ALÉM DE JOGAR,
EU FAÇO JOGOS!

Instituto COOPERFORTE
20 anos transformando realidades

cpdi
Comitê para Democratização da Informática

O que vamos aprender hoje?

- Finalização funções aritméticas
- Introdução à Lua
- Configurações iniciais de um projeto

Introdução à Lua

Lua é uma linguagem de programação de extensão projetada para oferecer suporte a programação processual geral com recursos de descrição de dados. Lua destina-se a ser usada como uma linguagem de script poderosa e leve para qualquer programa que precise de uma

Convenções

Nomes (também chamados de identificadores) em Lua podem ser qualquer combinação de letras, dígitos e caracteres, não começando com um dígito. Isso coincide com a definição de nomes na maioria dos idiomas. A definição de "letra" depende da localidade atual: qualquer caractere considerado alfabético pela localidade atual pode ser usado em um identificador. Identificadores são usados para nomear variáveis e campos de tabela.

Palavras reservadas

As seguintes palavras-chave são reservadas e não podem ser usadas como nomes:

and	**break**	**do**	**else**	**elseif**	**end**	**false**
for	**function**	**if**	**in**	**local**	**nil**	**not**
or	**repeat**	**return**	**then**	**true**	**until**	**while**

As strings a seguir denotam outros tokens:

+	**-**	*****	**/**	**,**	**}**	**[**	**>**	**#**
==	**~=**	**<=**	**>=**	**.**	**%**	**..**	**]**	**=**
(	**)**	**;**	**:**	**{**	**<**	**^**	**...**	

Convenções

Lua é uma linguagem que diferencia maiúsculas de minúsculas: and é uma palavra reservada, mas And e AND são dois nomes diferentes e válidos. Como convenção, nomes começando com sublinhado seguido por letras maiúsculas como (_VERSION) são reservados para variáveis globais internas usadas por Lua.

Tipos e valores

Lua é uma linguagem de tipagem dinâmica. Isso significa que as variáveis não possuem tipos; apenas os valores o fazem. Não há definições de tipo na linguagem. Todos os valores carregam seu próprio tipo.

Todos os valores na linguagem são valores de primeira classe. Isso significa que todos os valores podem ser armazenados em variáveis, passados como argumentos para outras funções e retornados como resultados.

Tipos e valores

Os tipos básicos com os quais precisamos nos preocupar são:

- nil - o tipo do valor nil, cuja propriedade principal é ser diferente de qualquer outro valor; geralmente representa a ausência de um valor útil.
- boolean - o tipo dos valores false e true. Nil e false tornam uma condição falsa; qualquer outro valor o torna verdadeiro.
- number - representa números reais.
- string - representa matrizes de caracteres. Lua é limpa em 8 bits, incluindo zeros embutidos.

Tipos e valores

Lua fornece conversão automática entre valores de string e número em tempo de execução. Qualquer operação aritmética aplicada a uma string tenta converter essa string em um número, seguindo as regras de conversão usuais. Por outro lado, sempre que um número é usado onde uma string é esperada, o número é convertido em uma string, em um formato razoável.

Operadores relacionais

Os operadores relacionais em Lua são:

==	**igual a**
~=	**diferente de**
<	**menor que**
>	**maior que**
<=	**menor ou igual a**
>=	**maior ou igual a**

Eles sempre resultam em true ou false.

Operadores lógicos

Os operadores lógicos em Lua são and, or e not. Todos os operadores lógicos consideram nil e false como falso e tudo o mais como verdadeiro.

and

Esse operador de conjunção retorna o primeiro argumento se esse valor for false ou nil; caso contrário, retorna o segundo argumento.

or

Esse operador de conjunção retorna seu primeiro argumento se esse valor for diferente de nil ou false; caso contrário retorna o segundo.

not

O operador de negação sempre retorna false ou true.

Concatenação

O operador de concatenação de strings em Lua é indicado por dois pontos (..). Se ambos os operandos forem strings ou números, eles serão convertidos em strings de acordo com as regras de convenção faladas anteriormente.

