ENTWURFSDOKUMENT

(V. 1.0)

$\begin{array}{c} \textbf{KNOT}^3 \\ \textbf{PSE WS } 2013/14 \end{array}$

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	Ein	leitung		4
2	Auf 2.1 2.2 2.3	Archit Klasse	ektur	5 6 6
3	Kla	ssenüb	ersicht	7
	3.1	Klasse	n	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
				19
				19
			*	20
				20
			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

	Klasse GameModel	
	Klasse GameModelInfo	24
3.1.33	Klasse GameObjectInfo	24
	Klasse GameScreen	25
	Klasse GameScreenComponent	26
3.1.36	Klasse GraphicsSettingsScreen	26
3.1.37	Klasse Input	27
	Klasse InputItem	28
	Klasse KeyInputItem	28
	Klasse Knot	28
	Klasse Knot3Game	30
	Klasse KnotFileIO	3
	Klasse KnotMetaData	32
	Klasse KnotStringIO	32
	Klasse Localizer	$\frac{32}{33}$
	Klasse Menu	33
	Klasse MenuButton	34
	Klasse MenuItem	34
	Klasse MenuScreen	35
	Klasse ModelFactory	35
	Klasse ModelkeyHandler	36
	Klasse ModelMouseHandler	36
	Klasse ModelRenderer	36
	Klasse MousePointer	38
	Klasse NodeMap	38
	Klasse NodeModel	38
3.1.57	Klasse NodeModelInfo	36
3.1.58	Klasse OptionInfo	39
3.1.59	Klasse Options	40
3.1.60	Klasse PauseDialog	40
	Klasse PipeModel	40
	Klasse PipeModelInfo	4
	Klasse PipeMovement	4
	Klasse PrinterIO	42
	Klasse ProfileSettingsScreen	43
	Klasse RenderEffect	
	Klasse RenderEffectStack	
	Klasse SettingsScreen	
	Klasse ShadowGameModel	45
	Klasse ShadowGameObject	
		46
	Klasse SliderItem	47
	Klasse StandardEffect	47
	Klasse T	48
	Klasse TextInputDialog	48
	Klasse TutorialChallengeMode	49
	Klasse VerticalMenu	49
	Klasse Widget	49
	Klasse WidgetKeyHandler	50
	Klasse WidgetMouseHandler	50
3.1.80	Klasse World	5.
3 1 81	Klasse XNA DrawableGameComponent	5

		3.1.82	Klasse XNA.Game	2
		3.1.83	Klasse XNA.GameComponent	2
	3.2	Schnitt	stellen	3
		3.2.1	Schnittstelle IChallengeIO	3
		3.2.2	Schnittstelle ICloneable	3
			Schnittstelle IEnumerable	4
		3.2.4	Schnittstelle IEnumerable 1	4
		3.2.5	Schnittstelle IEquatable	4
		3.2.6	Schnittstelle IEquatable 1	4
		3.2.7	Schnittstelle IGameObject	5
			Schnittstelle IGameStateComponent	5
			Schnittstelle IJunction	-
			Schnittstelle IKeyEventListener	
			Schnittstelle IKnotIO	7
			Schnittstelle IMouseEventListener	
			Schnittstelle XNA.IGameComponent	8
	3.3	Enume	rationen	2
4	Abl	äufe	6	3
	4.1		zdiagramme	3
5	Klas	ssenind	ex 6	4
6	Ann	nerkun	gen 6	5
7	Glos		e	c
1	G108		6 sdrücke	
				-
	7.2	ADKUIZ	ungen	. (

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.

Arrow Model

- + Info: Arrow ModelInfo
- + Draw (GameTime GameTime) : void + Intersects (Ray ray) : GameObjectDistance
- + Arrow Model (GameScreen screen, Arrow ModelInfo info): void
- + Update (GameTime GameTime) : void

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

AudioSettingsScreen # settingsMenu : void + Update (GameTime time) : void + Entered (GameScreen previousScreen, GameTime GameTime) : void

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.

