

## Fundamentos de Java

### 2-10: Variáveis

### Projeto:

Este projeto será desenvolvido por você ao longo do curso. Após cada lição, haverá mais a ser acrescentado, até que seja produzida uma animação completa que você poderá exportar para o Alice Player.

#### Objetivos da Lição:

- Usar números aleatórios para randomizar o movimento
- Entender variáveis
- Entender como as variáveis são usadas na programação
- Exibir código Alice como Código Java ao lado

#### Instruções:

1. Abra o Alice 3 no computador.
2. Use a guia My Projects ou File System, procure e abra o arquivo Fish\_9.a3p.
3. Usando o comando Save As do menu do arquivo, renomeie-o para Fish\_10.a3p.
4. Se você ainda não estiver no editor de código, use o botão Edit Code para ir para o editor de código.

Para esta lição, você examinará variáveis e o comportamento aleatório em seu código.

Atualmente, você tem um bloco de código no myFirstMethod que faz todos os peixes balançarem a cabeça ao mesmo tempo. Mas eles só fazem isso uma vez, e seria bom se os peixes balançassem as cabeças um número de vezes quando o código fosse executado.

Suas animações ficarão muito mais interessantes se exibirem um comportamento aleatório quando adequado.

5. Arraste uma instrução count bem acima dessa instrução do together e escolha 3 como o valor de espaço reservado.
6. Arraste a instrução do together para o loop de contagem.
7. Isso faria com que os peixes balançassem a cabeça três vezes toda vez que o código fosse executado. Para tornar o valor aleatório, clique na seta ao lado do valor de espaço reservado, em seguida, escolha a opção aleatória que permite selecionar o valor inferior e superior inclusivos a serem usados como seu intervalo aleatório. Escolha 1 e 3 como os valores de argumento.

Esse código permitirá que os peixes balancem a cabeça, no mínimo, uma vez e, no máximo, três vezes.

8. Teste o código executando-o várias vezes. Conte o número de vezes que os peixes balançam a cabeça.

9. O próximo passo é tornar seus procedimentos mais úteis adicionando variáveis ao seu código. Você já usou esse conceito por meio dos argumentos que vem empregando em seus procedimentos.

Clique na guia de nado. Se não estiver disponível, clique no botão de classes e, em seguida, em FISH e clique em nadar.

10. A classe de nado atualmente é muito restrita porque só permite que um peixe nade uma distância definida (1,2).

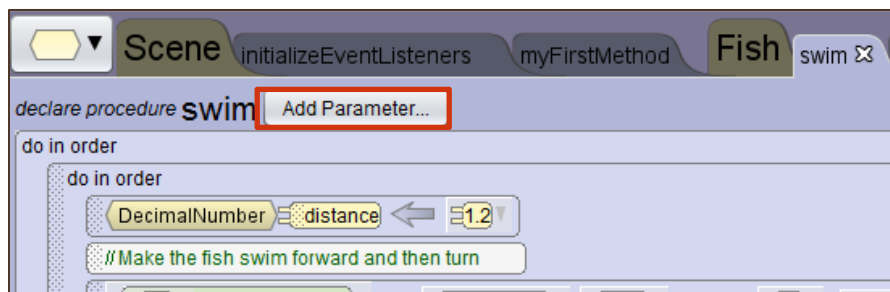
Para alterar isso, você pode adicionar uma variável ao topo do bloco de código.

11. Arraste a instrução da variável da parte inferior da tela e coloque-a sobre o código. Lembre-se de que a linha verde mostra onde ela será colocada ao soltá-la. Ela ficará bem na parte superior da instrução do in order.
12. Selecione a variável, escolha DecimalNumber como o tipo de dados e dê a ela o nome distância. Defina o valor inicial como 0 por meio da opção de menu customDecimalNumber.
13. Desse modo, você terá uma variável declarada e inicializada em sua distância nomeada de código com um valor inicial de 0,0.
14. Salve seu programa.
15. Para atribuir a variável ao valor, arraste o nome da variável (distância) sobre os valores de espaço reservado atuais para 1,2. Você precisará fazer isso duas vezes para as instruções forward move para o peixe palhaço.
16. Execute e teste o código.
17. O peixe palhaço não vai a lugar nenhum porque definimos a distância como 0. Altere o valor para 1,2 e teste o código novamente.

