representar botões, personagens, inimigos, itens e perigos ou qualquer outra coisa que possa ser exibida no jogo.

Um Object(Objeto) é...

...qualquer elemento que pode ser exibido na janela de um jogo.

Uma Scene(Cena) é...

...uma janela que exibe os objetos que foram criados para desenvolver o seu jogo. Essa janela, pode ser por exemplo, menus e as fases do seu jogo.



Termos chaves do GDevelop 5

Um Event(Evento) ...

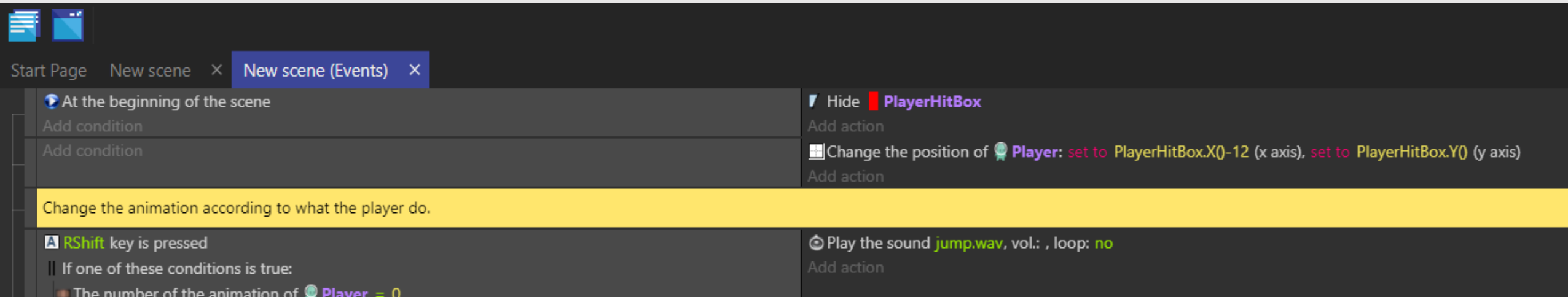
...são regras que você cria para o seu jogo e que são disparados enquanto seu jogo estiver executando. Os eventos são compostos de condições e ações.

Uma Condition(Condição) ...

...é utilizada para testar os objetos na cena, como por exemplo, verificar se houve colisões entre os objetos.

Uma Action(Ação) ...

...são comandos para manipular os objetos, como por exemplo, mudar a posição de um objeto, cor e entre outros. Ações geralmente são executadas junto com condições, caso seja utilizada sem uma condição, a ação irá referenciar todos os objetos da cena.



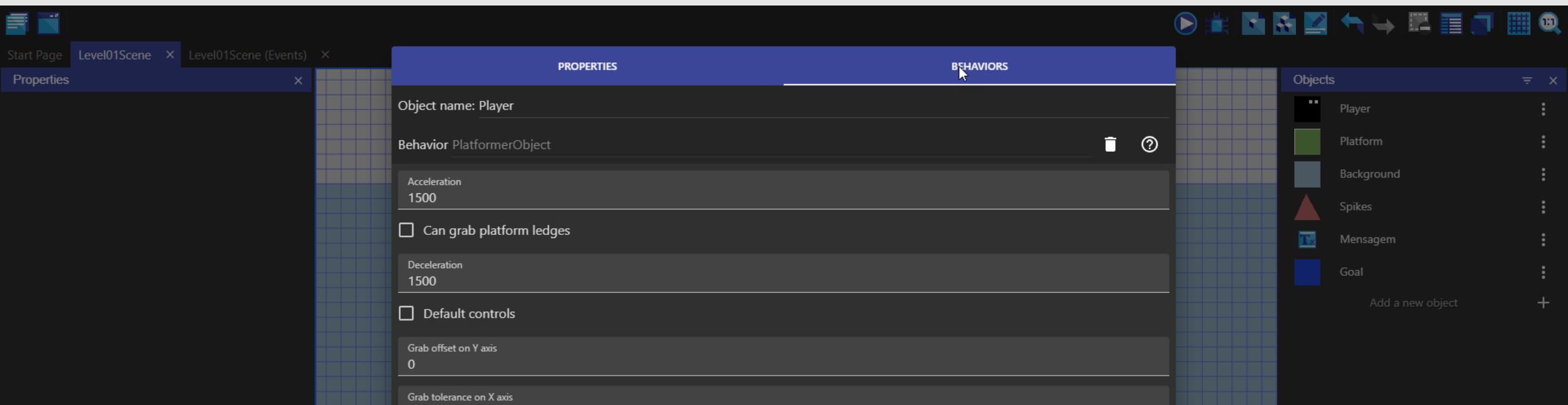
Termos chaves do GDevelop 5

Um Behavior (Comportamento) ...

