



DIGEO: UMA ONTOLOGIA DE JOGOS DIGITAIS



Filipe Ghesla Silvestrim
Orientador: Prof. Msc. João Ricardo Bittencourt

SUMÁRIO

- Desenvolvimento de Jogos
- Ontologia
- Trabalhos Relacionados
- Classificação
- DIGEO
- MuGE

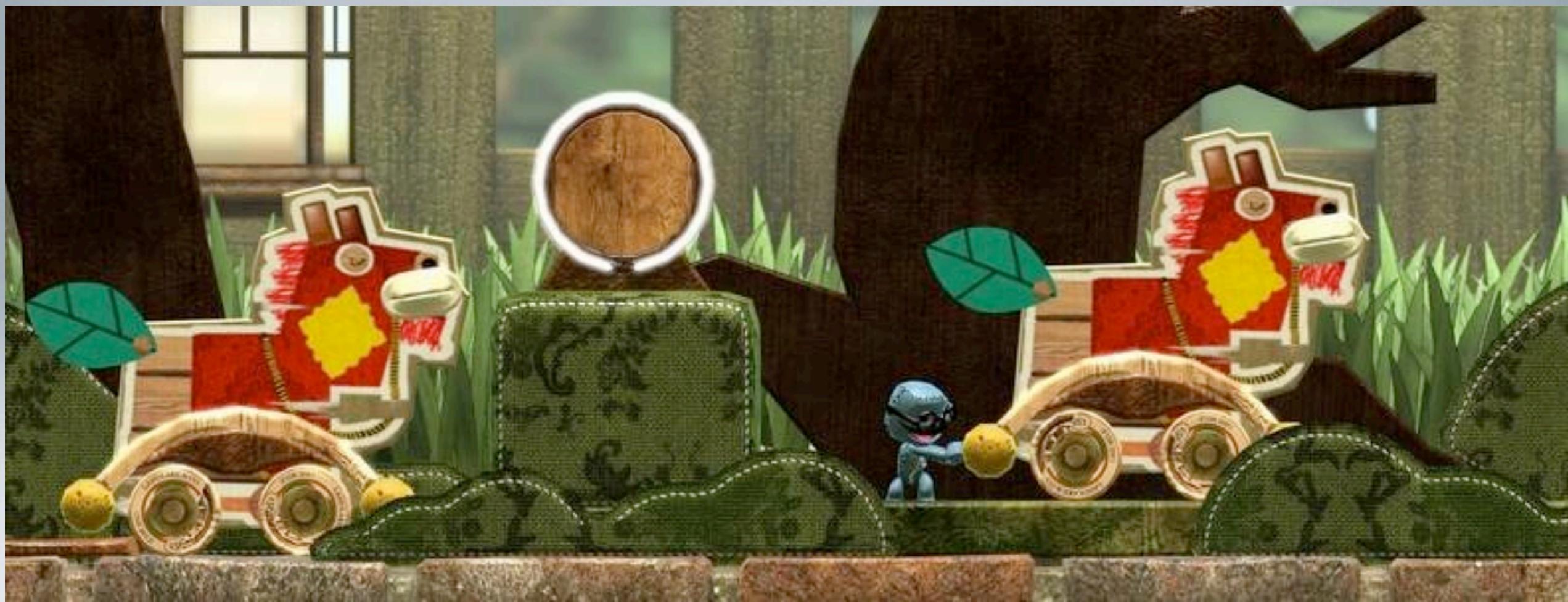
DIGEO

DIGITAL GAME ELEMENTS ONTOLOGY

INTRODUÇÃO



INTRODUÇÃO



INTRODUÇÃO



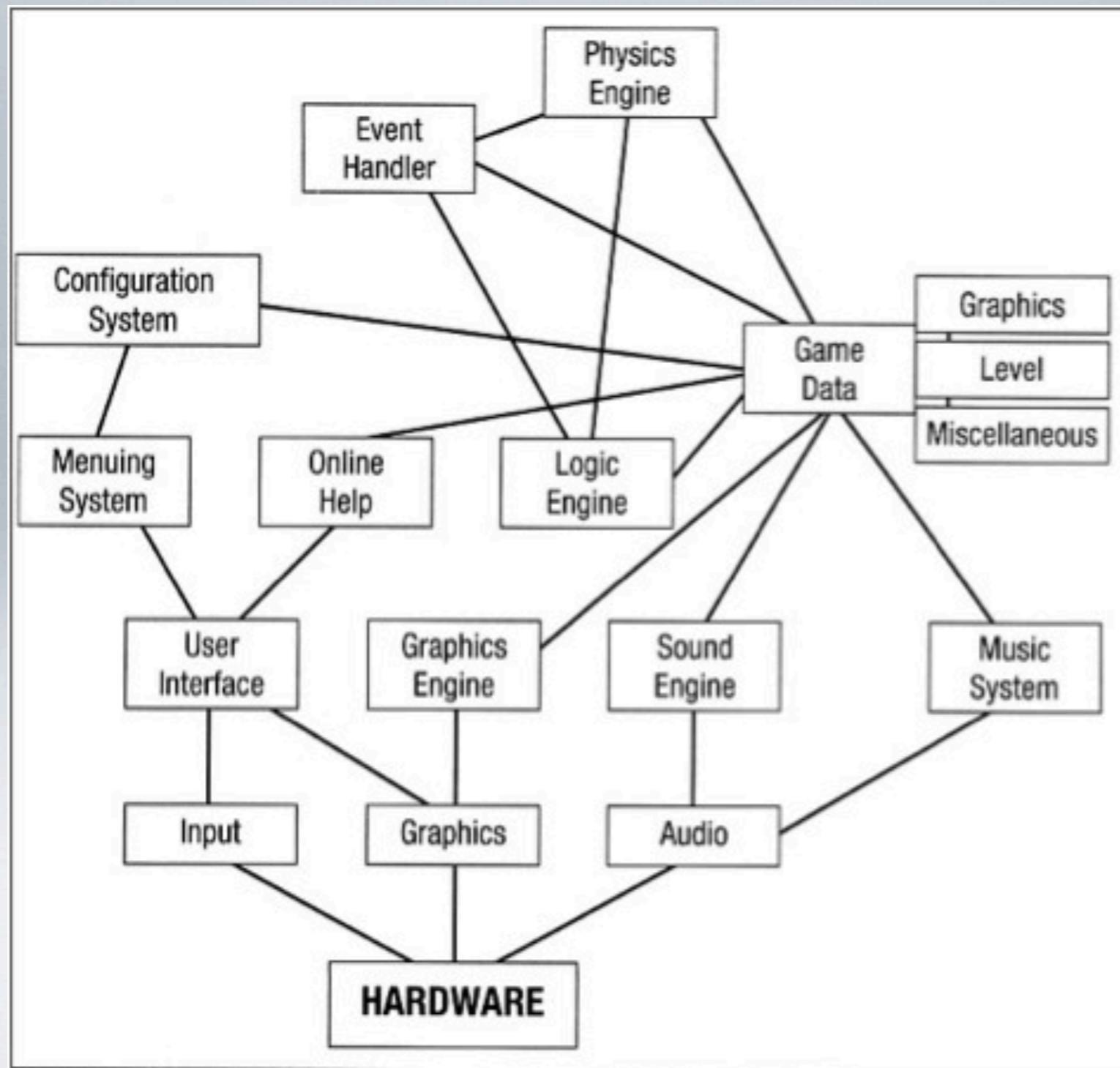
INTRODUÇÃO



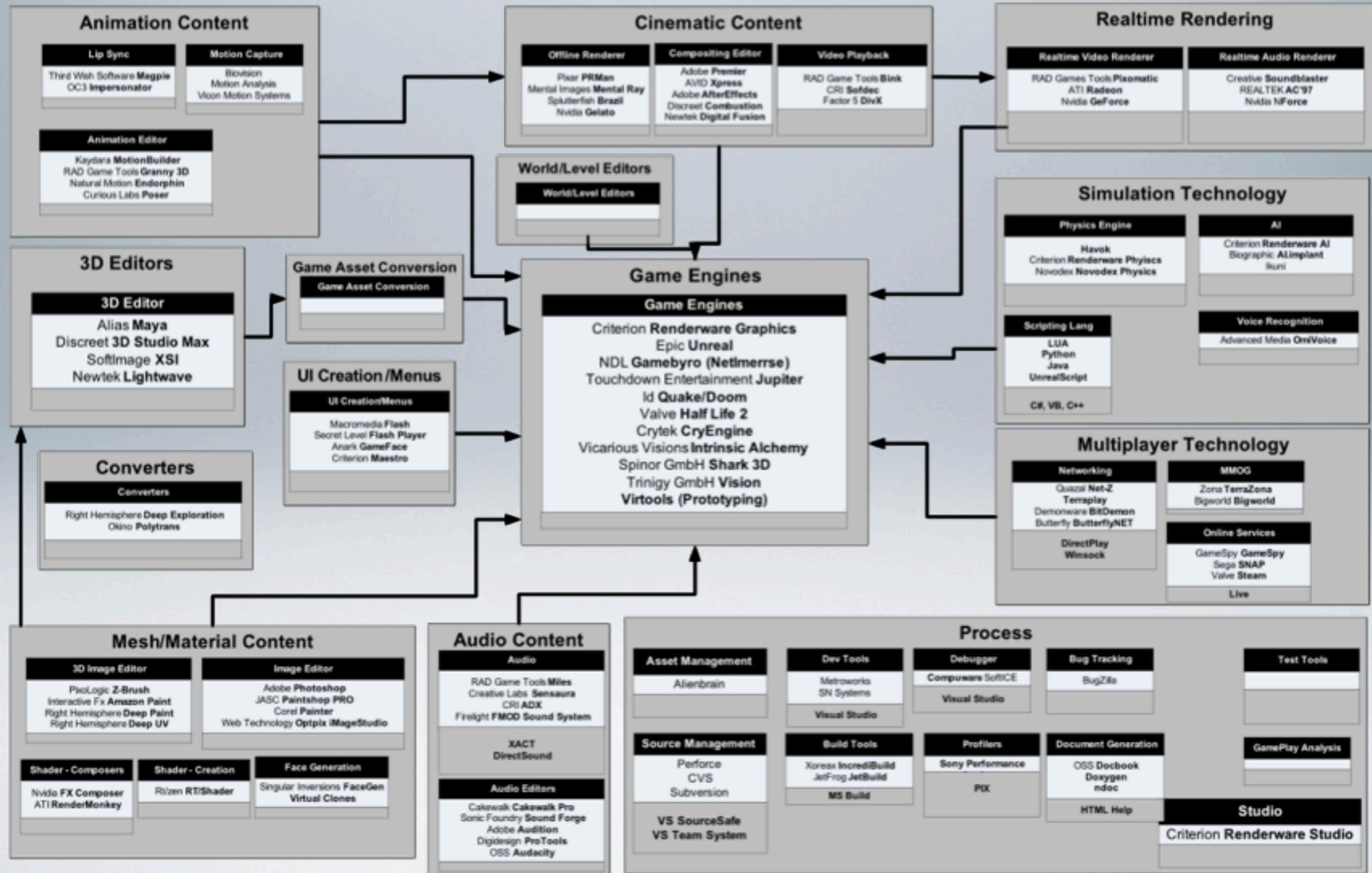
INTRODUÇÃO



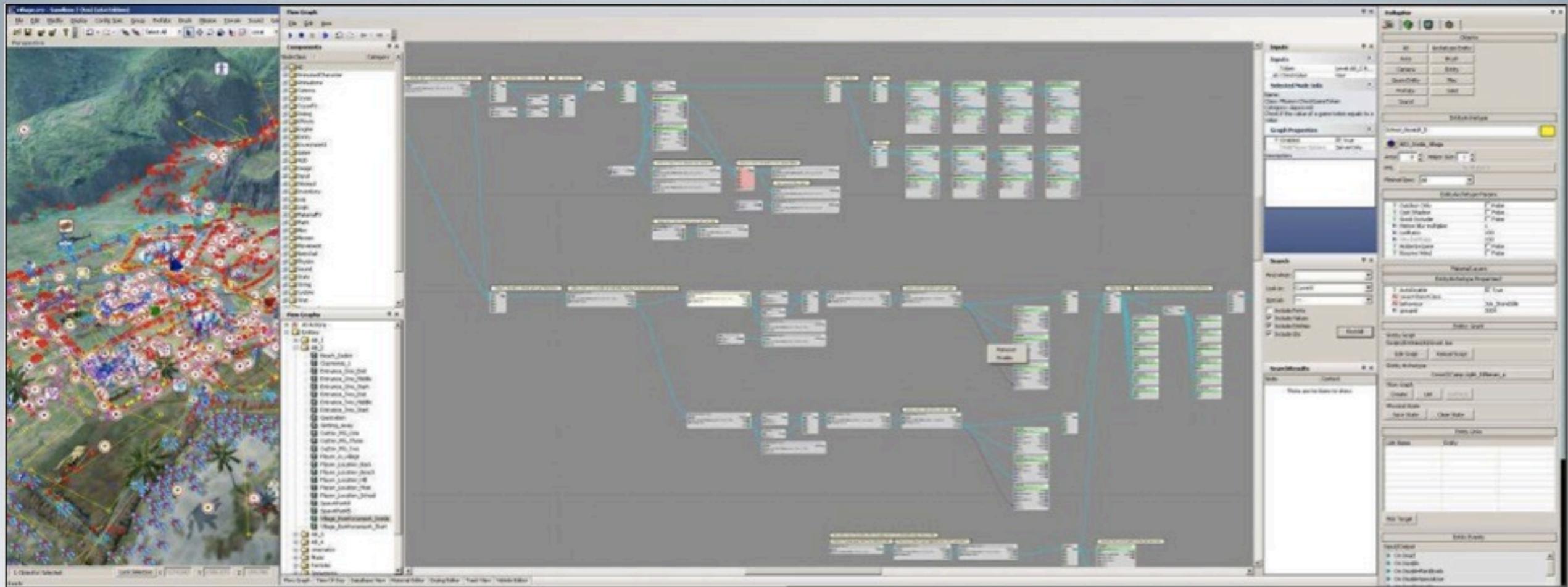
INTRODUÇÃO



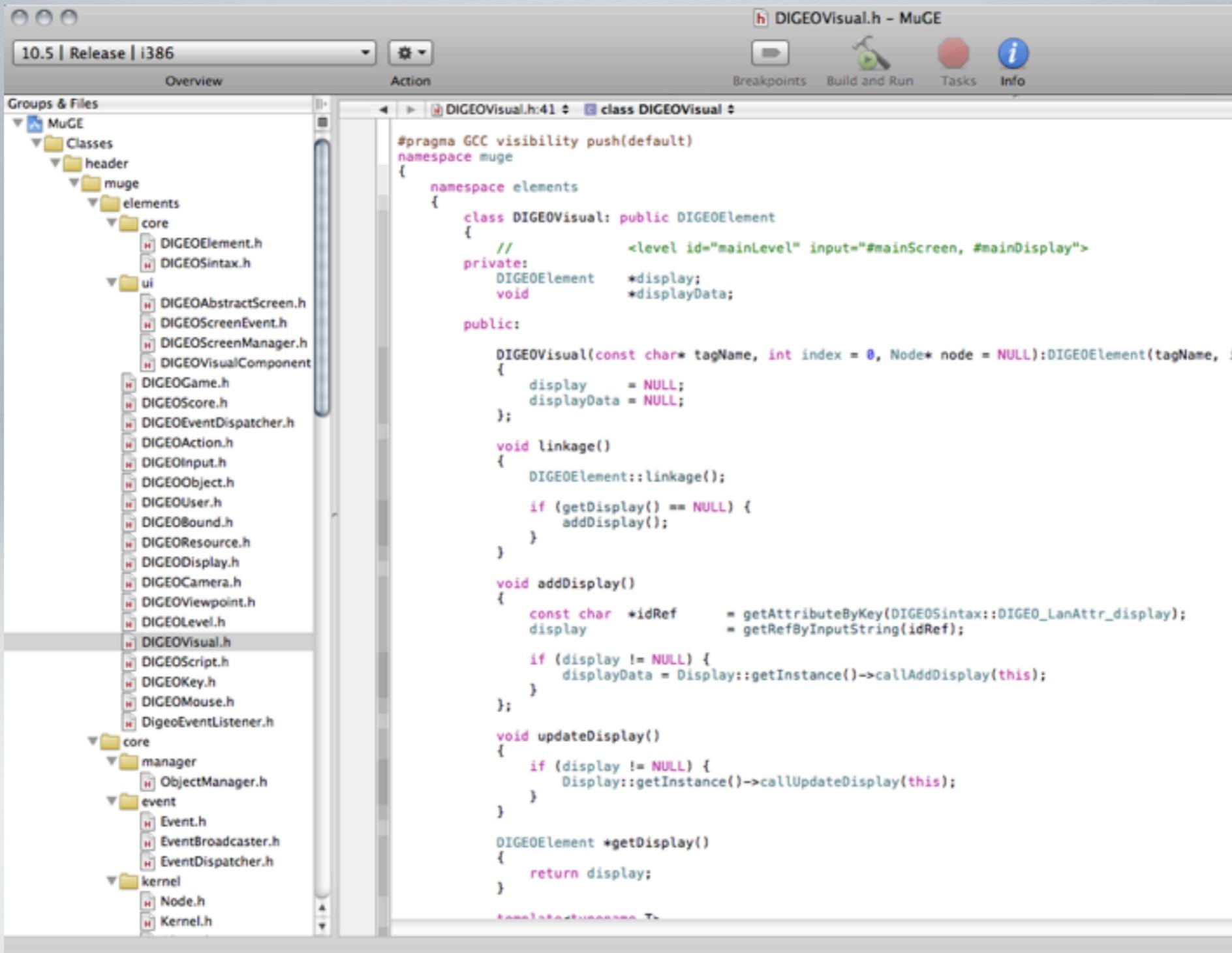
INTRODUÇÃO



INTRODUÇÃO



INTRODUÇÃO



The screenshot shows the Xcode IDE interface. The top menu bar displays "10.5 | Release | i386". The main window has tabs for "Overview", "Action", "Breakpoints", "Build and Run", "Tasks", and "Info". The "Action" tab is selected, showing the file "DIGEOVisual.h" at line 41, which defines the class "DIGEOVisual". The code is as follows:

```
#pragma GCC visibility push(default)
namespace muge
{
    namespace elements
    {
        class DIGEOVisual: public DIGEOElement
        {
            // <level id="mainLevel" input="#mainScreen, #mainDisplay">
            private:
                DIGEOElement *display;
                void *displayData;

            public:

                DIGEOVisual(const char* tagName, int index = 0, Node* node = NULL):DIGEOElement(tagName, i
                {
                    display = NULL;
