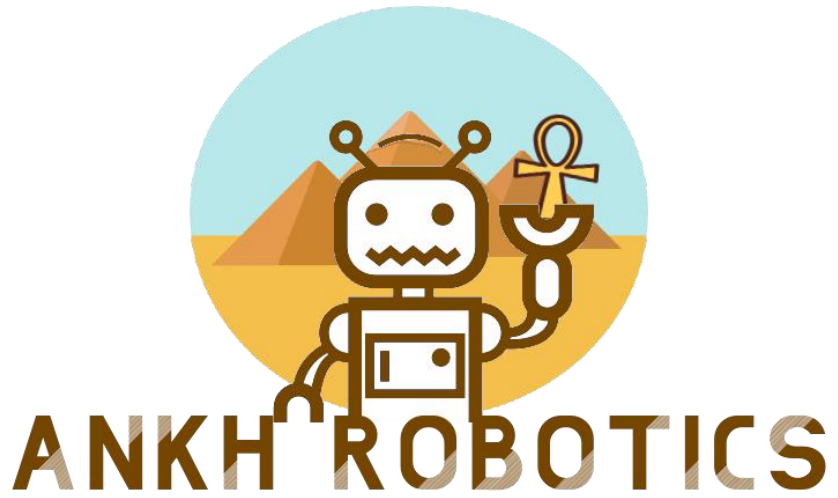


ROS2 Lecture 0



Today's Lecture

01

ROS History

02

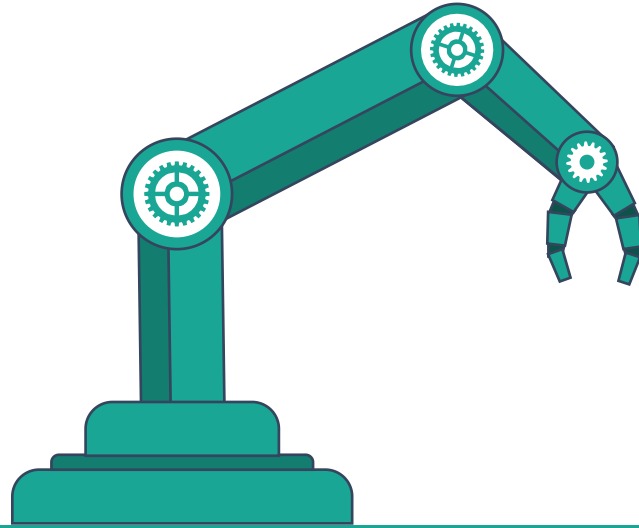
Why ROS2

03

Prerequisites

04

Prepare the enviroment

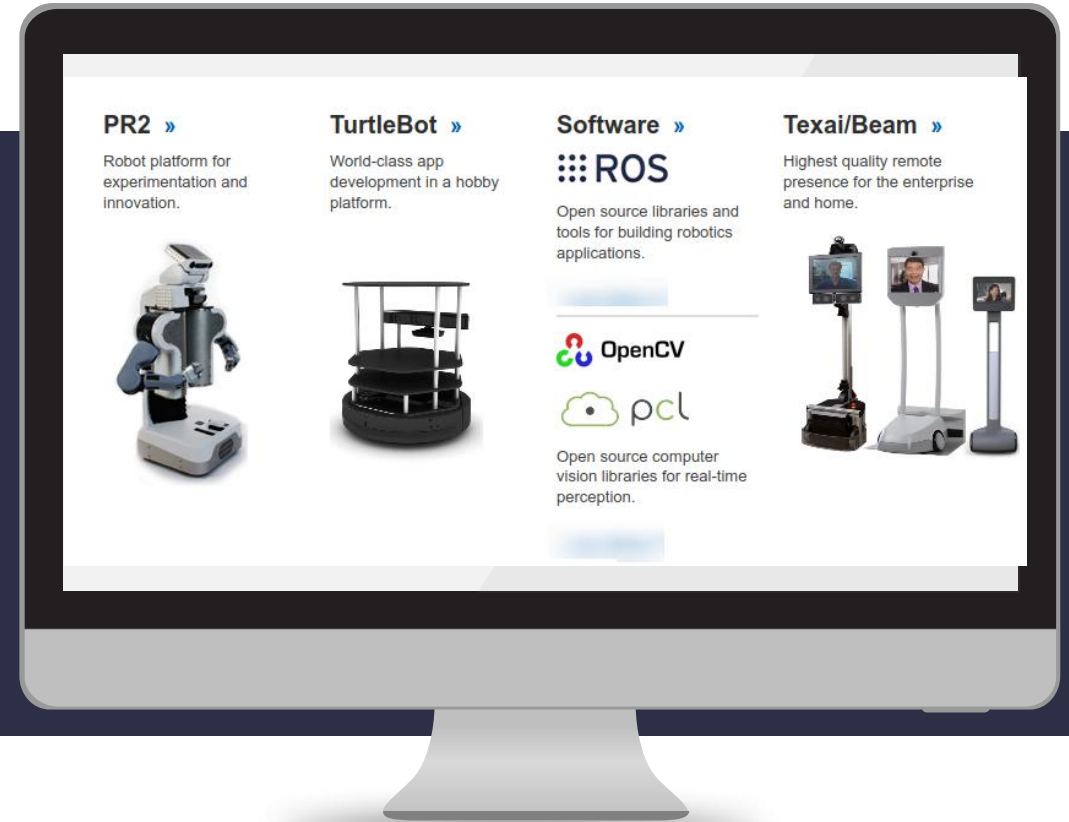


ROS history, and why ROS2?

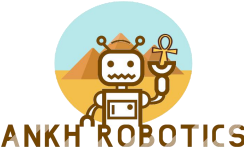
Willow Garage



- in 2006, a former Google VP started Willow Garage.
- ROS began life as the development environment for the Willow Garage PR2 robot.
- in 2012 Open Robotics became responsible for the development of ROS



Why ROS2?



ROS was made for PR2 and guided by it's use case

- a single robot
- workstation-class computational resources on board
- no real-time requirements (or, any real-time requirements would be met in a special-purpose manner)
- excellent network connectivity (either wired or close-proximity high-bandwidth wireless)
- applications in research, mostly academia
- maximum flexibility, with nothing prescribed or proscribed

Despite all of this ROS is very successful and was adapted in many great products currently in the market.

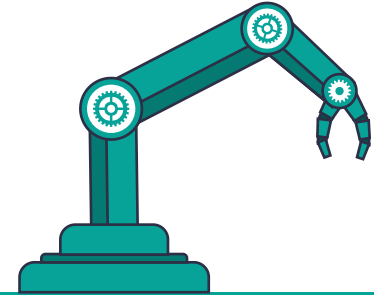
wheeled robots of all sizes, legged humanoids, industrial arms, outdoor ground vehicles (including self-driving cars), aerial vehicles, surface vehicles, and more.

Why ROS2?

New use cases



- Small embedded platforms
- Real-time systems
- Non-ideal networks
- Production environments
- Prescribed patterns for building and structuring systems
- Teams of multiple robots

