Arbeitsübersicht Softwareprojekt - Ecosimulator

Gruppenmitglieder

- Imojen Mason
- Stefan Warm

Liste der arbeitsteiligen Zuordnungen

Alle Packete/Datein befinden sich im Verzeichnis ./EcoSim/

Pakets cats	**bearbeitet von**
`ToAnimals(cats *[]Cat) *[]entity.Animal`	Stefan
`New(w *world.World) *data`	Stefan
`(a *data) Update(others *[]Cat, preys1 *[]rabbits.Rabbit, preys2 *[]grass.Grass) (offSpring *data)`	Stefan
`(a *data) GetOffspring() *data`	Stefan

Pakets config	**bearbeitet von**
`const`	Stefan

Pakets entity	**bearbeitet von**
`init()`	Imojen
`rand_ab(a, b int) float64`	Stefan
`NewDrawable(w *world.World) DrawableData`	Stefan
`(a *DrawableData) SetImageFromFile(file string, size, x, y int)`	Imojen
`(a *DrawableData) makeAnimal()`	Imojen
`(a *DrawableData) SetColorRGB(r, g, b uint8)`	Stefan
`(a *DrawableData) GetPosition() vec`	Stefan
`NewMoveable(w *world.World) MoveableData`	Stefan
`(a *MoveableData) randomStep()`	Stefan
`NewHealth() HealthData`	Stefan
`(a *HealthData) IsAlive() bool`	Stefan
`(a *HealthData) SetLifeSpan(ls int)`	Stefan
`a *HealthData) SetHealthLoss(e float64)`	Stefan

`(a *HealthData) SetHealth(e float64)`	Stefan
`(a *HealthData) GetHealth() float64`	Stefan
`(a *HealthData) SetHealthWhenEaten(e float64)`	Stefan
`(a *HealthData) GetHealthWhenEaten() float64`	Stefan
`(a *HealthData) SetMatureAge(mAge int)`	Stefan
`(a *HealthData) IncAge()`	Stefan
`New(w *world.World) *AnimalData`	Imojen
`(a *AnimalData) IsSame(b *AnimalData) bool`	Imojen
`(a *AnimalData) SetMoveable(m bool)`	Stefan
`(a *AnimalData) SetViewAngle(ang float64)`	Stefan
`(a *AnimalData) SetMaxVel(v float64)`	Stefan
`(a *AnimalData) SetViewMag(mag float64)`	Stefan
`(a *AnimalData) GetAge() int`	Stefan
`(a *AnimalData) GetDateOfLastBirth() int`	Stefan
`(a *AnimalData) SetDateOfLastBirth(d int)`	Stefan
`(a *AnimalData) GetMatureAge() int`	Stefan
`(a *AnimalData) GetWorld() *world.World`	Stefan
`(a *AnimalData) ApplyMove(others *[]Animal, preys *[]Animal)`	Stefan
`(a *AnimalData) avoidCollisionWithSeenObjects(others *[]Animal)`	Stefan
`(a *AnimalData) searchFood(others *[]Animal)`	Stefan
`(a *AnimalData) eatFood(others *[]Animal)`	Stefan
`(a *AnimalData) repelFromWater()`	Stefan
`(a *AnimalData) SeeOthers(others *[]Animal) (*[]Animal, *[]vec)`	Stefan
`(a *AnimalData) Draw(screen *ebiten.Image)`	Stefan
`(a *AnimalData) drawStats(screen *ebiten.Image)`	Stefan
`(a *AnimalData) drawView() *ebiten.DrawImageOptions`	Stefan
`(a *AnimalData) drawAnimal(screen *ebiten.Image)`	Stefan
`(a *AnimalData) makeArc(img *ebiten.Image, radius float32, startAngle, endAngle float32, c color.NRGBA, line bool)`	Stefan

Paket foxes	**bearbeitet von**
`ToAnimals(foxes *[]Fox) *[]entity.Animal`	Stefan
`New(w *world.World) *data`	Stefan
`(a *data) Update(others *[]Fox, preys *[]rabbits.Rabbit) (offSpring *data)`	Stefan
`(a *data) GetOffspring() *data`	Stefan