...são comportamentos para os objetos, adicionando capacidades extras. Todos os objetos podem possuir um ou mais behaviors. Um exemplo de behavior, chamado, “Platformer character ou PlatformerObject” é utilizado para permitir controlar um personagem no jogo que pode pular e andar sobre as plataformas.

Você pode saber mais sobre behaviors na página oficial do GDevelop:

<http://wiki.compilgames.net/doku.php/gdevelop5/behaviors>



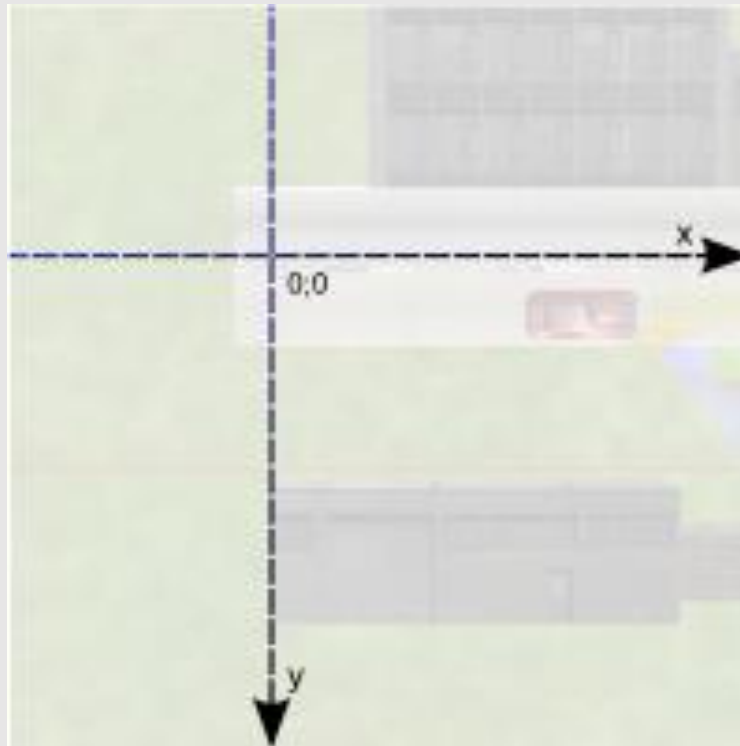


Conceitos básicos do GDevelop 5

Vamos conhecer alguns deles...