Você pode ver a vantagem de usar variáveis aqui. Se quiser alterar a distância que o peixe nada, só precisará alterar o valor em um local (no nível da variável), e as alterações serão executadas em todo o procedimento.

É possível executar mais uma etapa transmitindo um parâmetro para o procedimento. Parâmetro são informações transmitidas de um procedimento para outro. Você pode criar código que permita a cada peixe especificar a distância que ele nadará ao chamarem o procedimento de nado.

18. Clique no botão add Parameter



19. Você o usará em vez da variável de distância. É necessário ser um DecimalNumber para funcionar no bloco de código e ter um nome que ainda não tenha sido usado no código. Nomeie o parâmetro dist e marque a caixa para dizer que entendeu que precisará atualizar as invocações (chamadas) para esse procedimento. Clique em OK.

20. Exclua a linha de código que criou a variável de distância.

As instâncias da variável existentes no código agora são exibidas em vermelho.

21. Para atribuir o valor do parâmetro para funcionar dentro do código, arraste o nome do parâmetro (dist) para os locais vermelhos. Em seu bloco de código.

22. Clique na guia myFirstMethod

23. A chamada do método de nado agora tem um valor vermelho para o parâmetro dentro do procedimento.



24. Use o menu customDecimalNumber para alterá-lo para o valor 1,2.

25. Execute e teste o código.

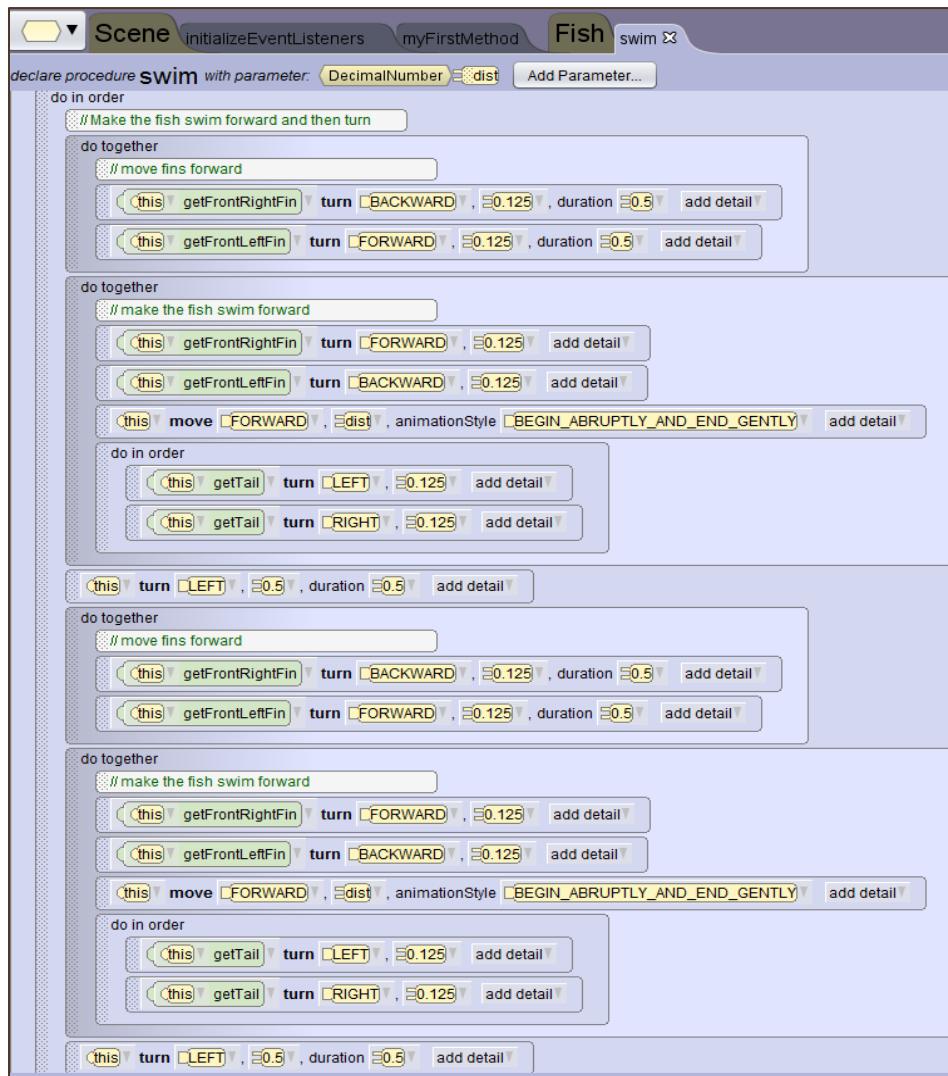
Agora você tem um procedimento que permite deixar qualquer peixe nadar. O peixe cirurgião-patela atualmente nada para frente e para trás. Agora você pode usar o procedimento de nado para torná-lo mais real.