Boolean Option Info
+ Value : bool
+ BooleanOptionInfo (String section, String name, String defaultValue, ConfigFile configFile) : void

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.

Camera + Position: Vector3 + Target: Vector3 + FoV: float + View Matrix: Matrix + WorldMatrix: Matrix + ProjectionMatrix: Matrix + ArcballTarget: Vector3 + View Frustum: BoundingFrustum - World: World + Rotation: Angles3 + TargetDirection (): Vector3 + TargetDistance (): float + Camera (GameScreen screen, World w orld): void

+ GetMouseRay (Vector2 mousePosition) : Ray

9

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

Konstruktoren: public CelShadingEffect (GameScreen screen) Methoden: protected void DrawRenderTarget (GameTime)

CelShadingEffect

- # Draw RenderTarget (GameTime): void
- + Draw Model (GameTime, GameModel GameModel) : void
- + RemapModel (GameModel GameModel) : void
- + CelShadingEffect (GameScreen screen) : void

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

Challenge

- + Start : Knot
- + Target : Knot
- highscore : SortedList<Integer, String>
- + Name : String
- file : IChallengeIO
- + Highscore : IEnumerator<KeyValuePair<String, Integer>>
- + Info : ChallengeMetaData
- + Challenge (ChallengeInfo info) : Challenge
- + Challenge (IChallangeIO file) : Challenge
- + CreateChallenge (Knot start, Knot target, String name, IChallengeIO file): Boolean
- + AddToHighscore (String name, Integer time): void

public String Name

private IChallengeIO file

public IEnumerator; Key Value Pair; 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; Highscore

+ Highscore : IEnumerable<KeyValuePair<String, Integer>> + Name : String + StartKnot : Knot

ChallengeFileIO

public String Name

+ ChallengeFileIO (String path) : void + Save (Challenge challenge) : void

- targetParser : KnotStringIO + Meta : ChallengeMetaData

+ TargetKnot : Knot - startParser : KnotStringIO

 ${\bf public} \ {\bf Knot} \ {\bf StartKnot}$

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)

ChallengeLoadScreen

public void Entered (GameScreen previousScreen, CameTime CameTime): void

public void Entered (GameScreen previousScreen, CameTime CameTime): void

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:

ChallengeMetaData + Name: String + Start: KnotMetaData + Target: KnotMetaData + File: IChallengeIO

Eigenschaften: ChallengeMode public void PlayerKnot + PlayerKnot : void + ChallengeKnot : void + PlayerKnot : World - PlayerWorld : World - ChallengeKnotRenderer : ModelRenderer public void ChallengeKnot - PlayerKnotRenderer : ModelRenderer - PlayerKnotMovement : PipeMovement + Undo : Stack<Knot> + Redo : Stack<Knot> public World PlayerKnot + Update (GameTime time) : void + Entered (GameScreen previousScreen, GameTime GameTime): void private World PlayerWorld private ModelRenderer ChallengeKnotRenderer ${\bf private\ Model Renderer\ Player Knot Renderer}$ private PipeMovement PlayerKnotMovement public Stack; Knot; Undo public Stack; Knot; Redo Methoden: public void Update (GameTime time) public void Entered (GameScreen previousScreen, GameTime GameTime)

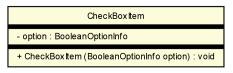
14

Klasse CheckBoxItem

3.1.12

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:

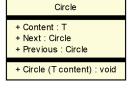
public Circle (T content)

3.1.14 Klasse ColorPicker

Beschreibung:

${\bf Eigenschaften:}$

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 ColorPickItem + Color : Color Klasse ConfigFile 3.1.16 Beschreibung: Methoden: public void SetOption (String section, String option, String value) ConfigFile + SetOption (String section, String option, String value) : void + GetOption (String section, String option, Boolean defaultValue): Boolean public Boolean GetOption (String section, String section, Stri 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

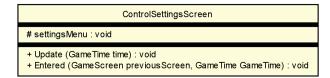
Beschreibung:

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 CameTime GameTime): void

+ Entered (GameScreen previousScreen, GameTime GameTime): void

+ Update (GameTime time): void

3.1.20 Klasse CreativeMode

Eigenschaften: CreativeMode public void Knot + Knot : void + Knot : World - KnotRenderer : ModelRenderer + Undo : Stack<Knot> + Redo : Stack<Knot> public World Knot + Update (GameTime time) : void + Entered (GameScreen previousScreen, GameTime time) : void ${\bf private\ Model Renderer\ Knot Renderer}$ public Stack; Knot; Undo public Stack; Knot; Redo Methoden: public void Update (GameTime time) public void Entered (GameScreen previousScreen, GameTime time)

Methoden:

Beschreibung:

3.1.21

3.1.22 Klasse Dialog

Klasse CreditsScreen

Eigenschaften: Dialog public String Name + Name : String + Text : String + OnKeyEvent (): void + OnLeftClick (): void public String Text + OnRightClick (): void + OnKeyEvent (): Rectangle Methoden: public void OnKeyEvent () public void OnLeftClick () public void OnRightClick () public Rectangle OnKeyEvent () 3.1.23 Klasse DistinctOptionInfo Beschreibung:

Konstruktoren:

public String Value

Eigenschaften:

public HashSet; string; ValidValues

public DistinctOptionInfo (String section, String name, String defaultValue, IEnumerable; string; validValue

+ Value : String

+ ValidValues : HashSet<string>

DistinctOptionInfo

+ DistinctOptionInfo (String section, String name, String defaultValue, IEnumerable<string> validValue

3.1.24 Klasse DrawableGameScreenComponent

Eigenschaften:

public GameScreen Screen

public DisplayLayer Index

Draw able Game Screen Component

- + Screen : GameScreen
- + Index : Display Layer
- + SubComponents (GameTime GameTime) : IEnumerable
- + Draw ableGameStateComponent (GameScreen screen, DisplayLayer index) : void

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:

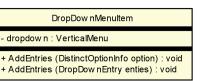
Eigenschaften:

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

FadeEffect

- IsFinished : Boolean

- PreviousRenderTarget : RenderTarget2D

+ FadeEffect (GameScreen new Screen, GameScreen oldScreen) : void # Draw RenderTarget (GameTime) : void

Konstruktoren:

public FadeEffect (GameScreen newScreen, GameScreen oldScreen)

Edge

+ Edge (Direction direction) : void + Get3DDirection () : Vector3

+ Color : Color + Direction : Direction + Rectangles : List<int>

Methoden:

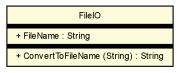
protected void DrawRenderTarget (GameTime)

