Framsticks simulation

- Simulation goals
- Building blocks
- Environment
- Forces
- Simulation step
- Collisions
- *MechaStick* vs. *ODE*
- Muscles
- Energy balance

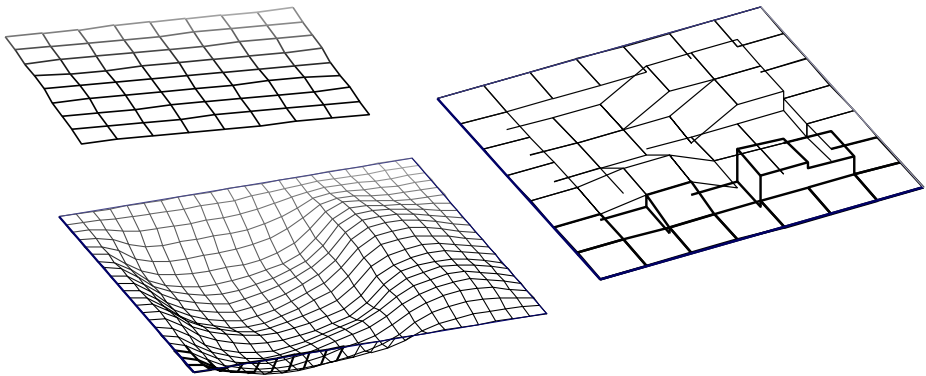- Physics-based: create real-world feeling to intuitively understand behaviors
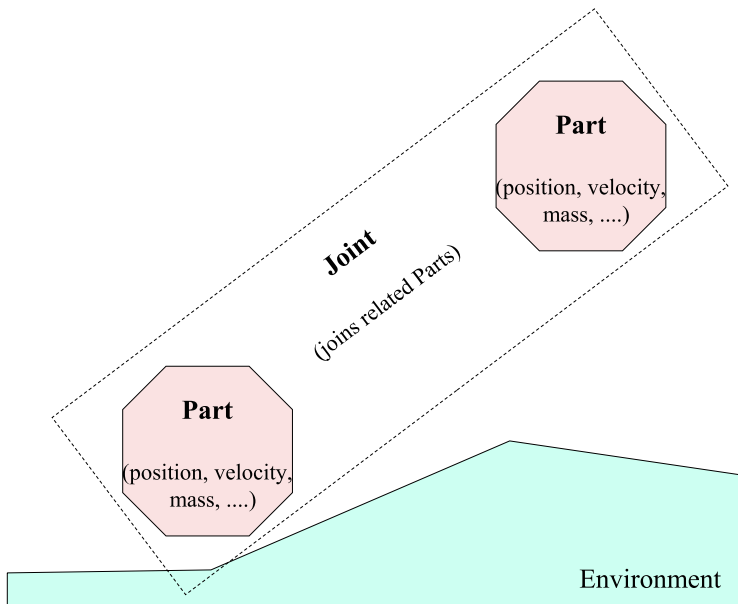- Not necessarily very accurate but fast — performance matters