26. Arraste um procedimento de nado do peixe cirurgião-patela para a parte superior do código. Use a mesma expressão para fazer com que o peixe nade três vezes sua própria extensão.

27. Desative o código existente que faz o peixe cirurgião-patela se mover, execute e teste o código.

28. Exclua o código desativado se você estiver satisfeito com os resultados do teste.

29. O procedimento de nado foi feito para o peixe palhaço quando é possível ver apenas um lado por vez. Se for usá-lo para outro peixe, precisará mover as duas nadadeiras ao mesmo tempo e também a cauda. Adapte o procedimento de nado da seguinte forma:



30. Adicione comentários adicionais ao código para explicar o que cada seção faz.

A próxima seção deste exercício é sobre a limpeza do código e a remoção de duplicações de código onde for possível. Se você examinar o procedimento de nado, poderá ver que o mesmo código aparece várias vezes. Isso é algo que você precisa se esforçar para evitar como programador.

Você identificará o código duplicado, o removerá e o colocará em seu próprio procedimento.

The screenshot shows the NetBeans IDE with the 'Fish' class selected. The 'swim' procedure is defined with a parameter 'dist' of type 'DecimalNumber'. The code is as follows:

```

declare procedure swim with parameter: DecimalNumber = dist
do in order
  // Make the fish swim forward and then turn
  do together
    // move fins forward
    (this) getFrontRightFin turn BACKWARD, 0.125, duration 0.5 add detail
    (this) getFrontLeftFin turn FORWARD, 0.125, duration 0.5 add detail
  do together
    // make the fish swim forward
    (this) getFrontRightFin turn FORWARD, 0.125 add detail
    (this) getFrontLeftFin turn BACKWARD, 0.125 add detail
    (this) move FORWARD, dist, animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
    do in order
      (this) getTail turn LEFT, 0.125 add detail
      (this) getTail turn RIGHT, 0.125 add detail
    (this) turn LEFT, 0.5, duration 0.5 add detail
  do together
    // move fins forward
    (this) getFrontRightFin turn BACKWARD, 0.125, duration 0.5 add detail
    (this) getFrontLeftFin turn FORWARD, 0.125, duration 0.5 add detail
  do together
    // make the fish swim forward
    (this) getFrontRightFin turn FORWARD, 0.125 add detail
    (this) getFrontLeftFin turn BACKWARD, 0.125 add detail
    (this) move FORWARD, dist, animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
    do in order
      (this) getTail turn LEFT, 0.125 add detail
      (this) getTail turn RIGHT, 0.125 add detail
    (this) turn LEFT, 0.5, duration 0.5 add detail

