Ve druhém dnu se naučíme do projektu přidat zdrojový kód a podíváme se podrobněji na různé možnosti pohybu.

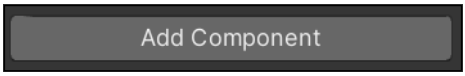
7 Skripty

Už umíme v Unity vytvořit scénu a na konci minulého dne jsme dokonce přidali i nějaký pohyb, ale ten pohyb vyplýval z toho, že jsme jen nastavili, že se nějaký objekt má chovat jako pevné těleso – a zbytek zařídily (simulované) zákony fyziky. Ale jsou i mocnější nástroje!

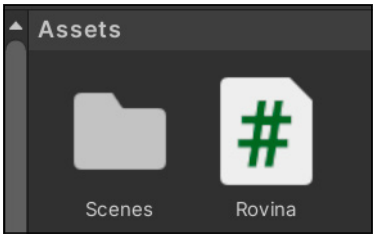
7.1 První skript

Ke každému objektu v Unity můžeme připojit kousek zdrojového kódu, těm kouskům budeme říkat **skripty** a jsou napsané v jazyku C#.

Ty kousky kódu jsou také **komponenty** a přidávají se stejně jako ostatní komponenty v okně **Inspector**, za poslední komponentou v okně **Inspector** stiskneme tlačítko **Add Component**:

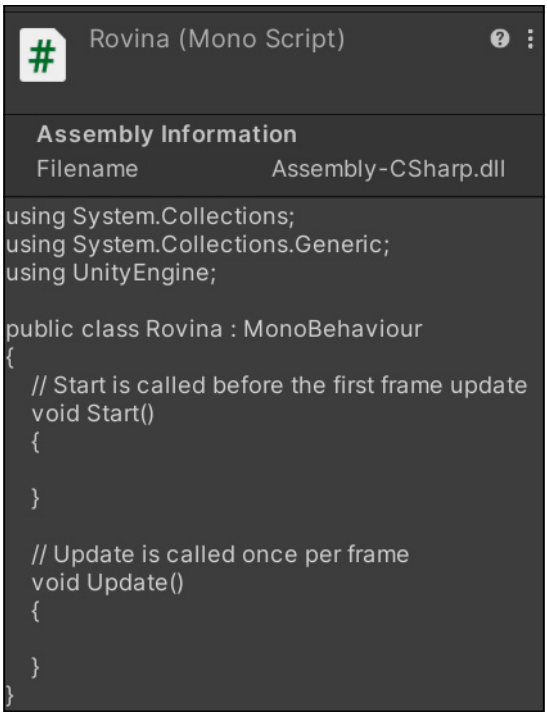


a z nabídky vybereme **New script** úplně dole a potom do políčka **Name** napíšeme jméno nového skriptu, v našem případě „Rovina“, a potvrdíme tlačítkem **Create and Add**. Nový skript se nám objeví v okně **Assets**:



Podobně jako u jiných assets bychom si ho v případě většího projektu měli ukládat do zvláštní složky pro skripty, ale u tohoto projektu ho necháme tam, kde je.

Když si skript v okně **Assets** vybereme, v okně **Inspector** uvidíme jeho náhled – a protože tento skript je prázdný, uvidíme ho celý:



Když na něj poklepáme, otevře se nám v editoru, ale k tomu ještě jeden tip:

Tip 18: Editor skriptů

V nabídce **Edit >>> Preferences >>> External Tools** u vlastnosti **External Script Editor** změňte výchozí hodnotu **Open by file extension** na **Visual Studio Community...** podle vaší verze Visual Studia.

Nevím proč to tak je, ale bez toho nám Visual Studio nenabízí při psaní kódu správnou nápovědu.

Jak vidíme, zdrojový kód je definice třídy pojmenované tak, jak jsme pojmenovali komponentu. Ta třída má připravené dvě funkce a jak říká nápověda v komentáři: funkce **Start()** se volá jednou na začátku, zatímco funkce **Update()** se volá při každé aktualizaci.

Úkol 20: Debug.Log

- Vytvořte nový projekt **Skript**,
- do scény vložte **Rovinu** a na ni **Krychli**,
- objektu **Krychle** přidejte skript **Krychle** (.cs),
- do metod **Start()** a **Update()** přidejte kontrolní tisky pomocí volání funkce **Debug.Log()**:

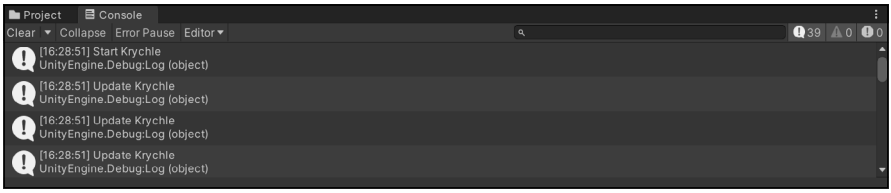
```
Debug.Log("Start " + this.name);
```

a

```
Debug.Log("Update " + this.name);
```

- zkuste spustit hru.

Funkce **Log()** třídy **Debug** vypisuje hlášení do okna **Console** (záložka **Console** v okně **Project/Console**) a když hru spustíme, můžeme vidět, že náš skript jednou vypsal zprávu **Start** a potom opakovaně zprávu **Update**:



Nad výpisem zpráv lze zvolit, jak se budou zprávy vypisovat a mazat a vpravo vidíme počty zpráv různých typů.

Tip 19: Data objektu

Možná jste si všimli, že skript má přístup k datům objektu, kterému je přiřazený!

V tomto úkolu pomocí **this.name** čteme jeho jméno, ale kromě čtení můžeme i dosazovat, a to i do zajímavějších vlastností, než je jméno. Za chvíli toho využijeme.