**KAT Virtual Runtime Usage Documentation**

-2021.09.14

Virtual Runtime is a simulation of a KAT treadmill that can be set to walk in a set direction and at a set speed. The purpose of using it is to allow developers to (in the absence of a KAT treadmill) quickly test whether their project is successfully adapted to the KAT treadmill.

**Caution.**

1. Virtual Runtime can only run and simulate the KAT treadmill on applications that have been successfully adapted to the KAT SDK; applications that have not been successfully adapted to the KAT SDK cannot use any runtime programs.

2, for developers who already have a treadmill (with dongle and KAT Industry (industry application)), please turn off Virtual Runtime when using KAT Industry (KAT series software including KAT IO B, KAT IO, KAT Industry, Virtual Runtime and KAT Gateway cannot be turned on at the same time).

3. open to set up the parameters to do so, with no sequential start-up sequence between the test games.

4. For unity developers.

(1) If you get an error such as "The specified module cannot be found" or similar during debugging or program operation, copy the 5 dll files included in the ToSystem32 package to the C:\Windows\System32 folder on the system disk (default is C disk). in the system disk (default is C). If prompted for the files, select Overwrite source files.

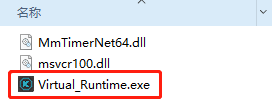
(2) If there are error reporting issues related to the SteamVR plugin during debugging, please go to the official resource store to download the latest SteamVR plugin.

5. For UE4 developers.

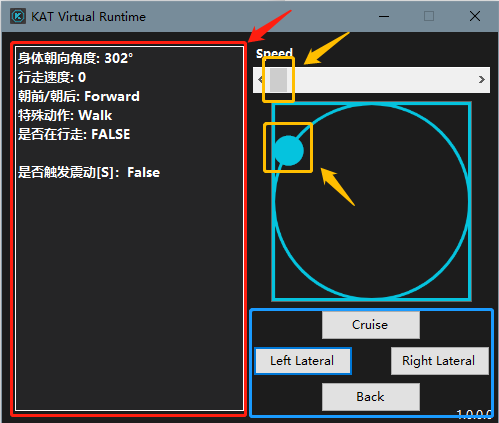
(1) If you can't read the data in runtime correctly or can't walk during debugging, please check if there are 5 dll files included in ToSystem zip package under C:\Windows\System32 path.

**Features.**

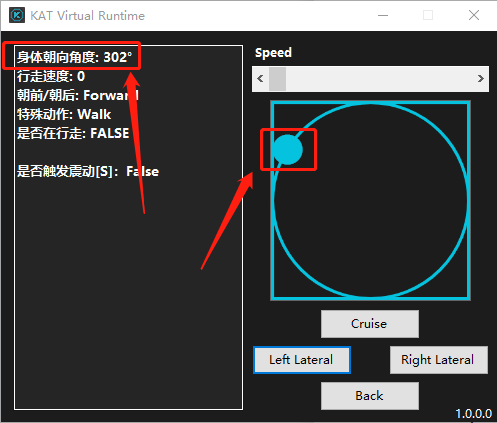
Double-click to open the Virtual Runtime application



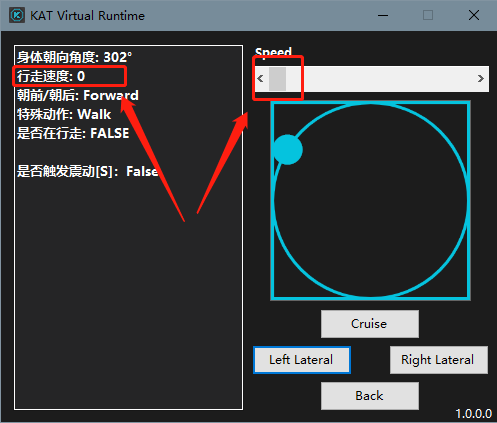
The display screen is as follows



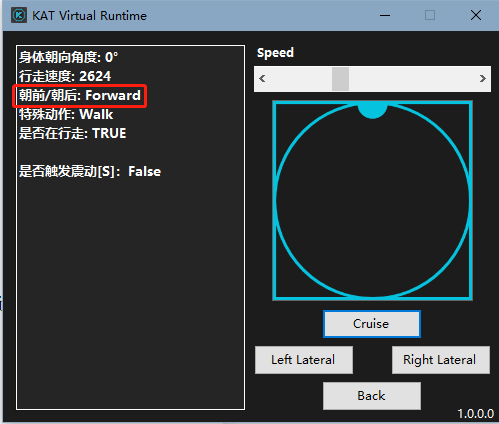
The red boxed area shows the parameters of Virtual Runtime, which need to be adjusted by Speed and the angle ball below and a few special actions in the blue box.



