

*Zukunft in  
Bewegung*



# NGAI



Technische **Hochschule**  
**Ingolstadt**

Fakultät für Elektrotechnik  
und Informatik

*Projekt RTS AI*

*auf Galaxy Basis für Starcraft 2*

**Michael Ernst M.Sc**



# Vorstellung

Michael Ernst M.Sc. (Michael.Ernst@thi.de)



- **Diplom Informatik (Datenkommunikation)**
- **Master of Science (Informatik)**
- **Doktorand TU Chemnitz**
  - Realtime Hierarchical Asynchronous Multicore Scheduling
- **Lehrbeauftragter EI**
  - Bussysteme, Grundlagen der Programmierung I / II, Grafikesysteme, 3D Grafik ...
- **Projekte Seminare für Spiele-Entwicklung für Fak. EI**

- **Doppelte ECTS -> Doppelte Wertung <- Arbeit !**
- **Projekt Bericht und Präsentation**
- **Einzelbewertung ?**
  
- **Dienstag 14:00 – 17.20**
  - Für Besprechungen
  - Für Ideen-Management
  - Für gemeinsames Brainstorming
  - Gemeinsames Entwickeln ?



# Next Gen AI (NGAI)

- Erlernen des Arbeiten im Team
- Erlernen von AI Entwicklung
- Erlernen von Galaxy
  
- Es soll in Teamarbeit eine 1on1 AI für das PC-Spiel „Starcraft 2 – LotV“ entwickelt werden. Welche entweder
  - AI vs AI als Demonstartor spielen kann (Möglichst UNMENSCHLICH, IMBALANCED)
  - AI vs Mensch als 1on1 Train dient. (Möglichst Menschlich)

- **Eigener Bewertungsbogen für jeden Studenten (Wöchentlich aktualisieren)**
- **Bewertungskriterien**
  - Teamarbeit
  - Kommunikation
  - Ideen-einbringung, Diskussion, Mitarbeit
  - Nutzung der gegebenen Tools (Wiki, Trello, Git, etc)
  - Entwicklung
  - Dokumentation
  - Ergebnis



Opening

Legacy of the Void

# *Starcraft 2 - Races*

## Zerg



# ZERG

## TECH-TREE AND UNIT COUNTERS



# *Starcraft 2 - Races*

## Terran



# TERAN

## TECH-TREE AND UNIT COUNTERS




# *Starcraft 2 - Races*

*Prot*



# PROTOSS

## TECH-TREE AND UNIT COUNTERS



# Wahl der Teams ?



Quelle: Pikmin 3



Technische **Hochschule**  
**Ingolstadt**

Fakultät für Elektrotechnik  
und Informatik

*Zukunft in  
Bewegung*

## *Projekt Management*

**Michael Ernst M.Sc.**





- **Zerlegung:**
  - Der Weg zur Lösung wird in einzelne gut überprüfbare Schritte zerlegt.
- **Transparenz:**
  - Der Fortschritt und die Hindernisse eines Projektes werden wöchentlich und für alle sichtbar festgehalten. (Docu, Präsentationen ?)
- **Überprüfung:**
  - In regelmäßigen Abständen werden Produktfunktionalitäten geliefert und vom Team beurteilt.
- **Anpassung:**
  - Die Anforderungen an das Produkt werden nicht ein für alle Mal festgelegt, sondern nach jeder Lieferung neu bewertet und bei Bedarf durch weitere Iteration angepasst.

Quelle: Wikipedia, Scrum



*Zukunft in  
Bewegung*



Technische **Hochschule**  
**Ingolstadt**

Fakultät für Elektrotechnik  
und Informatik

# *Code Versionierung und Verwaltung*

**Michael Ernst M.Sc**



# Code Verwaltung

Git oder SVN ?



## ■ Problem:

- Hosting ?
  - Sicherheit, Manipulation, Open Src, Dropbox ?
- Einzelbewertung
  - Wer hat was (nicht) gemacht ?
- Git
  - Dezentral, komplex
  - Tortoise Git & mySYS
- SVN
  - Serverbasierend, nicht ganz so komplex
  - Tortoise SVN

- **Tafel -> So funktioniert Git / SVN**

# Code Verwaltung

Wie „commite“ ich richtig ?



Nex	56fb3ee	* mem leak fix in gfxhandler.cpp for Messages	2014-01-27
Nex	c3d4e5b	* lol welcher depp hat den kompletten header ordner in die gitignore reingepackt ?	2014-01-27
Nex	424b1d5	* added memdebug into every cpp file to find mem leaks !	2014-01-27
Nex	d2ec3be	* this patch fixes many memory leaks	2014-01-27
Nex	592c2cf	* added memdebugging for memleaks (Visual Studio 2012 only ) #define _MEMDEBUG	2014-01-27
Dying Devil	52170e3	ich habs gefixed ich habs gefixed!!! :D	2014-01-23
Nex	17df713	* added a first version for DeleteChunk	2014-01-23
Dying Devil	4d383de	added removechunk method, but still buggy...	2014-01-23
Dying Devil	d6cadb9	created features.h is not empty now, everything disabled on default	2014-01-21

- Gleiche Sprache.
- Gleicher Syntax.
- Was wurde geändert ?
- Wie genau sollte geschrieben werden ?
- Wichtig fürs Projekt : Wer war noch beteiligt ?