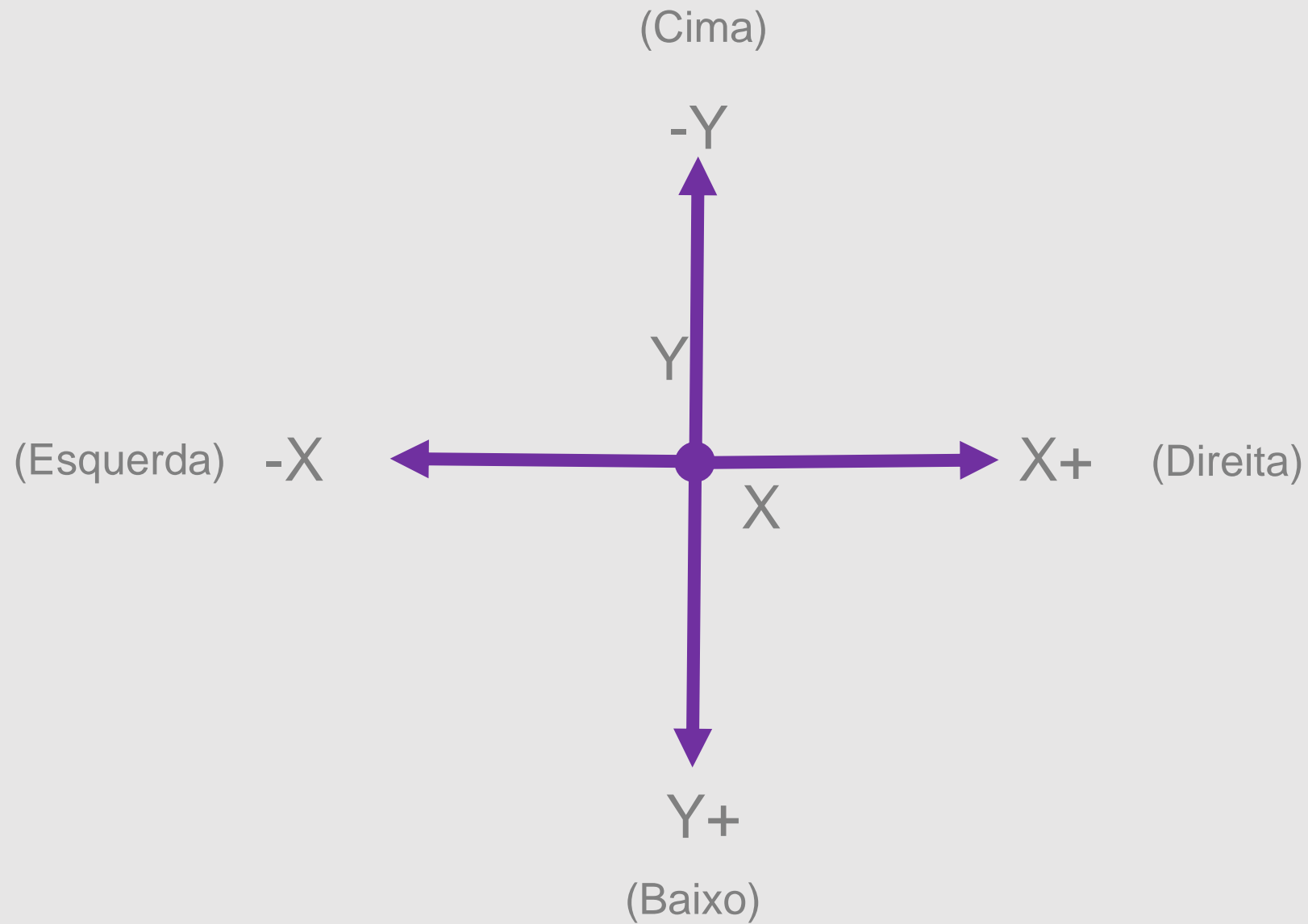
Coordenadas no GDevelop

- Os objetos que são exibidos na cena possuem as coordenadas X e Y. Essas coordenadas correspondem a posição horizontal (eixo X) e vertical (eixo Y) no plano cartesiano.
- A coordenada X diminui a medida que você vai para a esquerda e aumenta a medida que você vai para a direita. A coordenada Y aumenta a medida que você vai para baixo e diminui a medida que você vai para cima.