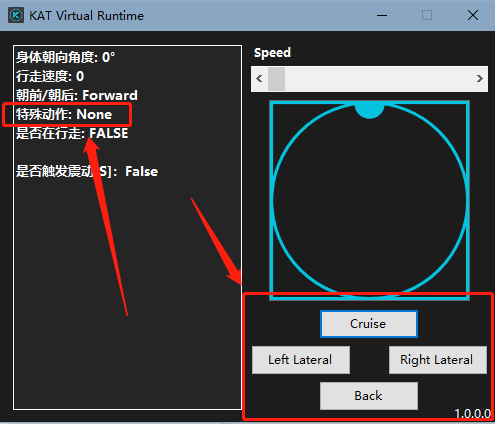
The value in the red box with the red arrow pointing in the first row is Yaw, which represents the body facing angle, or direction of travel. The angle Yaw is adjusted by the blue ball in the red box on the right, with 0° representing forward (the default direction in the scene), 90° to the right, 180° to the back, and 270° to the left. The angles and so on are freely adjustable.



The value in the red box with the red arrow pointing in the second row is Speed, which represents the walking speed. Speed is adjusted by the grey slider in the red box on the right, the higher the value the faster the speed.

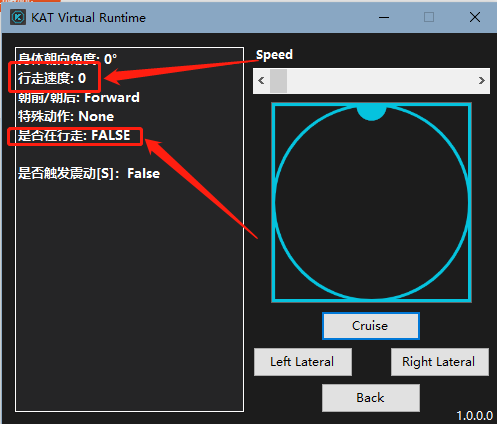


The red box in the third row represents the current walking state (forward/backward), 朝前 is forward and 朝后 is backward, click “back” to toggle between these two states.

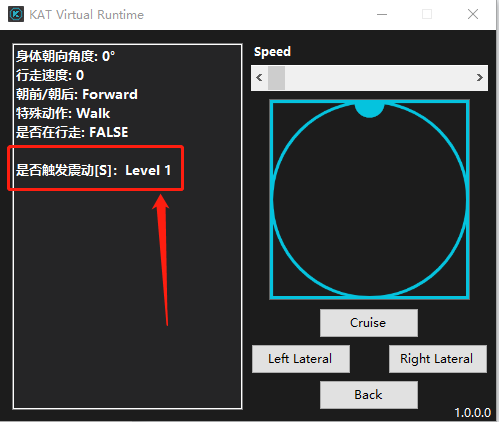
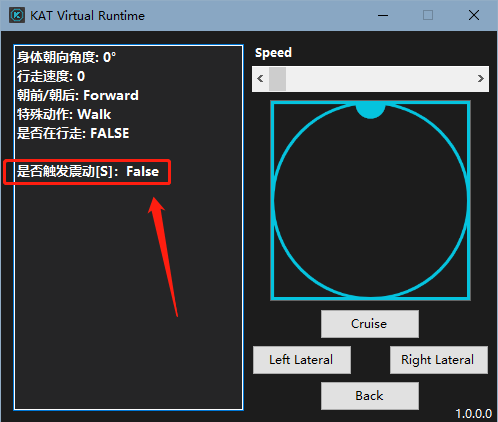


The fourth row represents the current state of the special action, with four states, each toggled by the four buttons in the red box on the right.

1. Cruise: cruising, this state moves towards the current body direction at a walking speed of 3000.
2. Left/Right Lateral: left/right panning, this state moves sideways at 2500 towards the current body direction of 90° counterclockwise/clockwise.
3. Back: Backward, this state will back up at 2000 in the opposite direction of the current body facing.



The fifth line shows whether it is walking or not, with two states TRUE and FALSE, indicating whether it is currently moving or not, expressed as FALSE when the walking speed is 0, and TRUE when the walking speed is not 0.



The sixth line shows whether the current vibration is triggered and the vibration level, when no vibration-related plug-ins are referenced it shows False, when vibration plug-ins are referenced it shows the vibration level, up to 5 (the call to debug will show a vibration level greater than 5, but the actual vibration level is not greater than 5 ).