Framsticks simulation

Szymon Ulatowski

www.framsticks.com

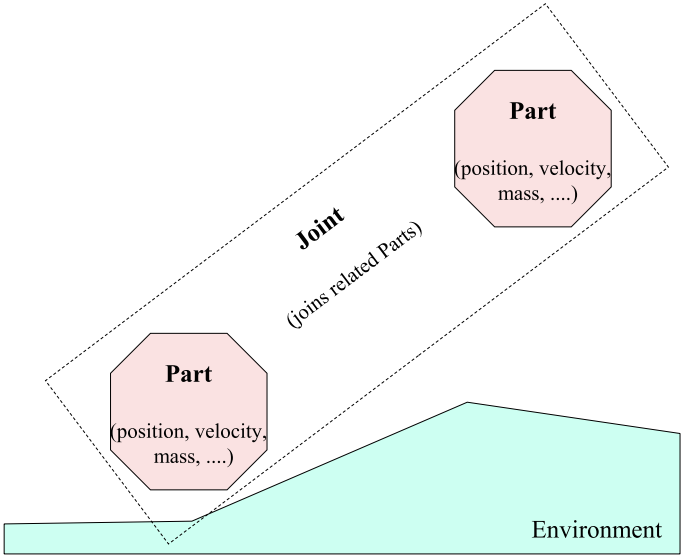
Simulation goals

- Physics-based: create real-world feeling to intuitively understand behaviors
- Not necessarily very accurate, but fast – performance matters

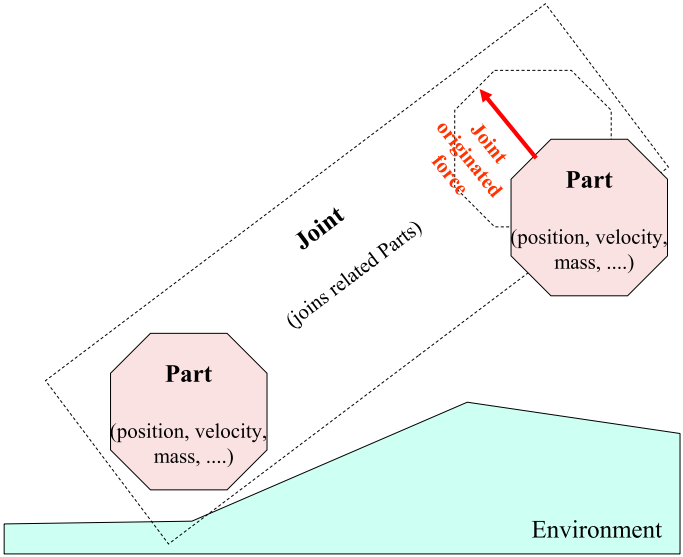
Building blocks

- "Parts" (atomic physical objects)
- "Joints" (description of internal forces and constraints, visualized as sticks)
- Environment (static objects, water)