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)
                                                                             GameModel
                                                               + Alpha : float
                                                               + BaseColor : Color
                                                               + HightlightColor : Color
                                                               + HighlightIntensity : float
                                                               + Info : GameModelInfo
                                                               + Model : XNA . Model
Methoden:
                                                               + World : World
                                                               + WorldMatrix : Matrix
public Vector3 Center ()
                                                               + Center (): Vector3
                                                               + Update (GameTime GameTime) : void
                                                               + Draw (GameTime GameTime) : void
                                                               + Intersects (Ray Ray) : GameObjectDistance
public void Update (GameTime GameTime)
                                                               + GameModel (GameScreen, GameModelInfo): void
public void Draw (GameTime GameTime)
public GameObjectDistance Intersects (Ray Ray)
```

3.1.32 Klasse GameModelInfo

GameModelInfo public String Modelname + Modelname : String + Rotation : Angles3 + Scale : Vector3 public Angles3 Rotation + GameModelInfo (String modelname, Angles3 rotation) : void public Vector3 Scale Konstruktoren: public GameModelInfo (String modelname, Angles3 rotation) 3.1.33 Klasse GameObjectInfo Beschreibung: Eigenschaften: public Boolean IsMovable Game Object Info + IsMovable : Boolean + IsSelectable : Boolean + lsVisible : Boolean public Boolean IsSelectable + Position : Vector3 + Equals (GameObjectInfo GameObjectInfo): Boolean + Equals (Tother) : Boolean public Boolean IsVisible public Vector3 Position Methoden: public Boolean Equals (GameObjectInfo GameObjectInfo)

3.1.34 Klasse GameScreen

public Boolean Equals (T other)

Beschreibung:

Eigenschaften:

Eigenschaften: GameScreen public Knot3Game Game + Game : Knot3Game + Input : Input + PostProcessingEffect : RenderEffect + CurrentRenderEffects: RenderEffectStack public Input Input + Entered (GameScreen previousScreen, GameTime time) : void + BeforeExit (GameScreen nextScreen, GameTime time) : void + Update (GameTime time) : void + GameScreen (Knot3Game game) : void + AddGameComponents (IGameStateComponent[] components) : void public RenderEffect PostProcessingEffect + RemoveGameComponents (IGameStateComponent[] components) : void 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 (IGameStateComponents)

3.1.35 Klasse GameScreenComponent

Beschreibung:

public void RemoveGameComponents (IGameStateComponents)

Eigenschaften:

public DisplayLayer Index

public GameScreen Screen

GameScreenComponent

- + Index : Display Layer
- + Screen : GameScreen
- + SubComponents (GameTime GameTime) : IEnumerable
- + GameStateComponent (GameScreen screen, DisplayLayer index) : void

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)

GraphicsSettingsScreen

settingsMenu : void

- + Update (GameTime time) : void
- + Entered (GameScreen previousScreen, GameTime GameTime) : void

public void Entered (GameScreen previousScreen, GameTime GameTime)

3.1.37 Klasse Input

Eigenschaften: public ClickState RightMouseButton public ClickState LeftMouseButton public MouseState CurrentMouseState public KeyboardState CurrentKeyboardState public MouseState PreviousMouseState public KeyboardState PreviousKeyboardState public Boolean GrabMouseMovement Konstruktoren:

3.1.38 Klasse InputItem

public Input (GameScreen screen)

public void Update (GameTime time)

 ${\bf Beschreibung:}$

Methoden:

Input

- + RightMouseButton : ClickState
- + LeftMouseButton : ClickState
- + CurrentMouseState : MouseState
- + CurrentKeyboardState : KeyboardState
- + PreviousMouseState : MouseState
- + PreviousKeyboardState : KeyboardState
- + GrabMouseMovement : Boolean
- + Input (GameScreen screen) : void
- + Update (GameTime time) : void

Eigenschaften:

public String InputText

InputItem + InputText : String

Klasse KeyInputItem 3.1.39

Beschreibung:

Eigenschaften:

private OptionInfo option

Methoden:

public void OnKeyEvent ()

Klasse Knot 3.1.40

Beschreibung:

Eigenschaften:

public String Name

private Circle edges

public KnotMetaData Info

private IKnotIO file

public Action EdgesChanged

public IEnumerable; Edge; SeletedEdges

KeyInputItem - option : OptionInfo + OnKeyEvent (): void

Knot

- + Name : String - edges : Circle
- + Info : KnotMetaData
- file : IKnotIO
- + EdgesChanged : Action
- + SeletedEdges : IEnumerable<Edge>
- + Knot () : void
- + Save () : void
- + ClearSelection (): void
- + Knot (IKnotIO file): void
- + Knot (KnotMetaData info) : void
- + IsValidMove (Direction dir, Integer distance): Boolean
- + Move (Direction dir, Integer distance) : Boolean
- + (Knot knotA, Knot knotB) : Boolean + (Knot knotA, Knot knotB) : Boolean
- + GetEnumerator () : IEnumerator<Edge>
- + Save (IKnotInfo file): void
- + Save (): Object
- + AddToSelection (Edge edge) : void
- + RemoveFromSelection (Edge edge) : void
- + ClearSelection (Edge edge) : void
- + IsSelected (Edge edge) : Boolean

```
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

 ${\bf Beschreibung:}$

Knot3Game + IsFullScreen: Boolean + Screens: Stack<GameScreen> + VSync: Boolean + Graphics: GraphicsDeviceManager + Game (): void + Initialize (): void + UnloadContent (): void + Draw (GameTime): void + Game (GameTime): void