```

Three red boxes highlight duplicated code blocks:

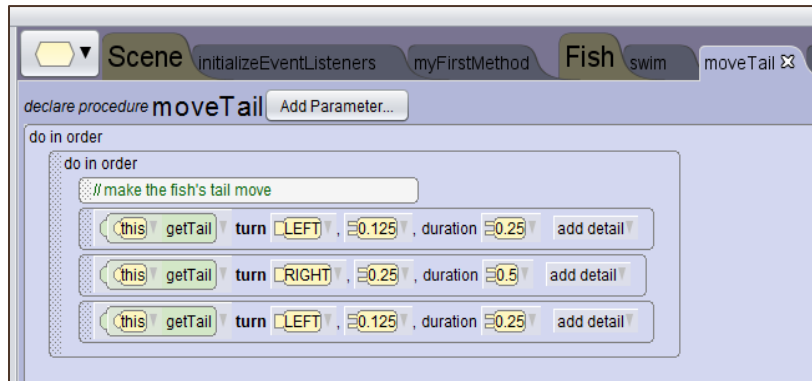
- Box 1 (top): `do together` block containing `// move fins forward` and two `turn` blocks.
- Box 2 (middle): `do in order` block containing `(this) getTail turn LEFT, 0.125 add detail` and `(this) getTail turn RIGHT, 0.125 add detail`.
- Box 3 (bottom): `do together` block containing `// make the fish swim forward` and two `turn` blocks.

Three callout boxes with arrows point to these blocks:

- Green box: "Mover as nadadeiras para frente" (Move the fins forward) - points to the first and last `do together` blocks.
- Red box: "Nadar para frente" (Swim forward) - points to the middle `do in order` block.
- Blue box: "Mover a cauda" (Move the tail) - points to the middle `do in order` block.

31. Você pode ver que há três partes nesse código: mover as nadadeiras para frente, mover a cauda e nadar para frente. É possível remover todas e colocá-las em seus próprios procedimentos.
32. O uso do botão de classes cria um novo procedimento no nível de FISH denominado moveTail.
33. Arraste uma das instruções do in order que controla a cauda na área de transferência.

34. Clique na guia moveTail e arraste o código para ela. Isso produz um movimento simples que você pode melhorar. Para isso, basta criar um fluxo de eventos semelhante ao procedimento shakeHead. Altere o código no procedimento moveTail da seguinte forma:



35. Clique novamente na guia de nado.
36. Remova o outro bloco de código que movia a cauda e substitua-o pelos procedimentos moveTail
37. Crie outro procedimento FISH e nomeie-e como moveFinsForward
38. Arraste um dos blocos de código para a área de transferência que movem as nadadeiras para frente.
39. Clique na guia moveFinsForward e arraste o código da área de transferência para o editor de código.
40. Clique na guia de nado e remova o código para mover as nadadeiras para frente e substitua-o em todos os lugares onde ele faz as nadadeiras serem movidas para frente pelo novo procedimento moveFinsForward.
41. Crie outro procedimento FISH e nomeie-e como swimForward
42. Arraste para a área de transferência um dos blocos de código do together que movem as nadadeiras para frente.
43. Clique na guia swimForward e arraste o código da área de transferência para o editor de código.
44. Esse é um pouco diferente porque exige um parâmetro, então, crie um parâmetro dist aqui exatamente como você fez antes. Arraste o nome do parâmetro para o rótulo vermelho a fim de completar o procedimento.
45. Clique na guia de nado e novamente substitua todas as instâncias do código de nado pelo novo procedimento swimForward.

Ao arrastá-lo para o editor de código, será solicitado que você forneça um valor para o parâmetro. Escolha dist na lista.