Nex / NGAI

## Wiki

Clone wiki

Create

NGAI / AI UserData

View History Edit

Grundfunktion:

```
AI GetUserINT( int player, int index);  
AI SetUserINT( int player, int index, int data);
```

Beschreibung: Diese Funktion kann bis zu 256 ints (erster Index = 1) pro verfügbaren Player auf der Map speichern. Vorteil : Global verfügbar und per Player !

Niemals MagicNumbers deswegen :

```
const int g_NGUD_INIT = 1;
```

<line> Beispiel

```
AI SetUserInt(player, c_NGUD_INIT, 1337);  
  
if (AI SetUserInt(player, c_NGUD_INIT) != 1337)  
{  
    bla();  
    bla();  
}
```

Updated 32 minutes ago



Nex / NGAI

## Issues



# Welcome to the issue tracker

Bugs stack up in any code, so track them quickly with Bitbucket.

If you're interested in a more powerful issue tracker, check out [Atlassian JIRA](#).

[Create your first issue](#)

## Forum vs Trello



The screenshot shows the Trello homepage for a user named 'NGAI'. The profile picture is a grey icon with three people. The board name is 'NGAI' with a 'Privat' (Private) status. Below it is the text 'Starcraft 2'. There is a button labeled 'Profil bearbeiten' (Edit Profile). The top navigation bar includes 'Boards' (selected), a search icon, and the 'Trello' logo. To the right are icons for '+' (New Board), a user profile, 'Nex' (Notifications), 'i' (Info), and a red bell icon. Below the profile are tabs for 'Boards', 'Mitglieder', 'Einstellungen', and 'Business Class'. Four colored boxes represent different Starcraft 2 factions: 'AI' (teal), 'Protoss' (blue), 'Terran' (purple), and 'Zerg' (green). A light grey box at the bottom left contains the text 'Neues Board erstellen...' (Create New Board...).

<https://trello.com/invite/ngai2/2f5e6d3a49b40485f9a828f13bdb5615>



Technische **Hochschule**  
**Ingolstadt**

Fakultät für Elektrotechnik  
und Informatik

*Zukunft in  
Bewegung*

*Design*

Michael Ernst M.Sc

15.03.16



- Ideen auf Knopfdruck !



- Wie kann man Ideen austauschen und sinnvoll diskutieren ?

- Wer verhindert gegenseitiges „flamen“ „mobbing“ ?

- Wie bewertet man gute / schlechte Ideen ?

- Demokratie, Anarchie ?



- Galaxy ....
- Was ist Galaxy ?
- IDE vs Notepad
- Erfahrungen im Team mit Scriptsprachen

## ■ Einheitliche Syntax

- if (a > b) { a = 5; } else { a =0; }
- a>b?a=5:a=0;

## ■ Einheitlicher Programmierstyle

- Klammern, Var / Class / Func names,
- namespaces, enums Tabs, new lines, Docu

## ■ Einheitliche Namensgebung

- Unit, Entity, Einheit, Dings ?

## ■ Optimierung vs Geek vs Sauber

- Sollte man überhaupt am Anfang Optimieren ?

## ■ Inline Code Docu & Function / Class Docu und Wiki ?

## ■ Testing und wie und wann commite ich meinen Code ?



```
void Virtualworld::SetChunks(Chunk* _c, int _i)
{
    if (_i % V_WORLD_SIZE != 0)
        _c->SetNeighborChunk(E_DIR_LEFT, GetChunkAt(_i-1));
    if (_i % V_WORLD_SIZE != V_WORLD_SIZE-1)
        _c->SetNeighborChunk(E_DIR_RIGHT, GetChunkAt(_i+1));
    if ((_i % (V_WORLD_SIZE*V_WORLD_SIZE)) + V_WORLD_SIZE < (V_WORLD_SIZE*V_WORLD_SIZE))
        _c->SetNeighborChunk(E_DIR_BACK, GetChunkAt(_i + V_WORLD_SIZE));
    if ((_i % (V_WORLD_SIZE*V_WORLD_SIZE)) - V_WORLD_SIZE >= 0)
        _c->SetNeighborChunk(E_DIR_FRONT, GetChunkAt(_i - V_WORLD_SIZE));
    if (_i >= (V_WORLD_SIZE * V_WORLD_SIZE))
        _c->SetNeighborChunk(E_DIR_BOT, GetChunkAt(_i - (V_WORLD_SIZE*V_WORLD_SIZE)));
    if (_i < (V_WORLD_SIZE * V_WORLD_SIZE * V_WORLD_SIZE - V_WORLD_SIZE))
        _c->SetNeighborChunk(E_DIR_TOP, GetChunkAt(_i+ (V_WORLD_SIZE*V_WORLD_SIZE)));
}
```



```
// prerender and add new Chunks
switch (_dir)
{
    case E_DIR_LEFT:
        // first mark every Chunk to be removed from render process and free the memory of the chunks!
        for (int y = 0; y < V_WORLD_SIZE; y++)
        {
            for (int z = 0; z < V_WORLD_SIZE; z++)
            {
                // generate removeChunk Message and store it in GfxHandler MessageQueue ( x = V_WORLD_SIZE-1 is static)
                Gfxhandler::getHandle()->AddMessage(Gfxhandler::getHandle()
                    ->Generate_RemoveChunkMessage(this->GetChunkAt(V_WORLD_SIZE-1,y,z)));
            }
        }

        // second move the whole scene to the right !
        Gfxhandler::getHandle()->MoveScene(E_DIR_RIGHT, CHUNK_SIZE * CUBE_SIZE);

        // now add Messages to MQ -> to prerender the new Chunks
        for (int y = 0; y < V_WORLD_SIZE; y++)
        {
            for (int z = 0; z < V_WORLD_SIZE; z++)
            {
                this->AddMessage(this->Generate_PrerenderChunkMessage(0,y,z));
            }
        }
    break;
}
```



