Platformer Tutorial, Part 8: Add Checkpoints to the Game

This part of the tutorial explains how to add checkpoints to the game. Then, when the player dies, they'll be sent to the most recent checkpoint.

Series

You are reading **Part 8** of the Platformer Tutorial.

1. Platformer Tutorial, Part 1
2. Platformer Tutorial, Part 2
3. Platformer Tutorial, Part 3
4. Platformer Tutorial, Part 4
5. Platformer Tutorial, Part 5
6. Platformer Tutorial, Part 6
7. Platformer Tutorial, Part 7
8. Platformer Tutorial, Part 8

Step 1: Create an object for the checkpoint

The first step is to create a “Checkpoint” object. By this point in the tutorial, this should be a familiar process. If you need a refresher though, refer to Objects.

To create an object for the checkpoint:

1. Create a Sprite object named “Checkpoint”.
2. Use the “bush.png” asset as the image for the checkpoint.
3. Drag one or more instances of the object into the scene.

Step 2: Save the player's coordinates when they reach a checkpoint

When the player reaches a checkpoint, you can use variables to save the coordinates of that checkpoint. Then, when the player dies, you can send them back to those coordinates.

You can access the X and Y coordinates of an object with the following syntax:

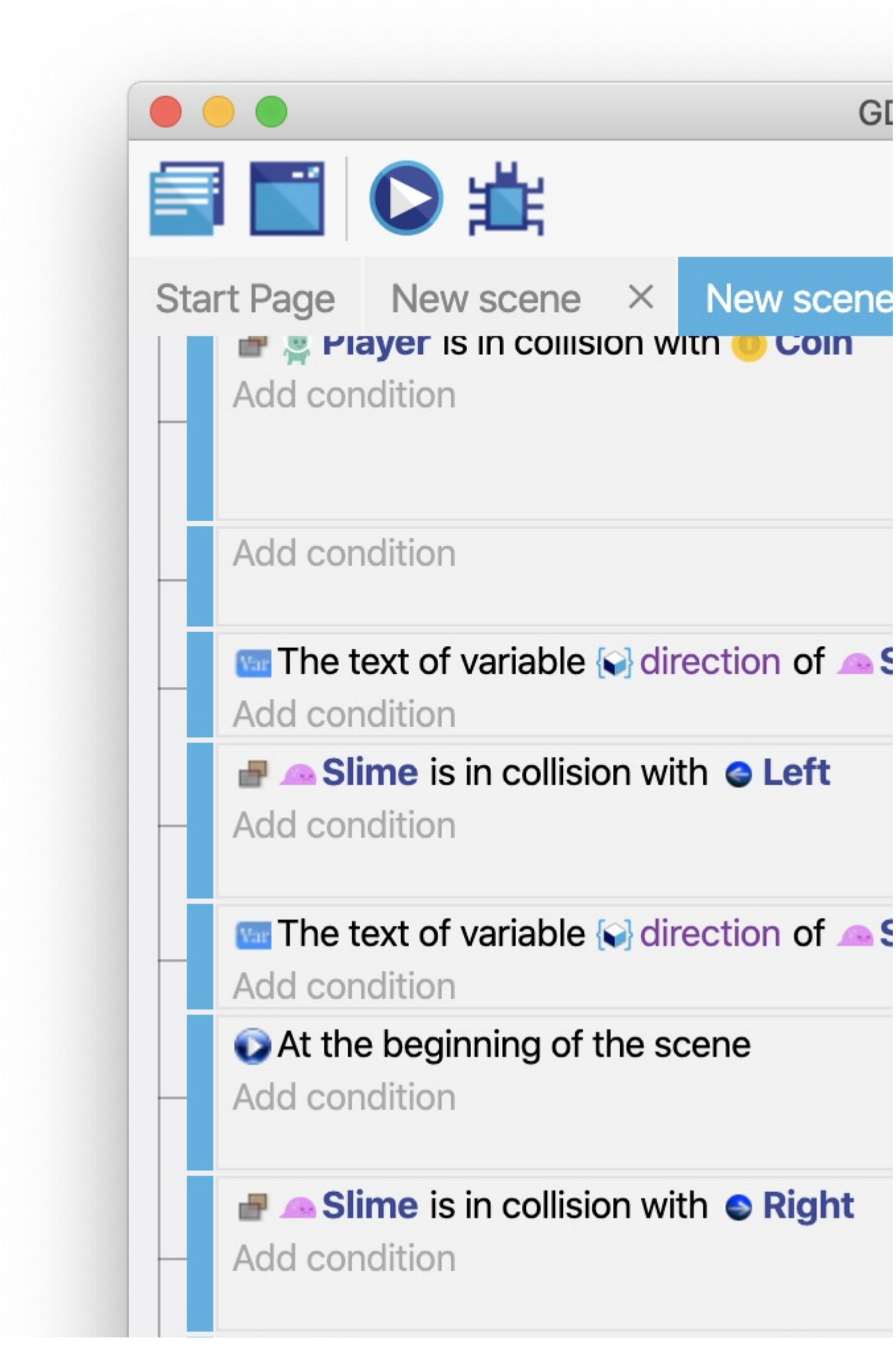
```
ObjectName.X()
ObjectName.Y()
```

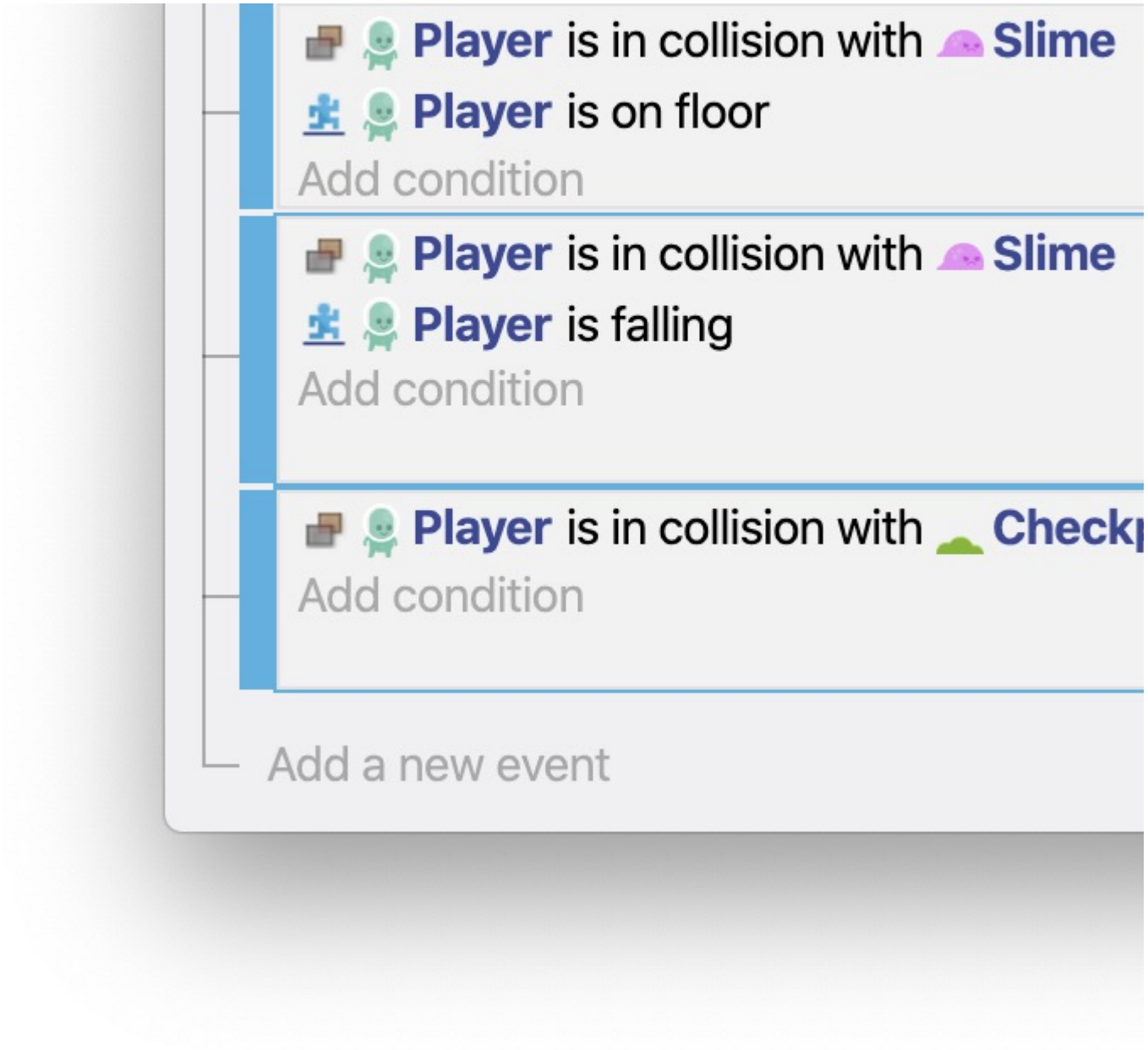
These statements are both examples of expressions.

