# MODEL DRIVEN ENGINEERING

Simulation-based performance evaluation of PacMan NPC's

Can Babek Ilgaz

Universiteit Antwerpen

## About the game

## INTRODUCTION

## SPECIFYING NPC BEHAVIOUR

Universiteit Antwerpen

## About the game

- PacMan
  - PacMan dies if touched
  - PacMan eats apples and score gets incremented
  - PacMan can not move to a tile occupied by an obstacle
- PacMan eat apple, Ghosts chase

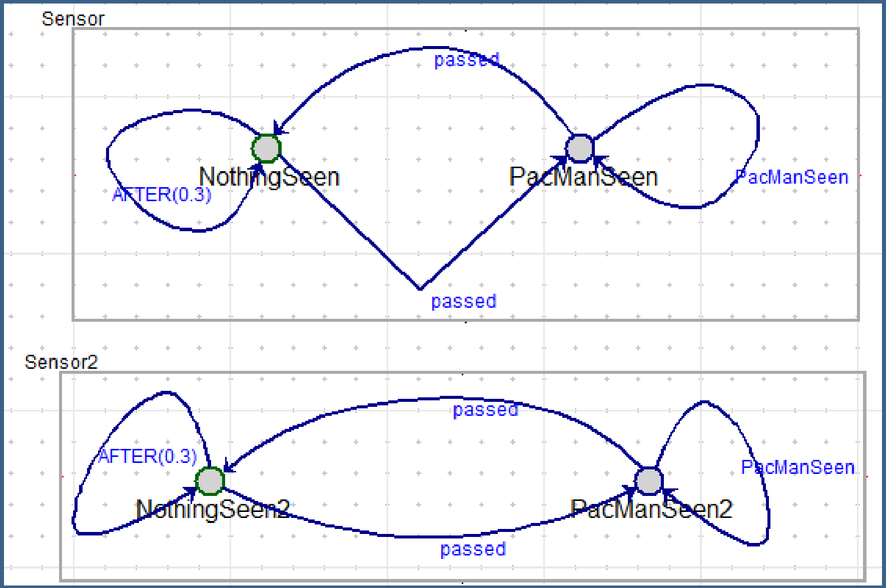
## Statecharts

- Based on [1]
- Allows for modular statecharts
  - Keeps them interchangeable
  - Reusable
  - Simple
- Statecharts made in atom3

[1] - Programmed Graph Rewriting with Time for Simulation-Based Design – Eugene Syriani and Hans Vangheluwe

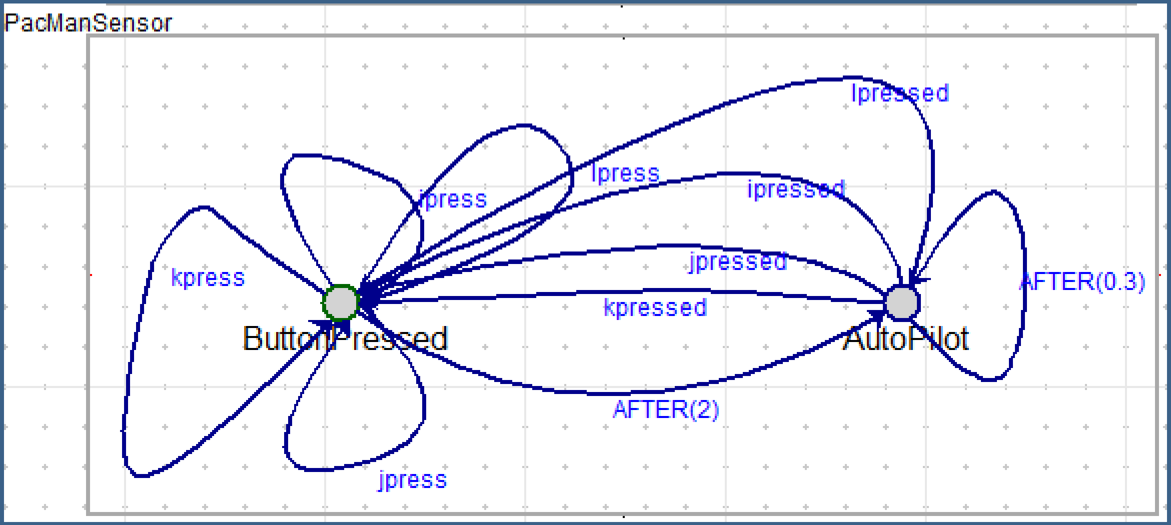
## Sensor

- Eyes
- Same for Ghosts
- Event generated if PacMan sighted



## PacMan

- Controlled by User
- If in 2 seconds no key pressed moves randomly



## Putting it all together

- All orthogonal components
  - Put together in one composite component
- Allows behaviour like:
  - After 2 seconds, move random tile.
    - If PacMan seen start chasing
  - If key pressed, go back to a different state

