ENTWURFSDOKUMENT

(V. 1.0)

KNOT^3

PSE WS 2013/14

Auftraggeber:
Karlsruher Institut für Technologie
Institut für Betriebs- und Dialogsysteme
Prof. Dr.-Ing. C. Dachsbacher

Betreuer:
Dipl.-Inf. Thorsten Schmidt
Dipl.-Inf. M. Retzlaff

Auftragnehmer:
Tobias Schulz, Maximilian Reuter, Pascal Knodel,
Gerd Augsburg, Christina Erler, Daniel Warzel

15. Dezember 2013

Inhaltsverzeichnis

1 Einleitung **4**

2 Aufbau **5**
2.1 Architektur . . . 5
2.2 Klassendiagramm . . . 6
2.3 Verwendete Entwurfsmuster . . . 6

3 Klassenübersicht **7**
3.1 Klassen . . . 7
3.1.1 Klasse ArrowModel . . . 7
3.1.2 Klasse ArrowModelInfo . . . 8
3.1.3 Klasse AudioSettingsScreen . . . 8
3.1.4 Klasse BooleanOptionInfo . . . 8
3.1.5 Klasse Camera . . . 9
3.1.6 Klasse CelShadingEffect . . . 10
3.1.7 Klasse Challenge . . . 11
3.1.8 Klasse ChallengeFileIO . . . 12
3.1.9 Klasse ChallengeLoadScreen . . . 13
3.1.10 Klasse ChallengeMetaData . . . 13
3.1.11 Klasse ChallengeMode . . . 14
3.1.12 Klasse CheckBoxItem . . . 15
3.1.13 Klasse Circle . . . 15
3.1.14 Klasse ColorPicker . . . 15
3.1.15 Klasse ColorPickItem . . . 16
3.1.16 Klasse ConfigFile . . . 16
3.1.17 Klasse ConfirmDialog . . . 17
3.1.18 Klasse ControlSettingsScreen . . . 17
3.1.19 Klasse CreativeLoadScreen . . . 17
3.1.20 Klasse CreativeMode . . . 18
3.1.21 Klasse CreditsScreen . . . 18
3.1.22 Klasse Dialog . . . 19
3.1.23 Klasse DistinctOptionInfo . . . 19
3.1.24 Klasse DrawableGameScreenComponent . . . 20
3.1.25 Klasse DropDownEntry . . . 20
3.1.26 Klasse DropDownMenuItem . . . 21
3.1.27 Klasse Edge . . . 21
3.1.28 Klasse FadeEffect . . . 22
3.1.29 Klasse FileIO . . . 22
3.1.30 Klasse Game . . . 23

3.1.31 Klasse GameModel . . . 23
3.1.32 Klasse GameModelInfo . . . 24
3.1.33 Klasse GameObjectInfo . . . 24
3.1.34 Klasse GameScreen . . . 25
3.1.35 Klasse GameScreenComponent . . . 26
3.1.36 Klasse GraphicsSettingsScreen . . . 26
3.1.37 Klasse Input . . . 27
3.1.38 Klasse InputItem . . . 28
3.1.39 Klasse KeyInputItem . . . 28
3.1.40 Klasse Knot . . . 28
3.1.41 Klasse Knot3Game . . . 30
3.1.42 Klasse KnotFileIO . . . 31
3.1.43 Klasse KnotMetaData . . . 32
3.1.44 Klasse KnotStringIO . . . 32
3.1.45 Klasse Localizer . . . 33
3.1.46 Klasse Menu . . . 33
3.1.47 Klasse MenuButton . . . 34
3.1.48 Klasse MenuItem . . . 34
3.1.49 Klasse MenuScreen . . . 35
3.1.50 Klasse ModelFactory . . . 35
3.1.51 Klasse ModelkeyHandler . . . 36
3.1.52 Klasse ModelMouseHandler . . . 36
3.1.53 Klasse ModelRenderer . . . 36
3.1.54 Klasse MousePointer . . . 38
3.1.55 Klasse NodeMap . . . 38
3.1.56 Klasse NodeModel . . . 38
3.1.57 Klasse NodeModelInfo . . . 39
3.1.58 Klasse OptionInfo . . . 39
3.1.59 Klasse Options . . . 40
3.1.60 Klasse PauseDialog . . . 40
3.1.61 Klasse PipeModel . . . 40
3.1.62 Klasse PipeModelInfo . . . 41
3.1.63 Klasse PipeMovement . . . 41
3.1.64 Klasse PrinterIO . . . 42
3.1.65 Klasse ProfileSettingsScreen . . . 43
3.1.66 Klasse RenderEffect . . . 43
3.1.67 Klasse RenderEffectStack . . . 44
3.1.68 Klasse SettingsScreen . . . 45
3.1.69 Klasse ShadowGameModel . . . 45
3.1.70 Klasse ShadowGameObject . . . 46
3.1.71 Klasse SliderItem . . . 47
3.1.72 Klasse StandardEffect . . . 47
3.1.73 Klasse T . . . 48
3.1.74 Klasse TextInputDialog . . . 48
3.1.75 Klasse TutorialChallengeMode . . . 49
3.1.76 Klasse VerticalMenu . . . 49
3.1.77 Klasse Widget . . . 49
3.1.78 Klasse WidgetKeyHandler . . . 50
3.1.79 Klasse WidgetMouseHandler . . . 50
3.1.80 Klasse World . . . 51
3.1.81 Klasse XNA.DrawableGameComponent . . . 51

3.1.82 Klasse XNA.Game . . . 52
3.1.83 Klasse XNA.GameComponent . . . 52
3.2 Schnittstellen . . . 53
3.2.1 Schnittstelle IChallengeIO . . . 53
3.2.2 Schnittstelle ICloneable . . . 53
3.2.3 Schnittstelle IEnumerable . . . 54
3.2.4 Schnittstelle IEnumerable1 . . . 54
3.2.5 Schnittstelle IEquatable . . . 54
3.2.6 Schnittstelle IEquatable1 . . . 54
3.2.7 Schnittstelle IGameObject . . . 55
3.2.8 Schnittstelle IGameStateComponent . . . 55
3.2.9 Schnittstelle IJunction . . . 56
3.2.10 Schnittstelle IKeyEventListener . . . 56
3.2.11 Schnittstelle IKnotIO . . . 57
3.2.12 Schnittstelle IMouseEventListener . . . 57
3.2.13 Schnittstelle XNA.IGameComponent . . . 58
3.3 Enumerationen . . . 62

4 Abläufe **63**
4.1 Sequenzdiagramme . . . 63

5 Klassenindex **64**

6 Anmerkungen **65**

7 Glossar **66**
7.1 Fachausdrücke . . . 66
7.2 Abkürzungen . . . 67

Kapitel 1

Einleitung

Das Knobel- und Konstruktionsspiel Knot[3], welches im Auftrag des IBDS Dachsbacher ausgearbeitet und wie im Pflichtenheft spezifiziert angefertigt wird.

Kapitel 2

Aufbau

2.1 Architektur

Die grundlegende Architektur des Spiels basiert auf der Spielkomponenten-Infrastruktur des XNA-Frameworks, die mit Spielzuständen kombiniert wird. Die abstrakten Klassen GameStateComponent und DrawableGameStateComponent erben von den von XNA bereitgestellten Klassen GameComponent und DrawableGameComponent implementieren zusätzlich die Schnittstelle IGameStateComponent. Sie unterscheiden sich von den XNA-Basisklassen dadurch, dass sie immer eine Referenz auf einen bestimmten Spielzustand halten und nur in Kombination mit diesem zu verwenden sind.

Die Spielzustände erben von der abstrakten Basisklasse GameScreen und halten eine Liste von IGameStateComponent-Objekten. Wird ein Spielzustand aktiviert, indem von einem anderen Spielzustand aus zu ihm gewechselt wird oder indem er der Startzustand ist, dann weist er seine Liste von IGameStateComponent-Objekten dem Components-Attribut der Game-Klasse zu, die von der vom XNA-Framework bereitgestellten abstrakten Klasse Game erbt. So ist zu jedem Zeitpunkt während der Laufzeit des Spiels ein Spielzustand aktiv, der die aktuelle Liste von Spielkomponenten verwaltet.

Die Spielkomponenten, die nicht gezeichnet werden und nur auf Eingaben reagieren, haben nur eine Update()-Methode und erben von GameStateComponent. Dies sind vor allem verschiedene Input-Handler, welche Tastatur- und Mauseingaben verarbeiten und beispielsweise die Kameraposition und das Kameratarget ändern oder Spielobjekte bewegen.

Spielkomponenten, die neben der Update()-Methode auch eine Draw()-Methode besitzen, erben von DrawableGameStateComponent. Dies sind vor allem die Elemente, aus denen die grafische Benutzeroberfläche zusammengesetzt ist, deren abstrakte Basisklasse Widget darstellt. [weitere Erklärungen zu Widgets...]

