**-= 22.03.2019 =-**

**First Session**

**TNG**

- DEF

- STOCK

- MTH (VEC, MATR, RAY, Hit?)

- WinApi

- ANIM (Anim\*\*\*)

- Timer

- Input: Keyboard (Async + Sync<-WM\_CHAR), Mouse, Joystick

- Unit + Unit Set

- Rnd: Base(OpenGL)

- Resourse

- Fonts

- Primitives + Support different vertex types

- Textures + Support different format

- Drawing to texture

- Shaders + ??? Variable mapping

- Materials (BRDF support)

- Lighting + Illumination (<-Materials) (deferred lighting)

- Shadow (<-Lighting)

- Rendering pipeline (FBO...)

- Deferred rendering (+(\*), GetDistance, Sort)

- Transparency

- Topology | G3DM ... | ...

===

- Particle queue [??? + shader]

- Noise [??? + shader]

- Scene graph ???

. . .

===

- Sample 1 level.

**-= 25.03.2019 =-**

**Second Session**

VG4 - DEF, WinApi, Timer

IK5 - STOCK

EH5 - MTH (VEC, MATR, RAY, Hit?) --> Noise [??? + shader]

SK5 - ANIM (Anim\*\*\*), Deferred rendering (+(\*), GetDistance, Sort), Unit + Unit Set

AF5 - Input: Keyboard (Async + Sync<-WM\_CHAR), Mouse, Joystick

AV3, SK4 - Rnd: Base(OpenGL), Rendering pipeline (FBO...)

AS4 - Resourse, Materials (BRDF support), Textures (Support different format) --> Drawing to texture

AG6 - Shaders + ??? Variable mapping

VD6, PD2 - Shadow (<-Lighting), Lighting + Illumination (<-Materials) (deferred lighting)

SK3 - Primitives + Support different vertex types

MM2 - Topology | G3DM ... | ...

**The rest:**

- Fonts

- Particle queue [??? + shader]

