# **Functie – toonSterrenbeeld (Name, Selection, Duration, Art, ArtDuration)**

Deze functie laat toe om één of (door herhaaldelijk uit te voeren) meerdere sterrenbeelden te selecteren m.b.v. de sterrenbeeldlijnen. Eventueel kan voor deze selectie ook de sterrenbeeldkunst geactiveerd worden.

## **Parameterlijst:**

* Name
* Selection
* Duration
* Art
* ArtDuration

## **Parameters:**

**Name:**

* De naam van het sterrenbeeld.
* Ofwel een variabele, ofwel het sterrenbeeld (of zijn afkorting) tussen dubbele quotes.
* Gebruik de Latijnse namen of (nog beter) de officiële afkortingen.
* **Voorbeelden:**
  + Name = “Ursa Minor”
  + “UMi”

**Selection:**

* Laat toe voor één of meerdere sterrenbeelden de lijnen te tekenen.
* **Mogelijke waarden: “Start” – “” – “End” – “One”**
* Indien “One” dan zal enkel en alleen voor het opgegeven sterrenbeeld de lijnen getoond worden. Op het einde van de functie worden de lijnen weer verwijderd.
* Indien “Start” dan zal voor het opgegeven sterrenbeeld de lijnen getoond worden. Op het einde van de functie blijven de lijnen staan. Een aantal settings worden geactiveerd (voor het tonen van de lijnen).
* Indien “” dan zal voor het opgegeven sterrenbeeld de lijnen getoond worden. Op het einde van de functie blijven de lijnen staan. Wil je dus meerdere sterrenbeelden uitlijnen, dan begin je met het eerste sterrenbeeld met “Start” en de volgende met “”.
* Indien “End” dan zal voor het opgegeven sterrenbeeld de lijnen getoond worden. Op het einde van de functie worden alle lijnen verwijderd. Een aantal settings worden weer gedesactiveerd.

**Duration:**

* Na het activeren van de lijnen blijft het script een bepaald aantal seconden wachten.
* **Uitgedrukt in een aantal seconden (te schrijven zonder quotes). Is de duur gelijk aan 999 dan blijft het script wachten tot je het weer activeert.**

**Art:**

* Bepaalt of de sterrenbeeldfiguren getoond moeten worden (afhankelijk van de ingestelde “sky culture”).
* **Mogelijke waarden: true / false.**
* Indien “true” worden de figuren getoond. Zijn meerdere sterrenbeelden actief, dan zullen de figuren ook voor meerdere sterrenbeelden (tegelijkertijd) getoond worden.

**ArtDuration:**

* De duur dat de sterrenbeeldfiguren getoond worden.
* **Uitgedrukt in een aantal seconden (te schrijven zonder quotes). Is de duur gelijk aan 999 dan blijft het script wachten tot je het weer activeert.**

## **Script:**

(lijnen voorafgegaan met // zijn commentaarlijnen)

function toonSterrenbeeld(Name, Selection, Duration, Art, ArtDuration)

{ if(Selection == "Start" || Selection == "One")

{ ConstellationMgr.setFlagLines(true);

// ConstellationMgr.setFlagLabels(true)

}

core.selectObjectByName(Name, false);

core.wait(Duration);

if(Duration == 999)

{ core.pauseScript();

}

else

{ core.wait(ArtDuration);

}

if(Art == true)

{ ConstellationMgr.setFlagArt(true);

if(ArtDuration == 999)

{ core.pauseScript();

}

else

{ core.wait(ArtDuration);

}

ConstellationMgr.setFlagArt(false);

}

if(Selection == "End" || Selection == "One")

{ ConstellationMgr.setFlagLines(false);

// ConstellationMgr.setFlagLabels(false);

core.selectObjectByName("", false); // Does not work

// Trick to reset the selectObjectByName

core.setSkyCulture("inuit") ;

core.setSkyCulture("western") ;

}

}

## **Opmerkingen:**

* In het begin wordt bij “Start” en “One” de setting *ConstellationMgr.setFlagLines* geactiveerd. Eerst bij “End” of “One” wordt deze setting gedesactiveerd.
* De setting ConstellationMgr.setFlagLabels werd in commentaar gezet. Men opteerde om geen labels (de namen van de sterrenbeelden) te tonen.
* De Stellarium-functie *core.selectObjectByName* werd gebruikt om een sterrenbeeld te selecteren. De optie om het object weer te deselecteren (b)lijkt niet te werken. De enige oplossing om te deselecteren lijkt te zijn om van “Sky Culture” te switchen.