Alocação de memória

Os dispositivos móveis em particular, possuem memória limitada disponível para uso, portanto, deve-se tomar cuidado para garantir que o consumo total de memória no nosso jogo seja mínimo.

Lua realiza gerenciamento automático de memória. Isso significa que você não precisa se preocupar em alocar memória para novos objetos. Também não precisamos liberar memória explicitamente quando os objetos não forem mais necessários. Lua gerencia a memória automaticamente executando um coletor de lixo de tempos em tempos para coletar todos os objetos "mortos" (objetos que não são mais acessíveis a partir de Lua).

Alocação de memória

Toda memória usada por Lua está sujeira a gerenciamento automático. No entanto, cabe a nós dizer a Lua o que considerar lixo. Por exemplo, qualquer coisa armazenada em uma variável global não é considerado lixo, mesmo que seu jogo nunca mais o use. Da mesma forma, qualquer coisa armazenada em uma tabela não será considerada lixo se ela não puder ser removida pelo coletor de lixo, mesmo que a variável/objeto armazenado tenha sido inicialmente declarado no escopo local. Em ambos os casos, cabe a nós atribuir nil a esses cargos. Isso garante que a memória correspondente não seja bloqueada e possa ser liberada pelo coletor de lixo.

Configuração do projeto

As definições de configuração de um jogo são definidas usando o arquivo config.lua escrito na sintaxe Lua. Ele é criado automaticamente na pasta do projeto ao ser criado no Corona SDK e deve se manter armazenado nesse diretório.

Embora ele seja um arquivo Lua, sua função é apenas configurar o jogo antes de começar a funcionalidade principal main.lua. Não se deve fazer qualquer tipo de execução de ação no arquivo config.

Estrutura básica

O arquivo config.lua é configurado utilizando uma tabela content aninhada dentro de uma tabela application da seguinte maneira:

```
application =
{
    content =
    {
        width = 320,
        height = 480,
        scale = "letterbox",
        fps = 60,

        --[[
        imageSuffix =
        {
            ["@2x"] = 2,
            ["@4x"] = 4,
        },
        --]]
    },
}
```

Dimensionamento de conteúdo

Como a maioria dos apps e jogos são desenvolvidos para vários dispositivos e resoluções de tela, o Corona possui várias opções de dimensionamento de conteúdo. Isso vai permitir que usemos um conjunto comum de coordenadas de tela enquanto o Corona dimensiona automaticamente tudo que for sendo colocado na tela para diferentes resoluções, dependendo do dispositivo.

Área de conteúdo

Um conceito fundamental por trás do escalonamento de conteúdo é a área de conteúdo. Nós podemos definir nossa área de conteúdo como desejarmos, mas geralmente ela é baseada em uma proporção comum de largura/altura da tela como 2:3, por exemplo 320 x 480.

A área de conteúdo representa o "palco" geral onde tudo vai ser exibido. O posicionamento sempre vai acontecer em relação a essa área de conteúdo, que pode ou não corresponder à resolução de pixels da tela. A área de conteúdo será dimensionada para caber na tela, com diferenças sutis que são ditadas pela definição do parâmetro scale.

O sistema interno de coordenadas também vai depender da área de conteúdo.

Largura e altura

A área de conteúdo é definida pelos valores width e height que estão dentro da tabela content. Se omitirmos ou definirmos um desses valores como 0, o dimensionamento do conteúdo não será usado.

É muito importante lembrar que a área de conteúdo deve ser definida sempre em relação à orientação retrato, mesmo quando nosso jogo é projetado para orientação de paisagem. O controle da orientação do jogo é definido por parâmetros específicos no arquivo build.settings

Escala

O método de dimensionamento da área de conteúdo é determinado pelo parâmetro scale. Se omitirmos isso (não é recomendado), os valores de largura e altura serão ignorados e a área de conteúdo será definida para a largura e altura reais do pixel do dispositivo.