Technische **Hochschule**  
**Ingolstadt**

Fakultät für Elektrotechnik  
und Informatik

*Zukunft in  
Bewegung*

AI

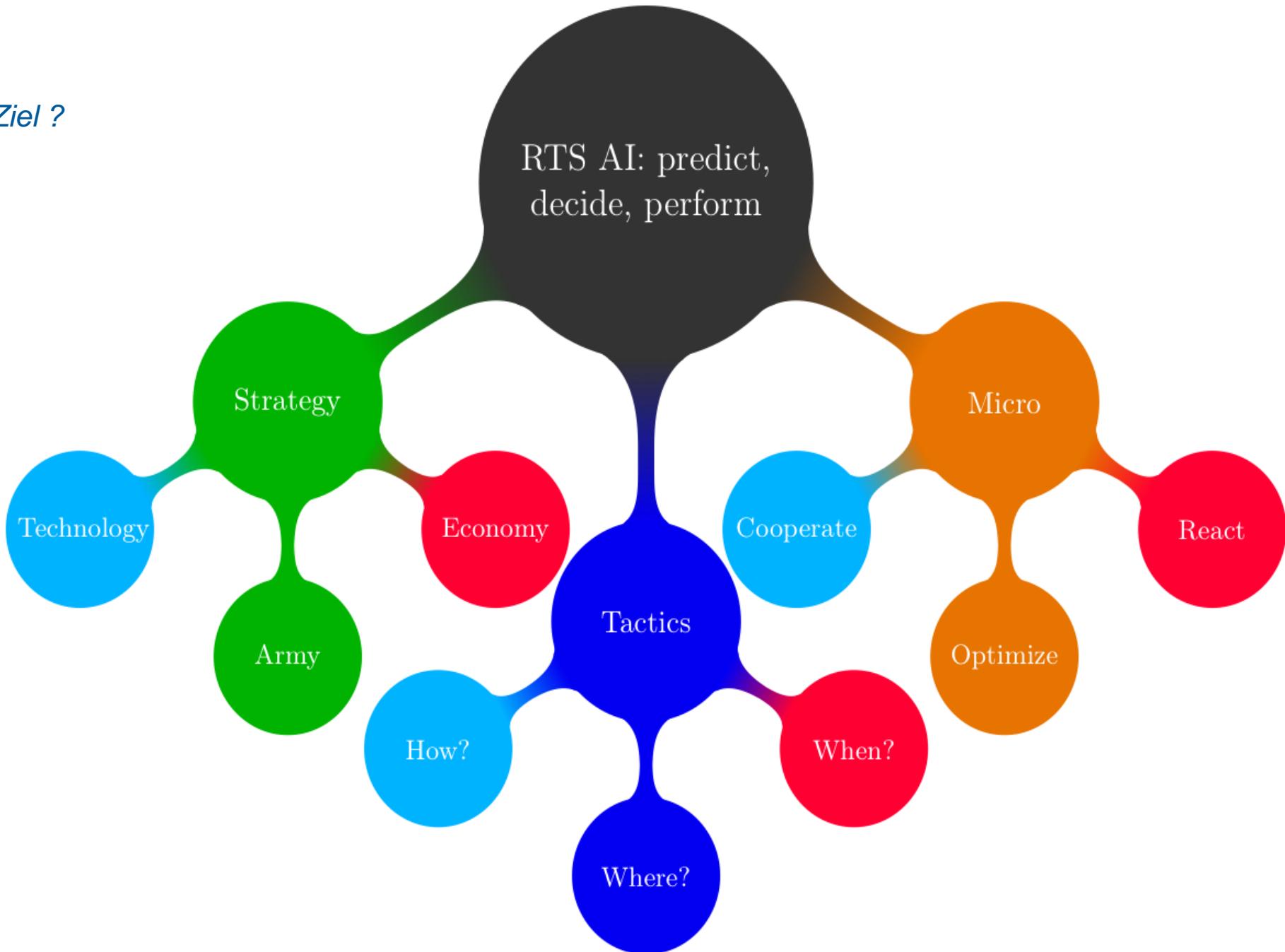