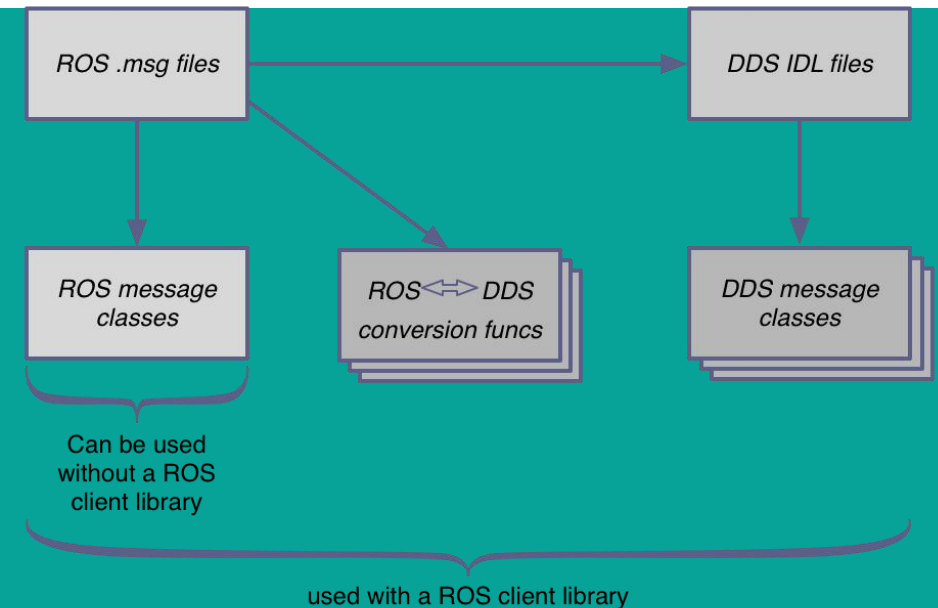
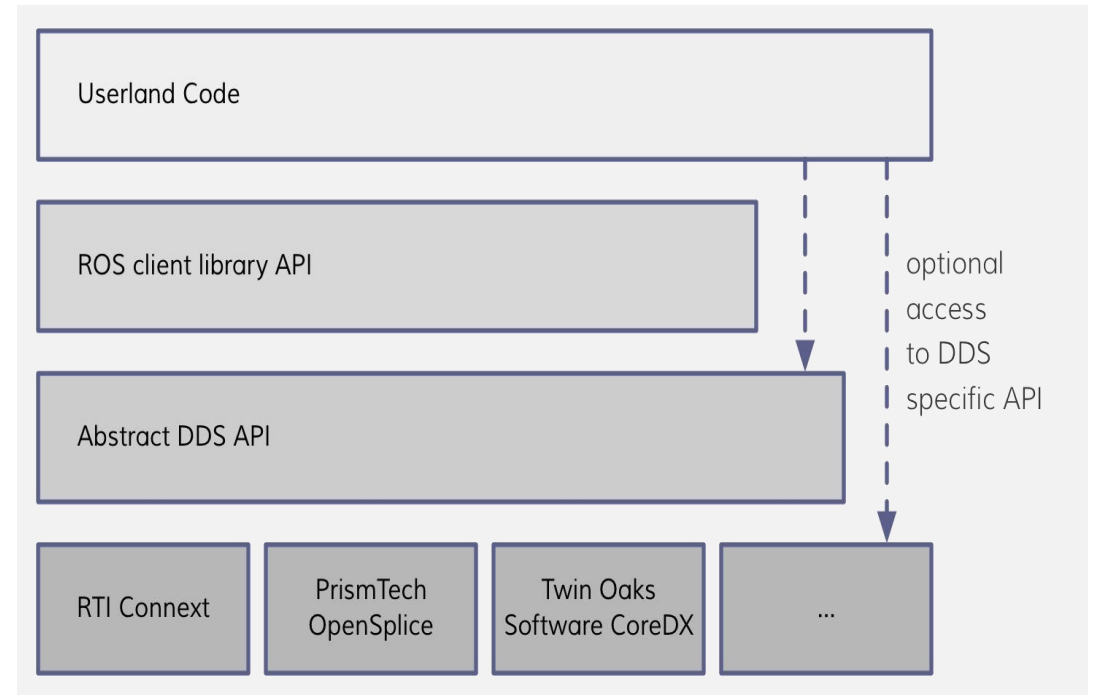


New technologies

- It is now possible to build a ROS-like middleware system using off-the-shelf open source libraries.
- maintain less code, especially non-robotics-specific code
- take advantage of features in those libraries that are beyond the scope of what we would build ourselves
- benefit from ongoing improvements that are made by others to those libraries
- can point to existing production systems that already rely on those libraries when people ask us whether ROS is “ready for prime time”.

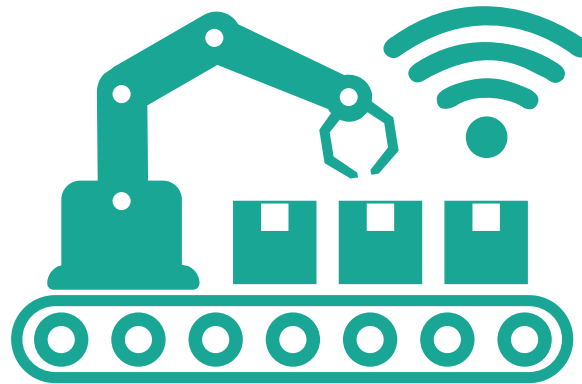
Data Distribution Service (DDS)

- DDS provides a publish-subscribe transport which is very similar to ROS's publish-subscribe transport
- DDS uses the "Interface Description Language (IDL)" for message definition and serialization
- DDS has a request-response style transport, which would be like ROS's service system.
- The default discovery system provided by DDS is a distributed discovery system. This allows any two DDS programs to communicate without the need for a tool like the ROS master.
- Efficient Transport Alternatives
- DDS Security Specification



