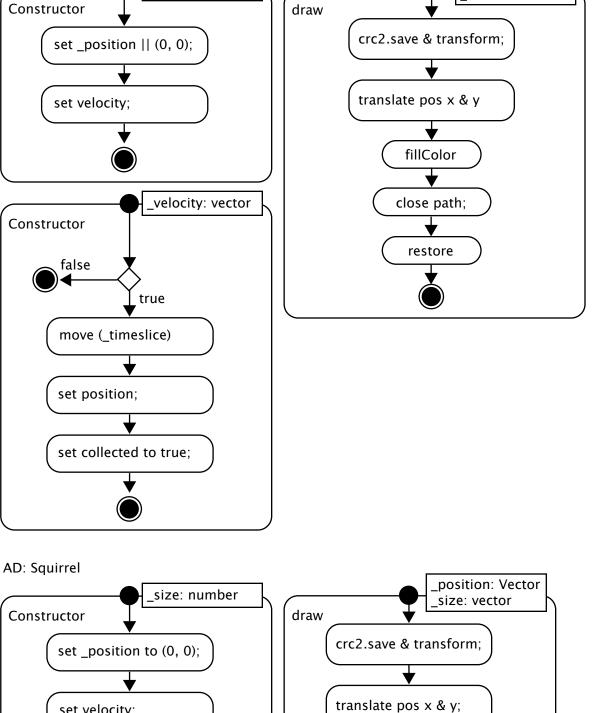
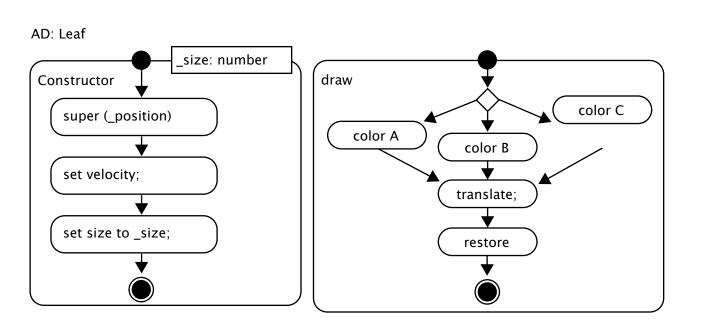
## 10.2 - Goldener Herbst Moveable Vector CanvasRenderingContext; position: Vector x: number velocity: Vector y: number constructor(\_position, \_velocity) Leaf constructor (\_x: number, \_y: number) move(\_timeslice) type: number set (\_x: number, \_y: number): void size: number scale (\_factor: number): void constructor add (\_added: number): void Nut drawLeaves(): void { random (\_minLength: number, eaten: boolean maxLength: number): void Squirrel constructor collect: boolean drawNuts(): void { size: nzmber constructor drawSquirrel(): void { AD: Moveable position: Vector timeslice: number Constructor move add velocity to \_timeslice and \_position set \_position to (0, 0); set velocity random; add canvas context remove canvas context AD: Main click moveable: Moveable [] load nSquirrel: Squirrels [] nNut: Nuts [ ] install load listener; ) handleLoad/ placeNut update handleLoad draw Mountainget crc2; add Colorstop; drawBackground(); size: number drawTree(); update getlmage; drawMountain(); move Moveables; drawLeaf(); draw; push.leaves/squirrel in Movable \_position: Vector \_treePosition: Array placeNut treeRadius: Array drawTree createNut crc2.save & transform; push.nuts; translate pos x & y collected = false false = true fillColor velocity = pos.nut velocity = pos.squirrel close path; set squirrel collected = false restore set nut = eaten = true create nut from canvas set Squirrel velocity; AD: Nuts \_position: Vector position: Vector size: vector Constructor draw crc2.save & transform; set \_position || (0, 0); translate pos x & y set velocity; fillColor \_velocity: vector close path; Constructor



set velocity;

set collected to false;

set size;



draw;

restore