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

- Ground: flat, blocks or heightfield
- Water:
  - Buoyant force (effectively "cancels" gravity for creatures)
  - Resistance depending on the orientation (creatures can push themselves forward)
- Complex, dynamic environment: not directly, can be made of other simulator objects (interactions handled by the experiment script)
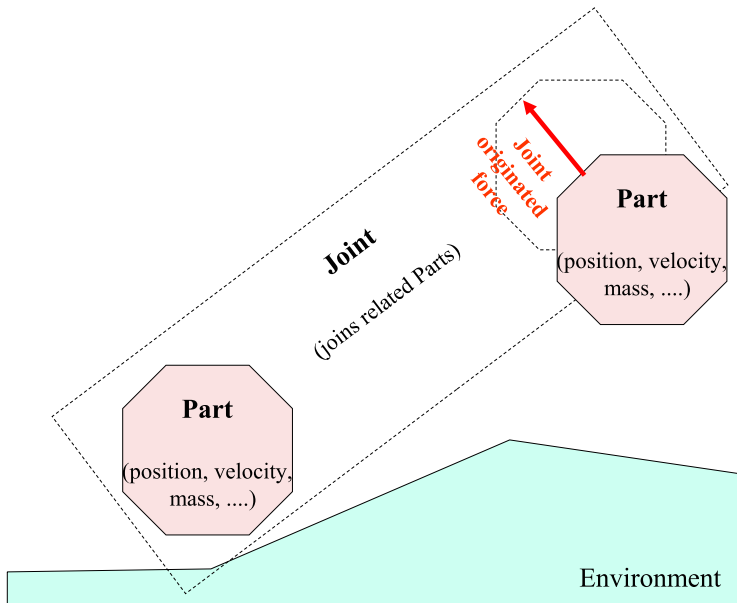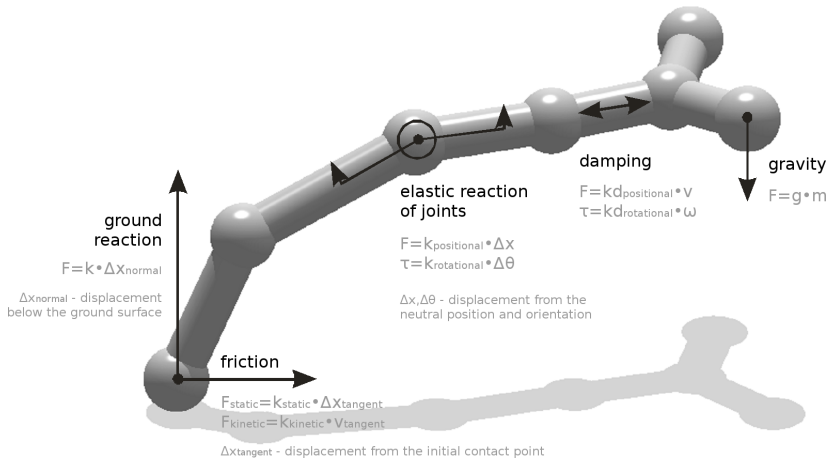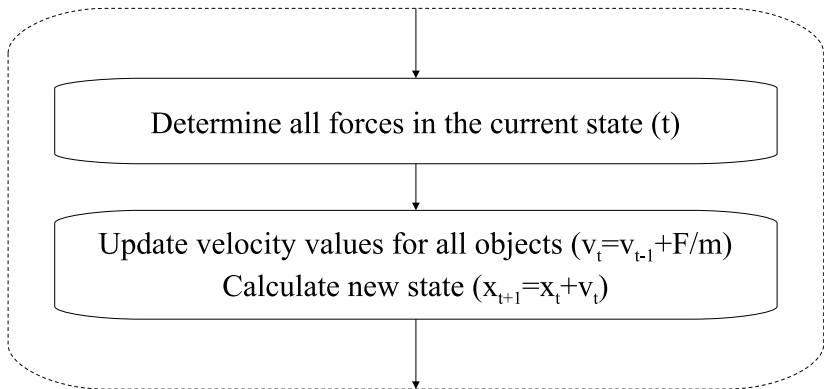
Part

(position, velocity, mass, ...)

Joint

(joins related Parts)

Part

(position, velocity, mass, ...)

Environment

ground
reaction

$F = k \cdot \Delta x_{normal}$

Δx_{normal} - displacement
below the ground surface

elastic reaction
of joints

$F = k_{positional} \cdot \Delta x$
$\tau = k_{rotational} \cdot \Delta \theta$

$\Delta x, \Delta \theta$ - displacement from the
neutral position and orientation

damping

$F = kd_{positional} \cdot v$
$\tau = kd_{rotational} \cdot \omega$

gravity

$F = g \cdot m$

friction

$F_{static} = k_{static} \cdot \Delta x_{tangent}$
$F_{kinetic} = k_{kinetic} \cdot v_{tangent}$

$\Delta x_{tangent}$ - displacement from the initial contact point

Determine all forces in the current state (t)

Update velocity values for all objects ($v_t = v_{t-1} + F/m$)
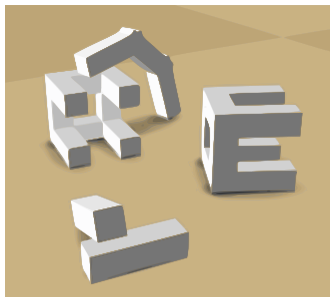Calculate new state ($x_{t+1} = x_t + v_t$)

Detection:

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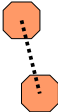- Part⇔Part (between different objects)

Effects:

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

Differences:

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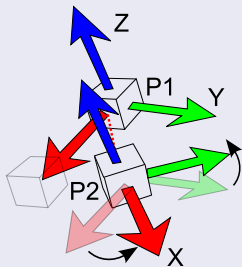- True solid bodies with accurate collisions
- Rigid stick connections

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

Muscles

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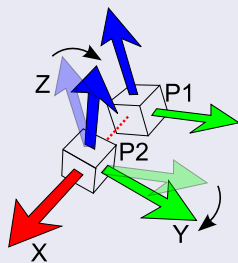 joint rotation (rx,ry,rz) + muscle rotation (mx,my,mz) · Signal

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

Scripts (step, collisions)

Muscle work (em_stat, em_dyn)

Assimilation (en_assim)

Idle power consumption (idleen)

Calculating
energy_p
and
energy_m

$$balance = energy\_p - energy\_m$$
$$energy := energy + balance$$

No

$energy < 0$
and Death enabled

Yes

Kill