- Transparency

**-= 26.03.2019 =-**

**Meetup (12:47)**

**- SK5:**

**Anim:**

**For subsystem implementation:**

**- add to tngANIM:**

**typedef struct tagtngANIM**

**{**

**. . .**

**tngSUBSYSTEM\_PRIMITIVES;**

**tngSUBSYSTEM\_TIMER;**

**. . .**

**} tngANIM;**

**#define TNG() (&TNG\_Anim)**

**extern TNG\_Anim;**

**tng\_timer.h:**

**typedef struct tagtngSUBSYSTEM\_TIMER**

**{**

**DBL GlobalTime, Time, DeltaTime;**

**VOID (\*TimerResponse)( VOID );**

**} tngSUBSYSTEM\_TIMER;**

**-----------**

**tng\_timer.c:**

**static VOID TNG\_TimerResponse( VOID )**

**{**

**}**

**VOID TNG\_SubsystemTimerInit( VOID )**

**{**

**TNG()->TimerResponse = TNG\_TimerResponse;**

**}**

**--- Alternate ---:**

**typedef struct tagtngANIM**

**{**

**. . .**

**tngSUBSYSTEM\_PRIMITIVES;**

**union**

**{**

**tngSUBSYSTEM\_TIMER SubsystemTimer;**

**tngSUBSYSTEM\_TIMER;**

**};**

**. . .**

**} tngANIM;**

**tng\_timer.c:**

**static VOID TNG\_TimerResponse( VOID )**

**{**

**}**

**static tngSUBSYSTEM\_TIMER Subsystem =**

**{**

**0, 0, 0,**

**TNG\_TimerResponse**

**};**

**VOID TNG\_SubsystemTimerInit( VOID )**

**{**

**TNG()->SubsystemTimer = Subsystem;**

**}**

**ANIM:**

**static VOID (\*TNG\_SystemRegistration[])( VOID ) =**

**{**

**TNG\_SubsystemTimerInit,**

**}**

**--- About primitives:**

**struct**

**{**

**VEC Pos;**

**INT NoTex;**

**FLOAT Size;**

**} PUK;**

**UNIT...:**

**tngPRIM \*Pr;**

**Init:**

**PUK p[2] =**

**{**

**{{0, 0, 0}, 1, 30.47},**

**{{1, 0, 0}, 2, 25.18}**

**};**

**Uni->Pr = Ani->PrimCreate(TNG\_PRIM\_POINT, "f3i1f1", p, 2, NULL, 0);**

**... Ani->Time ...**

**-= 28.03.2019 =-**

**\Units\**

**\NSF Samples\**

**\PD2\**

**tng\_unit\_pd2\_bounce\_ball.c**

**tng\_unit\_pd2\_car.c**

**\SK4\**

**--- later:**

**\Military Unit Set**

**\Tank**

**\BTR**

**\Environment\**

**\Water**

**\Plant**

**\Trees**

**\Grass**

**\**

**-->**

**\Units\**

**tng\_units\_register.c –**

**tnf\_UnitRegister =**

**{**

**{"PD2 Car", TNG\_UnitCreatePD2Car},**

**{}**

**}**

**tng\_units\_register.h:**

**#include "tng\_def.h"**

**/\* PD2 sample for car drawing demonstration unit creation function \*/**

**VOID TNG\_UnitCreatePD2Car( VOID );**

**\SRC\**

**. . .**

**\BIN\**

**\SHADERS\**

**\Default\**

**\RenderTarget**

**\FONTS\**

**\MODELS\**

**\NSF Samples\**

**\...**

**\Military Unit Set**

**\..**

**\TEXTURES\**

**\Lens Flares**

**Flare0.tga**

**. . .**

**\**

**-= 29.03.2019=-**

**:: 14:27**

**DeferredRendering:**

**Unit:**

**Render:**

**/\* Draw object . . . \*/**

**TNG()->RegisterDeferRendering(Uni);**

**DeferRender:**

**/\* Draw object deferred part . . . \*/**

**tngRTARGET \*Trg = TNG()->TargetCreate(1024, 1024, "DA(D24)");**

**. . .**

**TNG()->TargetApply(Trg);**

**TNG()->TargetApply(NULL);**

**----------------------------------------**

**RenderInit:**

**RenderTargetPass1 = TNG()->TargetCreate(1, 1, "6xCA(RGBA32F)","D()");**

**RenderTargetPass2 = TNG()->TargetCreate(1, 1, "CA(RGBA32F)","D()");**

**RenderStart:**

**glClear();**

**TNG()->TargetApply(RenderTargetPass1);**

**RenderStartDefer:**

**TNG()->TargetApply(RenderTargetPass2);**

**tngMATERIAL \*TNG\_TargetMtl . . .;**

**TNG\_TargetMtl->Tex[0] = RenderTargetPass1->Attachments[0];**

**. . .**

**TNG()->MaterialApply(TNG\_TargetMtl);**

**glDrawArrays(GL\_POINTS, 0, 1);**

**RenderEnd:**

**TNG()->TargetApply(NULL);**

**tngMATERIAL \*TNG\_TargetMtl . . .;**

**TNG\_TargetMtl->Tex[0] = RenderTargetPass2->Attachments[0];**

**. . .**

**TNG()->MaterialApply(TNG\_TargetMtl);**

**glDrawArrays(GL\_POINTS, 0, 1);**

tngSUBSYSTEM\_FBO->

**tngTARGET**

**{**

**INT W, H, NumOfA;**

**tngTEX\_FMT AttachmentsFmt[MAX];**

**tngTEXTURE \*Attachments[MAX];**

**UINT FBO, RBuf;**

**}**

**+ STOCK Targets;**

**tngTARGET \* TargetCreate(W, H, "CA(RGBA32F)DA(D24)RB(D)");**

**TargetFree(Trg);**

**TargetResize(Trg, NewW, NewH);**

**--------------------------------------**

**RenderPipelineStart();**

**RenderPipelineStartDeffer();**

**RenderPipelineStartLast();**

**RenderPipelineEnd();**

**tngSUBSYSTEM\_PIPELINE**

**tngSUBSYSTEM\_TARGET**

**================================= Primitives:**

**deferred primitive rendering:**

**PrimDrawDeffer()**

**STOCK PrimDrawDefered; <-- tngPRIM \* + Cam**

**==================================== CS**

**Название (рус/англ):**

**Abstract (аннотация)**

**1 предложение на русском и английском (развернутое).**

**Краткие тезисы (текст на 1 страницу – кратко обо всем):**

**Постановка задачи, как решали, как разбит проект, в чем преимущества, решения и т.п.**

**Полный текст работы:**

**Введение.**

**[вода + постановка задачи]**

**[общая структура проекта]**

**Описание проекта**

**-- по каждой части:**

**\* Система анимации**

**- [какие части]**

**- общая концепция (input/timer/Stocks)**

**\* Система вывода**

**- rendering pipeline**

**- resources**

**\* Primitives/Shaders/Textures/Materials/Lights**

**- G-buffers**

**- Topology**

**-**

**\* Вспомогательные модули**

**- Mth/Scanner/Audio**

**Заключение**

**[Что и зачем, какие результаты и т.п. Что нового и интересного]**

**Литература**

**Статистика:**

**-- по проекту**

**-- по людям**

**\*\*\*Каждая часть:**

**>>>Суть и концепция**

**>>>Полное развернутое описание**

**>>>2-3 схемы работы**

**>>>2-3 screen-shots**

**x12 человек**

**---------------------------**

**Current unclosed tasks:**

**-- audio: complete**

**-- input: GetChar, sample, UnitControl (WASD, Arrows, ...)**

**-- animation: complete**

**-- topology: TANGENT SPACE???**

**-- lights: shadow (DIR/SPOT)**

**-- SCAN: unsafe code.**

**-= 10.04.2019 =-**

**tngCUBEMAP \*SkyBox = CubeMapLoad("NABULA.BMP");**

**tngTEXTURE \*SkyBox = TextureCubeMapLoad("NABULA.BMP");**

**-->**

**\BIN\TEXTURE\NABULA\**

**XNEG.BMP**

**YNEG.BMP**

**. . .**

**tngEMITTER**

**InjectionSpeed**

**InjectionRest**

**ParticleQueue**

**tngPARTICLE**

**TotalTime, Age**

**EmitterCreate**

**EmitterFree**

**ParticleNew(Emi)**

**ParticlesHandleTime(Emi, DeltaTime)**

**Particles:**

**FLT**

**TotalLife**

**Age/LifeTime**

**V**

**Pos**

**CurPos = StartPos + V \* Age**

**CurTwist = StartTwist + Vtwist \* Age**