                    displayData = NULL;
                };

                void linkage()
                {
                    DIGEOElement::linkage();

                    if (getDisplay() == NULL) {
                        addDisplay();
                    }
                }

                void addDisplay()
                {
                    const char *idRef =getAttributeByKey(DIGEOSyntax::DIGO_LanAttr_display);
                    display = getRefByInputString(idRef);

                    if (display != NULL) {
                        displayData = Display::getInstance()->callAddDisplay(this);
                    }
                }

                void updateDisplay()
                {
                    if (display != NULL) {
                        Display::getInstance()->callUpdateDisplay(this);
                    }
                }

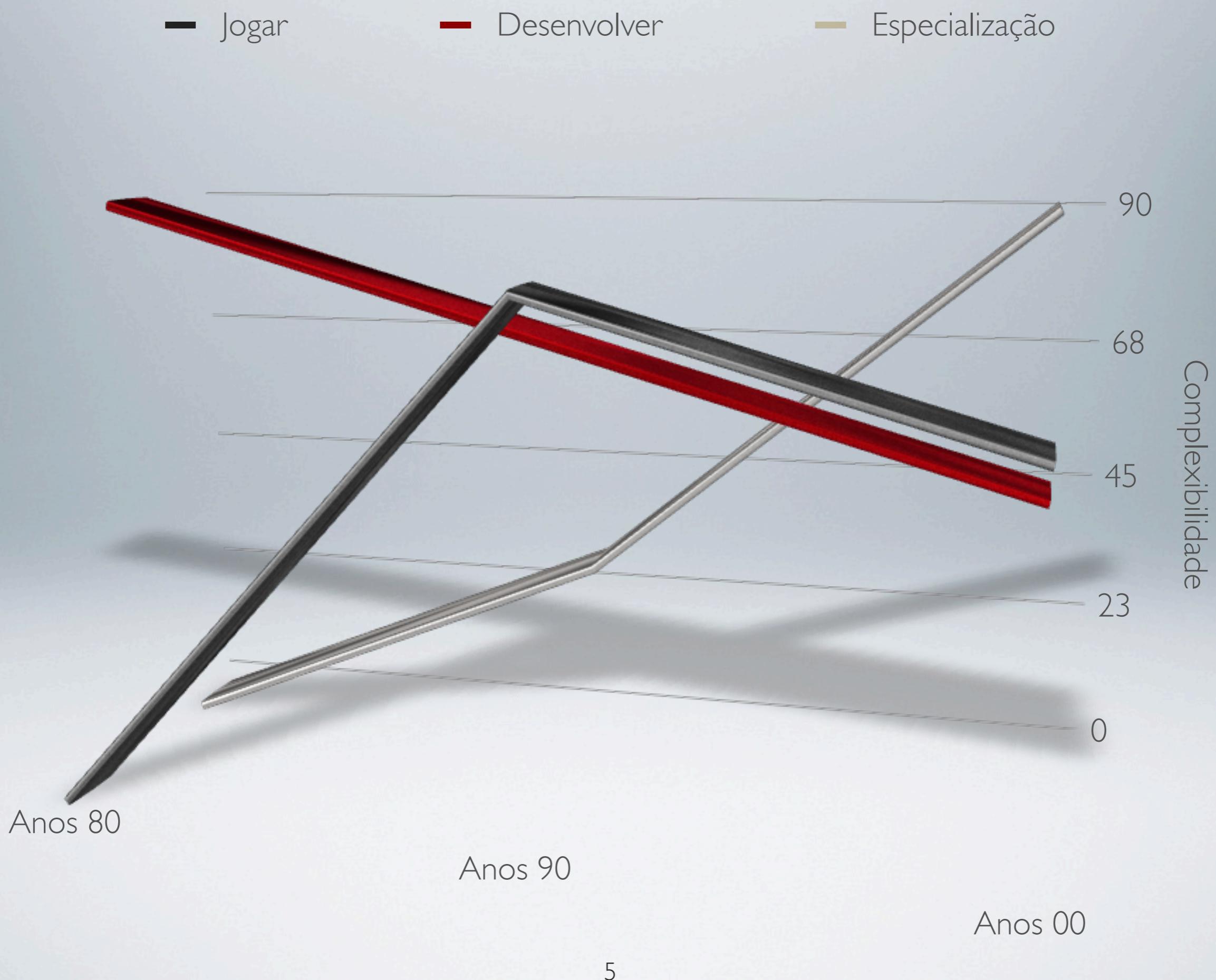
                DIGOEElement *getDisplay()
                {
                    return display;
                }

                ~DIGEOVisual()
                {
                    if (display != NULL) {
                        Display::getInstance()->callDeleteDisplay(display);
                    }
                }
        };
    }
}
```

— Jogar

— Desenvolver

— Especialização



DESEJOS