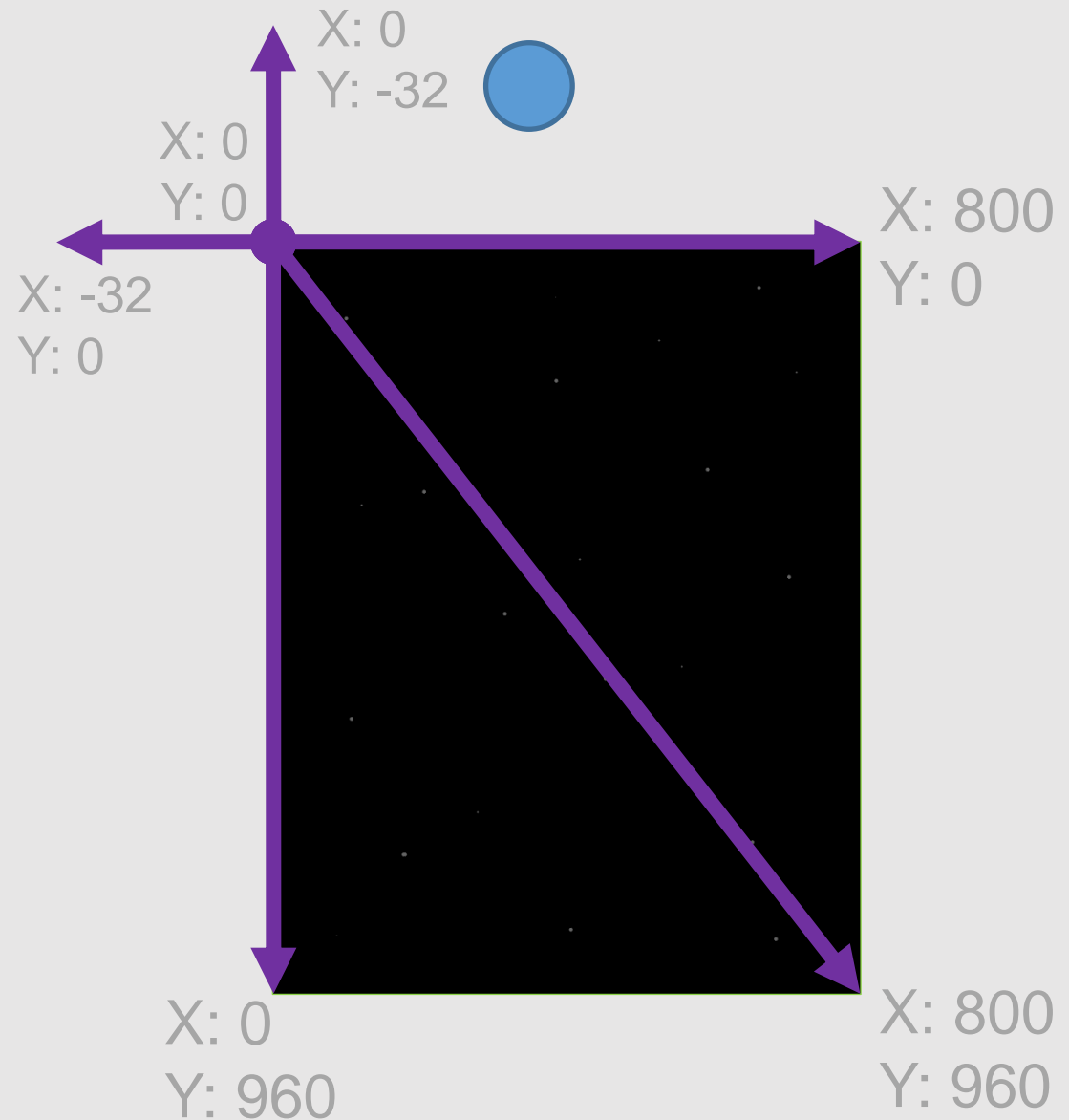
Coordenadas no GDevelop



Tamanho da Cena

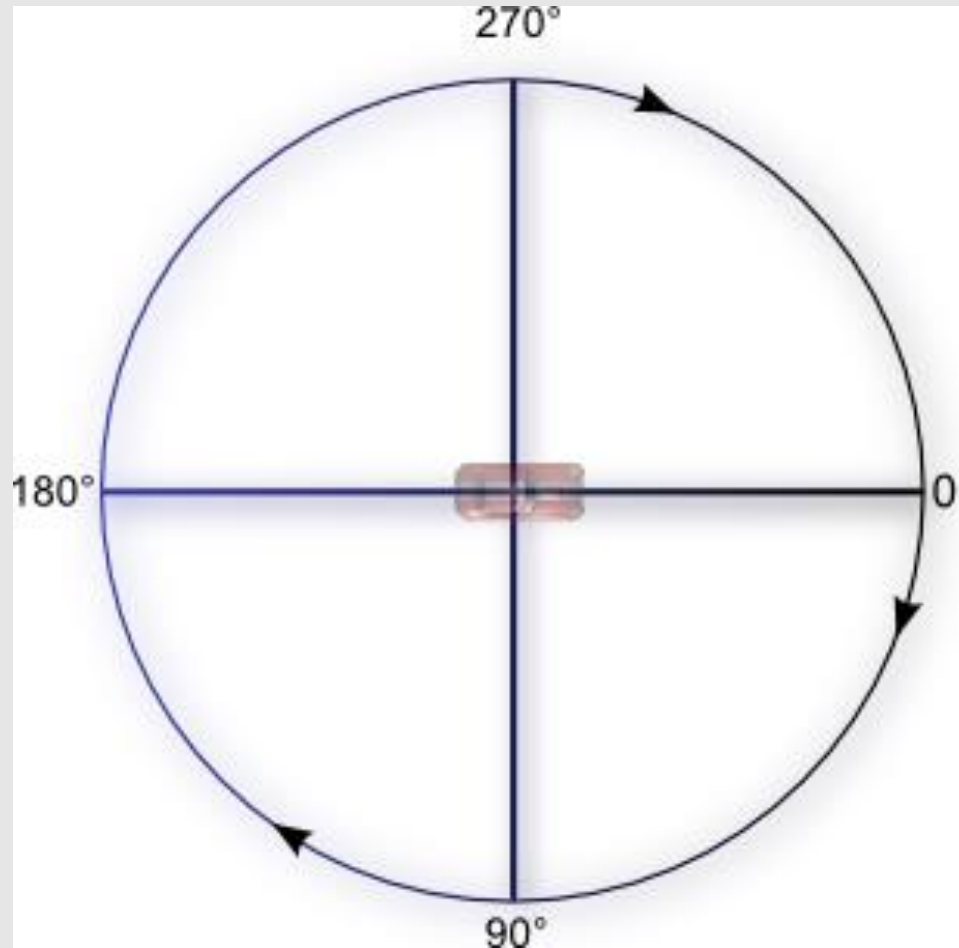
Largura: 800px

Altura: 960px

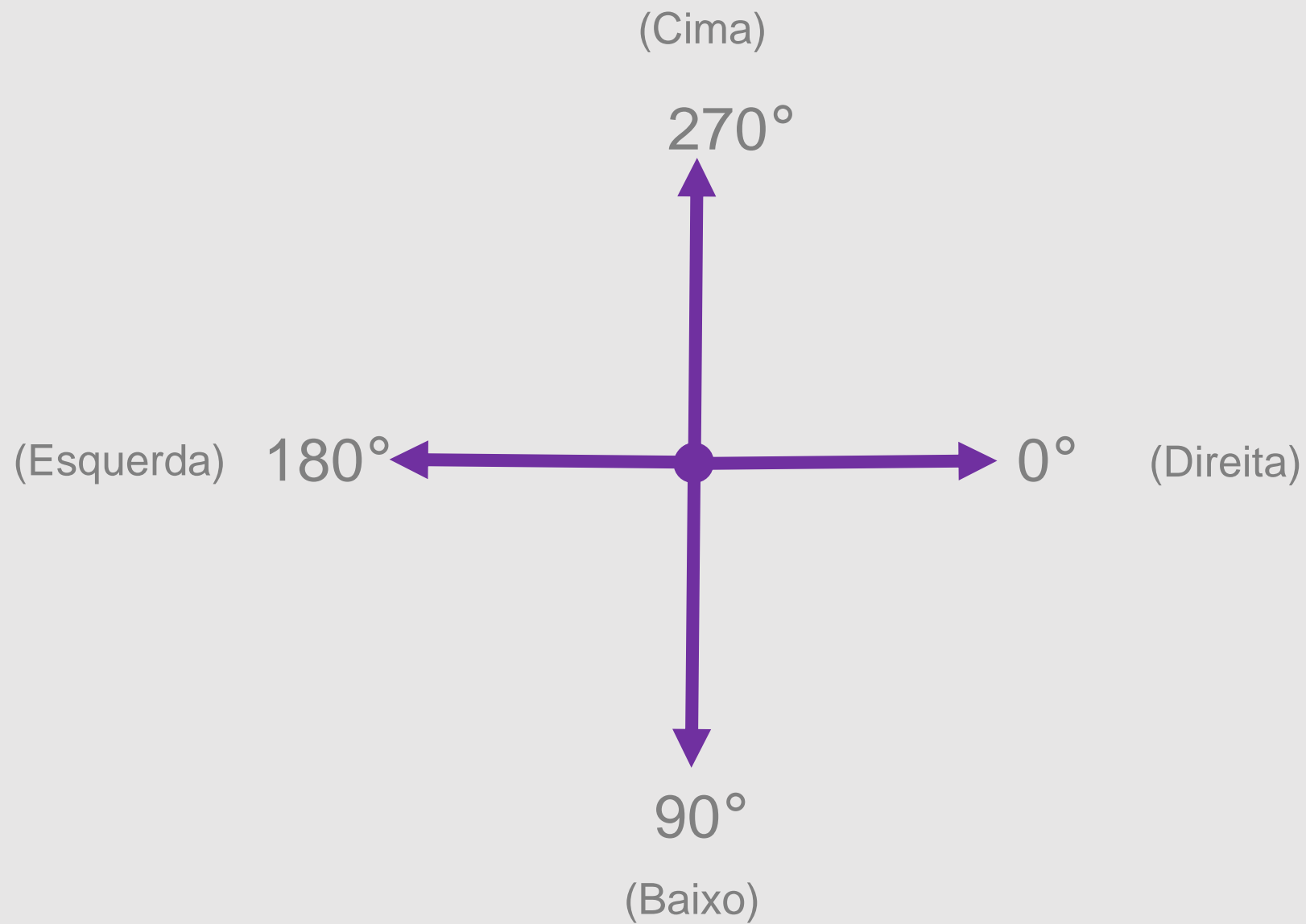


Ângulos no GDevelop

- Para mover ou girar objetos, você precisará especificar o ângulo desejado em graus. A ilustração abaixo demonstra como o GDevelop entende o ângulo de rotação:



Ângulos no GDevelop



Variáveis

- O GDevelop possui três tipos de escopo de variáveis: **Global**, **Scene(Cena)** e **Object(Objeto)**
- Diferentes tipos de valores podem ser armazenados: **Number**, **Text** e **Structure**(Uma lista que contém outras variáveis).

Global:

São variáveis que são armazenadas na memória durante toda a execução do jogo. Elas podem ser utilizadas quando queremos compartilhar algumas informações entre as cenas, como por exemplo, número de vidas ou pontuação.

Scene