Eigenschaften: public IEnumerable¡Edge¿ Edges public String Name

 ${\bf private~KnotString IO~parser}$

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

KnotMetaData + Name : String

+ File : IKnotlO + CountEdges : Integer

 $\hbox{\# KnotInfo (String name, Integer countEdges, IKnotIO file)}: KnotMetaData$

KnotFile IO

+ Edges : IEnumerable<Edge>

+ KnotFileIO (String path) : void + Save (Knot knot) : void

+ Name : String - parser : KnotStringIO + Meta : KnotMetaData

Methoden:

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

3.1.44 Klasse KnotStringIO

Beschreibung:

31

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:

KnotStringIO

+ Name : String

+ Edges : IErnumerable < Edge >

+ Content : String

+ Meta: KnotMetaData

+ Save (Knot knot) : void

+ KnotStringIO (String content) : void

Localizer

+ Localize (String text) : String

Eigenschaften: public String Name + Name : String + Relative temSize : Func<int, Vector2> + Relative ItemPosition : Func<int, Vector2> + ItemForegroundColor : Func<ItemState, Vector2> public Func; int, Vector2; RelativeItemSize + ItemBackgroundColor : Func<ItemState, Vector2> + ItemAlignX : HorizontalAlignment + ItemAlignY: VerticalAlignment + Add (Menultem item): void public Func;int, Vector2; RelativeItemPosition + Delete (Menultem item) : void + GetItem (Integer i) : MenuItem + Size (): Integer + GetEnumerator (): IEnumerator ${\bf public\ Func; Item State,\ Vector 2;\ Item Foreground Color}$ public Func; ItemState, Vector2; ItemBackgroundColor public Horizontal Alignment Item Align X 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 () Klasse MenuButton 3.1.47

Beschreibung:

Menu

${\bf Eigenschaften:}$

public String Name

MenuButton
+ Name : String
+ MenuButton (String name) : void

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:

Menultem + ItemState : ItemState + ItemOrder : Integer + Text : String + OnLeftClick () : void + OnRightClick () : void + OnKeyEvent () : void + OnLeftClick () : Rectangle

Methoden:

public void Update (GameTime time)

+ Update (GameTime time): void
+ Entered (GameScreen previousScreen, GameTime GameTime): void

- createModel: Func<GameScreen, GameModelInfo, GameModel>

public void Entered (GameScreen previousScreen, GameTime GameTin

3.1.50 Klasse ModelFactory

Beschreibung:

Eigenschaften:

private Dictionary; GameModelInfo, GameModel cache ModelFactory

- cache: Dictionary < GameModel | GameModel | Cache | Dictionary < GameModel | Diction

private Func; GameScreen, GameModelInfo, GameLore Func; GameScreen, GameModelInfo, GameLore Func; GameScreen, GameModelInfo, J: void

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

Methoden: ModelMouseHandler public void Update (GameTime time) + Update (GameTime time) : void Klasse ModelRenderer 3.1.53 Beschreibung: Eigenschaften: public GameObjectInfo Info ModelRenderer + Info : GameObjectInfo + World : World - arrow s : List<Arrow Model> public World World - nodes : List<NodeModel> - pipes : List<PipeModel> + Knot : Knot

private List; NodeModel; nodes private List; PipeModel; pipes

private List; Arrow Model; arrows

public Knot Knot private ModelFactory pipeFactory

private ModelFactory nodeFactory

private ModelFactory arrowFactory

Konstruktoren:

public ModelRenderer (GameScreen screen, GameObjectInfo info)

- pipeFactory : ModelFactory - nodeFactory : ModelFactory
- arrow Factory : ModelFactory
- + Center (): Vector3
- + Intersects (Ray Ray) : GameObjectDistance
- + OnEdgesChanged (): void
- + ModelRenderer (GameScreen screen, GameObjectInfo info) : void
- + Update (GameTime GameTime) : void
- + Draw (GameTime GameTime) : void
- + GetEnumerator () : IEnumerator