Alle Spielobjekte implementieren die Schnittstelle IGameObject. Die abstrakte Klasse GameModel repräsentiert dabei ein Spielobjekt, das aus einem 3D-Modell besteht, und hält zu diesem Zweck eine Referenz auf ein Objekt der Klasse Model aus dem XNA-Framework sowie weitere Eigenschaften wie Position, Drehung und Skalierung.

Spielobjekte sind keine Komponenten, sondern werden in einer Spielwelt zusammengefasst, die durch die Klasse World repräsentiert wird. Die Spielwelt ist ein DrawableGameStateComponent und ruft in ihren Update()- und Draw()-Methoden jeweils die dazugehörigen Methoden aller in ihr enthaltenen Spielobjekte auf.

Shadereffekte werden durch die abstrakte Klasse RenderEffect und die von ihr abgeleiteten Klassen gekapselt. Ein RenderEffect enthält ein Rendertarget vom Typ RenderTarget2D als Attribut und implementiert jeweils eine Begin()- und eine End-Methode. In der Methode Begin() wird das aktuell von XNA genutzte

Rendertarget auf einem Stack gesichert und das Rendertarget des Effekts wird als aktuelles Rendertarget gesetzt.

Nach dem Aufruf von Begin() werden alle Draw()-Aufrufe von XNA auf dem gesetzten Rendertarget ausgeführt. Es wird also in eine im RenderTarget2D-Objekt enthaltene Bitmap gezeichnet. Dabei wird von den Draw()-Methoden der GameModels die DrawModel(GameModel)-Methode des RenderEffects aufgerufen, der die Modelle mit bestimmten Shadereffekten in die Bitmap zeichnet.

In der End()-Methode wird schließlich das auf dem Stack gesicherte, vorher genutzte Rendertarget wiederhergestellt und das Rendertarget des RenderEffects wird, unter Umständen verändert durch Post-Processing-Effekte, auf dieses übergeordnete Rendertarget gezeichnet.

2.2 Klassendiagramm

2.3 Verwendete Entwurfsmuster

Kapitel 3

Klassenübersicht

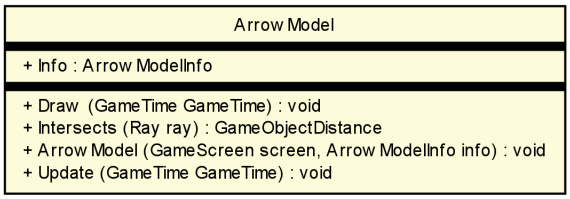
3.1 Klassen

3.1.1 Klasse ArrowModel

Beschreibung:

Diese Klasse repräsentiert ein 3D-Modell für einen Pfeil, der an selektierten Kanten erscheinen soll.

Eigenschaften:



public ArrowModelInfo Info

Das Info-Objekt, das die Position und Richtung des Pfeils enthält.

Konstruktoren:

public ArrowModel (GameScreen screen, ArrowModelInfo info)

Erstellt ein neues Pfeilmodell in dem angegebenen GameScreen mit einem bestimmten Info-Objekt, das Position und Richtung des Pfeils festlegt.

Methoden:

public void Draw (GameTime GameTime)

Zeichnet den Pfeil.

public GameObjectDistance Intersects (Ray ray)

public void Update (GameTime GameTime)

3.1.2 Klasse ArrowModelInfo

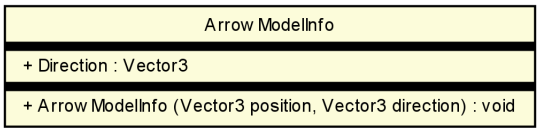
Beschreibung:

Ein Objekt dieser Klasse hält alle Informationen hält, die zur Erstellung eines Pfeil-3D-Modelles (ArrowModel) notwendig sind.

Eigenschaften:

public Vector3 Direction

Die Richtung, die die der Pfeil zeigen soll.



Konstruktoren:

public ArrowModelInfo (Vector3 position, Vector3 direction)

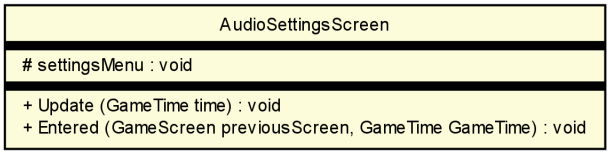
Erstellt ein neues ArrowModelInfo-Objekt an einer
bestimmten Position im 3D-Raum, das in eine bestimmte Richtung zeigt.

3.1.3 Klasse AudioSettingsScreen

Beschreibung:

Eigenschaften:

protected void settingsMenu



Methoden:

public void Update (GameTime time)

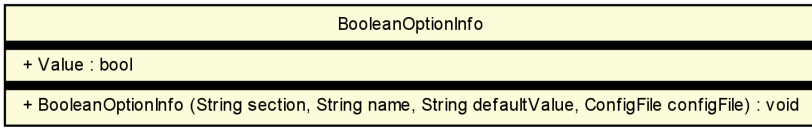
public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.4 Klasse BooleanOptionInfo

Beschreibung:

Diese Klasse repräsentiert eine Option, die die Werte Wahr oder Falsch annehmen kann.

Eigenschaften:



public bool Value

Ein Property, das den aktuell abgespeicherten Wert zurück gibt.

Konstruktoren:

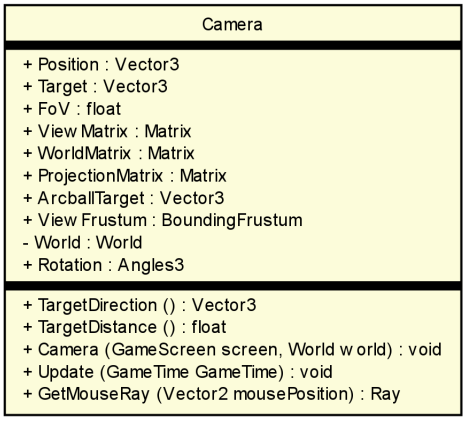
public BooleanOptionInfo (String section, String name, String defaultValue, ConfigFile configFile)

3.1.5 Klasse Camera

Beschreibung:

Jede Instanz der World-Klasse hält eine für diese Spielwelt verwendete Kamera als Attribut. Die Hauptfunktion der Kamera-Klasse ist das Berechnen der drei Matrizen, die für die Positionierung und Skalierung von 3D-Objekten in einer bestimmten Spielwelt benötigt werden, der View-, World- und Projection-Matrix. Um diese Matrizen zu berechnen, benötigt die Kamera unter Anderem Informationen über die aktuelle Kamera-Position, das aktuelle Kamera-Target und das Field of View.

Eigenschaften:



public Vector3 Position

Die Position der Kamera.

public Vector3 Target

Das Ziel der Kamera.

public float FoV

Das Field of View.

public Matrix ViewMatrix

Die View-Matrix wird über die statische Methode CreateLookAt der Klasse Matrix des XNA-Frameworks mit Matrix.CreateLookAt (Position, Target, Vector3.Up) berechnet.

public Matrix WorldMatrix

Die World-Matrix wird mit Matrix.CreateFromYawPitchRoll und den drei Rotationswinkeln berechnet.

public Matrix ProjectionMatrix

Die Projektionsmatrix wird über die statische XNA-Methode Matrix.CreatePerspectiveFieldOfView berechnet.

public Vector3 ArcballTarget

Eine Position, um die rotiert werden soll, wenn der User die rechte Maustaste gedrückt hält und die Maus bewegt.

public BoundingFrustum ViewFrustum

Berechnet ein Bounding-Frustum, das benötigt wird, um festzustellen, ob ein 3D-Objekt sich um Blickfeld des Spielers befindet.

private World World

Eine Referenz auf die Spielwelt, für die die Kamera zuständig ist.

public Angles3 Rotation

Konstruktoren:

public Camera (GameScreen screen, World world)

Methoden:

public Vector3 TargetDirection ()

public float TargetDistance ()

public void Update (GameTime GameTime)

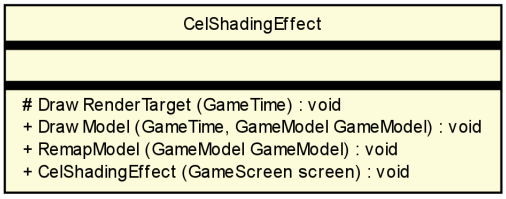
public Ray GetMouseRay (Vector2 mousePosition)

3.1.6 Klasse CelShadingEffect

Beschreibung:

Konstruktoren:

public CelShadingEffect (GameScreen screen)