## USING THE STATECHARTS

## Python code

- 3 different entity
  - Updated every pass of main loop (±30 ms)
    - Game hangs if not loop finished

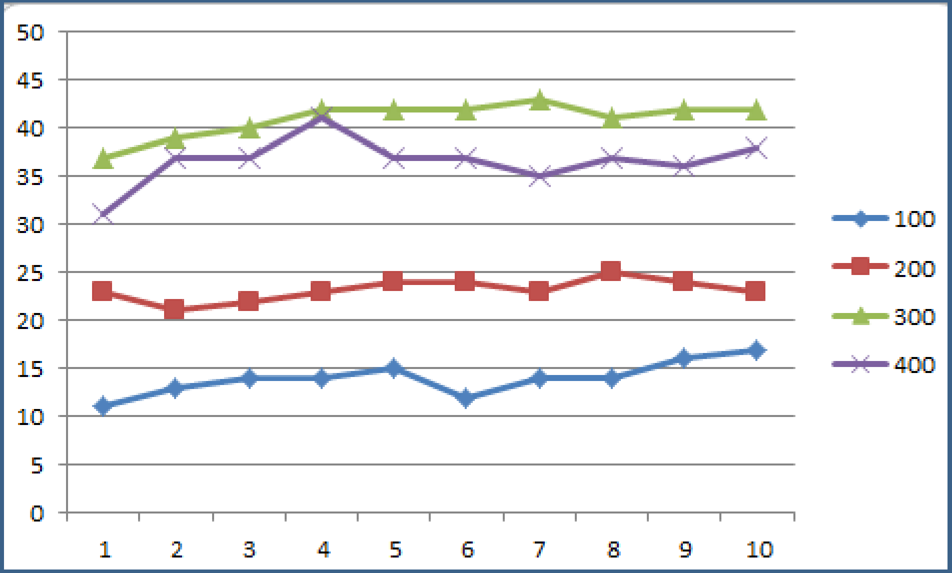
## Timing issues?

- Statechat can “miss” events
  - For example: seeing PacMan
  - Normally: If reached, next loop detected
  - But: Takes 2 or 3 loops
    - Ghosts keeps moving but on wrong state

## Solution?

- Correct the deviation
- Generate event in code  Just take it into account when designing
    - Avoid the problem!

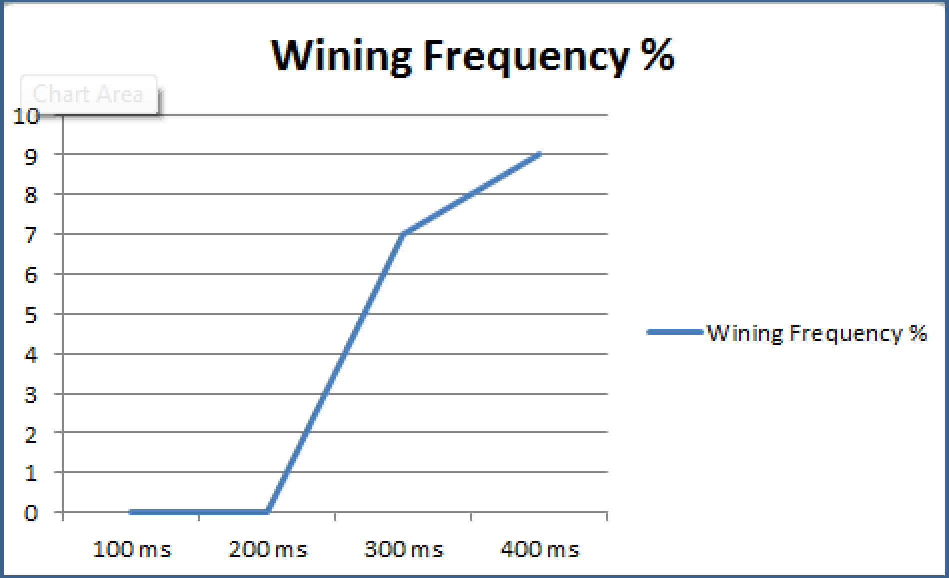
## Simulation



y - Game lasting time

x - Playing count

## Simulation



y - Playing count
x - Ghosts’ reaction time

## CONCLUSION

## Pros

- Easy to develop
  - Can be used by non-programmers!
- Higher abstraction
  - No knowledge of specific algorithms needed
- Easy to adapt
- Reusable
- Easy to represent complex structures and interactions
- Easy to read

## Cons

- Sometimes after a few steps game crashed
- Simulation with the current code generation simulation is not so obvious

## Do what when?

- Where do statecharts stop and does code begin?
- Made the mistake: coding first, statecharts later
  - Other way around!

## Conclusion

- Very usable for this type of problem
  - Some mistakes but can be solved

How we are going to beat Ghosts

## DEMO

Thank you for your attention.

# QUESTIONS?

Universiteit Antwerpen