*in Starcraft 2, Void of the Legacy*

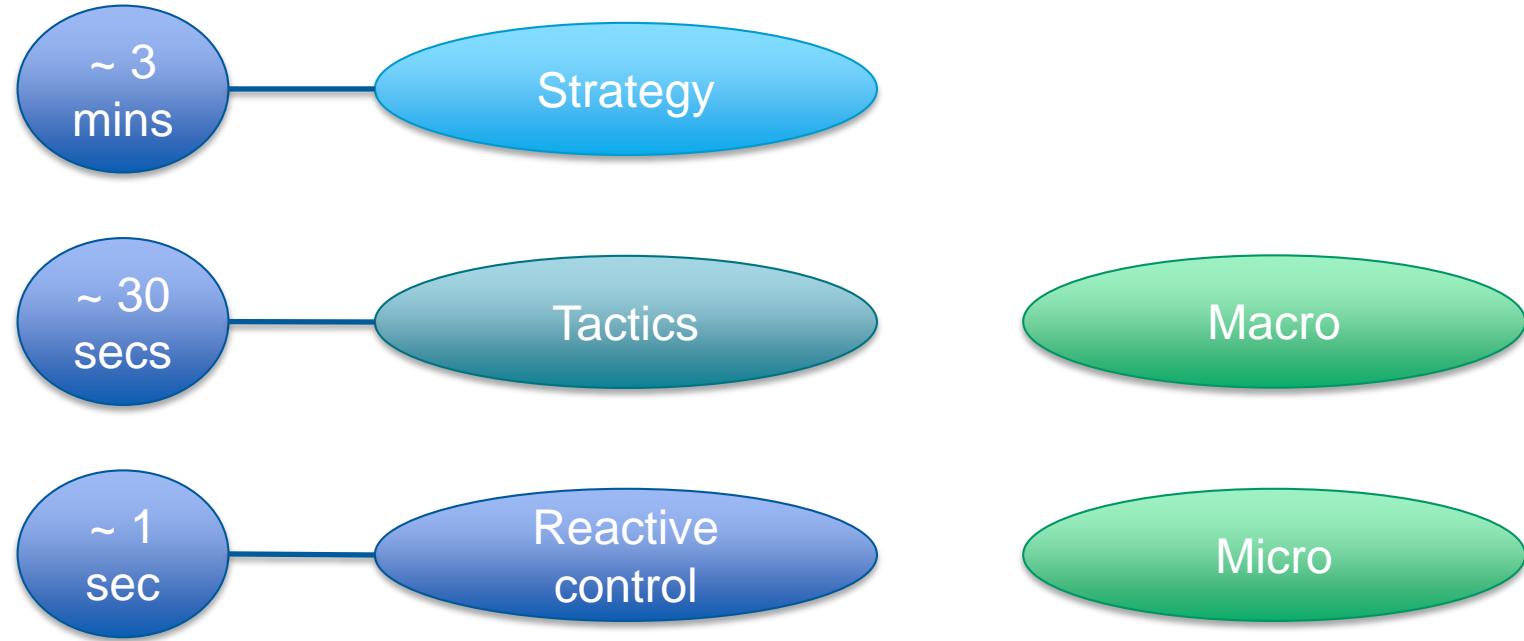
**Michael Ernst**



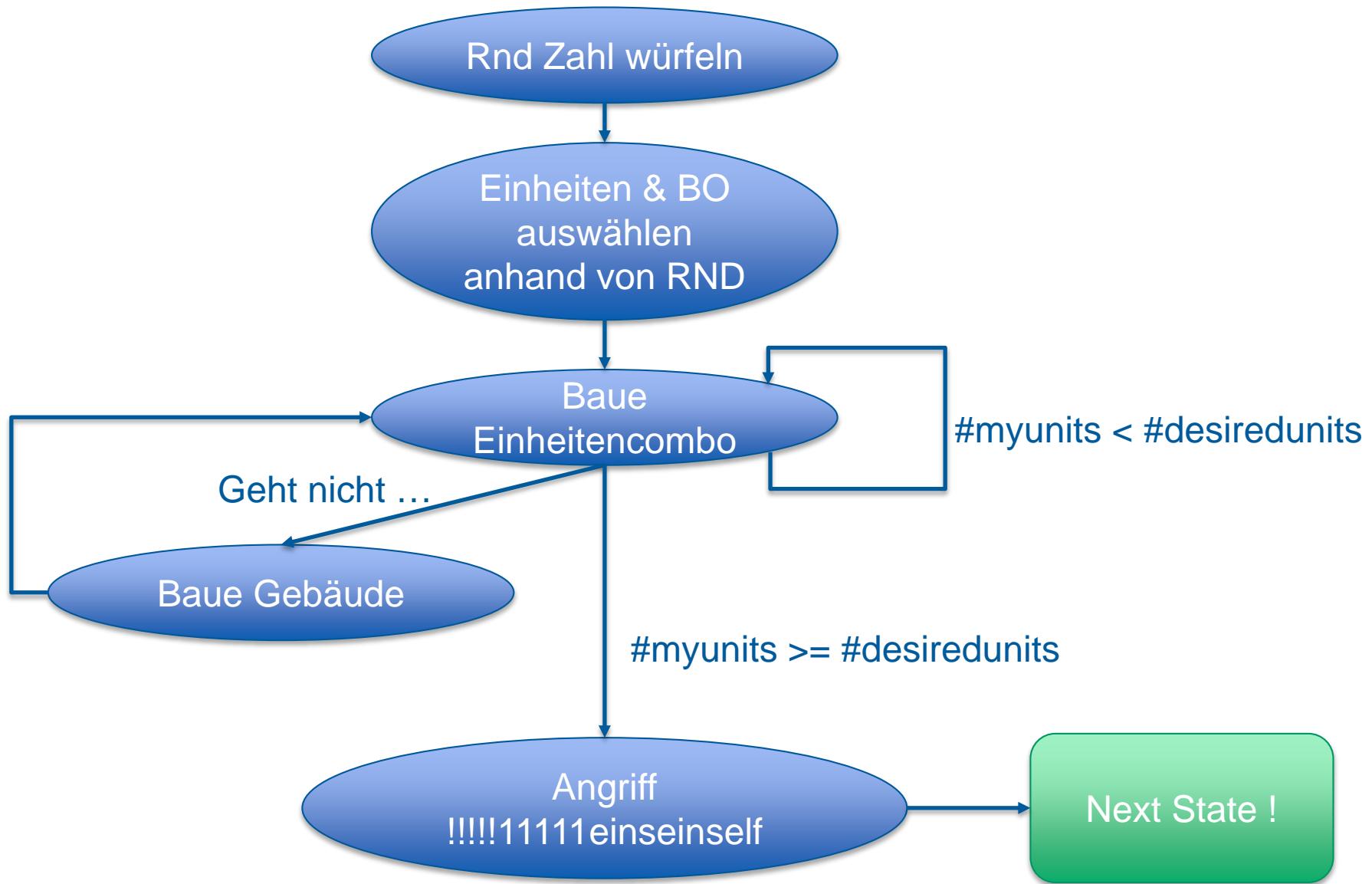
# Motivation

Was ist unser Ziel ?