Methoden:

protected void DrawRenderTarget (GameTime)

public void DrawModel (GameTime, GameModel GameModel)

public void RemapModel (GameModel GameModel)

3.1.7 Klasse Challenge

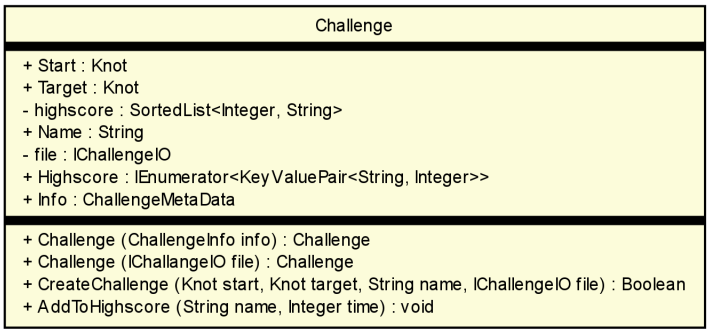
Beschreibung:

Eigenschaften:

public Knot Start

public Knot Target

private SortedList¡Integer, String¿ highscore



public String Name

private IChallengeIO file

public IEnumerator¡KeyValuePair¡String, Integer¿¿ Highscore

public ChallengeMetaData Info

Konstruktoren:

public Challenge (ChallengeInfo info)

public Challenge (IChallangeIO file)

Methoden:

public Boolean CreateChallenge (Knot start, Knot target, String name, IChallengeIO file)

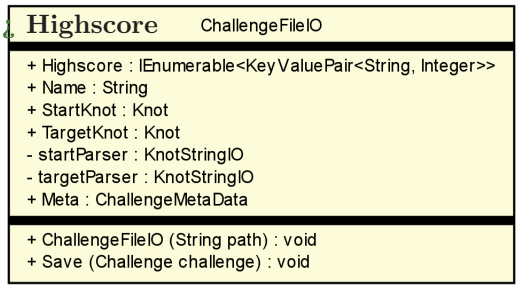
public void AddToHighscore (String name, Integer time)

3.1.8 Klasse ChallengeFileIO

Beschreibung:

Eigenschaften:

public IEnumerable¡KeyValuePair¡String, Integer¿¿



public String Name

public Knot StartKnot

public Knot TargetKnot

private KnotStringIO startParser

private KnotStringIO targetParser

public ChallengeMetaData Meta

Konstruktoren:

public ChallengeFileIO (String path)

Methoden:

public void Save (Challenge challenge)

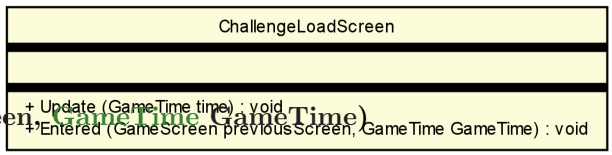
3.1.9 Klasse ChallengeLoadScreen

Beschreibung:

Methoden:

public void Update (GameTime time)

public void Entered (GameScreen previousScreen, GameTime GameTime)



3.1.10 Klasse ChallengeMetaData

Beschreibung:

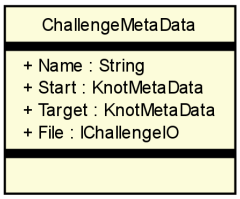
Eigenschaften:

public String Name

public KnotMetaData Start

public KnotMetaData Target

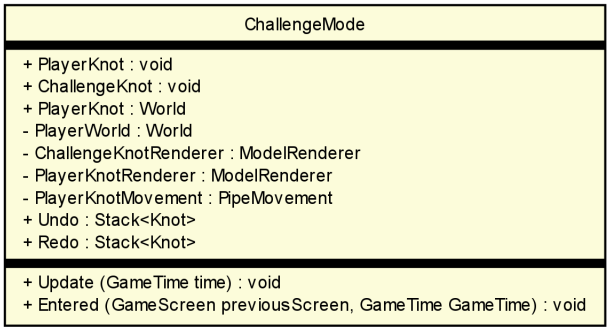
public IChallengeIO File



3.1.11 Klasse ChallengeMode

Beschreibung:

Eigenschaften:



public void PlayerKnot

public void ChallengeKnot

public World PlayerKnot

private World PlayerWorld

private ModelRenderer ChallengeKnotRenderer

private ModelRenderer PlayerKnotRenderer

private PipeMovement PlayerKnotMovement

public Stack¡Knot¿ Undo

public Stack¡Knot¿ Redo

Methoden:

public void Update (GameTime time)

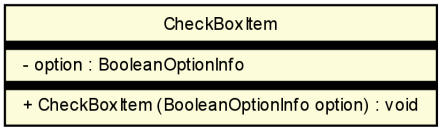
public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.12 Klasse CheckBoxItem

Beschreibung:

Eigenschaften:

private BooleanOptionInfo option



Konstruktoren:

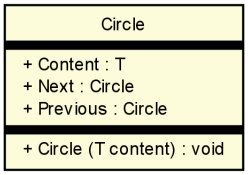
public CheckBoxItem (BooleanOptionInfo option)

3.1.13 Klasse Circle

Beschreibung:

Eigenschaften:

public T Content



public Circle Next

public Circle Previous

Konstruktoren:

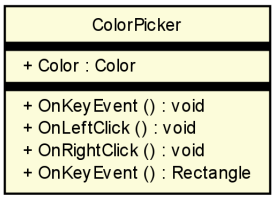
public Circle (T content)

3.1.14 Klasse ColorPicker

Beschreibung:

Eigenschaften:

public Color Color



Methoden:

public void OnKeyEvent ()

public void OnLeftClick ()

public void OnRightClick ()

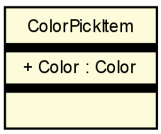
public Rectangle OnKeyEvent ()

3.1.15 Klasse ColorPickItem

Beschreibung:

Eigenschaften:

public Color Color



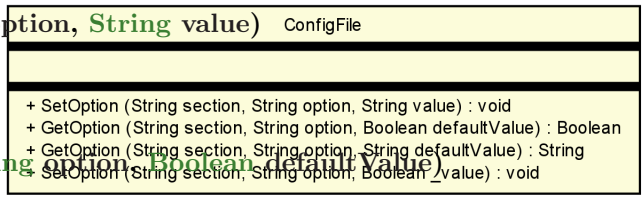
3.1.16 Klasse ConfigFile

Beschreibung:

Methoden:

public void SetOption (String section, String option, String value)

public Boolean GetOption (String section, String option, Boolean defaultValue)



public String GetOption (String section, String option, String defaultValue)

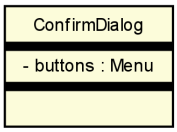
public void SetOption (String section, String option, Boolean $_value$)

3.1.17 Klasse ConfirmDialog

Beschreibung:

Eigenschaften:

private Menu buttons

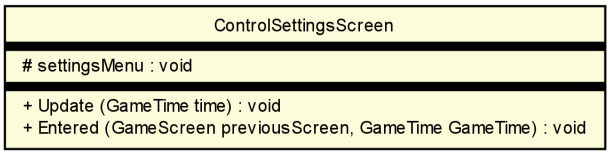


3.1.18 Klasse ControlSettingsScreen

Beschreibung:

Eigenschaften:

protected void settingsMenu



Methoden:

public void Update (GameTime time)

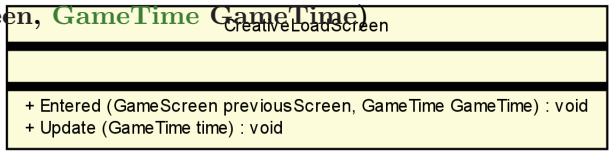
public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.19 Klasse CreativeLoadScreen

Beschreibung:

Methoden:

public void Entered (GameScreen previousScreen, GameTime GameTime)

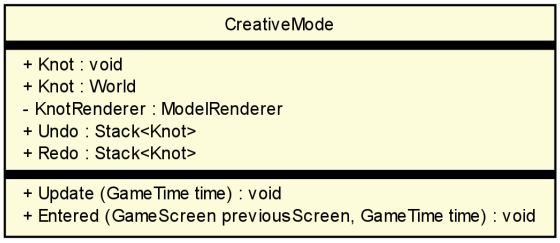


public void Update (GameTime time)

3.1.20 Klasse CreativeMode

Beschreibung:

Eigenschaften:



public void Knot

public World Knot

private ModelRenderer KnotRenderer

public Stack¡Knot¿ Undo

public Stack¡Knot¿ Redo

Methoden:

public void Update (GameTime time)

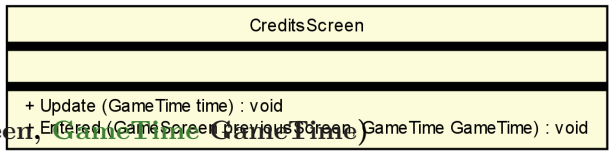
public void Entered (GameScreen previousScreen, GameTime time)