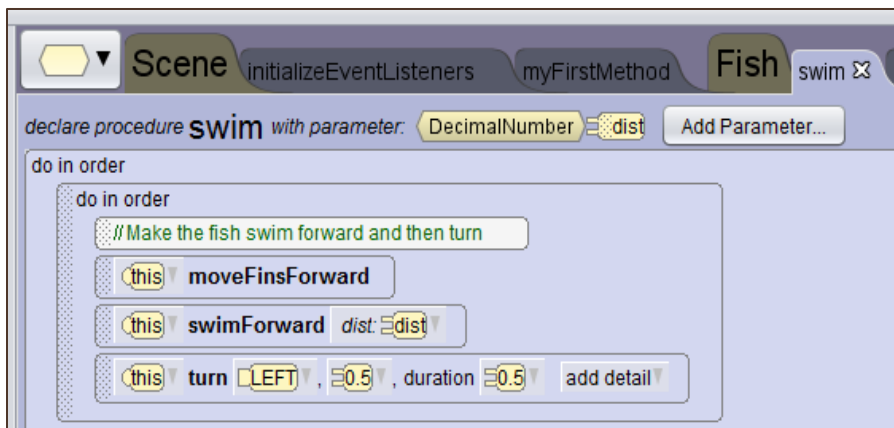
46. Com isso, você obtém uma classe de driver de nado que une todos os elementos dos diferentes procedimentos disponíveis.



Também significa que você tem muito mais flexibilidade em relação ao que é possível fazer com um peixe.

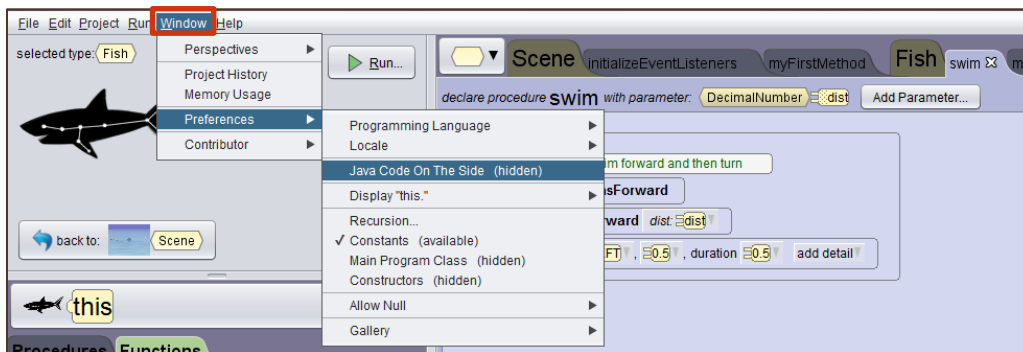
Um último ponto a observar aqui é que o procedimento de nado faz o peixe nadar, girar ao redor e nadar novamente antes de fazer um giro final. O que você pode fazer aqui é reduzir esse código excluindo novamente as duplicações.

47. Exclua as linhas de código até que o procedimento de nado seja o seguinte:

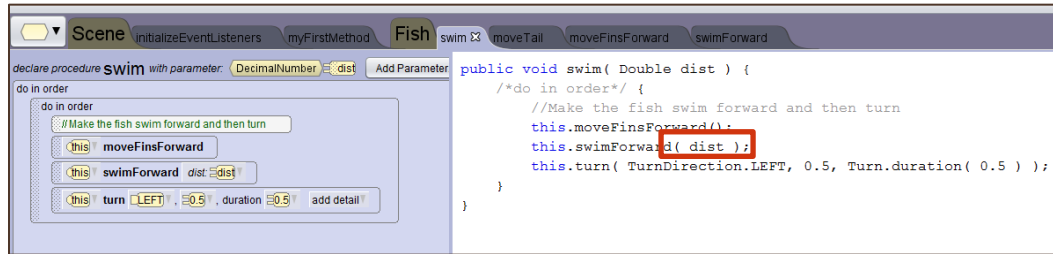


Muito menor aqui do que quando iniciamos e muito mais fácil de ler e entender. A diferença aqui é que o peixe só executará uma operação de nado simples.

