# Gui layout

De gui is opgebouwd als een volledig onafhankelijk venster die de, door andere klassen voorziene code, implementeert. Voor het maken van het venster zelf werd gebruik gemaakt van de geïmporteerd *module* Tkinter, welke verzien wordt door Python.

Het applicatievenster bestaat uit verschillende *frames* die allemaal in 1 grote *container* geplaatst worden. Deze container wordt op het 1ste niveau ingedeeld in 2 blokken, het linker en rechter blok, die elk 1 frame bevatten met verschillende *widgets*. Het rechter blok wordt nog verder onderverdeeld op een 2de niveau in 3 afzonderlijke frames die allemaal hun eigen widgets bevatten. Doorheen de volledige applicatie is er gebruik gemaakt van een *grid-layout*

# Bespreking linker frame:

Aan het linker frame werden er 3 widget toegevoegd, de *canvas*, het *progressionlabel* en de *progressbar*. Het canvas widget wordt gebruikt om figuren en bewegende objecten op te tekenen. Binnen de applicatie houdt dit in dat hier de planeten op geconstrueerd worden en bewegen. Het progressionlabel en de progressbar geven hoever de species ontwikkeld zijn.

# Bespreking rechter frame:

Het rechter frame is zoals eerder vermeld onderverdeeld in een top-, center- en bottomframe. Het bovenste frame bestaat uit 4 knoppen waar de gebruiker een *researchfocus* kan zetten, het middelste frame staat dan weer in voor de weergave van events en het onderste frame geeft meer informatie over de geselecteerde planeet/*mainplanet*.

# Methodes in gui:

## research\_focus:

De gebruiker kan een focus kiezen voor verdere ontwikkeling.

## create\_canvas:

Canvas object wordt aangemaakt. Zowel achtergrond, grootte en plaats wordt opgegeven.

## Thread\_make:

Aparte thread voor het afzonderlijk besturen van de canvas wordt aangemaakt.

## Show\_planet:

Deze methode krijgt een lijst van planeet objecten mee en een canvas object. Aan de hand van de meegegeven planeten wordt een object van type planet\_drawing aangemaakt en getekend op de canvas samen met de bijhorende bewegingscirkels.

## Get\_random\_angle:

Krijgt een aantal mee en geeft een lijst van gegenereerde getallen terug. Er worden “aantal” hoeken gegenereerd die nooit minder dan 20° van elkaar verwijderd zijn. Deze methode wordt gebruikt bij show\_planet om ervoor te zorgen dat overlap niet mogelijk is.

## Update\_canvas:

Zorgt ervoor dat de getekend planeten binnen het canvas object bewegen.

## Create\_progressbar:

Creëert de progressbar die de progressie van de simulatie weergeeft.

## Create\_message\_window:

Zorgt voor de aanmaken van het frame die de opgeteerde event toont.

## create\_info\_frame\_planet:

Deze methode krijgt 2 parameter mee: een dictionary en een boolean. Aan de hand van het aantal meegegeven elementen in de dictionary worden de labels verdeeld over de plaats die ter beschikking is, tevens wordt de weer te geven tekst ook uit de dictionary gehaald. Door de waarde van de boolean bepaalt de methode of de er een “set main planet” knop moet getoond worden.

## update\_info\_frame\_label:

Elke beurt worden de nieuwe berekende waarde opgehaald en aangepast indien nodig, tevens toont deze methode het verschil t.o.v. de vorige beurt.

## set\_main\_planet:

Om de achterliggende code duidelijk te maken welke planeet er al mainplanet moet aangeduid worden, is deze methode ontworpen. Hier gebeurt ook de aanroep van de thread\_make en de create\_info\_frame\_planet. Tijdens de oproep van deze methode wordt een popup-venster opgeroepen die aan de gebruiker vraagt om zijn puntenverdeling, over de verschillende techs, te geven.

## Next\_turn:

Indien een mainplanet is gekozen, kan deze methode opgeroepen worden door op de “next turn” knop te drukken. Alle nieuwe waarden worden opgevraagd en naar de corresponderende functies doorgestuurd zoals de progressbar.

# Definitions

## frame

In Tkinter, the Frame widget is the basic unit of organization for complex layouts. A frame is a

rectangular area that can contain other widgets.

## widget

The generic term for any of the building blocks that make up an application in a graphical user interface. Examples of widgets: buttons, radiobuttons, text fields, frames, and text labels.

## window

This term has different meanings in different contexts, but in general it refers to a rectangular area somewhere on your display screen.

## Grid layout

This layout manager treats every window or frame as a table—a gridwork of rows and columns.

## Canvas wigdet

A canvas is a rectangular area intended for drawing pictures or other complex layouts. On it you can place graphics, text, widgets, or frames.

## Module

a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code. (https://www.tutorialspoint.com/python/python\_modules.htm)