3.1.21 Klasse CreditsScreen

Beschreibung:

Methoden:

public void Update (GameTime time)

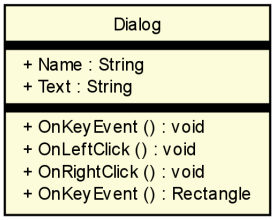


public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.22 Klasse Dialog

Beschreibung:

Eigenschaften:



public String Name

public String Text

Methoden:

public void OnKeyEvent ()

public void OnLeftClick ()

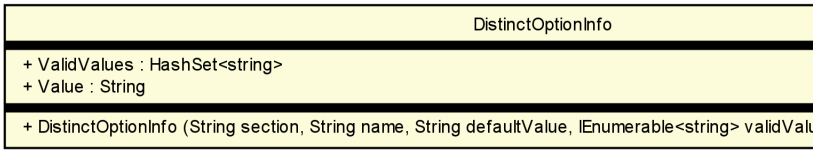
public void OnRightClick ()

public Rectangle OnKeyEvent ()

3.1.23 Klasse DistinctOptionInfo

Beschreibung:

Eigenschaften:



public HashSet¡string¿ ValidValues

public String Value

Konstruktoren:

public DistinctOptionInfo (String section, String name, String defaultValue, IEnumerable¡string¿ validValue

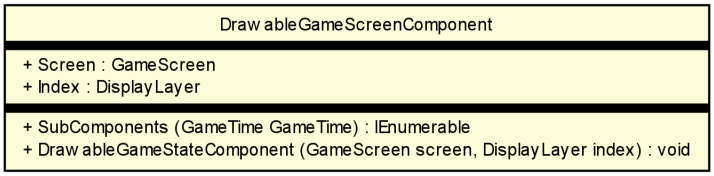
3.1.24 Klasse DrawableGameScreenComponent

Beschreibung:

Eigenschaften:

public GameScreen Screen

public DisplayLayer Index



Methoden:

public IEnumerable SubComponents (GameTime GameTime)

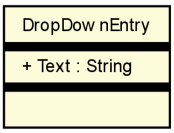
public void DrawableGameStateComponent (GameScreen screen, DisplayLayer index)

3.1.25 Klasse DropDownEntry

Beschreibung:

Eigenschaften:

public String Text

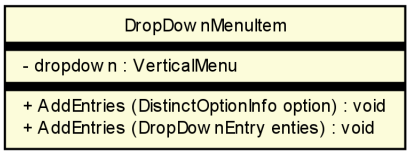


3.1.26 Klasse DropDownMenuItem

Beschreibung:

Eigenschaften:

private VerticalMenu dropdown



Methoden:

public void AddEntries (DistinctOptionInfo option)

public void AddEntries (DropDownEntry enties)

3.1.27 Klasse Edge

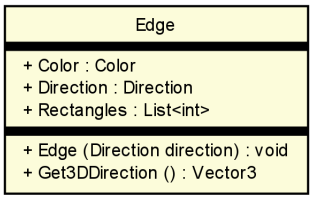
Beschreibung:

Eigenschaften:

public Color Color

public Direction Direction

public List¡int¿ Rectangles



Konstruktoren:

public Edge (Direction direction)

Methoden:

public Vector3 Get3DDirection ()

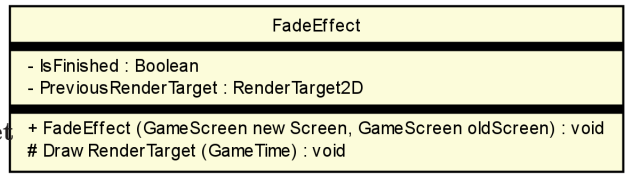
3.1.28 Klasse FadeEffect

Beschreibung:

Eigenschaften:

private Boolean IsFinished

private RenderTarget2D PreviousRenderTarget



Konstruktoren:

public FadeEffect (GameScreen newScreen, GameScreen oldScreen)

Methoden:

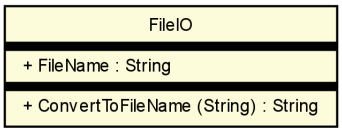
protected void DrawRenderTarget (GameTime)

3.1.29 Klasse FileIO

Beschreibung:

Eigenschaften:

public String FileName



Methoden:

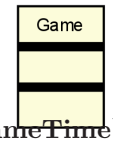
public String ConvertToFileName (String)

3.1.30 Klasse Game

Beschreibung:

Methoden:

public void Update (GameTime time)



public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.31 Klasse GameModel

Beschreibung:

Eigenschaften:

public float Alpha

public Color BaseColor

public Color HightlightColor

public float HighlightIntensity

public GameModelInfo Info

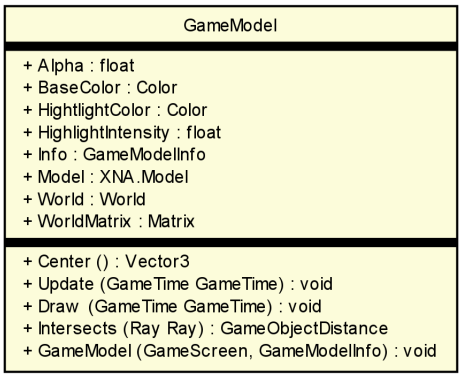
public XNA.Model Model

public World World

public Matrix WorldMatrix

Konstruktoren:

public GameModel (GameScreen, GameModelInfo)



Methoden:

public Vector3 Center ()

public void Update (GameTime GameTime)

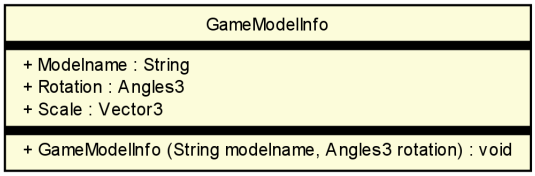
public void Draw (GameTime GameTime)

public GameObjectDistance Intersects (Ray Ray)

3.1.32 Klasse GameModelInfo

Beschreibung:

Eigenschaften:



public String Modelname

public Angles3 Rotation

public Vector3 Scale

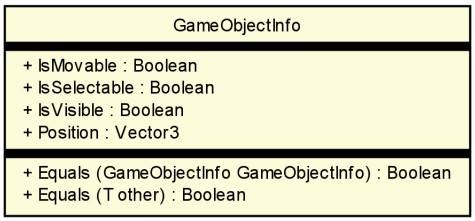
Konstruktoren:

public GameModelInfo (String modelname, Angles3 rotation)

3.1.33 Klasse GameObjectInfo

Beschreibung:

Eigenschaften:



public Boolean IsMovable

public Boolean IsSelectable

public Boolean IsVisible

public Vector3 Position

Methoden:

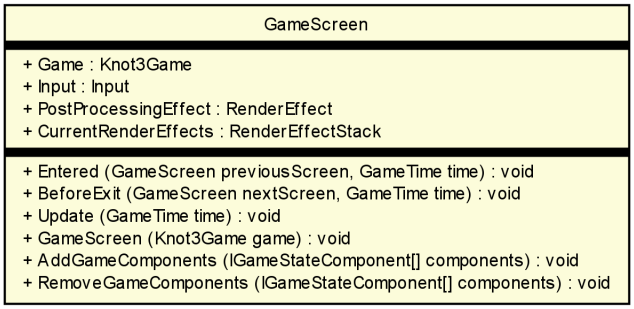
public Boolean Equals (GameObjectInfo GameObjectInfo)

public Boolean Equals (T other)

3.1.34 Klasse GameScreen

Beschreibung:

Eigenschaften:



public Knot3Game Game

public Input Input

public RenderEffect PostProcessingEffect

public RenderEffectStack CurrentRenderEffects

Konstruktoren:

public GameScreen (Knot3Game game)

Methoden:

public void Entered (GameScreen previousScreen, GameTime time)

public void BeforeExit (GameScreen nextScreen, GameTime time)

public void Update (GameTime time)

public void AddGameComponents (IGameStateComponent[] components)

public void RemoveGameComponents (IGameStateComponent[] components)

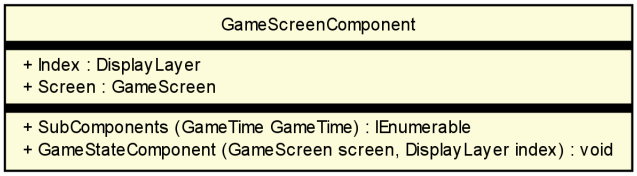
3.1.35 Klasse GameScreenComponent

Beschreibung:

Eigenschaften:

public DisplayLayer Index

public GameScreen Screen



Methoden:

public IEnumerable SubComponents (GameTime GameTime)