To save the coordinates of the “Player” object when they reach a checkpoint:

1. Create a new event.
2. Create a **Collision** condition that checks if the “Player” object has collided with the “Checkpoint” object.

3. Create a **Value of a scene variable** action that creates a “CheckpointX” variable with a value of `Checkpoint.X()`.
4. Create a **Value of a scene variable** action that creates a “CheckpointY” variable with a value of `Checkpoint.Y()`.


Start Page
New scene
New scene
Player is in collision with Coin
Add condition
Add condition
The text of variable direction of
Add condition
Slime is in collision with Left
Add condition
The text of variable direction of
Add condition
At the beginning of the scene
Add condition
Slime is in collision with Right
Add condition

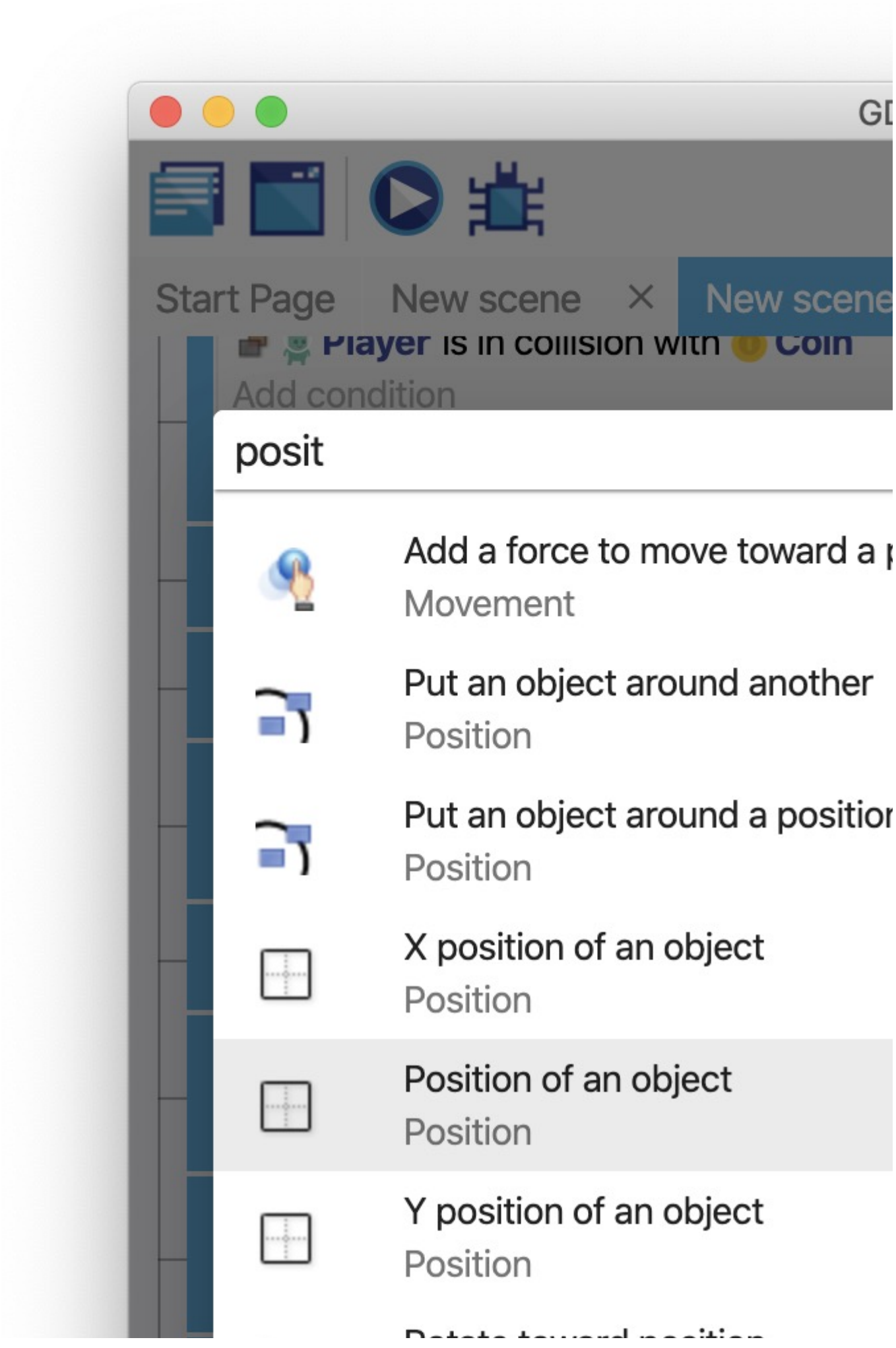


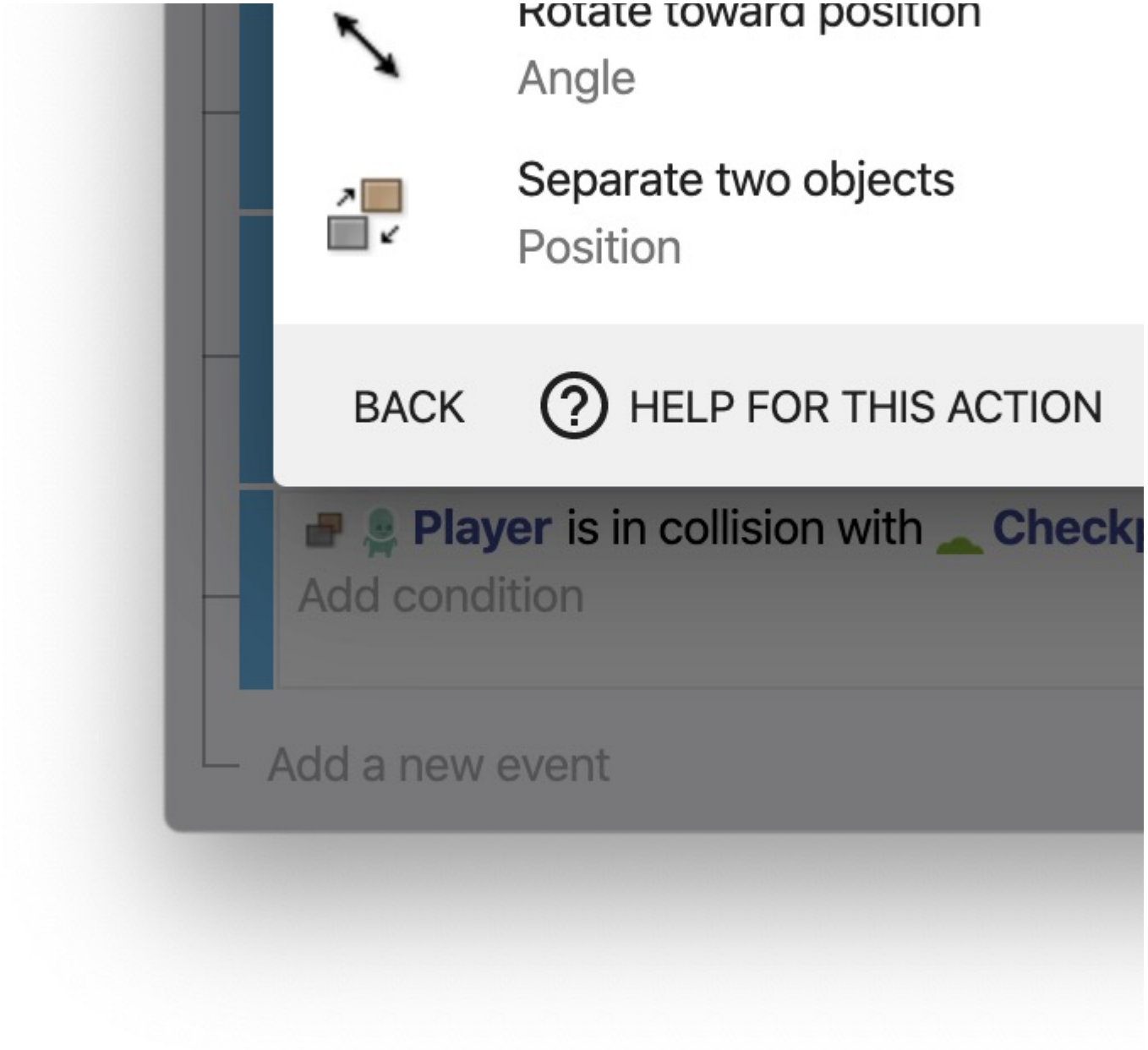
Step 3: Send the player to the previous checkpoint

