UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Gruppenfindung

Einführung in FlatRedBall

FlatRedBall ist

- 2.5D game engine
- Developer Kit
  - SpriteEditor
    - Plazieren und Editieren von Sprites
    - Erstellen von Spiele-Level, GUIs
    - http://www.flatredball.com/frb/forum/viewtopic.php?t=436
  - Animation Editor
    - Animationsketten erstellen
- Community
  - http://www.flatredball.com/frb/forum/
- Template für XNA
- Frei verfügbar

- Sprite Editor
  - Laden und positionieren von Sprites in der Szene
    - Move, scale, rotate
    - Copy, delete, undo
  - Gruppierung von Sprites
    - Group control mode / Hierarchy control mode
  - Kamera
    - 3D-Camera / orthogonal view
  - Farbmanipulation
    - Color, blend
  - Texturen

- Erstellen eines neuen Projektes
  - Starten des XNA Game Studio Express
  - Neues Projekt
  - Unter „Meine Vorlagen“ „FlatRedBall XNA Template“ auswählen
  - Namen festlegen
  - Datei > Alles speichern

Einbinden einer mit dem SpriteEditor erstellten Szene

```
using FlatRedBall.Content;

// deklarieren einer Klassenvariablen
Scene _myScene;

// laden in LoadGraphicsContent
_myScene =
    SpriteEditorScene.FromFile(@"..\..\..\content\scenes\myTest.scn
    x").ToScene("myTest.scnx");
SpriteManager.AddScene(_myScene);

// freigeben in UnloadGraphicsContent
_myScene.Clear();
```

Basisklasse Screens

- Lädt und zerstört Ressourcen
  - .scnx Files
  - Erstellt die Oberfläche
  - Kontrolliert den Programmfluß

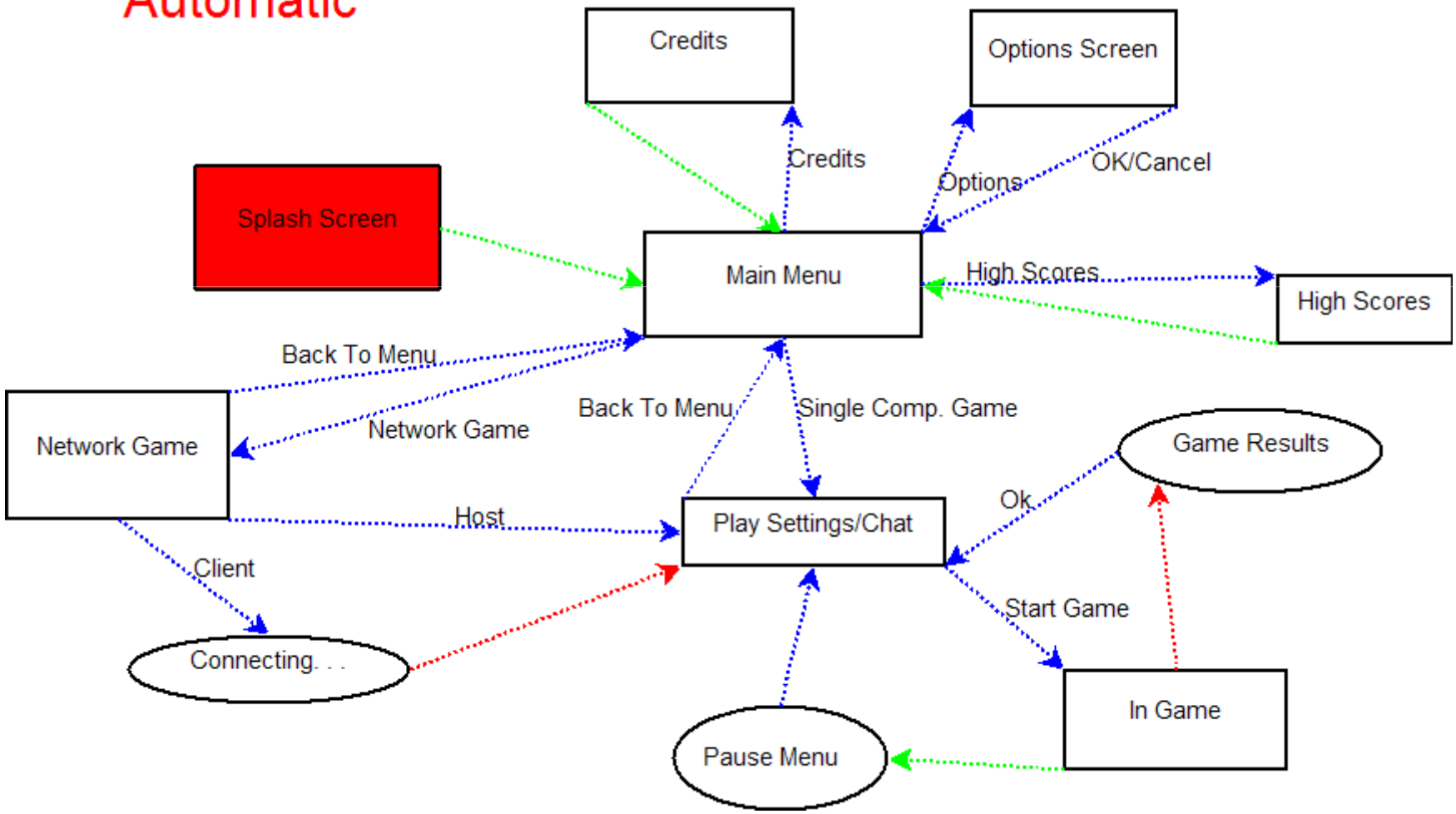
- Kopieren der Klasse ScreenTemplate
- Aufruf des Konstruktors

```
public SplashScreen() :
    base(@"Assets\Scenes\SplashScreen.scnx", "SplashScreen")
```