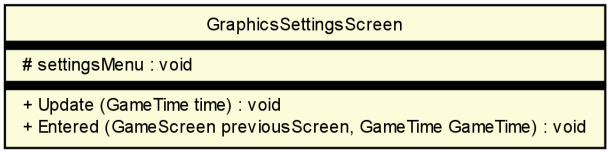
public void GameStateComponent (GameScreen screen, DisplayLayer index)

3.1.36 Klasse GraphicsSettingsScreen

Beschreibung:

Eigenschaften:

protected void settingsMenu



Methoden:

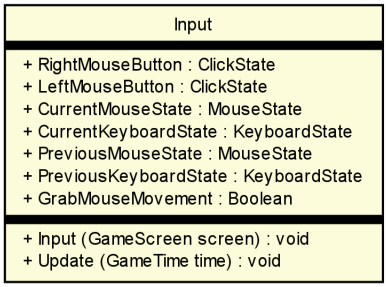
public void Update (GameTime time)

public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.37 Klasse Input

Beschreibung:

Eigenschaften:



public ClickState RightMouseButton

public ClickState LeftMouseButton

public MouseState CurrentMouseState

public KeyboardState CurrentKeyboardState

public MouseState PreviousMouseState

public KeyboardState PreviousKeyboardState

public Boolean GrabMouseMovement

Konstruktoren:

public Input (GameScreen screen)

Methoden:

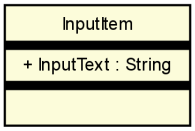
public void Update (GameTime time)

3.1.38 Klasse InputItem

Beschreibung:

Eigenschaften:

public String InputText

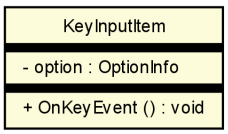


3.1.39 Klasse KeyInputItem

Beschreibung:

Eigenschaften:

private OptionInfo option



Methoden:

public void OnKeyEvent ()

3.1.40 Klasse Knot

Beschreibung:

Eigenschaften:

public String Name

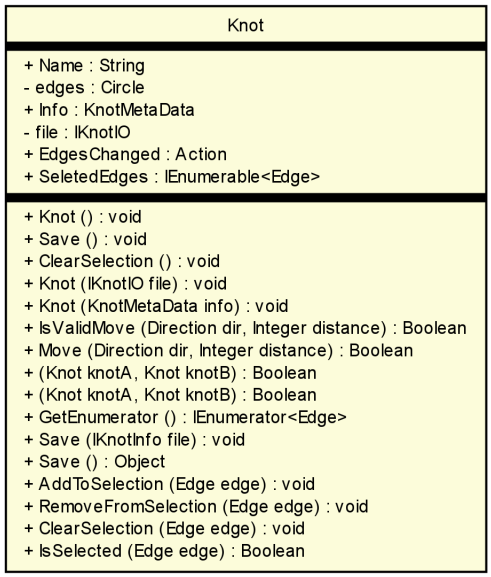
private Circle edges

public KnotMetaData Info

private IKnotIO file

public Action EdgesChanged

public IEnumerable¡Edge¿ SeletedEdges



Konstruktoren:
public Knot ()

public Knot (IKnotIO file)

public Knot (KnotMetaData info)

Methoden:
public void Save ()

public void ClearSelection ()

public Boolean IsValidMove (Direction dir, Integer distance)

public Boolean Move (Direction dir, Integer distance)

public Boolean (Knot knotA, Knot knotB)

public Boolean (Knot knotA, Knot knotB)

public IEnumerator¡Edge¿ GetEnumerator ()

public void Save (IKnotInfo file)

public Object Save ()

public void AddToSelection (Edge edge)

public void RemoveFromSelection (Edge edge)

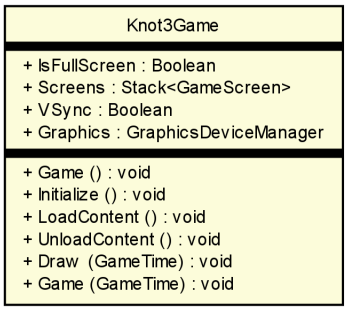
public void ClearSelection (Edge edge)

public Boolean IsSelected (Edge edge)

3.1.41 Klasse Knot3Game

Beschreibung:

Eigenschaften:



public Boolean IsFullScreen

public Stack¡GameScreen¿ Screens

public Boolean VSync

public GraphicsDeviceManager Graphics

Methoden:

public void Game ()

public void Initialize ()

public void LoadContent ()

public void UnloadContent ()

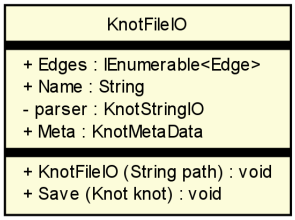
public void Draw (GameTime)

public void Game (GameTime)

3.1.42 Klasse KnotFileIO

Beschreibung:

Eigenschaften:



public IEnumerable¡Edge¿ Edges

public String Name

private KnotStringIO parser

public KnotMetaData Meta

Konstruktoren:

public KnotFileIO (String path)

Methoden:

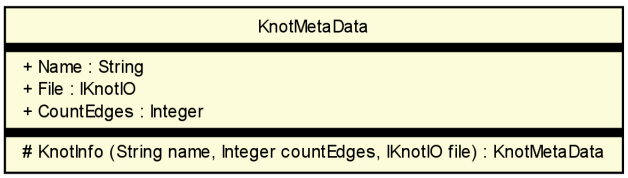
public void Save (Knot knot)

3.1.43 Klasse KnotMetaData

Beschreibung:

Eigenschaften:

public String Name



public IKnotIO File

public Integer CountEdges

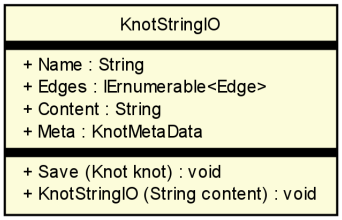
Methoden:

protected KnotMetaData KnotInfo (String name, Integer countEdges, IKnotIO file)

3.1.44 Klasse KnotStringIO

Beschreibung:

Eigenschaften:



public String Name

public IErnumerable¡Edge¿ Edges

public String Content

public KnotMetaData Meta

Konstruktoren:

public KnotStringIO (String content)

Methoden:

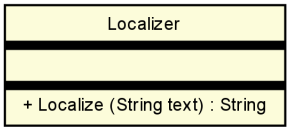
public void Save (Knot knot)

3.1.45 Klasse Localizer

Beschreibung:

Methoden:

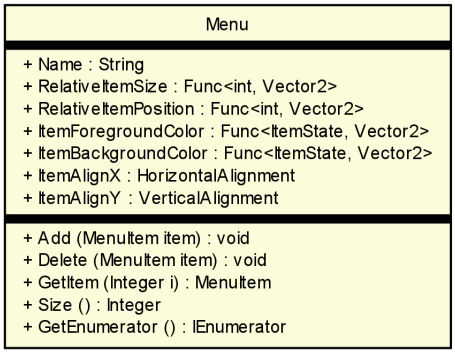
public String Localize (String text)



3.1.46 Klasse Menu

Beschreibung:

Eigenschaften:



public String Name

public Func¡int, Vector2¿ RelativeItemSize

public Func¡int, Vector2¿ RelativeItemPosition

public Func¡ItemState, Vector2¿ ItemForegroundColor

public Func¡ItemState, Vector2¿ ItemBackgroundColor

public HorizontalAlignment ItemAlignX

public VerticalAlignment ItemAlignY

Methoden:

public void Add (MenuItem item)

public void Delete (MenuItem item)

public MenuItem GetItem (Integer i)

public Integer Size ()

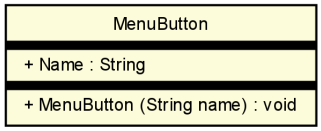
public IEnumerator GetEnumerator ()

3.1.47 Klasse MenuButton

Beschreibung:

Eigenschaften:

public String Name



Konstruktoren:

public MenuButton (String name)

3.1.48 Klasse MenuItem

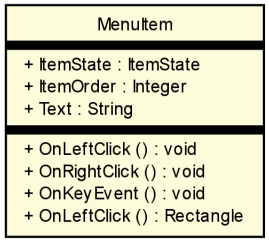
Beschreibung:

Eigenschaften:

public ItemState ItemState

public Integer ItemOrder

public String Text



Methoden:

public void OnLeftClick ()

public void OnRightClick ()

public void OnKeyEvent ()

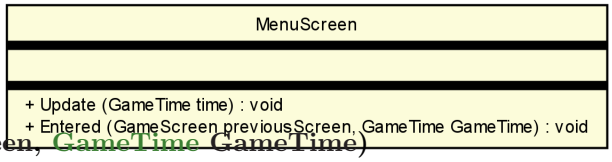
public Rectangle OnLeftClick ()