Paket graphics	**bearbeitet von**
`init()`	Imojen
`New(x,y float64) *data`	Imojen
`(u *data) drawLine(dst *ebiten.Image, history []int, c color.RGBA)`	Imojen
`drawText(dst *ebiten.Image, x,y float64, c color.RGBA, str string)`	Imojen
`(u *data) Draw(dst *ebiten.Image)`	Imojen
`(u *data) Update(nG,nR,nC,nF int)`	Imojen

Paket grass	**bearbeitet von**
`ToAnimals(gras *[]Grass) *[]entity.Animal`	Stefan
`New(w *world.World) *data`	Stefan

Paket rabbits	**bearbeitet von**
`ToAnimals(rabbits *[]Rabbit) *[]entity.Animal`	Stefan
`New(w *world.World) *data`	Imojen
`(a *data) Update(others *[]Rabbit, food *[]grass.Grass) (offSpring *data)`	Stefan
`(a *data) GetOffspring() *dat`	Stefan

Paket ui	**bearbeitet von**
`init()`	Imojen
`(c *CheckBox) Update()`	Imojen
`drawText(dst *ebiten.Image, x, y float64, size float64, str string)`	Imojen
`(c *CheckBox) Draw(dst *ebiten.Image)`	Imojen
`(s *Slider) Update()`	Imojen
`(s *Slider) Draw(dst *ebiten.Image)`	Imojen
`New() *data`	Imojen
`(u *data) GetNumberOfGrass() int`	Imojen
`(u *data) GetNumberOfBunnies() int`	Imojen
`(u *data) GetNumberOfCats() int`	Imojen

Paket ui	**bearbeitet von**
`(u *data) GetNumberOfFoxes() int`	Imojen
`(u *data) Draw(dst *ebiten.Image)`	Imojen
`(u *data) Update()`	Imojen

Paket world	**bearbeitet von**
`init()`	Stefan
`New(width float32, height float32, scale float32, img *ebiten.Image) *data`	Imojen
`(wo *data) ToggleGrid()`	Stefan
`(wo *data) ToggleDebug()`	Stefan
`(wo *data) ToggleStatistics()`	Stefan
`(wo *data) GetShowStats() bool`	Stefan
`(wo *data) GetDebug() bool`	Stefan
`(wo *data) GetXYTile(x, y int) (tileX, tileY int)`	Stefan
`(wo *data) GetTileSizeScaled() int`	Stefan
`(wo *data) GetTileBorders(x, y int) (n, no, o, so, s, sw, w, nw bool)`	Stefan
`(wo *data) GetTileDstToWater(x, y int) (n, s, e, w int)`	Stefan
`(wo *data) IsLand(x, y int) bool`	Stefan
`(wo *data) Width() float32`	Stefan
`(wo *data) Height() float32`	Stefan
`(wo *data) Margin() float32`	Stefan
`(wo *data) ToggleGround(mx, my int)`	Stefan
`(wo *data) Draw(dst *ebiten.Image, c int)`	Stefan
`(wo *data) getTileNumber(tileX, tileY int) (int, bool)`	Stefan
`(wo *data) areNeighborsGround(tileX, tileY int, layer []int) (n, no, o, so, s, sw, w, nw bool)`	Stefan
`getState(a, b, c, d bool) int`	Stefan
`boolToInt(bit bool) int`	Stefan
`(wo *data) setTileInLayer(x, y int, l []int, value int)`	Stefan
`(wo *data) getTileFromLayer(x, y int, l []int) int`	Stefan
`(wo *data) toggle(tileX, tileY int)`	Stefan

Funktion main.go	**bearbeitet von**
`init()`	Imojen
`(b *Button) Update()`	Imojen
`(b *Button) Draw(dst *ebiten.Image)`	Imojen
`(b *Button) SetOnPressed(f func(b *Button))`	Imojen
`(g *Game) Update() error`	Imojen
`drawGame(g *Game, dst *ebiten.Image)`	Imojen
`(g *Game) Draw(dst *ebiten.Image)`	Imojen
`(g *Game) Layout(outsideWidth, outsideHeight int) (int, int)`	Imojen
`resetGrass()`	Imojen
`resetBunnies()`	Imojen
`resetCats()`	Imojen
`resetFoxes()`	Imojen
`main()`	Imojen