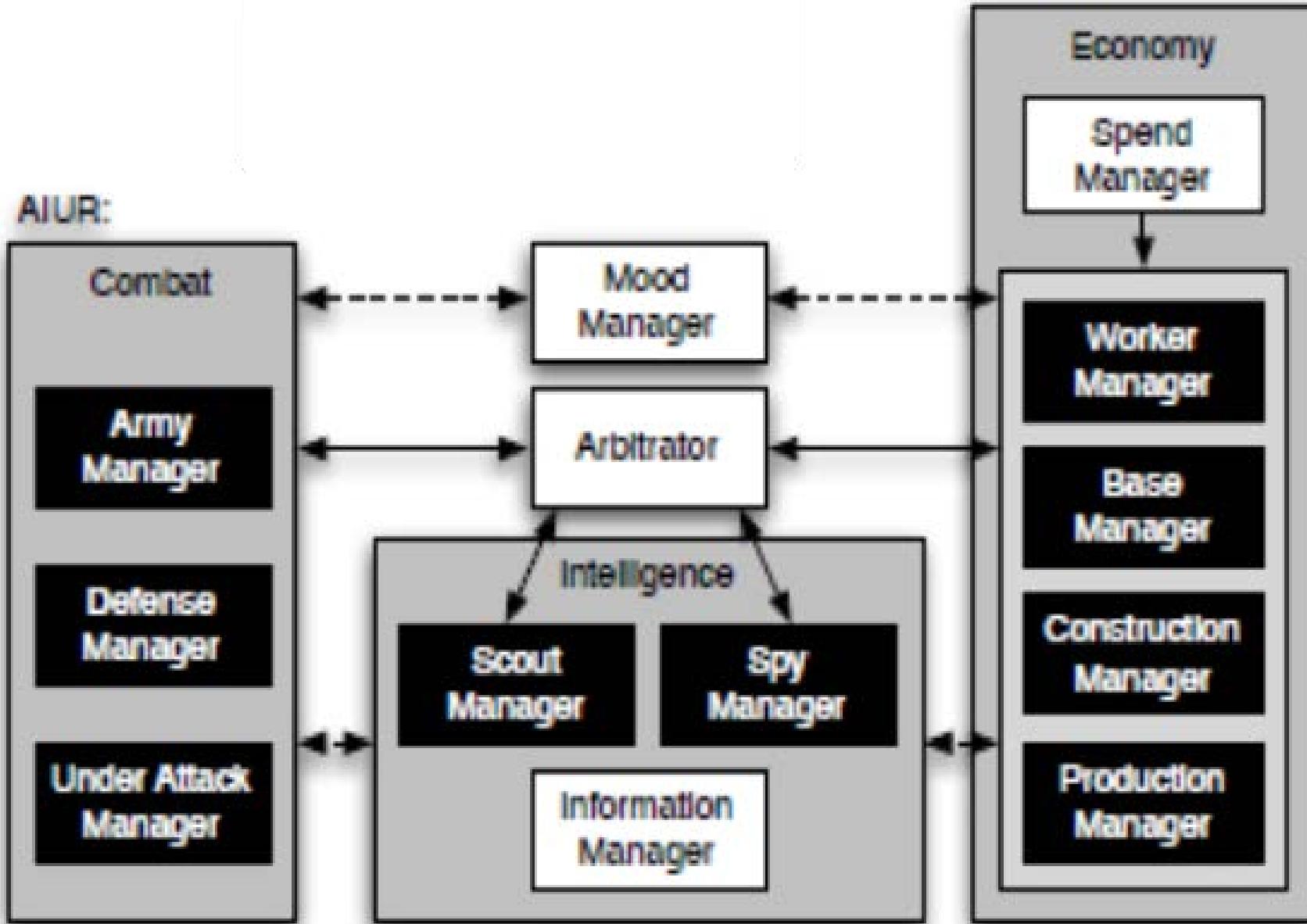




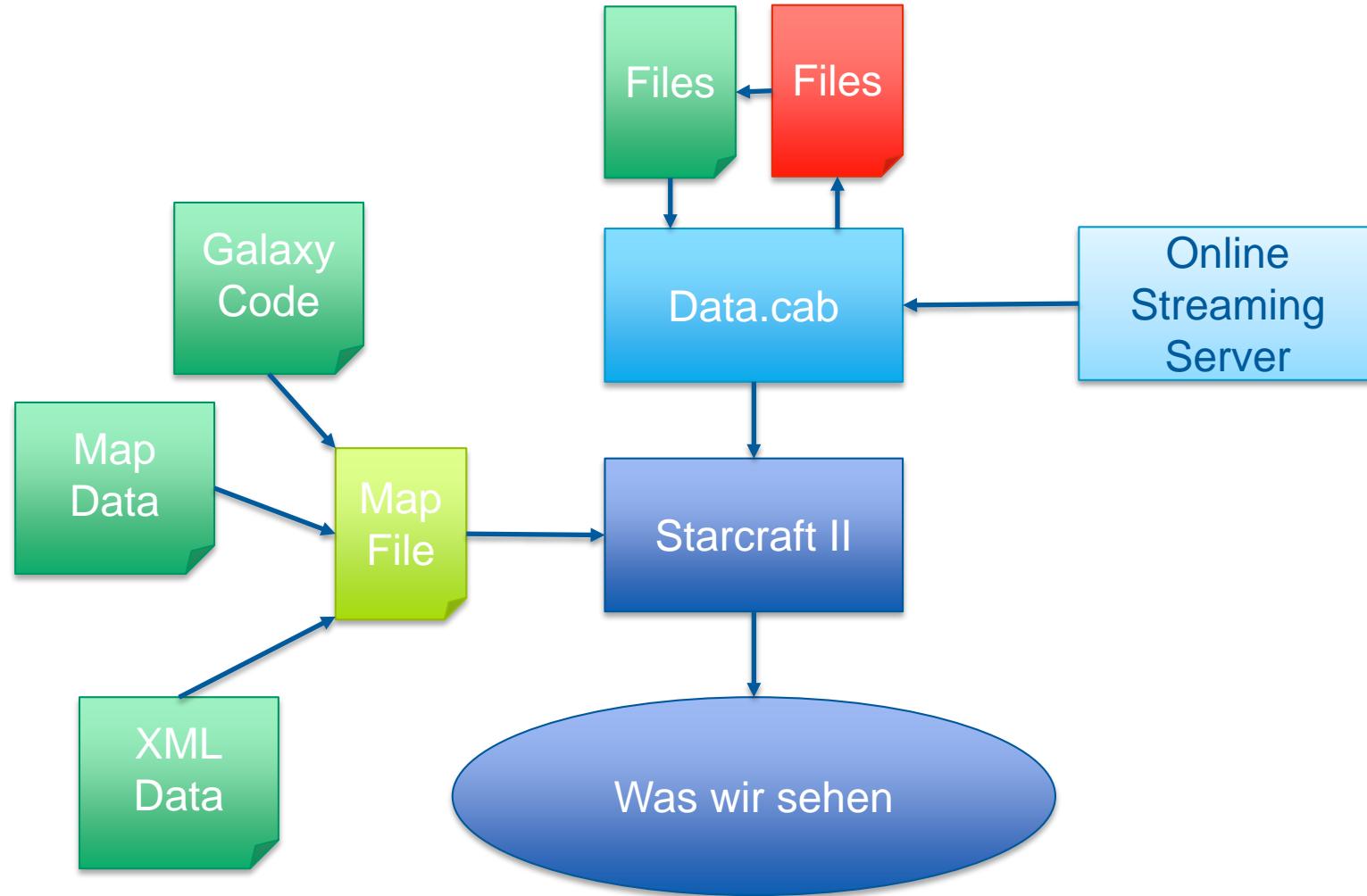
## Aktuelle Ausgangssituation: Die BlizzAI (Original KI von Blizzard)



AIUR:



# Aufbau von Starcraft 2





Technische **Hochschule**  
**Ingolstadt**

Fakultät für Elektrotechnik  
und Informatik

*Zukunft in  
Bewegung*

**Galaxy**

*in Starcraft 2, Void of the Legacy*

**Michael Ernst**





Primitives:

- int (32bit signedGanzzahl)
- Fixed (32bit signed fixed-point) ( Range: -524288 bis +524287,999 )
- Byte (8bit unsigned int)
- bool (8bit true false)
- Complex
  - string, text
  - unit, unitref, unitgroup, unitfilter, wave
  - playergroup
  - order, marker
  - point, region
  - Timer, Trigger
  - structs
- Fake types
  - Player = int
  - Unittype = string
  - Difficulty = int



```
//-----
@order OracleEnvision (int player, unit aiUnit, unitgroup scanGroup) {
    order ord;
    int unitCount;
    unit unitToCheck;
    bool nearCloakedEnemy = false;
    region r;
    unitgroup allNearbyEnemies;

    if (AIIsCampaign(player)) {
        return null;
    }
```