3.1.49 Klasse MenuScreen

Beschreibung:

Methoden:

public void Update (GameTime time)



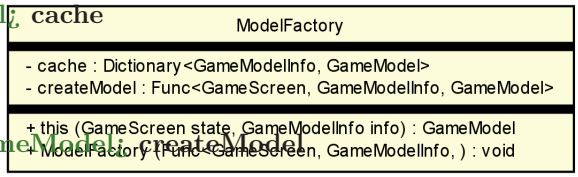
public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.50 Klasse ModelFactory

Beschreibung:

Eigenschaften:

private Dictionary¡GameModelInfo, GameModel¿ cache



private Func¡GameScreen, GameModelInfo, GameModel¿ createModel

Konstruktoren:

public ModelFactory (Func¡GameScreen, GameModelInfo,)

Methoden:

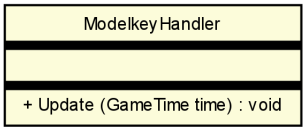
public GameModel this (GameScreen state, GameModelInfo info)

3.1.51 Klasse ModelkeyHandler

Beschreibung:

Methoden:

public void Update (GameTime time)

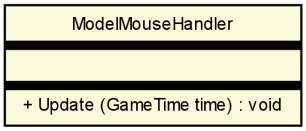


3.1.52 Klasse ModelMouseHandler

Beschreibung:

Methoden:

public void Update (GameTime time)



3.1.53 Klasse ModelRenderer

Beschreibung:

Eigenschaften:

public GameObjectInfo Info

public World World

private List¡ArrowModel¿ arrows

private List¡NodeModel¿ nodes

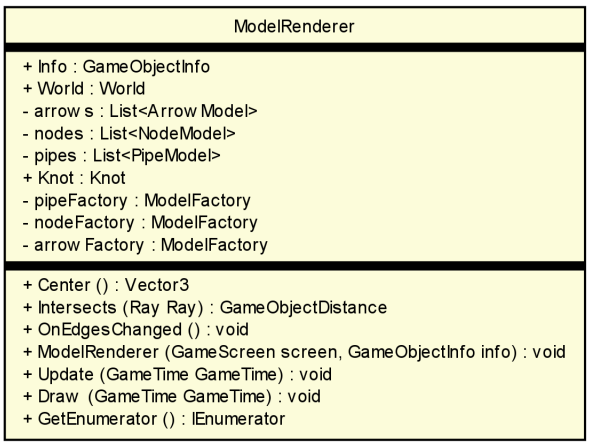
private List¡PipeModel¿ pipes

public Knot Knot

private ModelFactory pipeFactory

private ModelFactory nodeFactory

private ModelFactory arrowFactory



Konstruktoren:

public ModelRenderer (GameScreen screen, GameObjectInfo info)

Methoden:

public Vector3 Center ()

public GameObjectDistance Intersects (Ray Ray)

public void OnEdgesChanged ()

public void Update (GameTime GameTime)

public void Draw (GameTime GameTime)

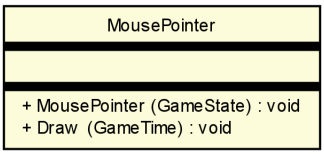
public IEnumerator GetEnumerator ()

3.1.54 Klasse MousePointer

Beschreibung:

Konstruktoren:

public MousePointer (GameState)



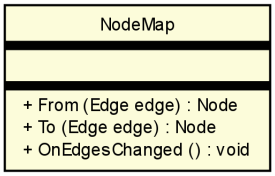
Methoden:

public void Draw (GameTime)

3.1.55 Klasse NodeMap

Beschreibung:

Methoden:



public Node From (Edge edge)

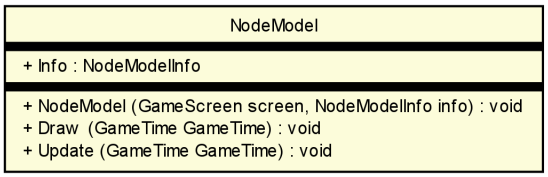
public Node To (Edge edge)

public void OnEdgesChanged ()

3.1.56 Klasse NodeModel

Beschreibung:

Eigenschaften:



public NodeModelInfo Info

Konstruktoren:

public NodeModel (GameScreen screen, NodeModelInfo info)

Methoden:

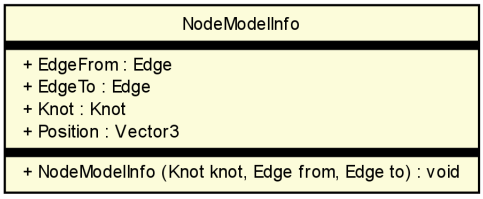
public void Draw (GameTime GameTime)

public void Update (GameTime GameTime)

3.1.57 Klasse NodeModelInfo

Beschreibung:

Eigenschaften:



public Edge EdgeFrom

public Edge EdgeTo

public Knot Knot

public Vector3 Position

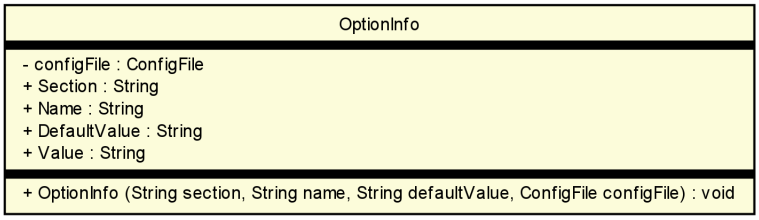
Konstruktoren:

public NodeModelInfo (Knot knot, Edge from, Edge to)

3.1.58 Klasse OptionInfo

Beschreibung:

Eigenschaften:



private ConfigFile configFile

public String Section

public String Name

public String DefaultValue

public String Value

Konstruktoren:

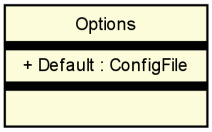
public OptionInfo (String section, String name, String defaultValue, ConfigFile configFile)

3.1.59 Klasse Options

Beschreibung:

Eigenschaften:

public ConfigFile Default

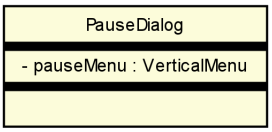


3.1.60 Klasse PauseDialog

Beschreibung:

Eigenschaften:

private VerticalMenu pauseMenu

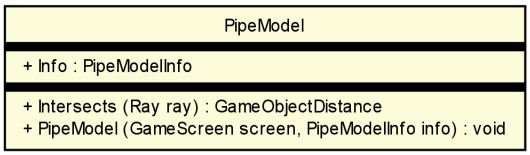


3.1.61 Klasse PipeModel

Beschreibung:

Eigenschaften:

public PipeModelInfo Info



Konstruktoren:

public PipeModel (GameScreen screen, PipeModelInfo info)

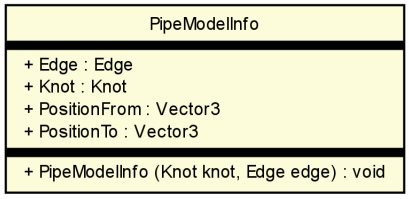
Methoden:

public GameObjectDistance Intersects (Ray ray)

3.1.62 Klasse PipeModelInfo

Beschreibung:

Eigenschaften:



public Edge Edge

public Knot Knot

public Vector3 PositionFrom

public Vector3 PositionTo

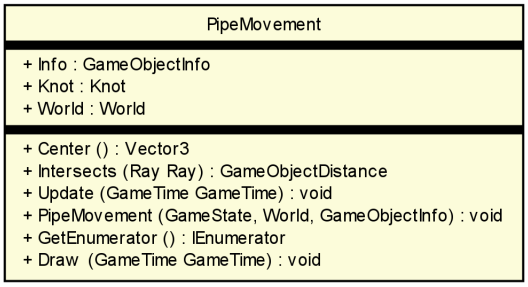
Konstruktoren:

public PipeModelInfo (Knot knot, Edge edge)

3.1.63 Klasse PipeMovement

Beschreibung:

Eigenschaften:



public GameObjectInfo Info

public Knot Knot

public World World

Konstruktoren:

public PipeMovement (GameState, World, GameObjectInfo)

Methoden:

public Vector3 Center ()

public GameObjectDistance Intersects (Ray Ray)

public void Update (GameTime GameTime)

public IEnumerator GetEnumerator ()

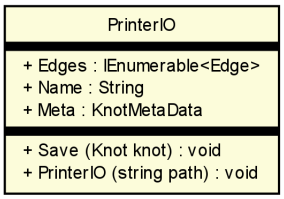
public void Draw (GameTime GameTime)

3.1.64 Klasse PrinterIO

Beschreibung:

Eigenschaften:

public IEnumerable¡Edge¿ Edges



public String Name

public KnotMetaData Meta

Konstruktoren:

public PrinterIO (string path)

Methoden:

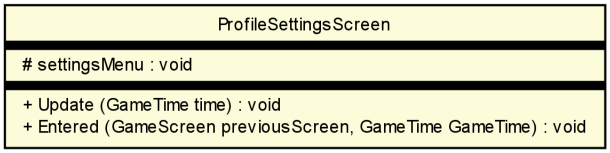
public void Save (Knot knot)

3.1.65 Klasse ProfileSettingsScreen

Beschreibung:

Eigenschaften:

protected void settingsMenu



Methoden:

public void Update (GameTime time)

public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.66 Klasse RenderEffect

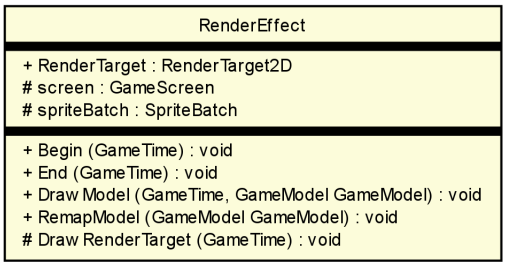
Beschreibung:

Eigenschaften:

public RenderTarget2D RenderTarget

protected GameScreen screen

protected SpriteBatch spriteBatch



Methoden:

public void Begin (GameTime)

public void End (GameTime)

public void DrawModel (GameTime, GameModel GameModel)

public void RemapModel (GameModel GameModel)

protected void DrawRenderTarget (GameTime)

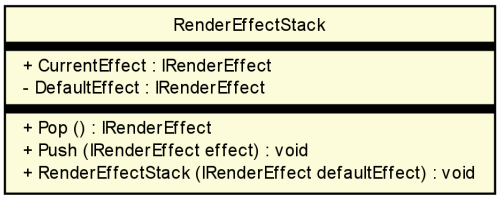
3.1.67 Klasse RenderEffectStack

Beschreibung:

Eigenschaften:

public IRenderEffect CurrentEffect

private IRenderEffect DefaultEffect



Konstruktoren:

public RenderEffectStack (IRenderEffect defaultEffect)

Methoden:

public IRenderEffect Pop ()

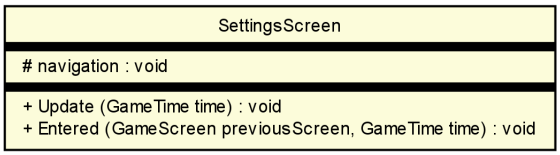
public void Push (IRenderEffect effect)

3.1.68 Klasse SettingsScreen

Beschreibung:

Eigenschaften:

protected void navigation



Methoden:

public void Update (GameTime time)

public void Entered (GameScreen previousScreen, GameTime time)

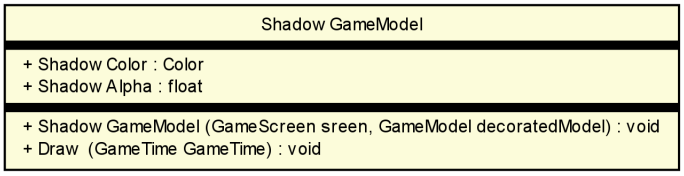
3.1.69 Klasse ShadowGameModel

Beschreibung:

Eigenschaften:

public Color ShadowColor

public float ShadowAlpha



Konstruktoren:

public ShadowGameModel (GameScreen sreen, GameModel decoratedModel)

Methoden:

public void Draw (GameTime GameTime)

3.1.70 Klasse ShadowGameObject

Beschreibung:

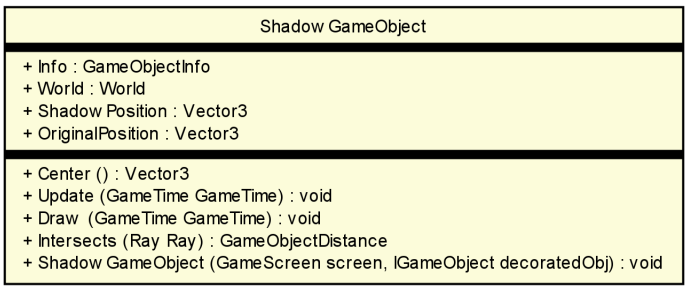
Eigenschaften:

public GameObjectInfo Info

public World World

public Vector3 ShadowPosition

public Vector3 OriginalPosition



Konstruktoren:

public ShadowGameObject (GameScreen screen, IGameObject decoratedObj)

Methoden:

public Vector3 Center ()

public void Update (GameTime GameTime)

public void Draw (GameTime GameTime)

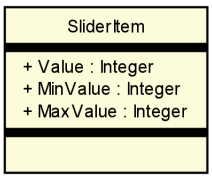
public GameObjectDistance Intersects (Ray Ray)

3.1.71 Klasse SliderItem

Beschreibung:

Eigenschaften:

public Integer Value



public Integer MinValue

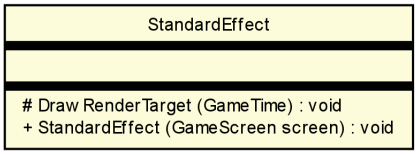
public Integer MaxValue

3.1.72 Klasse StandardEffect

Beschreibung:

Konstruktoren:

public StandardEffect (GameScreen screen)



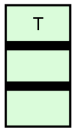
Methoden:

protected void DrawRenderTarget (GameTime)

3.1.73 Klasse T

Beschreibung:

Eigenschaften:



public float X

public float Y

public float Z

public Angles3 Zero

Methoden:

public IEnumerator GetEnumerator ()

public IEnumerator GetEnumerator ()

public Boolean Equals (T other)

public Boolean Equals (T other)

public Angles3 FromDegrees (float X, float Y, float Z)

public void Angles3 (float X, float Y, float Z)

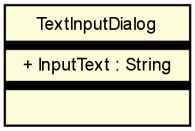
public void ToDegrees (float X, float Y, float Z)

3.1.74 Klasse TextInputDialog

Beschreibung:

Eigenschaften:

public String InputText

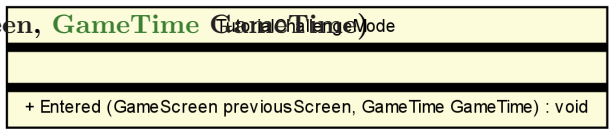


3.1.75 Klasse TutorialChallengeMode

Beschreibung:

Methoden:

public void Entered (GameScreen previousScreen, GameTime GameTime)

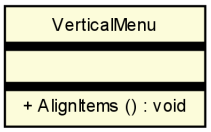


3.1.76 Klasse VerticalMenu

Beschreibung:

Methoden:

public void AlignItems ()



3.1.77 Klasse Widget

Beschreibung:

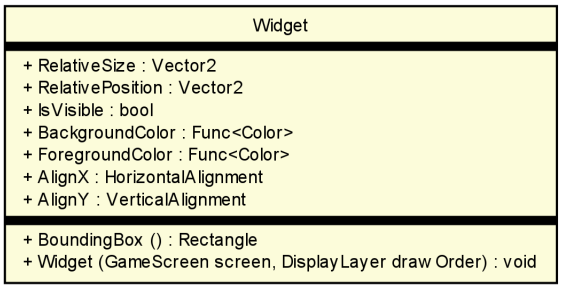
Eigenschaften:

public Vector2 RelativeSize

public Vector2 RelativePosition

public bool IsVisible

public Func¡Color¿ BackgroundColor



public Func¡Color¿ ForegroundColor

public HorizontalAlignment AlignX

public VerticalAlignment AlignY

Konstruktoren:

public Widget (GameScreen screen, DisplayLayer drawOrder)

Methoden:

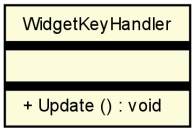
public Rectangle BoundingBox ()

3.1.78 Klasse WidgetKeyHandler

Beschreibung:

Methoden:

public void Update ()

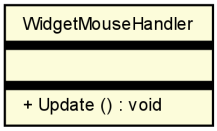


3.1.79 Klasse WidgetMouseHandler

Beschreibung:

Methoden:

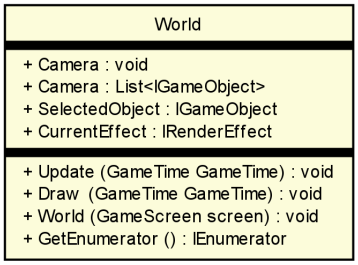
public void Update ()



3.1.80 Klasse World

Beschreibung:

Eigenschaften:



public void Camera

public List¡IGameObject¿ Camera

public IGameObject SelectedObject

public IRenderEffect CurrentEffect

Konstruktoren:

public World (GameScreen screen)