48. Se quiser ver como esse código se parece em Java, ative o código java ao lado da opção de menu da janela.

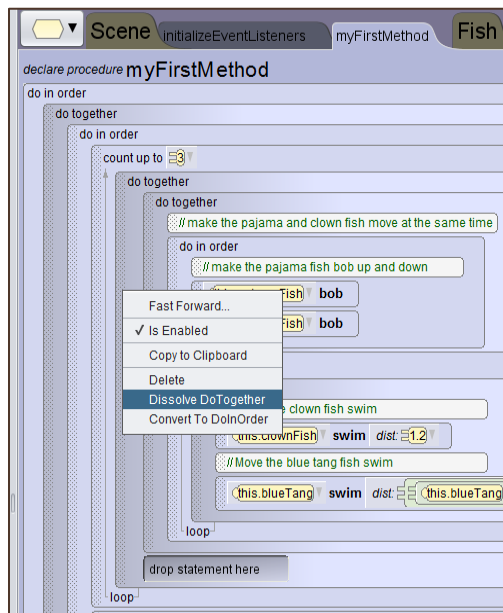


Pode ser útil verificar esse código antes de prosseguir para o Greenfoot e o Eclipse, pois você terá uma ideia de como o Java se parece e funciona.



Você pode observar que o parâmetro `dist` é transmitido em colchetes. É assim que os parâmetros são transmitidos em Java. Se você olhar o método de giro poderá ver que ao transmitir vários parâmetros, eles são separados por vírgulas. É possível obter dicas de codificação úteis ao estudar o código Java na opção lateral.

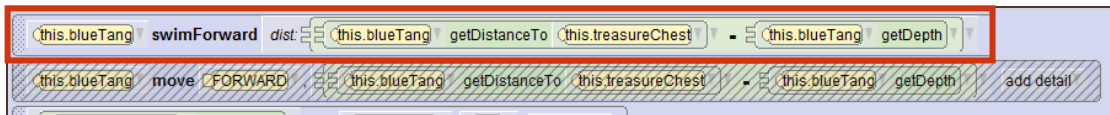
49. Se quiser desativar o código Java na lateral, siga as mesmas etapas para ativá-lo.
50. Para corrigir o fato de que os peixes só nadam uma vez, quando você precisa que eles nadem duas vezes para que a animação funcione corretamente, clique na guia `myFirstMethod`.
51. Agora que o procedimento só produz um nado simples, você pode fazer com que o código o chame uma vez usando uma instrução `count` e escolhendo 2 como argumento. Coloque-a acima do procedimento de nado e arraste a chamada de nado para ele.
52. Faça com que o peixe cirurgião-patela se mova da mesma forma arrastando o procedimento de nado do peixe para a instrução `count loop` ao lado do procedimento de nado do peixe palhaço.
53. Altere o valor do peixe cirurgião-patela de 3 para 2 para impedi-lo de nadar para fora da tela.
54. Adicione uma instrução `do together` de modo que os dois peixes nadem ao mesmo tempo.
55. Você pode ver que agora há duas instruções `do together`, sendo que uma delas é redundante. Para dissolver isso, clique com o botão direito do mouse fora da instrução `do together` e escolha `dissolver` no menu.





56. O peixe cirurgião-patela balança a cabeça e diz "chega de nadar por hoje!" Seria melhor se essas ações forem executadas juntas. No Alice, você pode facilmente converter essas instruções. Clique com o botão direito do mouse nessa instrução do in order e escolha converter em do together.
57. Você criou vários procedimentos que controlam o modo como um peixe se move na água. Agora, você pode usá-los para tornar o movimento dos peixes mais realista.