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 () Klasse MousePointer 3.1.54Beschreibung: Konstruktoren: public MousePointer (GameState) MousePointer + MousePointer (GameState) : void + Draw (GameTime) : void Methoden: public void Draw (GameTime)

3.1.55 Klasse NodeMap

Methoden: public Node From (Edge edge) + From (Edge edge): Node + To (Edge edge): Node + OnEdgesChanged(): void 3.1.56 Klasse NodeModel Beschreibung: Eigenschaften:

NodeModel

+ NodeModel (GameScreen screen, NodeModelInfo info) : void

+ Info : NodeModelInfo

+ Draw (GameTime GameTime) : void

+ Update (GameTime GameTime) : void

Konstruktoren:

public NodeModelInfo Info

public NodeModel (GameScreen screen, NodeModelInfo info)

Methoden:

public void Draw (GameTime GameTime)

public void Update (GameTime GameTime)

3.1.57 Klasse NodeModelInfo

Eigenschaften: NodeModelInfo public Edge EdgeFrom + EdgeFrom : Edge + EdgeTo : Edge + Knot : Knot + Position : Vector3 public Edge EdgeTo + NodeModelInfo (Knot knot, Edge from, Edge to): void public Knot Knot public Vector3 Position Konstruktoren: public NodeModelInfo (Knot knot, Edge from, Edge to) 3.1.58Klasse OptionInfo Beschreibung: Eigenschaften: private ConfigFile configFile OptionInfo - configFile : ConfigFile + Section : String + Name : String public String Section + DefaultValue : String + Value : String + OptionInfo (String section, String name, String defaultValue, ConfigFile configFile): void public String Name public String DefaultValue public String Value

Konstruktoren:

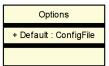
public OptionInfo (String section, String name, String defaultValue, ConfigFile configFile)

3.1.59 Klasse Options

Beschreibung:

Eigenschaften:

public ConfigFile Default



3.1.60 Klasse PauseDialog

Beschreibung:

Eigenschaften:

private VerticalMenu pauseMenu

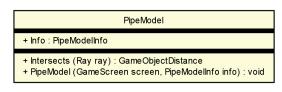


3.1.61 Klasse PipeModel

Beschreibung:

Eigenschaften:

public PipeModelInfo Info



Konstruktoren:

public PipeModel (GameScreen screen, PipeModelInfo info)

Methoden:

public GameObjectDistance Intersects (Ray ray)

3.1.62 Klasse PipeModelInfo

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:

 ${\bf public~GameObjectInfo~Info}$

public Knot Knot

public World World

PipeMovement

PipeModelInfo

+ PipeModelInfo (Knot knot, Edge edge) : void

+ Edge : Edge + Knot : Knot

+ PositionFrom : Vector3 + PositionTo : Vector3

- + Info : GameObjectInfo
- + Knot : Knot + World : World
- + Center (): Vector3
- + Intersects (Ray Ray) : GameObjectDistance
- + Update (GameTime GameTime) : void
- $+\ Pipe Movement\ (Game State,\ World,\ Game Object Info): void$
- + GetEnumerator () : IEnumerator
- + Draw (GameTime GameTime) : void

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) Klasse PrinterIO 3.1.64Beschreibung: Eigenschaften: public IEnumerable; Edge; Edges public String Name public KnotMetaData Meta Konstruktoren: public PrinterIO (string path) Methoden: public void Save (Knot knot)

Klasse ProfileSettingsScreen

3.1.65

Beschreibung:

PrinterIO

+ Edges: IEnumerable<Edge>
+ Name: String
+ Meta: KnotMetaData

+ Save (Knot knot): void
+ PrinterIO (string path): void

Eigenschaften: **ProfileSettingsScreen** protected void settingsMenu # settingsMenu : void + Update (GameTime time) : void + Entered (GameScreen previousScreen, GameTime GameTime) : void Methoden: public void Update (GameTime time) public void Entered (GameScreen previousScreen, GameTime GameTime) 3.1.66 Klasse RenderEffect Beschreibung: Eigenschaften: public RenderTarget2D RenderTarget RenderEffect + RenderTarget : RenderTarget2D # screen : GameScreen # spriteBatch : SpriteBatch protected GameScreen screen + Begin (GameTime) : void + End (GameTime) : void + Draw Model (GameTime, GameModel GameModel) : void protected SpriteBatch spriteBatch + RemapModel (GameModel GameModel) : void # Draw RenderTarget (GameTime) : void Methoden: public void Begin (GameTime)