Se houver uma evolução de paradigmas no **game design**, nós diríamos que isso será um passo a frente da criação de um design genérico... Tal design permitirá que designers começem a argumentar do geral para o específico, isso fará com que os conceitos de design fiquem mais portáveis e robustos.

— Andrew Rollings & Dave Morris

ONTOLOGIA

“Uma especificação explícita e formal de uma conceitualização compartilhada” [BORST,W.N]

- Arquétipos: modelo que simboliza um protótipo ideal a algo.
- Padronização.
- Classificação (Categorias).
- Nomenclatura (Dicionário).

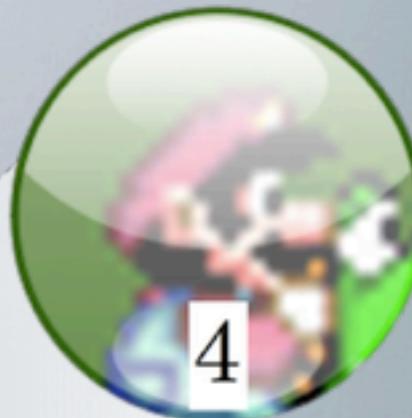
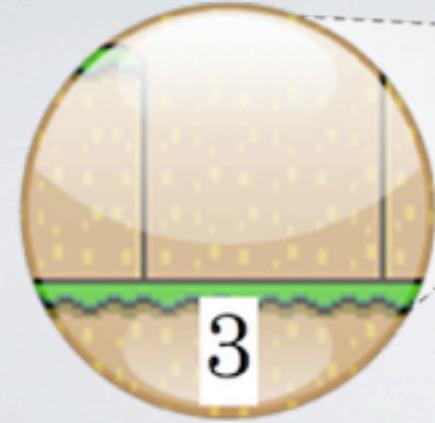
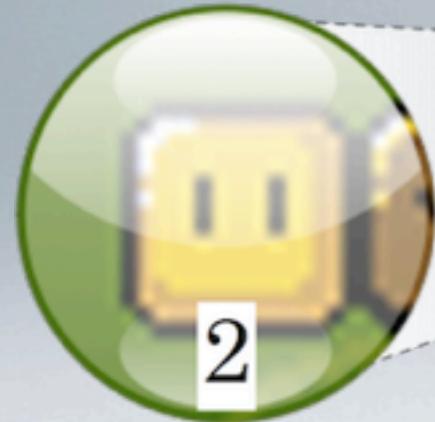
ONTOLOGIAS DE JOGOS

- **Game Ontology Project (GOP):**
Perspectiva de jogador.
- **Digital Game Ontology:**
Banco de dados.
- **Ontología de Videojuegos:**
Interação jogador Vs. jogo.
- **DIGEO:**
Perspectiva de um desenvolvedor + teoria de game design

DIAGRAMA METODOLÓGICO



ELEMENTOS



JOGOS ANALISADOS

JOGOS ANALISADOS

Pong



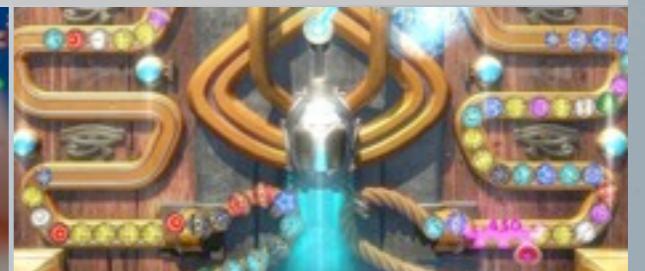
Halo : Combat Evolved



Spore



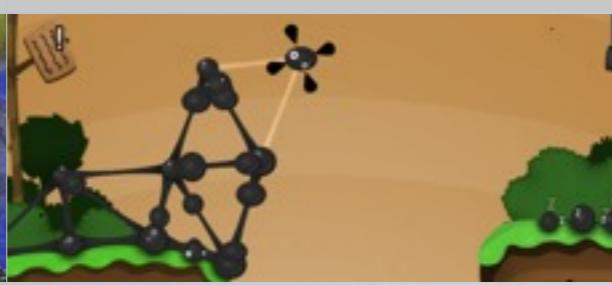
Luxor 3



WarCraft III - Reign of chaos



World of Goo



Tetris Zone



Full Throttle



Zelda : A link to the past



Super Mario World



Rock N Roll Racing



Pokemon Red



Guitar Hero : Legends of Rock



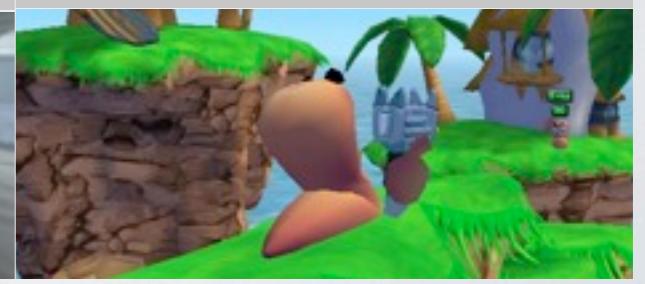
GTA II



Need For Speed Pro Street



Worms 3D



CATEGORIAS

Primárias

São aspectos que precedem os estereótipos de um jogo — são as definições que fazem do título um local habitável.