Qualquer procedimento de movimento para frente do peixe pode ser substituído por um procedimento swimForward para que as nadadeiras e a cauda se movam enquanto os peixes nadam. Desative todos os procedimentos de movimento para frente e substitua-os por procedimentos swimForward usando os mesmos argumentos.

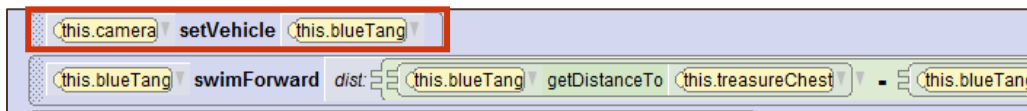


58. Exclua todos os procedimentos desativados quando você testou suas substituições swimForward.

Só faltam mais algumas inclusões cinemáticas para que o projeto seja concluído e esteja pronto para ser exportado para o Alice Player.

59. Quando o peixe cirurgião-patela se move em direção ao baú do tesouro, quero que a câmera se afaste simultaneamente. Para isso, basta ajustar o veículo da câmera da mesma forma que o peixe.

Arraste um procedimento setVehicle da classe Câmera para o editor de código antes que o peixe cirurgião-patela se mova e direção ao baú do tesouro.



60. A câmera deve permanecer em sua posição antes que o peixe cirurgião-patela se afaste

Use a tecla CTRL e arraste o mouse para copiar o código de ajuste do veículo da câmera e solte-o acima do código de giro. Altere o valor de argumento para isso. Isso libera a câmera do peixe cirurgião-patela.

61. Quero que a câmera se aproxime do baú do tesouro enquanto o peixe palhaço e o peixe pijama nadam em direção a ele. Arraste uma instrução moveToward para o código do together usando o baú do tesouro e 2 como argumentos.

62. Em vez de o peixe pijama apenas flutuar no final, inclua um procedimento de virar à esquerda e um procedimento swimForward usando 10 como o argumento no final da instrução if.



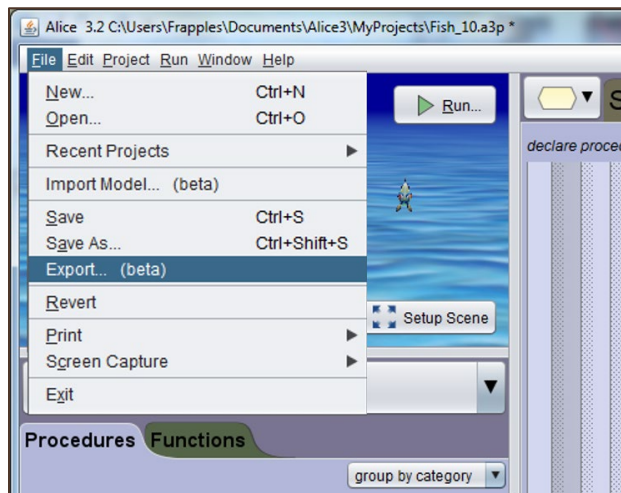
Falta apenas um segundo e último elemento neste projeto. O baú do tesouro deve aguardar alguns minutos antes de voltar para o fundo do mar. Enquanto o baú está se movendo para baixo, a câmera deve se afastar da cena.

63. Adicione uma instrução do in order após a instrução if.
64. Coloque uma instrução delay do baú do tesouro na do in order. Use 4 como o argumento.
65. Adicione uma instrução do together abaixo da instrução delay.
66. Arraste um procedimento move do baú do tesouro movendo-o para baixo pela própria altura (use uma função) e especifique 5 como a duração.
- Arraste um procedimento moveAwayFrom da câmera e escolha baú do tesouro e 5 como argumentos. Defina a duração como 7.
67. Execute e teste sua animação.
68. Uma última coisa. Adicione som quando o peixe palhaço desaparece. Escolha o peixe palhaço, arraste um procedimento play audio para a instrução do together onde o peixe palhaço desaparece.