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:

 ${\bf public} \ {\bf IRenderEffect} \ {\bf CurrentEffect}$

private IRenderEffect DefaultEffect

RenderEffectStack
+ CurrentEffect : IRenderEffect - DefaultEffect : IRenderEffect
+ Pop () : RenderEffect + Push (RenderEffect effect) : void + RenderEffectStack (RenderEffect defaultEffect) : void

Konstruktoren:

public RenderEffectStack (IRenderEffect defaultEffect)

Methoden:

public IRenderEffect Pop ()

public void Push (IRenderEffect effect)

3.1.68 Klasse SettingsScreen

Beschreibung:

Eigenschaften:

protected void navigation

SettingsScreen # navigation : void + Update (GameTime time) : void + Entered (GameScreen previousScreen, GameTime time) : void

${\bf Methoden:}$

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

Shadow GameModel

- + Shadow Color: Color
- + Shadow Alpha: float
- + Shadow GameModel (GameScreen sreen, GameModel decoratedModel) : void
- + Draw (GameTime GameTime): void

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

Shadow GameObject

- + Info : GameObjectInfo
- + World : World
- + Shadow Position : Vector3 + OriginalPosition : Vector3
- + Center (): Vector3
- + Update (GameTime GameTime) : void
- + Draw (GameTime GameTime) : void
- + Intersects (Ray Ray) : GameObjectDistance
- + Shadow GameObject (GameScreen screen, IGameObject decoratedObj) : void

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

SliderItem + Value : Integer + MinValue : Integer + Max Value : Integer

3.1.72Klasse StandardEffect

Beschreibung:

Konstruktoren:

public StandardEffect (GameScreen screen)

StandardEffect # Draw RenderTarget (GameTime): void + StandardEffect (GameScreen screen) : void

Methoden:

protected void DrawRenderTarget (GameTime)

3.1.73 Klasse T Beschreibung: Eigenschaften:

public float \mathbf{Y}

public float X

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

Eigenschaften: TextInputDialog public String InputText + InputText : String TutorialChallengeMo-3.1.75Klasse de Beschreibung: Methoden: public void Entered (GameScreen previousScreen, GameTime GameInen)ode + Entered (GameScreen previousScreen, GameTime GameTime): void Klasse VerticalMenu 3.1.76 Beschreibung: Methoden: public void AlignItems () VerticalMenu + AlignItems (): void 3.1.77Klasse Widget Beschreibung: Eigenschaften: public Vector2 RelativeSize Widget + RelativeSize : Vector2 + RelativePosition: Vector2 + lsVisible : bool + BackgroundColor : Func<Color> public Vector2 RelativePosition + ForegroundColor : Func<Color> + AlignX : HorizontalAlignment + AlignY: VerticalAlignment

+ BoundingBox () : Rectangle

+ Widget (GameScreen screen, DisplayLayer draw Order): void

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 () Klasse WidgetKeyHandler 3.1.78 Beschreibung: Methoden: public void Update () WidgetKeyHandler + Update () : void ${\bf Klasse\ Widget Mouse Handler}$ 3.1.79 Beschreibung: Methoden: public void Update () WidgetMouseHandler + Update () : void Klasse World 3.1.80

Eigenschaften: public void Camera public List; IGameObject; Camera public IGameObject SelectedObject public IRenderEffect CurrentEffect Konstruktoren: public World (GameScreen screen)

World

- + Camera : void
- + Camera : List<lGameObject>
- + SelectedObject : IGameObject + CurrentEffect : IRenderEffect
- + Update (GameTime GameTime) : void
- + Draw (GameTime GameTime) : void + World (GameScreen screen) : void
- + GetEnumerator () : IEnumerator