São variáveis que são armazenadas na memória somente enquanto a cena estiver em execução. Se a cena for trocada, as informações da variável serão perdidas.

Object

São variáveis que são armazenadas na memória de forma individual para cada instância de um objeto que está na cena, como por exemplo, podemos armazenar informações sobre o nível de energia de um inimigo ou personagem. Se o objeto for excluído da cena ou a cena for trocada, as variáveis e suas informações que estão armazenadas também serão eliminadas.

Recursos comuns a todos os objetos

- Os objetos no GDevelop compartilham alguns recursos em comum:

Position:

Os objetos são posicionados na cena com base nas coordenadas X e Y. Essas coordenadas podem ser alteradas nas propriedades do objeto pela IDE quando a instância do objeto está na cena ou através dos eventos.

Angle:

Da mesma forma que a Position podemos alterar o ângulo de um objeto na cena. O valor deve ser informado em graus de 0 a 360.

Z order:

É utilizado para especificar a ordem de como o objeto é visualizado na cena. Com isso, podemos dizer se o objeto pode ficar na frente ou atrás de outro objeto na cena.

Layer:

Layer(camada) pode ser utilizada para definir como os objetos estão localizados na cena, com isso podemos criar uma camada para exibir menus, mapas ou backgrounds. Desta forma, facilitando a organização dos objetos na cena.



Funções úteis dos objetos

Angle():

Permite retornar o ângulo em graus que um objeto está na cena.

X() e Y():

Permite retornar a posição do objeto na cena, respectivamente, as coordenadas X e Y.

Width() e Height():

Permite retornar o tamanho do objeto, respectivamente, a largura e altura.

ZOrder():

Permite retornar a ordem que um objeto está na cena.

Count(<nome do objeto>):

Permite retornar a quantidade de objetos que estão na cena: O nome do objeto deve ser informado entre parênteses.