When the “Player” object collides with the “Slime” object, the **Delete an object** action deletes the “Player” object from the scene. But since the game has checkpoints now, it doesn't make sense to delete the “Player” object.

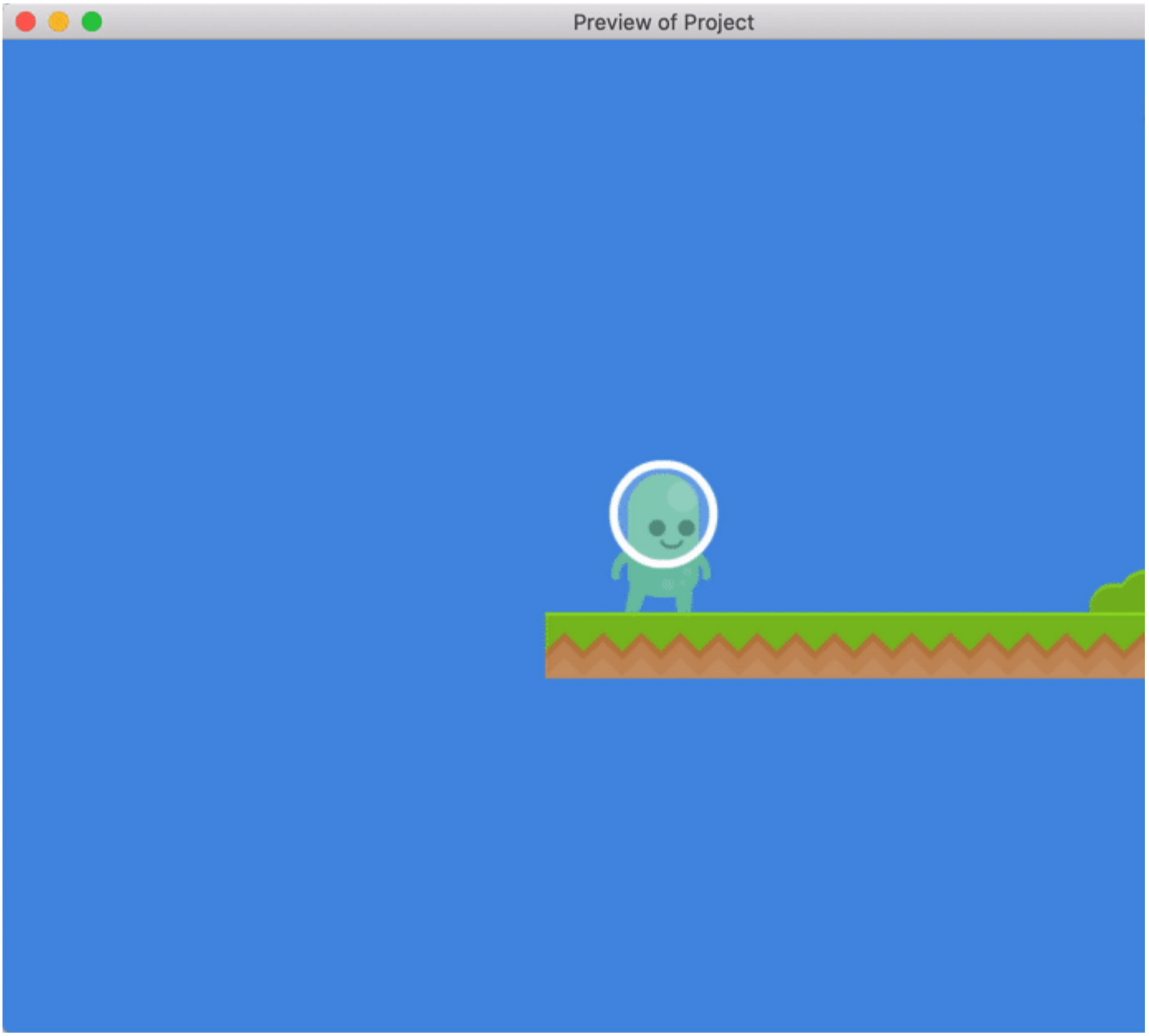
To send the “Player” object back to the previous checkpoint:

1. Remove the **Delete an object** action.
2. Create a **Position of an object** action for the “Player” object.
3. For each of the **Modification's sign** fields, select **= (set to)**.
4. Set the **X position** coordinate to `Variable(CheckpointX)`.
5. Set the **Y position** coordinate to `Variable(CheckpointY)`.


Start Page
New scene
New scene
Player is in collision with Coin
Add condition
posit
Add a force to move toward a
Movement
Put an object around another
Position
Put an object around a positior
Position
X position of an object
Position
Position of an object
Position
Y position of an object
Position



If you preview the game, dying after reaching a checkpoint sends you back to the checkpoint.



Step 4: Set the player's default coordinates

The `CheckpointX` and `CheckpointY` variables don't exist until the player reaches a checkpoint, so if a player dies before reaching a checkpoint, GDevelop sends the player back to the default coordinates of “0,0”.

This may cause a problem if:

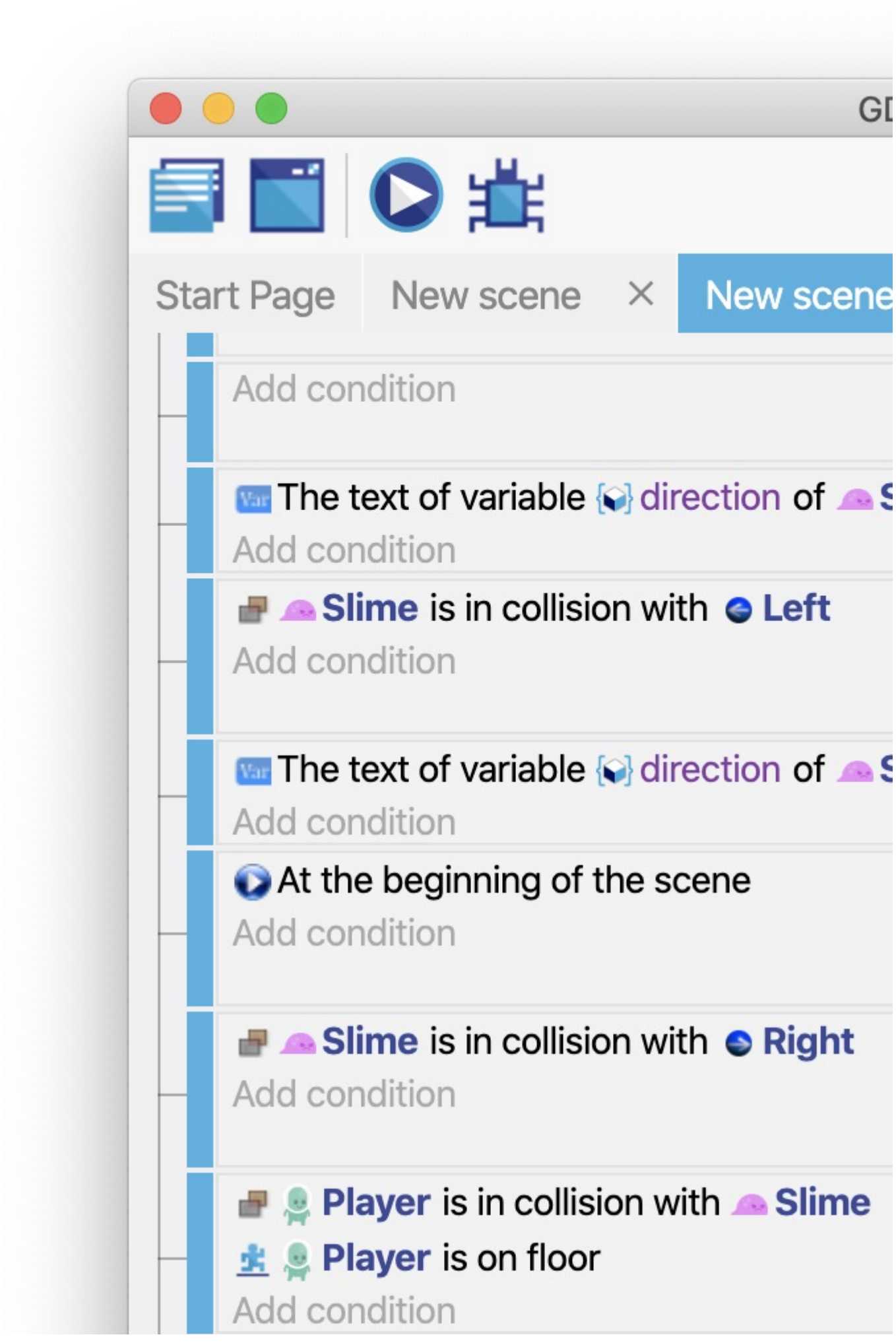
- Something else exists at the default coordinates (such as an enemy).
- There isn't a platform beneath the default coordinates.