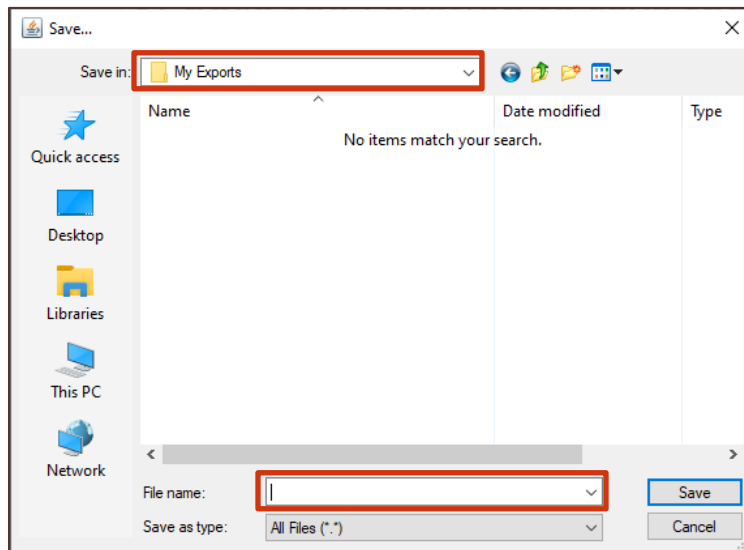
Escolha importar áudio, faixas musicais, e em seguida, a intensidade de seu áudio.

69. Teste sua animação.
70. Salve seu programa.
71. Agora que você tem uma animação completa, pode usar o Alice Player para compartilhar e executar seu programa Alice 3.

72. Clique em File e depois em Export.



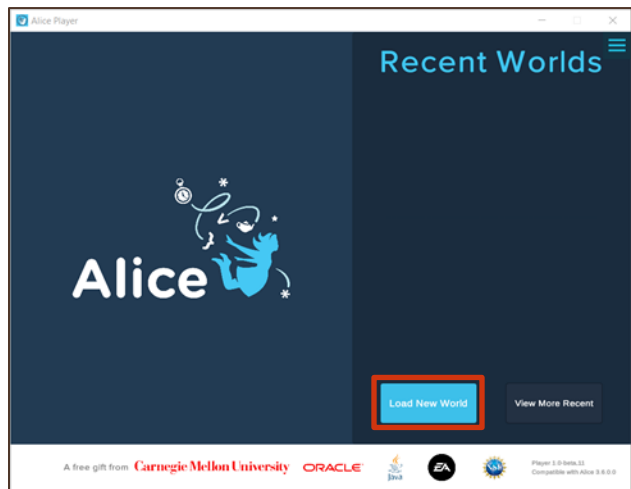
73. Escolha onde o arquivo deverá ser salvo e atribua um nome a ele.



74. Clique em Save. Seu mundo agora está pronto para ser carregado no Alice Player. Observação: O Alice Player é um aplicativo separado e pode ser encontrado aqui: <http://www.alice.org/get-alice/alice-3-beta-vr/>

75. Clique duas vezes no arquivo Alice Player.exe (ou no aplicativo Alice Player se a extensão .exe não estiver visível) encontrado na pasta Alice Player.

76. Selecione Load New World. Em seu computador, navegue até o local em que se encontra o arquivo a3w e selecione o arquivo desse projeto.



Existem controles para o player que permitem alterar a velocidade de reprodução, reiniciar o mundo e retornar ao menu principal para selecionar um novo mundo para jogar. Na primeira vez que você abrir o mundo, eles aparecerão rapidamente na parte superior do player. Para acessar esses controles, arraste o mouse sobre o centro superior do player.

Você terminou sua primeira animação completa e a exportou para o Alice 3!!

77. Saia do Alice 3 e do Alice Player.