Secundárias

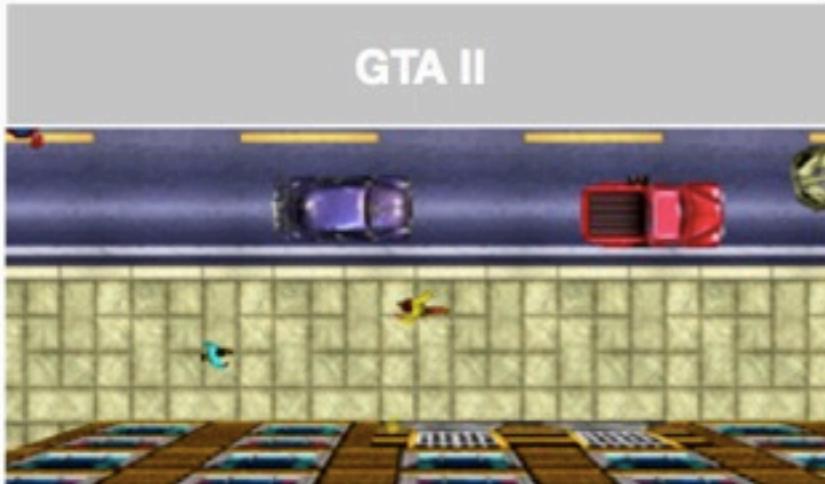
Descreve um título como um composto de objetos de gameplay.

CATEGORIA SECUNDÁRIA

	Secundary Category							
	Default	Platform	FPS	RTS	Action-Adventure	Racing	RPG	Artillery
Elements								
Camera	Fixed	Tracked	Tracked	Movable	Tracked	Tracked	Tracked	Tracked
Viewpoint	Top	Side	First Person	Isometric	Top	Top	Third Person	Third Person
Time	No	Free	No	Free	No	Limited	Turn	Turn
Map	Visible	Visible	Visible	Fog of War	Visible	Visible	Visible	Visible
Main Entity	Character	Character	Character	Character	Character	Car	Character	Character
Action Elements								
Collect	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE
Change Value (Combat)	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE
Show Message	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
Use (ie.: weapon)	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE
Release	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE

Descreve um título como um composto de objetos de gameplay

EXEMPLOS

GTA II	Zelda : A link to the past	Pokemon Red																																														
 <p>▼ Characteristics</p> <table> <tr> <td>Main Category</td><td>Action</td></tr> <tr> <td>Secondary Category</td><td>Action-Adventure</td></tr> </table> <p>▼ Main Categ. Customizations</p> <table> <tr> <td>Physics</td><td><input type="checkbox"/></td></tr> <tr> <td>Dialog</td><td><input checked="" type="checkbox"/></td></tr> </table> <p>▼ Sec. Categ. Customizations</p> <table> <tr> <td>Show Message</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Use (ie.: weapon)</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Release</td><td><input checked="" type="checkbox"/></td></tr> </table>	Main Category	Action	Secondary Category	Action-Adventure	Physics	<input type="checkbox"/>	Dialog	<input checked="" type="checkbox"/>	Show Message	<input checked="" type="checkbox"/>	Use (ie.: weapon)	<input checked="" type="checkbox"/>	Release	<input checked="" type="checkbox"/>	 <p>▼ Characteristics</p> <table> <tr> <td>Main Category</td><td>Action</td></tr> <tr> <td>Secondary Category</td><td>Action-Adventure</td></tr> </table> <p>▼ Main Categ. Customizations</p> <table> <tr> <td>Physics</td><td><input type="checkbox"/></td></tr> <tr> <td>Level</td><td><input type="checkbox"/></td></tr> <tr> <td>Dialog</td><td><input checked="" type="checkbox"/></td></tr> </table> <p>▼ Sec. Categ. Customizations</p> <table> <tr> <td>Camera</td><td>Movable</td></tr> <tr> <td>Show Message</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Use (ie.: weapon)</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Release</td><td><input checked="" type="checkbox"/></td></tr> </table>	Main Category	Action	Secondary Category	Action-Adventure	Physics	<input type="checkbox"/>	Level	<input type="checkbox"/>	Dialog	<input checked="" type="checkbox"/>	Camera	Movable	Show Message	<input checked="" type="checkbox"/>	Use (ie.: weapon)	<input checked="" type="checkbox"/>	Release	<input checked="" type="checkbox"/>	 <p>▼ Characteristics</p> <table> <tr> <td>Main Category</td><td>RPG</td></tr> <tr> <td>Secondary Category</td><td>RPG</td></tr> </table> <p>▼ Main Categ. Customizations</p> <table> <tr> <td>Score</td><td>Challenge</td></tr> </table> <p>▼ Sec. Categ. Customizations</p> <table> <tr> <td>Camera</td><td>Movable</td></tr> <tr> <td>Viewpoint</td><td>Top</td></tr> <tr> <td>Time</td><td>No</td></tr> <tr> <td>Use (ie.: weapon)</td><td><input checked="" type="checkbox"/></td></tr> </table>	Main Category	RPG	Secondary Category	RPG	Score	Challenge	Camera	Movable	Viewpoint	Top	Time	No	Use (ie.: weapon)	<input checked="" type="checkbox"/>
Main Category	Action																																															
Secondary Category	Action-Adventure																																															
Physics	<input type="checkbox"/>																																															
Dialog	<input checked="" type="checkbox"/>																																															
Show Message	<input checked="" type="checkbox"/>																																															
Use (ie.: weapon)	<input checked="" type="checkbox"/>																																															
Release	<input checked="" type="checkbox"/>																																															
Main Category	Action																																															
Secondary Category	Action-Adventure																																															
Physics	<input type="checkbox"/>																																															
Level	<input type="checkbox"/>																																															
Dialog	<input checked="" type="checkbox"/>																																															
Camera	Movable																																															
Show Message	<input checked="" type="checkbox"/>																																															
Use (ie.: weapon)	<input checked="" type="checkbox"/>																																															
Release	<input checked="" type="checkbox"/>																																															
Main Category	RPG																																															
Secondary Category	RPG																																															
Score	Challenge																																															
Camera	Movable																																															
Viewpoint	Top																																															
Time	No																																															
Use (ie.: weapon)	<input checked="" type="checkbox"/>																																															

DIGEO: DIGITAL GAME ELEMENTS ONTOLOGY

Objetivo:

Tornar inteligível e padronizado o conhecimento ligado a área do desenvolvimento de jogos digitais.

Função:

Evidenciar a existência dos elementos que caracterizam e categorizam um jogo digital.

Consequência:

Taxonomia padronizada e teoria aplicável.

Elementos:

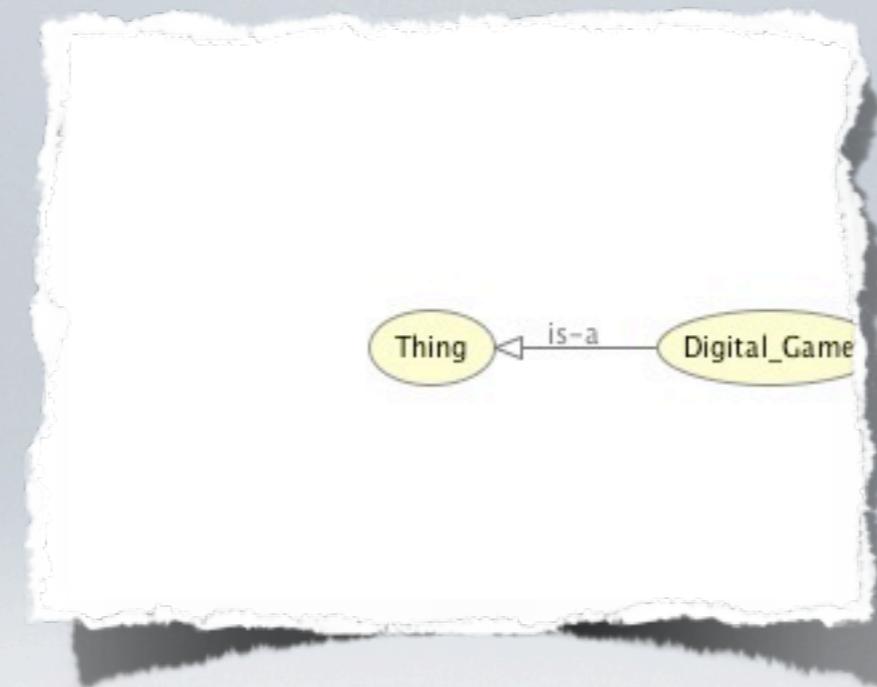
Descritivos, Interativos, Comportamentais.