## **Voorbeelden:**

* toonSterrenbeeld (“Ursa Minor”, “One”, 5, true, 10);
* De lijnen van het sterrenbeeld Kleine Beer gaan 5 seconden lang getoond worden, nadien volgt nog de sterrenbeeldkunst. Dat beeld blijft 10 seconden geprojecteerd. Dan verdwijnt de kunst alsook de sterrenbeeldlijnen (dat laatste omwille van “One”).
* toonSterrenbeeld (“Ursa Major”, “Start”, 999, false, 0);

toonSterrenbeeld (“Ursa Minor”, “End”, 999, true, 999);

* Eerst worden de lijnen van het sterrenbeeld de Grote Beer getoond. Het script blijft wachten tot de presenteerder actie onderneemt. Dan worden de lijnen van het sterrenbeeld de Kleine Beer getoond. Het script blijft opnieuw wachten tot de presenteerder actie onderneemt. De lijnen blijven staan, want eerst moet nog de sterrenbeeldkunst getoond worden. Dit gebeurt voor beide sterrenbeelden. Het script wacht weer tot het mag verder gaan. Zowel de kunst als de lijnen verdwijnen.
* toonSterrenbeeld (“Ursa Major”, “Start”, 999, false, 0);

toonSterrenbeeld (“Ursa Minor”, “”, 999, true, 999);

core.pauseScript();

toonSterrenbeeld (“Ursa Minor”, “End”, 0, false, 0);

* Het scenario loopt net hetzelfde als in het vorige voorbeeld, alleen zal bij het pauseren van het script nu de kunst wel verdwijnen, maar niet de sterrenbeeldlijnen. De parameter Selection krijgt immers een blanco waarde.. Eerst als het script door de presenteerder weer geactiveerd wordt, zullen de lijnen verdwijnen (parameter Selection = “End”).
* toonSterrenbeeld (“Ursa Major”, “Start”, 0, false, 0);

toonSterrenbeeld (“Ursa Minor”, “End”, 999, true, 999);

* Opnieuw gelijkaardig, maar na het tonen van de sterrenbeeldlijnen voor de Grote Beer word er deze keer NIET gewacht zodat eigenlijk de sterrenbeeldlijnen voor de Grote en Kleine Beer zowat tegelijkertijd geactiveerd worden. Eerst dan pauseert het script (om nadien ook de kunst te tonen voor beide sterrenbeelden).

# **Functie – toonObject**

Deze functie laat toe om een object aan te wijzen. Vaak zal men daarna op het object verder inzoomen. Dit lukt alleen maar indien voor dit object een foto gevonden wordt in de Stellarium database en directories.

**OPGELET**: Zorg er steeds voor dat geen enkel sterrenbeeld geselecteerd is. Mocht dat toch het geval zijn, dan zal bij selectie van het object plots ALLE sterrenbeelden mee geselecteerd worden.

## **Parameterlijst:**

* Name
* Pointer
* Duration1
* Zoom
* ZoomFOV
* ZoomDuration
* Duration2

## **Parameters:**

**Name:**

* De naam van het object (naam ster, Messier nummer, NGC nummer, ...).
* Ofwel een variabele, ofwel de naam van het object tussen dubbele quotes.
* **Voorbeelden:**
  + Name = “M42”
  + “M42”

**Pointer:**

* **“true” – “false”.**
* Indien “true” dan wordt het object op het scherm aangeduid met een knipperende “kruisdraad” voor een bepaalde duur (Duration 1).

**Duration1:**

* Het **aantal seconden** dat de pointer (zie vorige parameter) moet getoond worden.
* Is de waarde gelijk aan **999**, dan wacht het script op een gebruikersactie om weer verder te gaan.

**Zoom:**

* **“true” – “false”.**
* Indien “true” zal er op het object ingezoomd worden. De drie volgende parameters ZoomFOV, ZoomDuration en Duration2 zijn daarbij belangrijk.

**ZoomFOV**

* De Field Of View in graden. Gebruik een decimaal punt.
* **Voorbeeld:**
  + 2.3
  + 5

**ZoomDuration**

* Het **aantal seconden** dat Stellarium zal gebruiken om uit het getoond beeld naar het ingezoomde beeld te gaan.