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)

XNA . Draw able Game Component

- + Draw (GameTime time): void
- + Update (GameTime time) : void

Klasse XNA.Game 3.1.82

Methoden: XNA.Game public void Game () + Game (): void + Initialize (): void + LoadContent (): void + UnloadContent (): void public void Initialize () + Game (GameTime) : void + Draw (GameTime): void 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) XNA.GameComponent + Update (GameTime time) : void Schnittstellen 3.2 Schnittstelle IChallengeIO 3.2.1Beschreibung: 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 ()

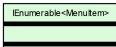
3.2.5 Schnittstelle IEquatable

IChallengeIO

- + Highscore : IEnumerable<KeyValuePair<String, Integer>>
- + Name : String
- + StartKnot : Knot
- + TargetKnot : Knot
- + Meta : ChallengeMetaData

+ Save (Challenge challenge) : void





Beschreibung: IEnumerable1 + GetEnumerator (): IEnumerator Methoden: public Boolean Equals (T other) 3.2.6 Schnittstelle IEquatable1 Beschreibung: |Equatable<GameObjectInfo> Methoden: public Boolean Equals (T other) Schnittstelle IGameObject 3.2.7Beschreibung: IEquatable1 Eigenschaften: + Equals (Tother) : Boolean ${\bf public~GameObjectInfo~Info}$ public World World Methoden: public Vector3 Center () **IGameObject** + Info : GameObjectInfo + World : World + Center (): Vector3 public void Update (GameTime time) + Update (GameTime time) : void + Draw (GameTime time) : void + Intersects (Ray ray) : GameObjectDistance public void Draw (GameTime time)

public GameObjectDistance Intersects (Ray ray)

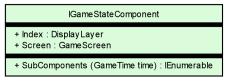
3.2.8 Schnittstelle IGameStateComponent

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 ()

3.2.11Schnittstelle IKnotIO

Beschreibung:

Eigenschaften:

public IEnumerable; Edge; Edges

public String Name

public KnotMetaData Meta

Methoden:

public void Save (Knot knot)

Schnittstelle IMouseEventListener 3.2.12

Beschreibung:

Eigenschaften:

public DisplayLayer Index

public Boolean IsMouseEventEnabled

Methoden:

public void OnLeftClick ()

public void OnRightClick ()

public Rectangle Bounds ()

lKeyEventListener

- + Index : DisplayLayer
- + IsKeyEventEnabled : Boolean
- + ValidKeys : List<Keys>
- + OnKeyEvent (): void

IKnotIO

- + Edges : IEnumerable<Edge>
- + Name : String + Meta : KnotMetaData
- + Save (Knot knot) : void

MouseEventListener

- + Index : DisplayLayer
- + IsMouseEventEnabled : Boolean
- + OnLeftClick (): void
- + OnRightClick (): void
- + Bounds (): Rectangle

${\bf 3.2.13}\quad {\bf Schnittstelle~XNA.IGame Component}$

Beschreibung:

public Edge EdgeTo

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 Knot Knot public Vector3 Position ${\bf public\ Pipe Model Info\ Info}$ public NodeModelInfo Info public List; IGameObject; Camera ${\bf public} \ {\bf IGameObject} \ {\bf SelectedObject}$ ${\bf public} \ {\bf IRenderEffect} \ {\bf CurrentEffect}$ public GameScreen Screen public DisplayLayer Index ${\color{red} \textbf{public DisplayLayer Index}}$ public GameScreen Screen public Vector3 Position public Vector3 Target public float FoV

 ${\color{red}\mathbf{public}}\ \mathbf{Matrix}\ \mathbf{ViewMatrix}$

```
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

Abläufe

4.1 Sequenzdiagramme

Klassenindex

Anmerkungen

Glossar

Test Test

7.1 Fachausdrücke

Test (Test-Beschreibung) ... 50

7.2 Abkürzungen

 $\mathbf{Test} \ \mathrm{Test} \ 50$