PAPÉIS PRINCIPAIS DA ONTOLOGIA

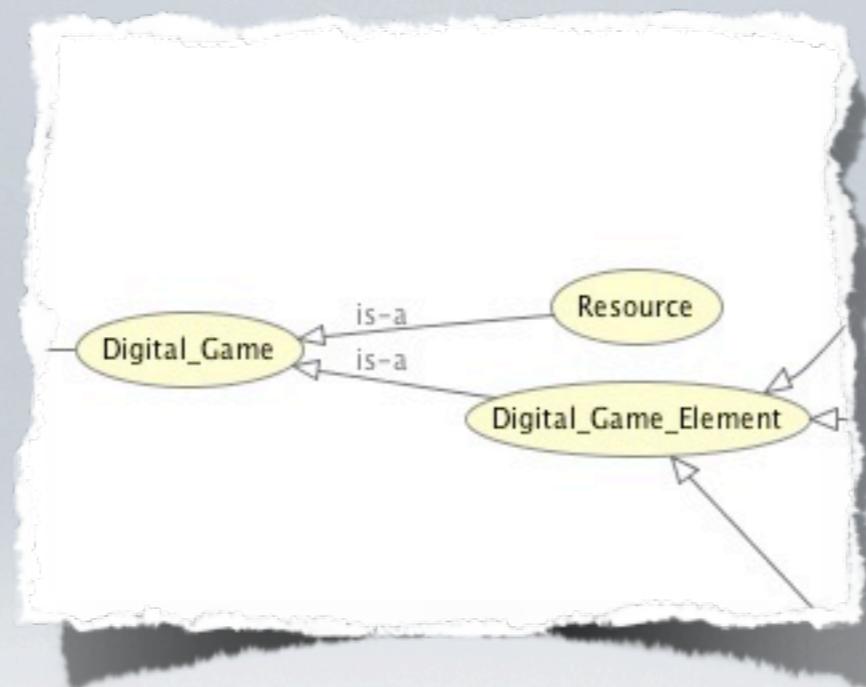
- I) Ter uma definição que precede, no escopo de desenvolvimento, definições de ontologias de jogos já existentes;**
- 2) Vocabulário comum e padrão**
- 3) Adicionar o entendimento do que são elementos;**
- 4) Possibilitar a descrição da composição de jogos digitais sob um olhar de game design por meio do uso de elementos;**
- 5) Tornar explícito e acessível as definições a respeito dos elementos necessários para o desenvolvimento de jogos digitais;**
- 6) Formalizar a composição de um jogo digital;**
- 7) Explicar o funcionamento semântico e as relações das classes da ontologia;**
- 8) Auxiliar o desenvolvimento de jogos para ser menos burocrático, diminuindo o tempo e tornando mais compreensível;**

Para leigos: “glossário de desenvolvimento de jogos”