**Duration2**

* Het **aantal seconden** dat het object getoond wordt.
* **OPGELET**: Dit getal moet minstens gelijk zijn aan de waarde van ZoomDuration. Indien beide gelijk, dan wordt onmiddellijk na het inzoomen weer terug het normale beeld gegeven. Indien groter, dan wordt het verschil in seconden nog gewacht vooraleer terug te keren naar het gewone scherm.
* Is de waarde gelijk aan **999**, dan wacht het script op een gebruikersactie om weer verder te gaan.

## **Script:**

(lijnen voorafgegaan met // zijn commentaarlijnen)

function toonObject(Name, Pointer, Duration1, Zoom, ZoomFOV, ZoomDuration, Duration2)

{ core.selectObjectByName(Name, Pointer);

if(Duration1 == 999)

{ core.pauseScript();

}

else

{ core.wait(Duration1);

} // Gaat inzoomen (FOV / Duration).

if(Zoom == true)

{ StelSkyDrawer.setFlagStarMagnitudeLimit(false);

StelMovementMgr.setFlagTracking(true);

StelMovementMgr.zoomTo (ZoomFOV, ZoomDuration);

if(Duration2 == 999)

{ core.pauseScript();

}

else

{ core.wait(Duration2);

}

StelSkyDrawer.setFlagStarMagnitudeLimit(true);

// Uitzoomen tot volledig fisheye zicht

StelMovementMgr.zoomTo(FOV,0);

StelMovementMgr.setFlagTracking(false);

// Met het zuiden onderaan het scherm.

core.moveToAltAzi(Altitude, Azimuth, 0);

}

core.selectObjectByName("", false);

// Trick to reset the selectObjectByName

core.setSkyCulture("inuit") ;

core.setSkyCulture("western") ;

}

## **Opmerkingen:**

* Bij het inzoomen wordt de (eventuele) beperking op grensmagnitude uitgeschakeld. Mocht deze bijvoorbeeld op vier staan, dan krijg je bij het inzoomen weinig tot geen sterren te zien. Waardoor het effect een beetje verloren gaat.
* Tracking moet geactiveerd worden, zo blijft het ingezoomde object centraal in beeld staan.
* Deze routine maakt gebruik van drie variabelen die buiten de routine een waarde zouden moeten krijgen. Ze zorgen er voor dat bij het uitzoomen de sterrenhemel weer getoond wordt zoals net voor het inzoomen.

Bij een full dome projectie zijn die waarden:

FOV = 180

Altitude = 89 en Azimuth = 0 zorgen er voor dat het noorden onderaan het scherm wordt getoond).

Deze zijn vooral van belang indien een script wordt ontwikkeld voor normale projectie. In een full dome projectie blijven deze drie waarden steeds dezelfde. Bij een gewone projectie zijn deze nogal variabel.

* De Stellarium-functie *core.selectObjectByName* werd gebruikt om een object te selecteren. De optie om het object weer te deselecteren (b)lijkt niet te werken. De enige oplossing om te deselecteren lijkt te zijn om van “Sky Culture” te switchen.

## **Voorbeelden:**

* toonObject("M42", true, 10, true, 3.3, 15, 30);
* De eerste “true” laat blauwe vierkantje haakjes flikkeren gedurende 10 seconden rond de positie van M42.
* De tweede “true” zorgt er voor dat er wordt ingezoomd op de nevel. Het zoomen zelf krijgt 15 seconden tijd tot een Field Of View is bereikt van 3,3°.
* Het ingezoomde beeld blijft dan nog eens 15 seconden staan. De laatste functieparameter heeft dan wel een waarde van 30, je moet er de zoomtijd van aftrekken. Met andere woorden de laatste hoeveelheid seconden moet minstens gelijk zijn aan de tijd voor het inzoomen, zoniet wordt het inzoomen vroegtijdig afgebroken.
* toonSterrenbeeld("Orion", "One", 10, false, 0);

toonObject("M42", true, 10, true, 3.3, 10, 30);

* Toont eerst het sterrenbeeld Orion m.b.v. de sterrenbeeldlijnen. Deze verdwijnen om dan in te zoommen op de Orionnevel.
* toonSterrenbeeld("Orion", "End", 10, false, 0);

toonObject("M42", true, 5, true, 3.3, 10, 999);

* Het *toonSterrenbeeld* statement zorgt ervoor m.b.v. “End” dat eerder geselecteerde sterrenbeelden niet meer op het scherm getoond worden.

