Computer Games Exercises: 2024s s02 (non-physics)

https://www.umm.uni-heidelberg.de/miism/

Contents

Answer header

Please put the author information in the header of all code files.

- `name (Name)`
- `coauthor list`

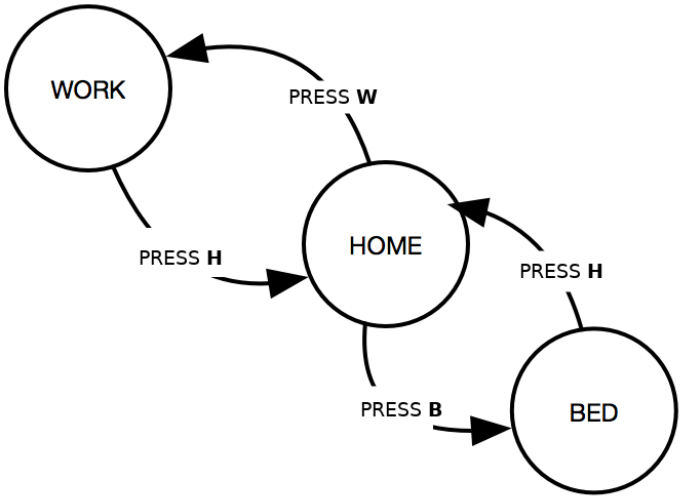
G03b: State machine

Preparation

Read the documentation further under *Tutorials*:

- *Inputs - Using InputEvent*
- *2D - Custom drawing in 2D*

Task



Implement the state machine shown above in an application.

- Create a scene with a panel and an attached label. Create a script for the panel.
- Add three actions for the keys `"H"`, `"W"`, and `"B"` and assign the corresponding keys (NOT physical keys) to them. In the function `_process()`, implement the state transitions corresponding to the figure above. Use a variable to store the current state and analyze the event `Input.is_action_pressed()`. Change the current state and change the text of the label to show the current state to the user.
- Visualize the different states with colors. In the function `_draw()`, change the color of the panel according to the current state. Use
  - green for the state `BED`
  - blue for the state `HOME`
  - red for the state `WORK`

  Do not forget to call `queue_redraw()` in each frame!

HINT: To create a state dictionary, add the following line:

```
const STATES = {"BED":0, "HOME":1, "WORK":2}
```

Questions

Write the corresponding answers in the script file.

- How do you come to this implementation?
- What could be further improved for this implementation when there are more than 100 states?