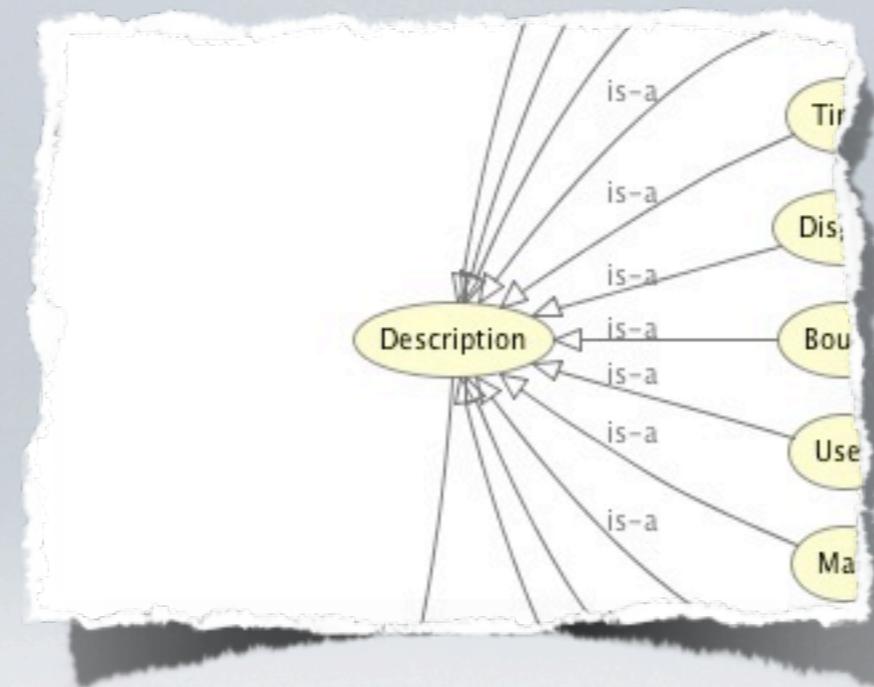
HIERARQUIA ONTOLÓGICA



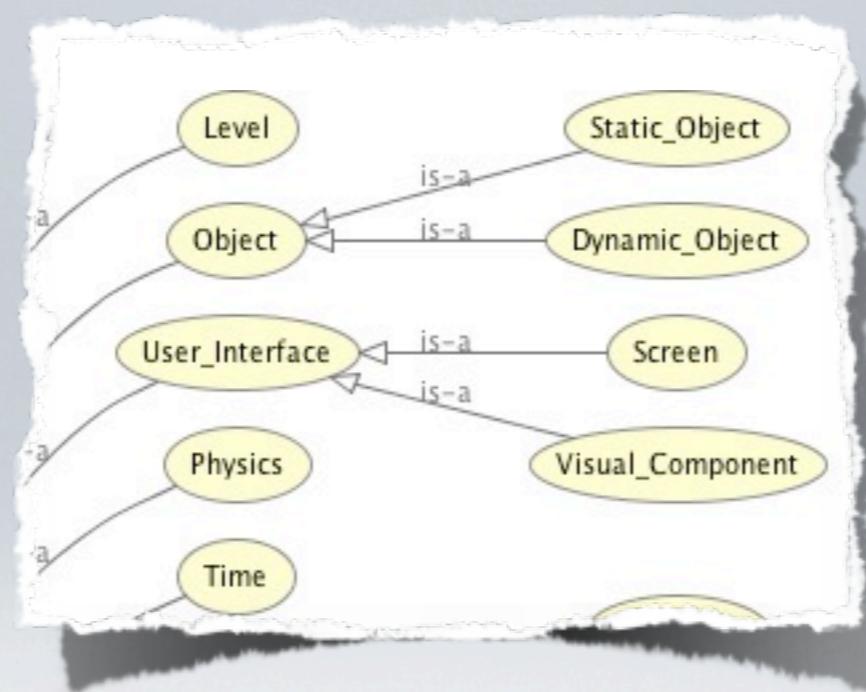
HIERARQUIA ONTOLÓGICA



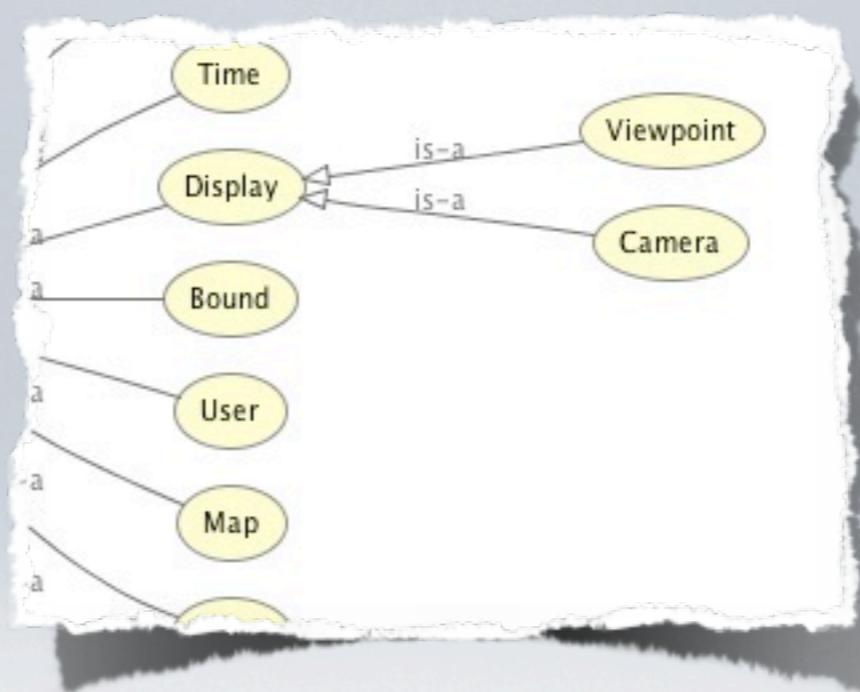
HIERARQUIA ONTOLÓGICA



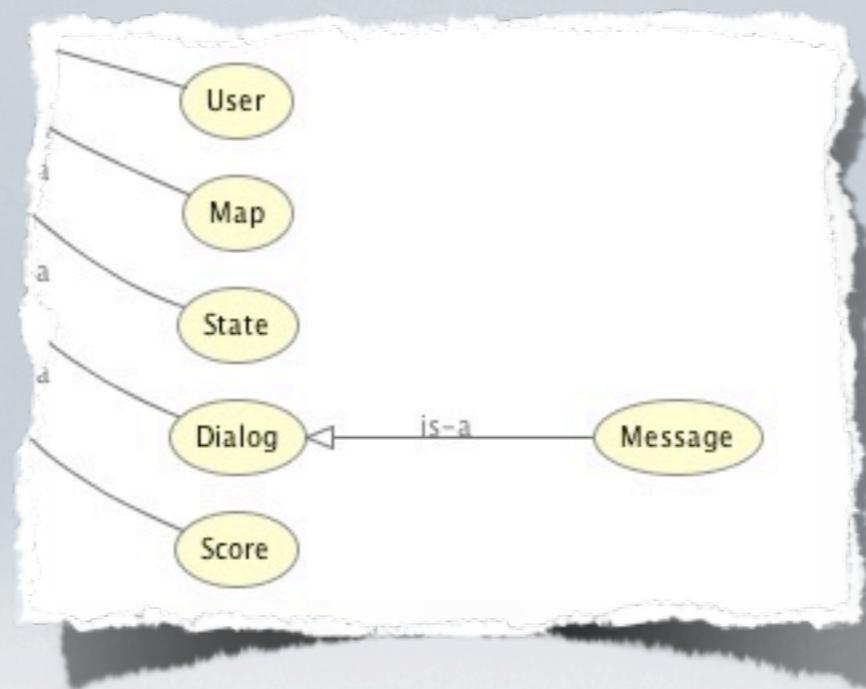
HIERARQUIA ONTOLÓGICA



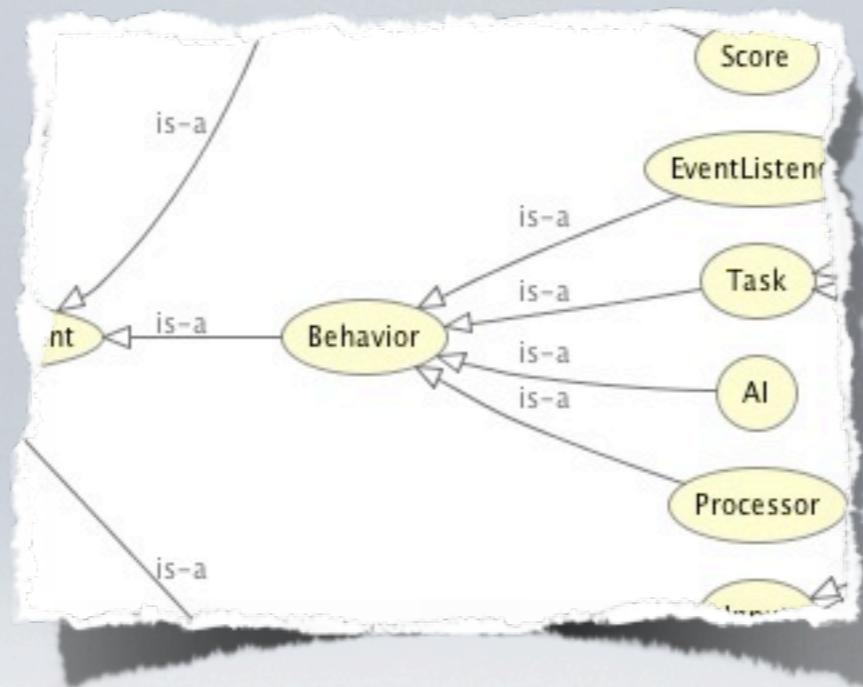
HIERARQUIA ONTOLÓGICA



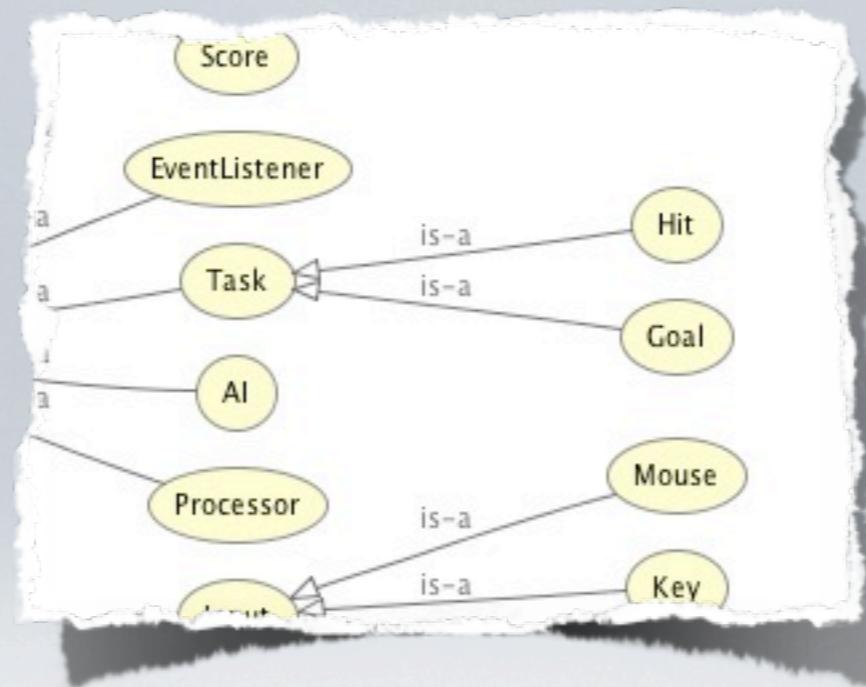
HIERARQUIA ONTOLÓGICA



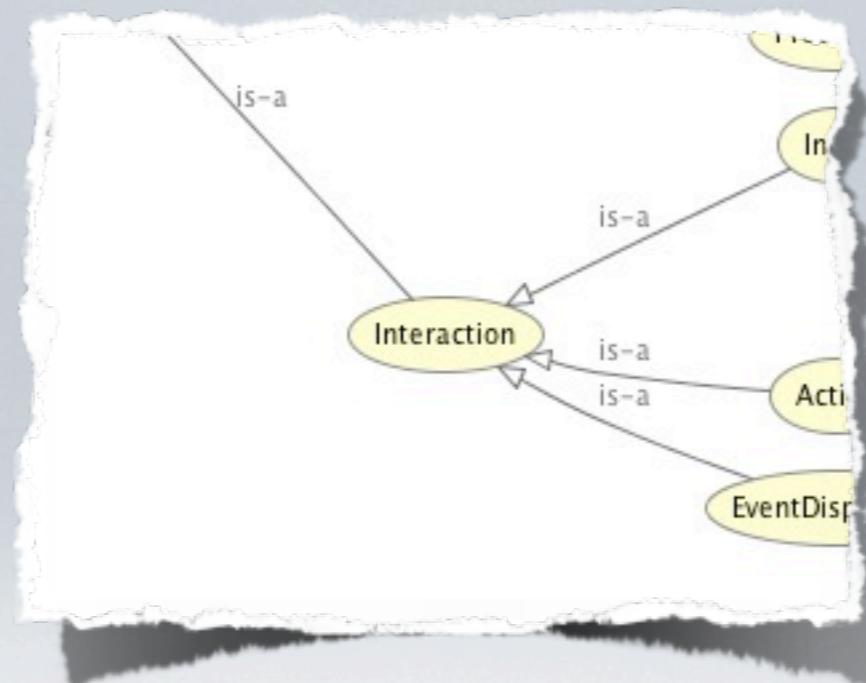
HIERARQUIA ONTOLÓGICA



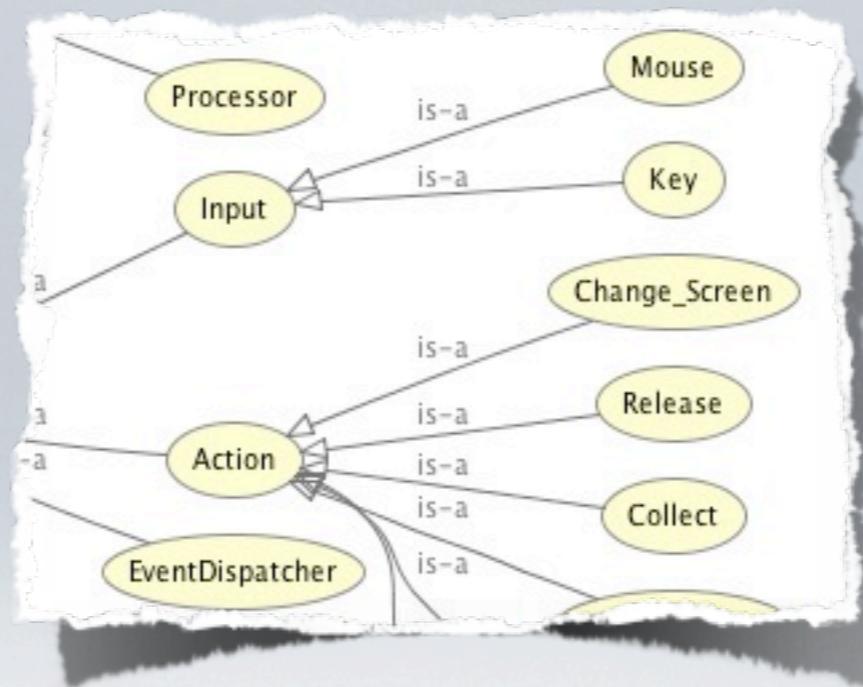
HIERARQUIA ONTOLÓGICA



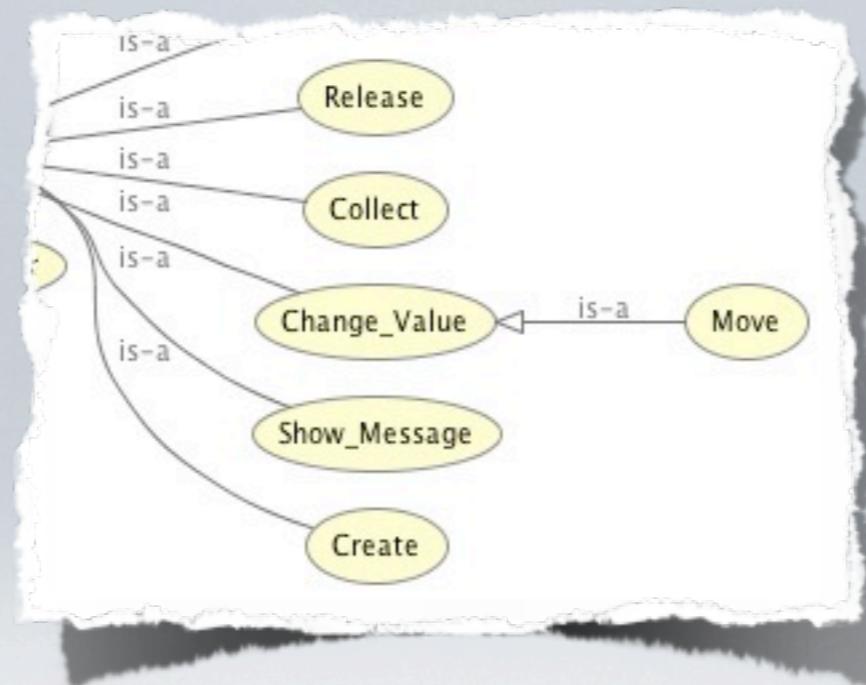
HIERARQUIA ONTOLÓGICA



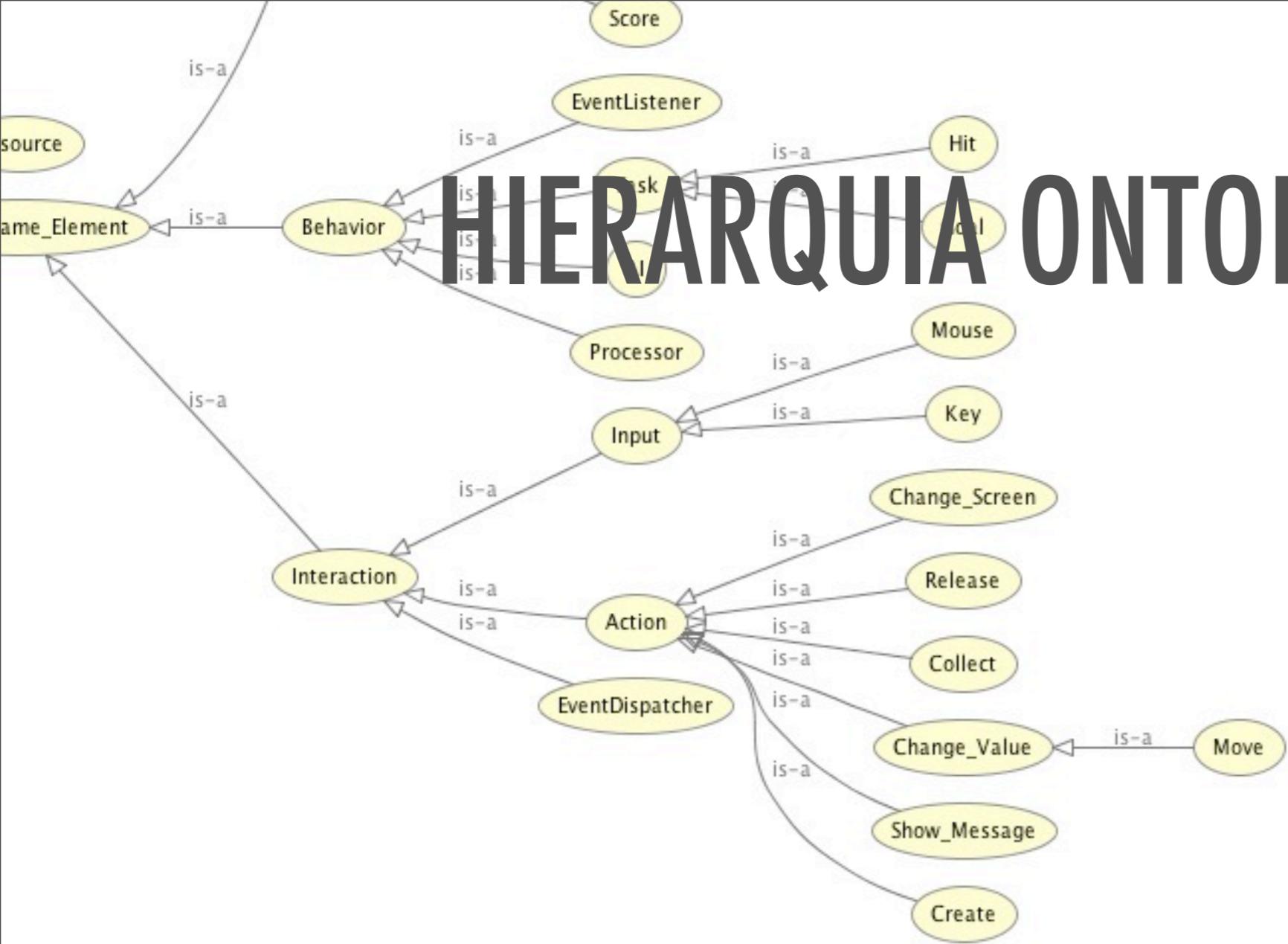
HIERARQUIA ONTOLÓGICA



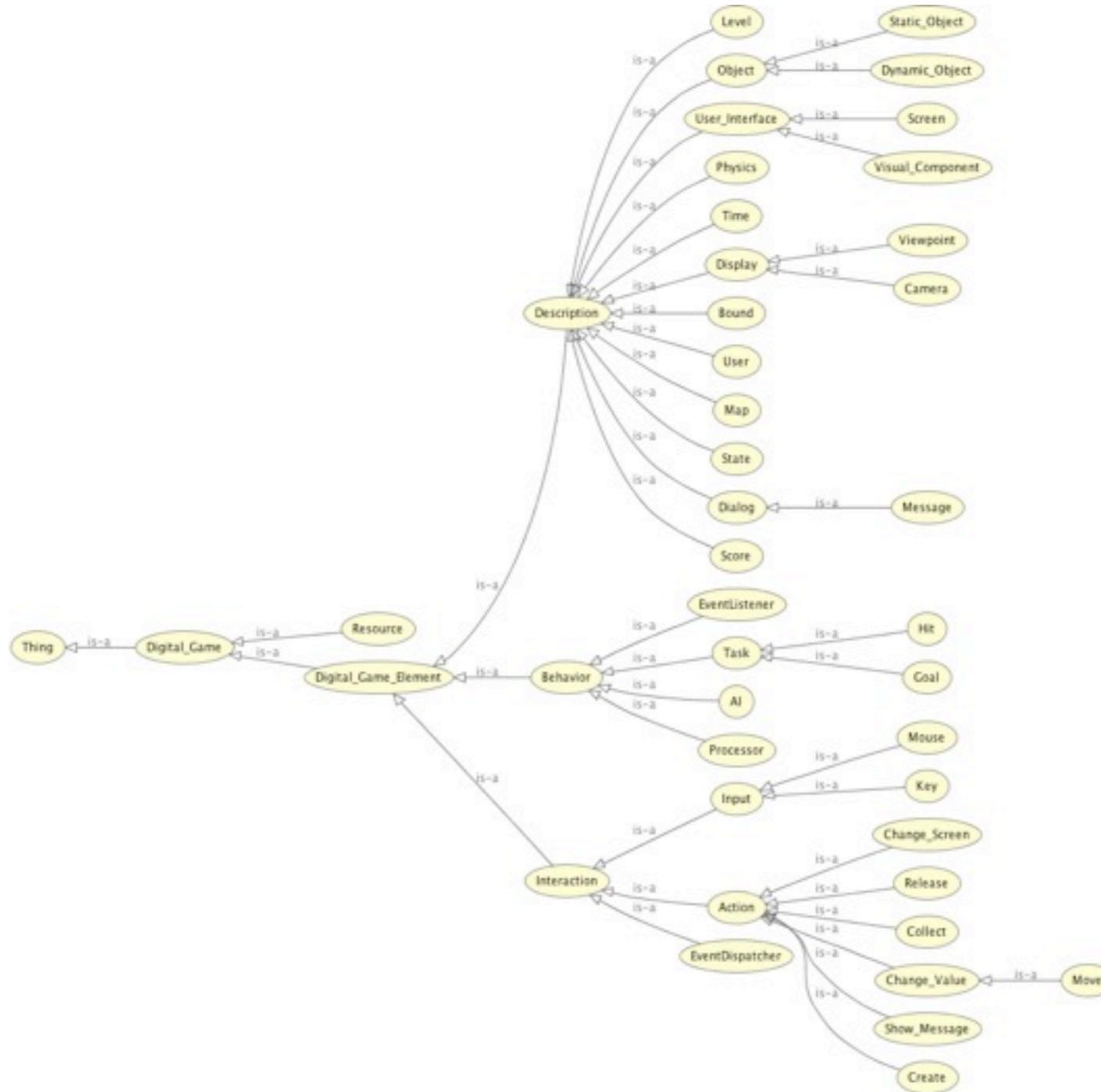
HIERARQUIA ONTOLÓGICA