Building blocks

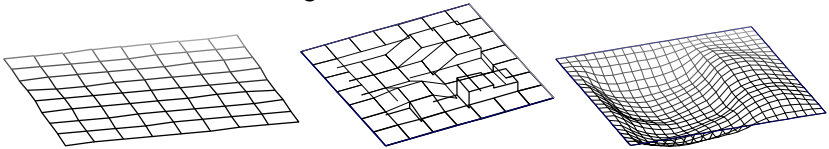


Building blocks



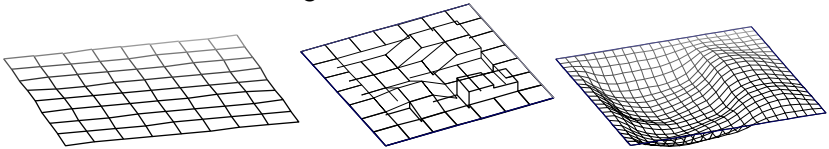
Environment

- Ground: flat, blocks, or heightfield

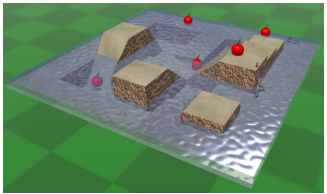


Environment

- Ground: flat, blocks, or heightfield

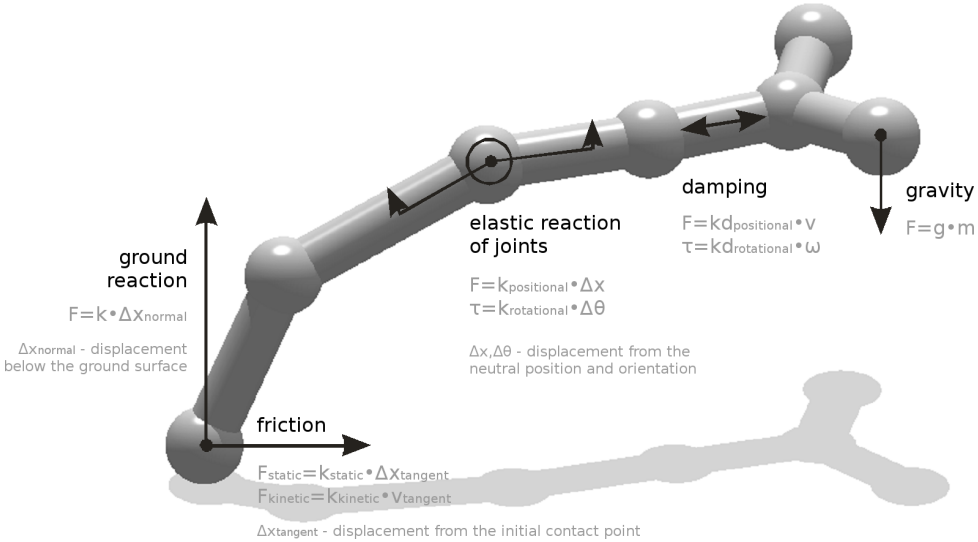


- Water:
  - Buoyant force (effectively "cancels" gravity for creatures)
  - Resistance depending on the orientation (creatures can push themselves forward)

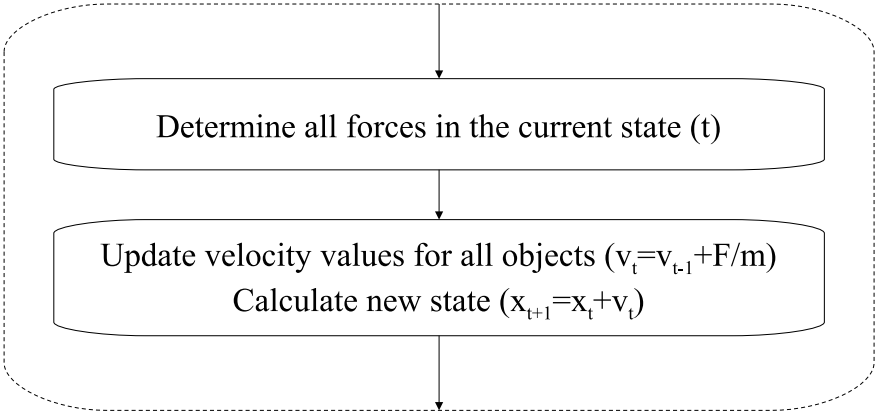


- Dynamic environment: not directly, can be made of other simulator objects (interactions handled by the experiment script)

Forces



Simulation step



Collisions

Detection:

- Part⇔Environment (including ground and water)
- Part⇔Part (between different objects)

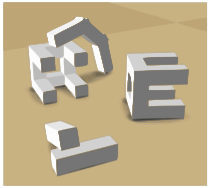
Effects:

- Physical: controlled directly by the simulator
- User-defined: can be handled by the experiment script

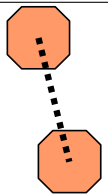
MechaStick vs. *ODE* (Open Dynamics Engine, www.ode.org)

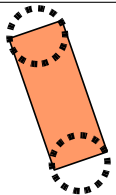
ODE:

- Much more realistic
- True solid bodies with accurate collisions
- Rigid stick connections
- Slower



	MechaStick	ODE
Part	physical point	imaginary point
Joint (stick)	imaginary line	physical body



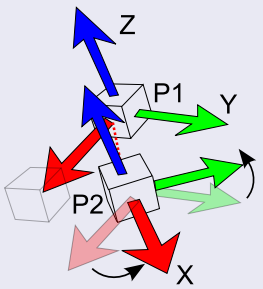


Muscles

Joint total rotation (Tx,Ty,Tz) =
joint rotation (rx,ry,rz) + muscle rotation (mx,my,mz) · Signal

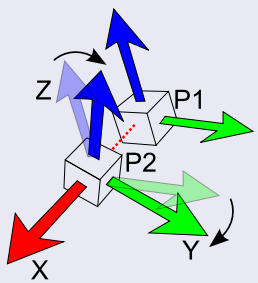
Bending

(mx,my,mz) = (0,0,180°·*Range*)



Rotating

(mx, my, mz) = (180°, 0, 0)



Can do a full 360° rotation for the input signal $-1 \,..\, +1$

Creature energy balance