Noetic Ninjemys: The Last Official ROS 1 Release

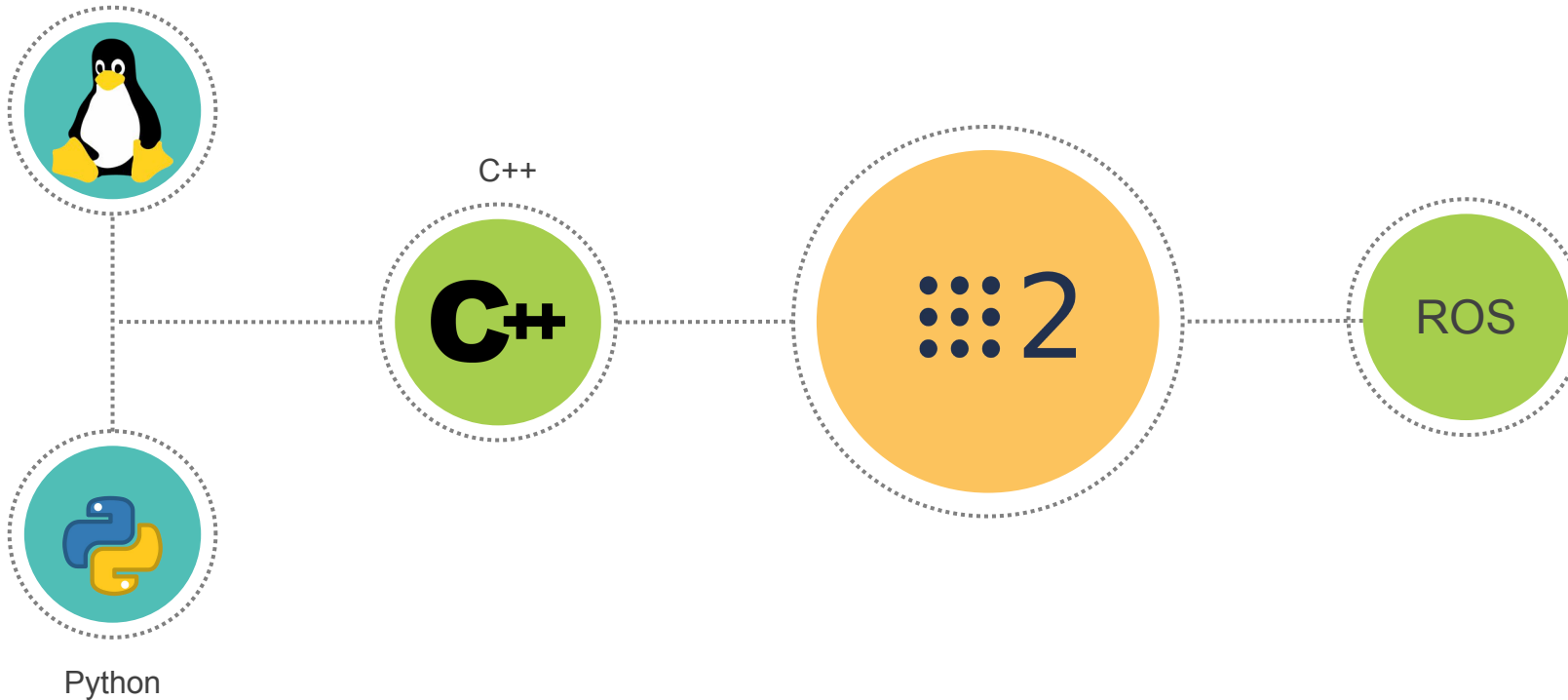




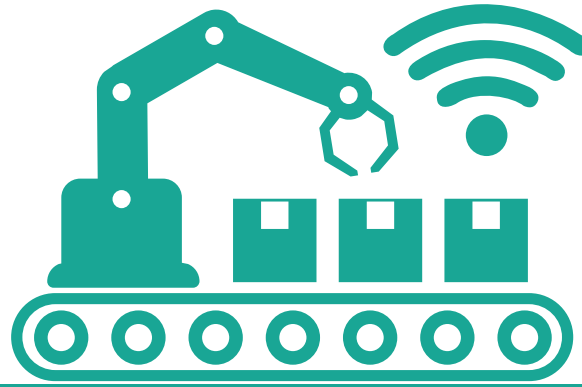
Prerequisites

Prerequisites

Basics of linux command line.