- Wie C90 !!! Kommentare nur EINZEILIG !
- Ohne Pointer, keine switches
- Func Order betrachten -> Prototypes funktionieren !
- Var + Init am Anfang der Funktion
- Magic Numbers nicht erwünscht

```
if (UnitTestState(unitToCheck, c_unitStateCloaked)) {
    nearCloakedEnemy = true;
    break;
}
```

- Natives.galaxy

```
const int c_unitStateDetector          = 2; // Read-only|
```

# Requirements.galaxy

Keine MagicNumbers !



c = constant

AB = Ability // use Blink

BF = Buff // Stimpack

U = Unit // Overlord

B = Building // Hatchery

R = Research // Burrow

Z = Zerg

T = Terran

P = Protos

```
const string c_ZU_Overlord = "Overlord";
```

## Debug Ausgaben



DebugAI ( String input)

IntToString, FixedToString ...

```
DebugAI("Player: " + IntToString(player) +
        "WaveType: " + IntToString(player) +
        " Und noch nen sinnloser Fixed: " + FixedToString(3.14 , 2));
```

## Globale vs Player\_UserData



```
// THIS CODE WILL RUN ONCE PER NG AI
if (AI GetUserInt(player, c_NGUD_INIT) != 1337 && AIPlayerDifficulty(player) == c_skirVeryEasy)
{
    AIResetUserData(player);
    AISetUserInt(player, c_NGUD_INIT, 1337);

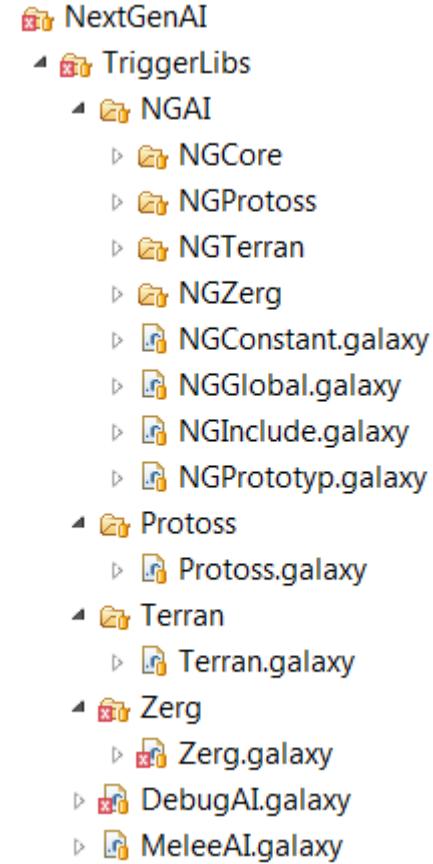
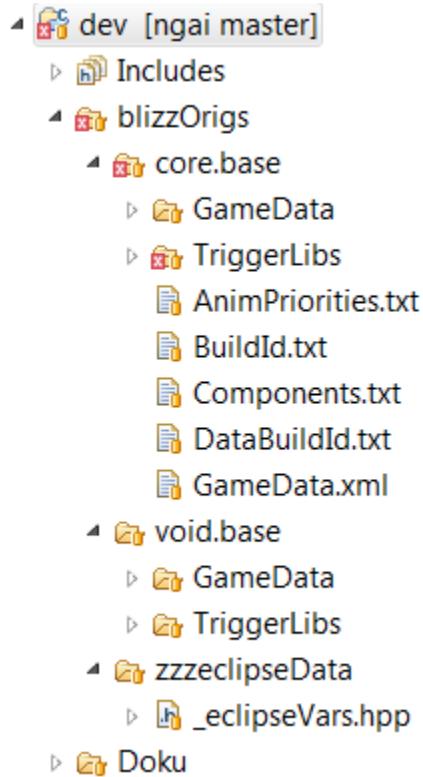
    DebugAI("This will run once for player: " + IntToString(player));
}
```

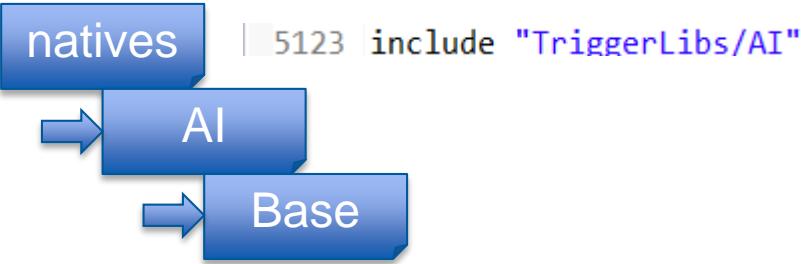
VS

```
// THIS CODE RUNS ONLY ONCE PER MAP NOT PER AI / PLAYER
if (g_NGTriggerRunOnce == false) {

    // iterate and search for humanplayer
    humanplayer = -1;
    while (humanplayer <= 15) {
    }
    cpuplayer = cpuplayer + 1;
}
```