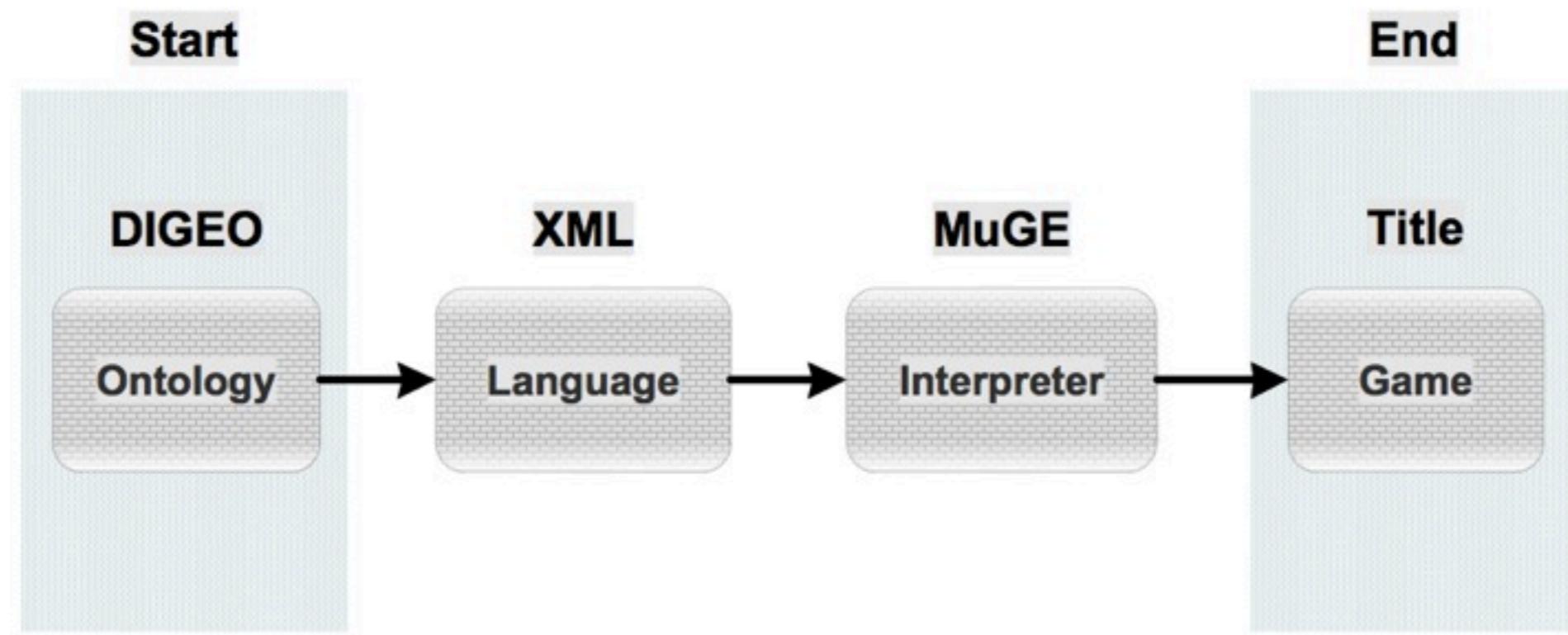
HIERARQUIA ONTOLÓGICA



HIERARQUIA ONTOLÓGICA



MUGE: MULTI-GENRE GAME ENGINE



MUGE: MULTI-GENRE GAME ENGINE

MuGE.framework



```
ace plugins  
  
mespace display  
  
    class Display: public Singleton  
    {  
        public:  
            void *ref;  
  
            const char *LIBRARY;  
            void init();  
  
            void setup(int screenW, in  
            void config();  
            void render();  
            void shutDown();
```

GamePrototype.app



MuGE.framework
IrrlichtPlugin.framework

```
#include <MuGE/Display.h>  
using namespace muge::plugins::  
  
#include <MuGE/Kernel.h>  
using namespace muge::core::ker  
  
#include <MuGE/GlobalEventRecei  