Shwyt Linux, Shwyt Python, Shwyt ROS, w shwyt C++ Kootar awwiii



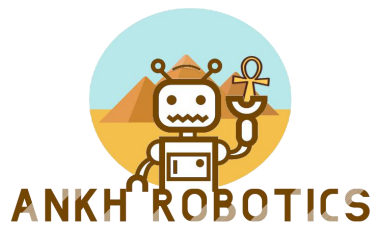
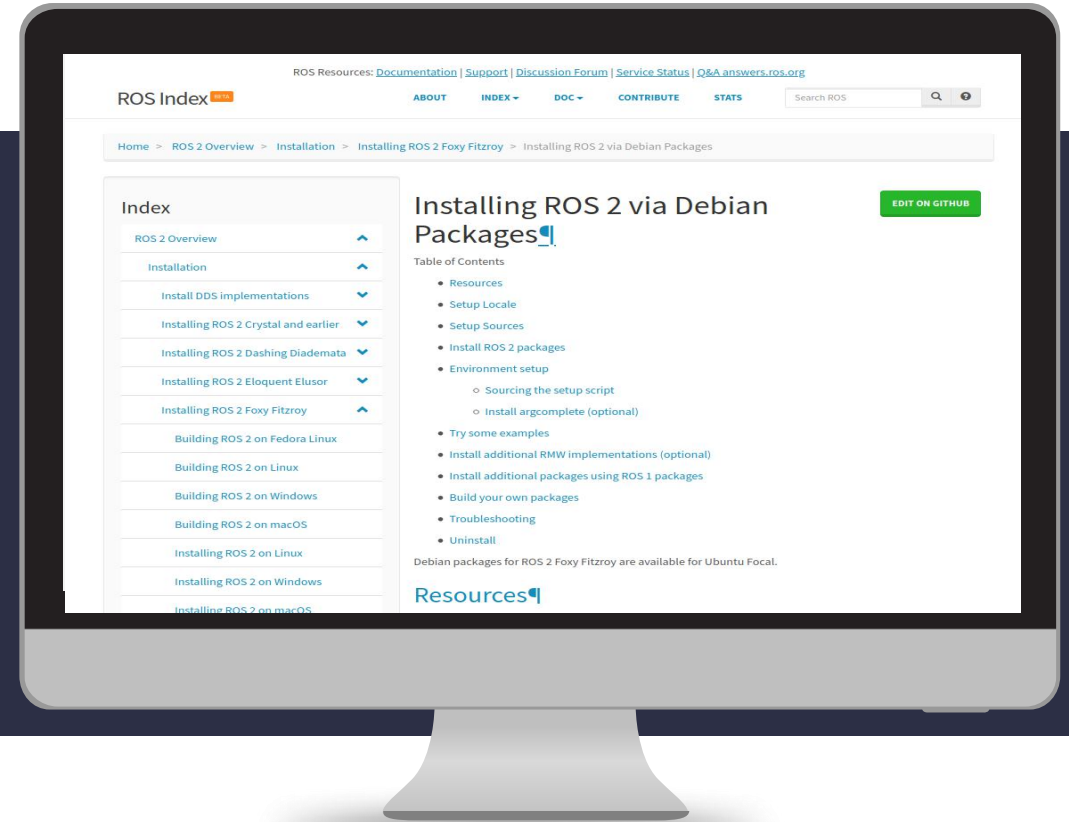
Preparing the Environment