Methoden:

public void Update (GameTime GameTime)

public void Draw (GameTime GameTime)

public IEnumerator GetEnumerator ()

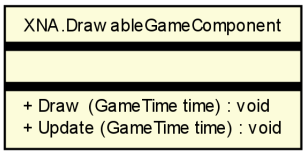
3.1.81 Klasse XNA.DrawableGameComponent

Beschreibung:

Methoden:

public void Draw (GameTime time)

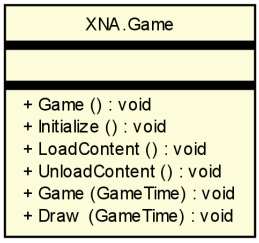
public void Update (GameTime time)



3.1.82 Klasse XNA.Game

Beschreibung:

Methoden:



public void Game ()

public void Initialize ()

public void LoadContent ()

public void UnloadContent ()

public void Game (GameTime)

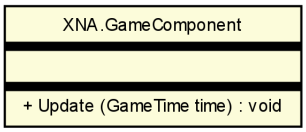
public void Draw (GameTime)

3.1.83 Klasse XNA.GameComponent

Beschreibung:

Methoden:

public void Update (GameTime time)



3.2 Schnittstellen

3.2.1 Schnittstelle IChallengeIO

Beschreibung:

Eigenschaften:

public IEnumerable¡KeyValuePair¡String, Integer¿¿ Highscore

public String Name

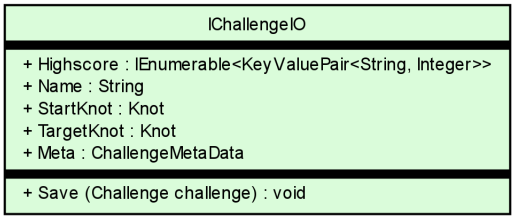
public Knot StartKnot

public Knot TargetKnot

public ChallengeMetaData Meta

Methoden:

public void Save (Challenge challenge)



3.2.2 Schnittstelle ICloneable

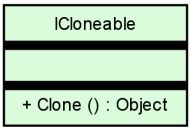
Beschreibung:

Methoden:

public Object Clone ()

3.2.3 Schnittstelle IEnumerable

Beschreibung:

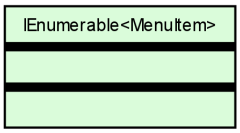


Methoden:

public IEnumerator GetEnumerator ()

3.2.4 Schnittstelle IEnumerable1

Beschreibung:

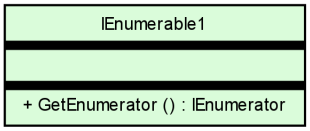


Methoden:

public IEnumerator GetEnumerator ()

3.2.5 Schnittstelle IEquatable

Beschreibung:

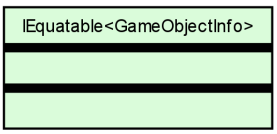


Methoden:

public Boolean Equals (T other)

3.2.6 Schnittstelle IEquatable1

Beschreibung:

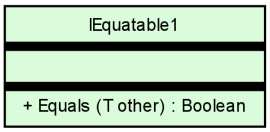


Methoden:

public Boolean Equals (T other)

3.2.7 Schnittstelle IGameObject

Beschreibung:



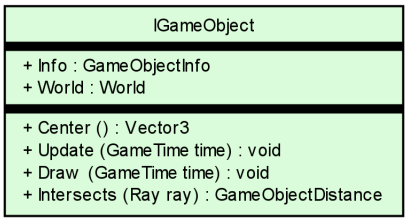
Eigenschaften:

public GameObjectInfo Info

public World World

Methoden:

public Vector3 Center ()



public void Update (GameTime time)

public void Draw (GameTime time)

public GameObjectDistance Intersects (Ray ray)

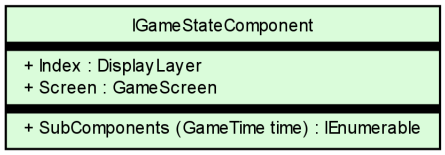
3.2.8 Schnittstelle IGameStateComponent

Beschreibung:

Eigenschaften:

public DisplayLayer Index

public GameScreen Screen



Methoden:

public IEnumerable SubComponents (GameTime time)

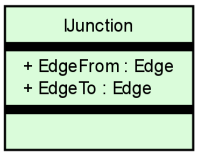
3.2.9 Schnittstelle IJunction

Beschreibung:

Eigenschaften:

public Edge EdgeFrom

public Edge EdgeTo



3.2.10 Schnittstelle IKeyEventListener

Beschreibung:

Eigenschaften:

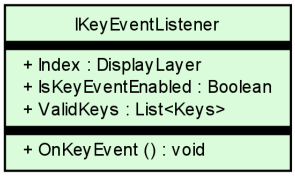
public DisplayLayer Index

public Boolean IsKeyEventEnabled

public List¡Keys¿ ValidKeys

Methoden:

public void OnKeyEvent ()



3.2.11 Schnittstelle IKnotIO

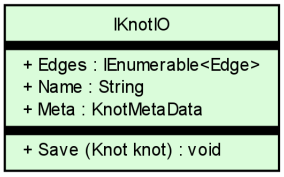
Beschreibung:

Eigenschaften:

public IEnumerable¡Edge¿ Edges

public String Name

public KnotMetaData Meta



Methoden:

public void Save (Knot knot)

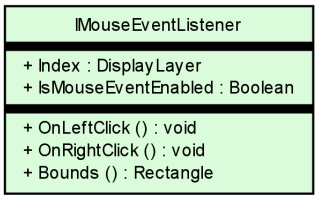
3.2.12 Schnittstelle IMouseEventListener

Beschreibung:

Eigenschaften:

public DisplayLayer Index

public Boolean IsMouseEventEnabled



Methoden:

public void OnLeftClick ()

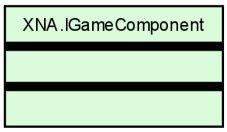
public void OnRightClick ()

public Rectangle Bounds ()

3.2.13 Schnittstelle XNA.IGameComponent

Beschreibung:

Eigenschaften:



public void Camera

public Boolean IsMovable

public Boolean IsSelectable

public Boolean IsVisible

public Vector3 Position

public String Modelname

public Angles3 Rotation

public Vector3 Scale

public Edge Edge

public Knot Knot

public Vector3 PositionFrom

public Vector3 PositionTo

public Edge EdgeFrom

public Edge EdgeTo

public Knot Knot

public Vector3 Position

public PipeModelInfo Info

public NodeModelInfo Info

public List¡IGameObject¿ Camera

public IGameObject SelectedObject

public IRenderEffect CurrentEffect

public GameScreen Screen

public DisplayLayer Index

public DisplayLayer Index

public GameScreen Screen

public Vector3 Position

public Vector3 Target

public float FoV

public Matrix ViewMatrix

public Matrix WorldMatrix

public Matrix ProjectionMatrix

public Vector3 ArcballTarget

public BoundingFrustum ViewFrustum

private World World

public Angles3 Rotation

public RenderTarget2D RenderTarget

Methoden:

public Boolean Equals (GameObjectInfo GameObjectInfo)

public Boolean Equals (T other)

public void GameModelInfo (String modelname, Angles3 rotation)

public void PipeModelInfo (Knot knot, Edge edge)

public void NodeModelInfo (Knot knot, Edge from, Edge to)

public GameObjectDistance Intersects (Ray ray)

public void PipeModel (GameScreen screen, PipeModelInfo info)

public void NodeModel (GameScreen screen, NodeModelInfo info)

public void Draw (GameTime GameTime)

public void Update (GameTime GameTime)

public void Update (GameTime GameTime)

public void Draw (GameTime GameTime)

public void World (GameScreen screen)

public IEnumerator GetEnumerator ()

public IEnumerable SubComponents (GameTime GameTime)

public void DrawableGameStateComponent (GameScreen screen, DisplayLayer index)

public IEnumerable SubComponents (GameTime GameTime)

public void GameStateComponent (GameScreen screen, DisplayLayer index)

public Vector3 TargetDirection ()

public float TargetDistance ()

public void Camera (GameScreen screen, World world)

public void Update (GameTime GameTime)

public Ray GetMouseRay (Vector2 mousePosition)

public void Begin (GameTime)

public void End (GameTime)

public void DrawModel (GameTime, GameModel model)

public void RemapModel (GameModel model)

3.3 Enumerationen

Kapitel 4

Abläufe

4.1 Sequenzdiagramme

Kapitel 5

Klassenindex

Kapitel 6

Anmerkungen

Kapitel 7

Glossar

Test Test

7.1 Fachausdrücke

Test (Test-Beschreibung) ... 50

7.2 Abkürzungen

Test Test 50