# **Functie – toonHelp**

Bij full-dome projectie worden de hoeken van het pc-scherm niet mee geprojecteerd. De functie toonHelp laat toe om in de linkerbovenhoek een tiental korte instructies of geheugensteuntjes te plaatsen voor diegene die het programma presenteert. Bij Urania in hoofdzaak gebruikt om de volgende stappen in het scenario op te lijsten.

Opmerking: Maak de teksten niet te lang. Zo vermijd je dat ze toch binnen de cirkel van de sterrenhemel komen (en bijgevolg op het scherm geprojecteerd zullen worden).

## **Parameterlijst:**

* T1
* T2
* T3
* T4
* T5
* T6
* T7
* T8
* T9
* T10

## **Parameters:**

**T1 t.e.m. T10:**

* De tekst die op het scherm moet komen in de linkerbovenhoek. T1 is de eerste lijn, T2, de tweede enzovoorts.
* Ofwel een variabele, ofwel de naam van het object tussen dubbele quotes.
* De 10 parameters moeten vermeld worden. Indien een parameter geen tekst moet bevatten, gebruik dan “” als waarde.
* **Voorbeelden:**
  + Tekst = “Inzoomene op M42”
  + “Sterrenbeelden Dierenriem worden getoond”

## **Script:**

(lijnen voorafgegaan met // zijn commentaarlijnen)

function toonHelp(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)

{ LabelMgr.deleteAllLabels(); // Vorige teksten verwijderen

id = LabelMgr.labelScreen(T1, 100, 100, true, 20);

id = LabelMgr.labelScreen(T2, 100, 130, true, 20);

id = LabelMgr.labelScreen(T3, 100, 160, true, 20);

id = LabelMgr.labelScreen(T4, 100, 190, true, 20);

id = LabelMgr.labelScreen(T5, 100, 220, true, 20);

id = LabelMgr.labelScreen(T6, 100, 250, true, 20);

id = LabelMgr.labelScreen(T7, 100, 280, true, 20);

id = LabelMgr.labelScreen(T8, 100, 310, true, 20);

id = LabelMgr.labelScreen(T9, 100, 340, true, 20);

id = LabelMgr.labelScreen(T10, 100, 370, true, 20);

}

## Voorbeelden:

* toonHelp("Grote Beer",

"Kleine Beer",

"Cassiopeia (P)",

"Alles KUNST",

"Gevolgd door PAUZE",

"",

"",

"",

"",

"");

* De eerste vijf lijntjes doen tekst verschijnen in de linkerbovenhoek verschijnen, de vijf laatste lijntjes blijven leeg.

# **Functie – startRotation**

Deze functie toont een versnelde dagbeweging tot een bepaalde datum/uur bereikt is. De snelheid wordt dan weer aangepast naar de reële snelheid van de dagbeweging.

## **Parameterlijst:**

* Rate
* Date

## **Parameters:**

**Rate:**

* Het aantal keer dat de tijd sneller zal gaan.
* Meestal gebruiken we **een geheel getal** tussen 1000 en 2000.
* Een negative waarde zou ook mogelijk zijn. In dat geval gaat de tijd achteruit.

**Date:**

* De versnelde dagbeweging blijft duren tot het opgegeven tijdstip bereikt is.
* Een tijd uitgedrukt in het formaat YYYY-MM-DDTHH:MM:SS
* Voorbeeld:
  + date = "2019-08-16T17:00:00"

## **Script:**

function startRotation(rate, date)

{ core.setTimeRate(rate);

core.waitFor(date, "local");

core.setTimeRate(1);

}

## **Voorbeelden:**

* datum = “2019-09-18T23:00:00;

startRotation(2000, datum);

* De dagbeweging wordt 2000 keer versneld tot de datum/tijd bereikt wordt zoals die in de variabele datum werd gezet
* Is de waarde van variabele datum kleiner dan de huidige datum in het script, dan zal er geen dagbeweging zijn (tenzij de waarde van rate negatief is en de tijd dus terug draait).