Installing ROS2

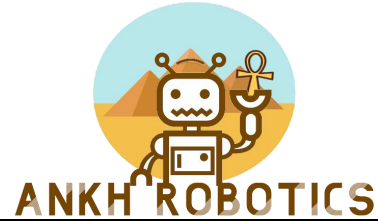
Ubuntu 18.04 -> Dashing Diademata

Ubuntu 20.04 -> Foxy Fitzroy

You can use docker -> `docker pull osrf/ros:dashing-desktop`



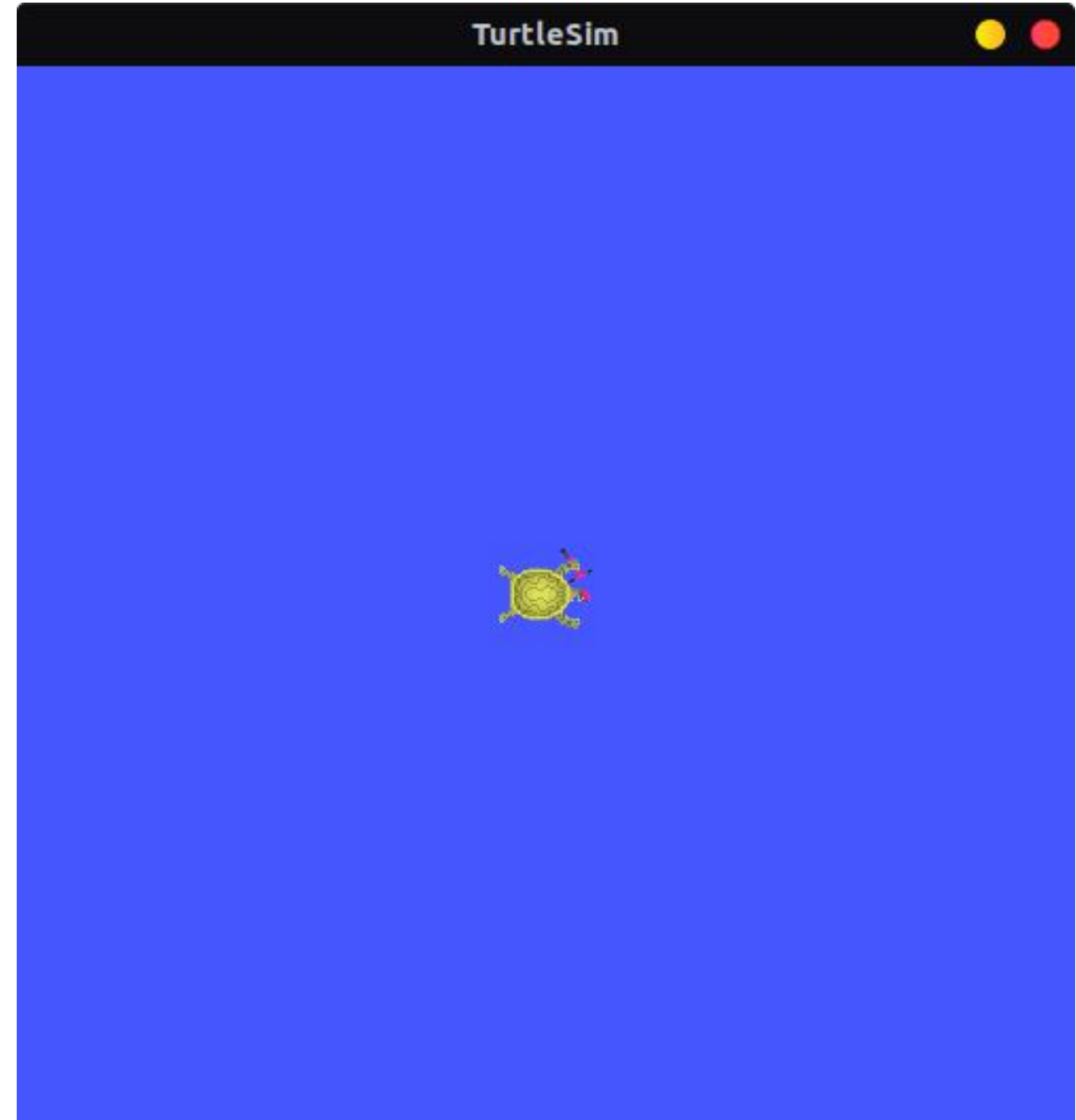
Lets Run the Great Turtle simulation

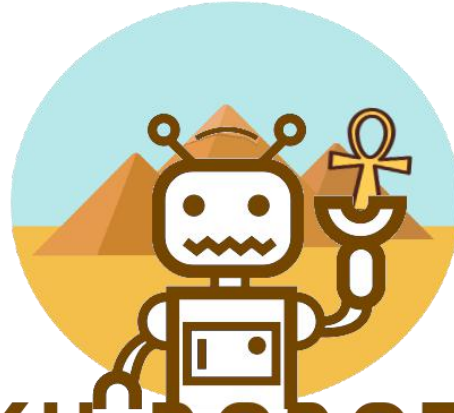


```
:~$ source /opt/ros/dashing/setup.bash  
:~$ sudo apt update  
:~$ sudo apt install ros-dashing-turtlesim  
:~$ sudo apt install ros-dashing-rqt-*
```

```
:~$ source /opt/ros/dashing/setup.bash  
:~$ ros2 run turtlesim turtlesim_node
```

```
:~$ source /opt/ros/dashing/setup.bash  
:~$ ros2 run turtlesim turtle_teleop_key
```





ANKH ROBOTICS

Thank You