# Aufbau der Files und Folder





- Computer.galaxy
  - DebugAI
  - RequirementsAI
  - BuildAI
  - SharedAI
  - Tactical/TacticalAI
    - include "TriggerLibs/Tactical/TactProtAI"
    - include "TriggerLibs/Tactical/TactTerrAI"
    - include "TriggerLibs/Tactical/TactZergAI"
    - include "TriggerLibs/Tactical/TactCampAI"
  - CampaignAI
  - MeleeAI
    - TriggerLibs/MeleeBuildAI
    - TriggerLibs/MeleeWaveAI
    - TriggerLibs/MeleeHighAI
      - "TriggerLibs/Protoss/ProtossHigh"
      - "TriggerLibs/Terran/TerranHigh"
      - "TriggerLibs/Zerg/ZergHigh"
    - TriggerLibs/MeleeLowAI
    - TriggerLibs/Terran/Terran
      - ...
    - TriggerLibs/Protoss/Protoss
      - ...
    - TriggerLibs/Zerg/Zerg
      - include "TriggerLibs/Zerg/ZergVyEy"
      - include "TriggerLibs/Zerg/ZergEasy"
      - include "TriggerLibs/Zerg/ZergMedi"
      - include "TriggerLibs/Zerg/ZergMdHd"



### Per Race Main

```
//-----  
//  AIMeleeZerg  
//-----  
void AIMeleeZerg (int player) {  
    int mainState = AIState(player, e_mainState);  
  
    DebugAI("--ZERG_MAIN called <" + "NEX WAS HERE" + ">");  
  
    if (AIPlayerDifficulty(player) >= c_skirChMoney) {  
        AIMeleeCheatResources(player);  
    }  
}
```

- Nur einmal pro Race
- Im Schnitt 3 mal pro Sec

# Aufbau einer Sc2 AI

Stichwort : Multitasking in AIs



## AI Think per Unit

```
//-----
void AIThinkVoidRay (int player, unit aiUnit, unitgroup scanGroup) {
    order ord;

    ord = VoidRayDamageBoost(player, aiUnit, scanGroup);
    if (ord != null) {
        AICast(aiUnit, ord, c_noMarker, c_castHold);
        return;
    }
}
```

- Pro Unit 1 mal
- Pro Sec circa 3 mal .. Kommt auf die Masse der Units an
- Garantierter Durchlauf
- Aber keine garantierte Zeit...



### Wave Main

```
//-----
//  AIWaveThinkZerg
//-----
void AIWaveThinkZerg (int player, wave w, int type) {
    AIWaveThinkDefault(player, w, type);
}
```

- Wave ist eine Unit-Gruppe .. Nicht zu verwechseln mit dem Datentyp Unitgroup
- 3 mal pro Sec pro Wave



### New Unit for Player

```
//-
//  AINewUnitZerg
//-
void AINewUnitZerg (int player, unit u) {
    wave w;
    unit drop;
    string type = UnitGetType(u);

    // ignored units
    //
    if (type == c_ZU_Larva ||
        type == c_ZU_Broodling ||
        type == c_ZU_Mantaling ||
        type == c_ZU_Locust) {
        return;
    }
}
```

- 1x pro neuer Unit die IRGENDWIE dem Player zugewiesen wird

## Aufbau unserer AI



```
//-----
//  AIMeleeTerr
//-----

void AIMeleeTerr (int player) {
    int mainState = AIState(player, e_mainState);

    if (AIPlayerDifficulty(player) >= c_skirChMoney) {
        AIMeleeCheatResources(player);
    }

    if (mainState == e_mainState_Init)          { TerranInit(player); }
    else if (mainState == e_mainState_Open)      { TerranOpen(player); }
    else if (mainState == e_mainState_Mid)       { TerranMid(player); }
    else if (mainState == e_mainState_Late)       { TerranLate(player); }
    else { ErrorMeleeScript(player, "Invalid mainState"); }
}

void TerranOpen (int player) {
    int diff = AIPlayerDifficulty(player);

    if (diff == c_skirVeryEasy)      { NGTerranMain(player); }
    else if (diff == c_skirEasy)     { TerranOpenEasy(player); }
    else if (diff == c_skirMedium)   { TerranOpenMedi(player); }
    else if (diff == c_skirMed_Hard) { TerranOpenMdHd(player); }
    else if (diff == c_skirHard)    { TerranOpenHard(player); }
    else if (diff == c_skirHard_VH) { TerranOpenHdVH(player); }
    else if (diff == c_skirVeryHard){ TerranOpenVyHd(player); }
    else if (diff == c_skirChVision) { TerranOpenChVi(player); }
    else if (diff == c_skirChMoney)  { TerranOpenChRe(player); }
    else                           { TerranOpenChIn(player); }
}
```



- Jeder:
  - Lieblingseinheit seiner zugewiesenen Race
  - Eigenes „Test“-Command erstellen für seine eigene Sandbox func
  - Include nicht vergessen
    - // include TriggerLibs/NGAI/NGSandbox/name,,
  - Sc2 Recherche seiner Race, Units -> Doku !
- Team
  - Aufbau der KI – Core – Race ?
  - Commander, Persönlichkeiten, Strategien



FIN

Hier könnte ihre Werbung stehen.