- Laden des Screens in der Initialize-Methode

```
Screens.ScreenManager.Start("frb_test.Screens.SplashScreen");
```

Selectable Links
Button Presses
Automatic



Verarbeitung von Benutzereingaben

```
using FlatRedBall.Input;
using Microsoft.Xna.Framework.Input; // for the Keys enum

if(InputManager.Keyboard.KeyPushed(Keys.A))
                                        KeyDown
                                        KeyReleased
                                        KeyTyped
```

Texteingaben

```
Text _myText; // Klassenweite Deklaration

// Initialisierung
_myText = TextManager.AddText("Mein Text : ");
_myText.HorizontalAlignment = HorizontalAlignment.Center;
_myText.Y = 4;
_myText.X = 1;

// Textdarstellung in Methode Activity
_myText.DisplayText +=
   InputManager.Keyboard.GetStringTyped();
```

Spritemanipulationen

- Screenklasse stellt Variablen zur Verfügung
  - protected SpriteList mSprites;
  - protected List<SpriteGrid> mSpriteGrids;
  - protected PositionedObjectList<SpriteFrame> mSpriteFrames;
  - protected PositionedObjectList<Text> mTexts;
  - …

```
if (InputManager.Keyboard.KeyPushed(Keys.D1))
{
    mSprites[0].RotationXVelocity += 1;
 }

mSprites.FindByName("Spritename"); // Sprite nach Namen
```

Kollisionserkennung

```
// FlatRedBall.Math.Geometry.AxisAlignedRectangle für jedes zu
    kontrollierende Sprite

FlatRedBall.Math.Geometry.AxisAlignedRectangle rectangle = new
    FlatRedBall.Math.Geometry.AxisAlignedRectangle();
rectangle.ScaleX = mSprites[1].ScaleX;
rectangle.ScaleY = mSprites[1].ScaleY;
mSprites[1].SetCollision(rectangle);

// Kollisionsüberprüfung in Methode Activity
if (mSprites[1].CollideAgainst(mSprites[2]))
{
    // Reaktion auf die Kollision
}
```