Os valores possíveis de scale incluem:

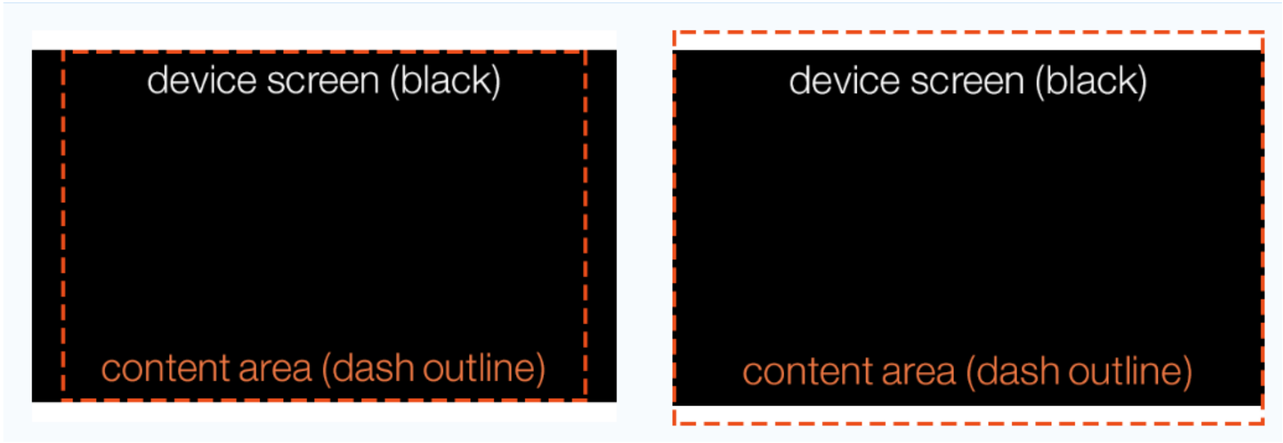
"letterbox"

Dimensiona a área de conteúdo para preencher a tela enquanto preserva a proporção. Toda a área de conteúdo aparecerá na tela, mas isso pode resultar em barras pretas em dispositivos que possuem proporção diferentes da do nosso conteúdo. É um modo de escala ideal se quisermos garantir que tudo em nossa área de conteúdo apareça nos limites da tela em todos os dispositivos.

Escala

"zoomEven"

Dimensiona a área de conteúdo para preencher a tela enquanto preserva a proporção. Alguns conteúdos podem sangrar nas bordas da tela em dispositivos com proporções diferentes da proporção do nosso conteúdo. Basicamente, zoomEven é uma boa opção para garantir que toda a tela seja preenchida pela área de conteúdo em todos os dispositivos (e o recorte de conteúdo próximo às bordas externas é aceitável).



Escala

"adaptive"

Ao invés de uma área de conteúdo estático, uma largura e altura de conteúdo dinâmico são escolhidas com base no dispositivo. Esse modo não é compatível com aplicativos desktop macOS ou Win32.

"zoomStretch"

Dimensiona a área de conteúdo para preencher a tela inteira em qualquer dispositivo, ignorando a proporção do conteúdo. Esse modo deve ser usado com cuidado, pois ele estica/distorce imagens e textos se a proporção do dispositivo não corresponder exatamente à proporção da área de conteúdo.

Escala

"adaptive"

Ao invés de uma área de conteúdo estático, uma largura e altura de conteúdo dinâmico são escolhidas com base no dispositivo. Esse modo não é compatível com aplicativos desktop macOS ou Win32.

"zoomStretch"

Dimensiona a área de conteúdo para preencher a tela inteira em qualquer dispositivo, ignorando a proporção do conteúdo. Esse modo deve ser usado com cuidado, pois ele estica/distorce imagens e textos se a proporção do dispositivo não corresponder exatamente à proporção da área de conteúdo.

Objetos de exibição

Os objetos de exibição abrangem uma ampla variedade de objetos visuais que colocamos no palco ou em grupos de exibição. Isso inclui imagens, texto, formas, linhas, sprites e etc.

Localizações X e Y

Ao incluir objetos na tela, é necessário definir as localizações x e y do mesmo na tela.

A linha X determina a localização horizontal, já a linha y define a localização vertical.