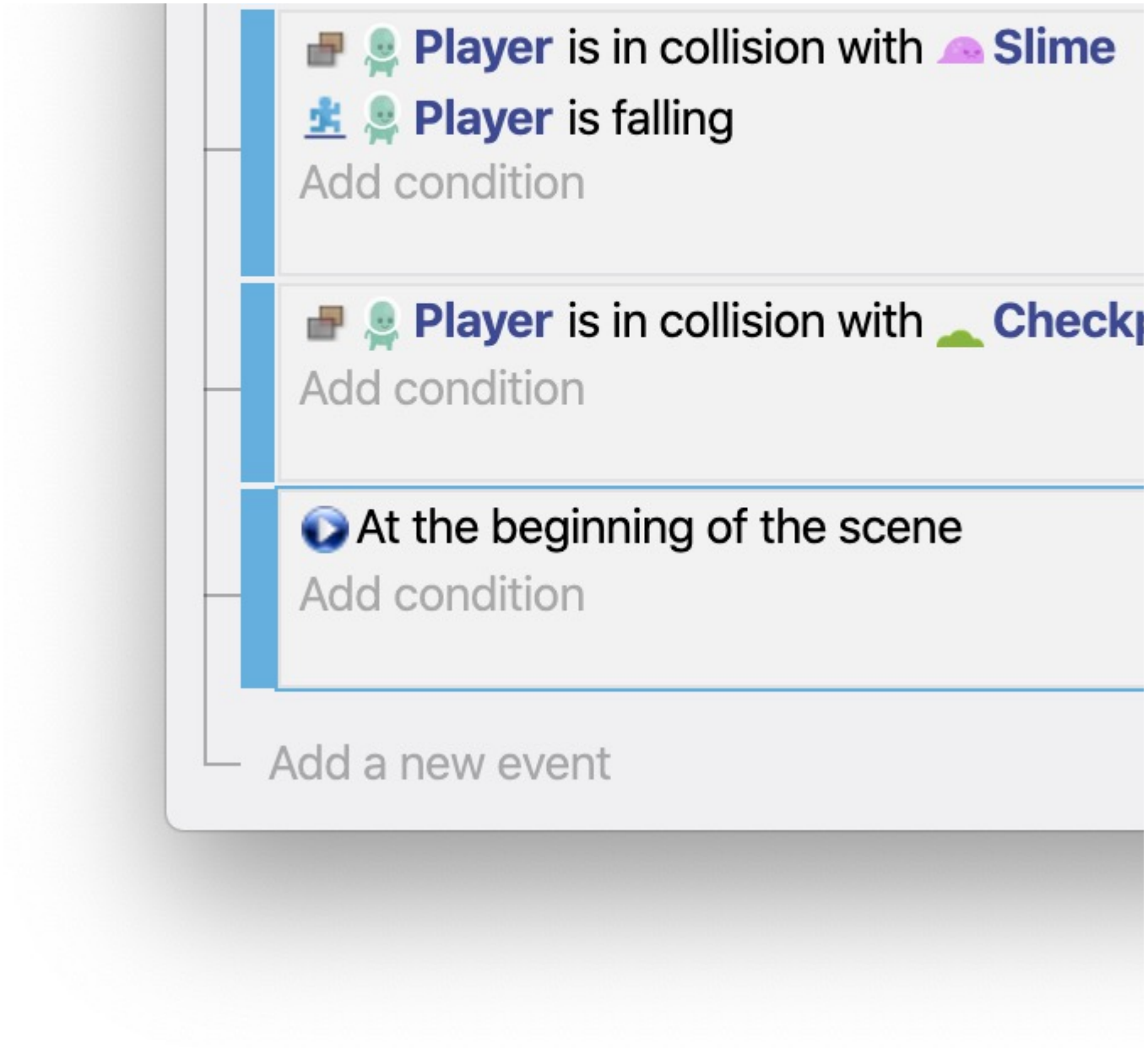
To fix this, set the value of the `CheckpointX` and `CheckpointY` variables to the initial coordinates of the “Player” object. Then the default coordinates are defined by where you place the “Player” object in the Scene editor.

To set the default coordinates:

1. Create a new event.
2. Add the **At the beginning of the scene** condition to the event.
3. Create a **Value of a scene variable** action that creates a “CheckpointX”

variable with a value of `Player.X()`.
4. Create a **Value of a scene variable** action that creates a “CheckpointY” variable with a value of `Player.Y()`.}}


Start Page
New scene
New scene
Add condition
The text of variable direction of
Add condition
Slime is in collision with Left
Add condition
The text of variable direction of
Add condition
At the beginning of the scene
Add condition
Slime is in collision with Right
Add condition
Player is in collision with Slime
Player is on floor
Add condition



Next steps

You've reached the end of this tutorial. Congratulations!

From here, either dive into another tutorial, such as the Geometry Monster tutorial, or continue playing around with your platforming game. Based on what you've learned, how unique can you make it?