using namespace muge::plugins::  
  
Display *disp    = Display::getI  
Kernel *ker     = Kernel::getIn  
  
void pluginDisplay ()  
{  
    disp->init();  
    disp->setup(800, 600, (char  
    disp->config());
```

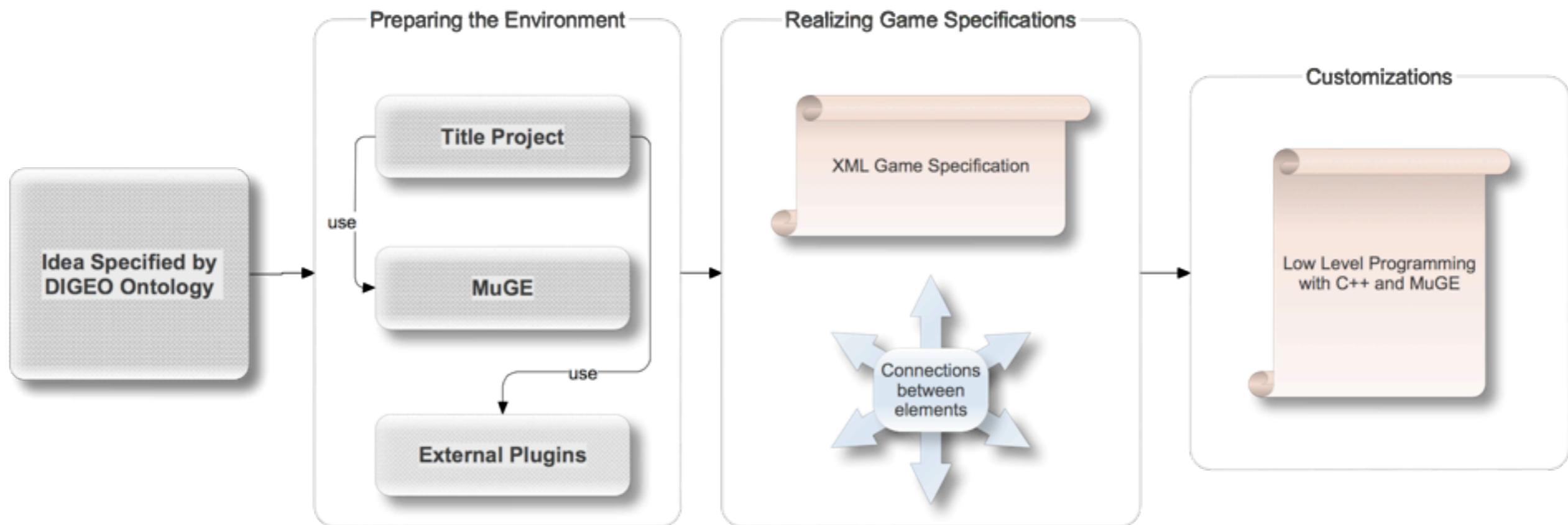
IrrlichtPlugin.framework



MuGE.framework

```
ITexture          *rt = 0;  
ISceneNode       *test;  
ICameraSceneNode *fpsCamera;  
  
IrrGlobalEventReceiver receiver;  
  
void Display::setup(int screenW,  
{  
    device = createDevice( video  
    device->setWindowCaption(Str  
  
    receiver.init();  
    Kernel::getInstance()->recei  
}  
  
void Display::config()  
{...}
```

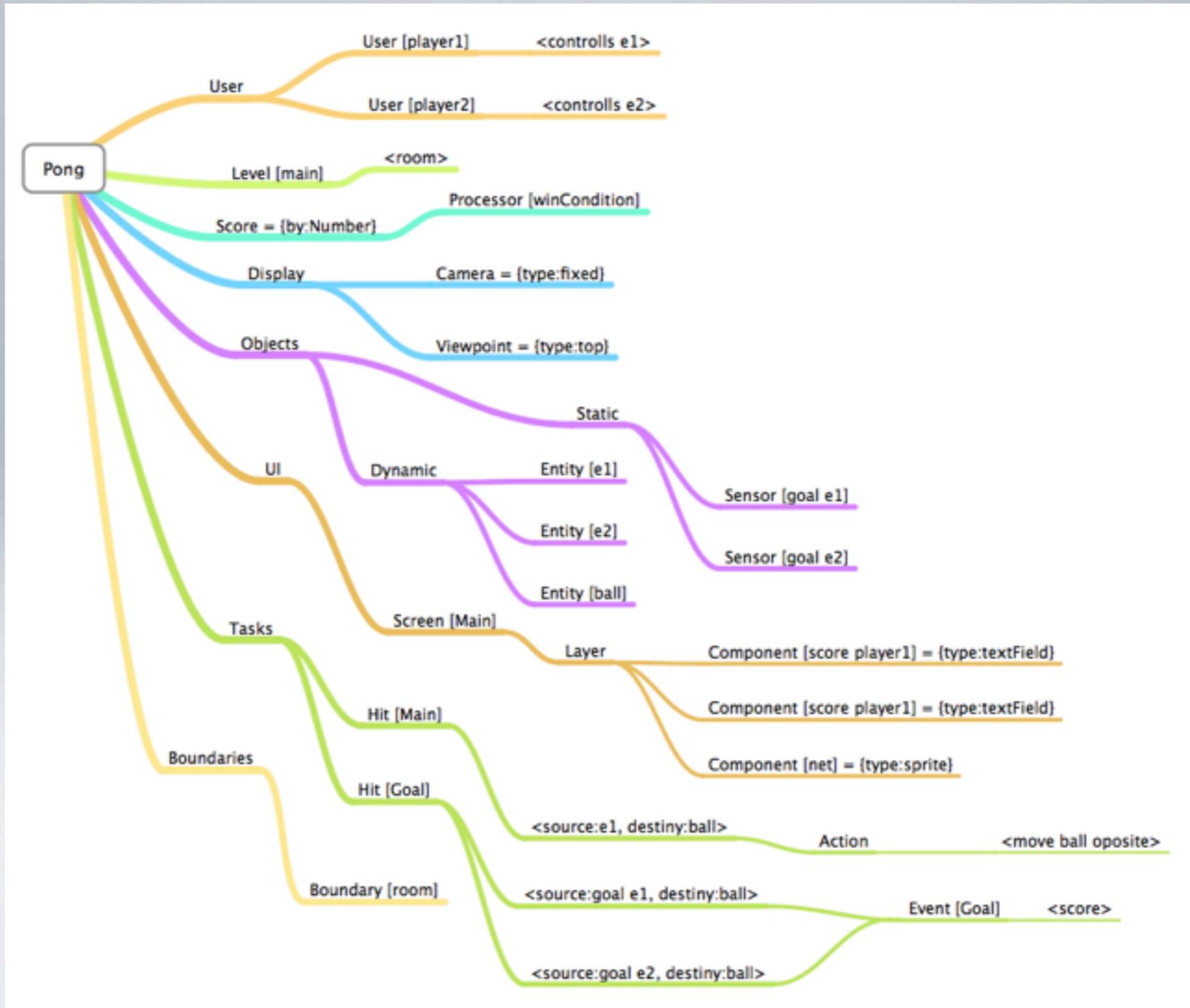
MUGE: MULTI-GENRE GAME ENGINE



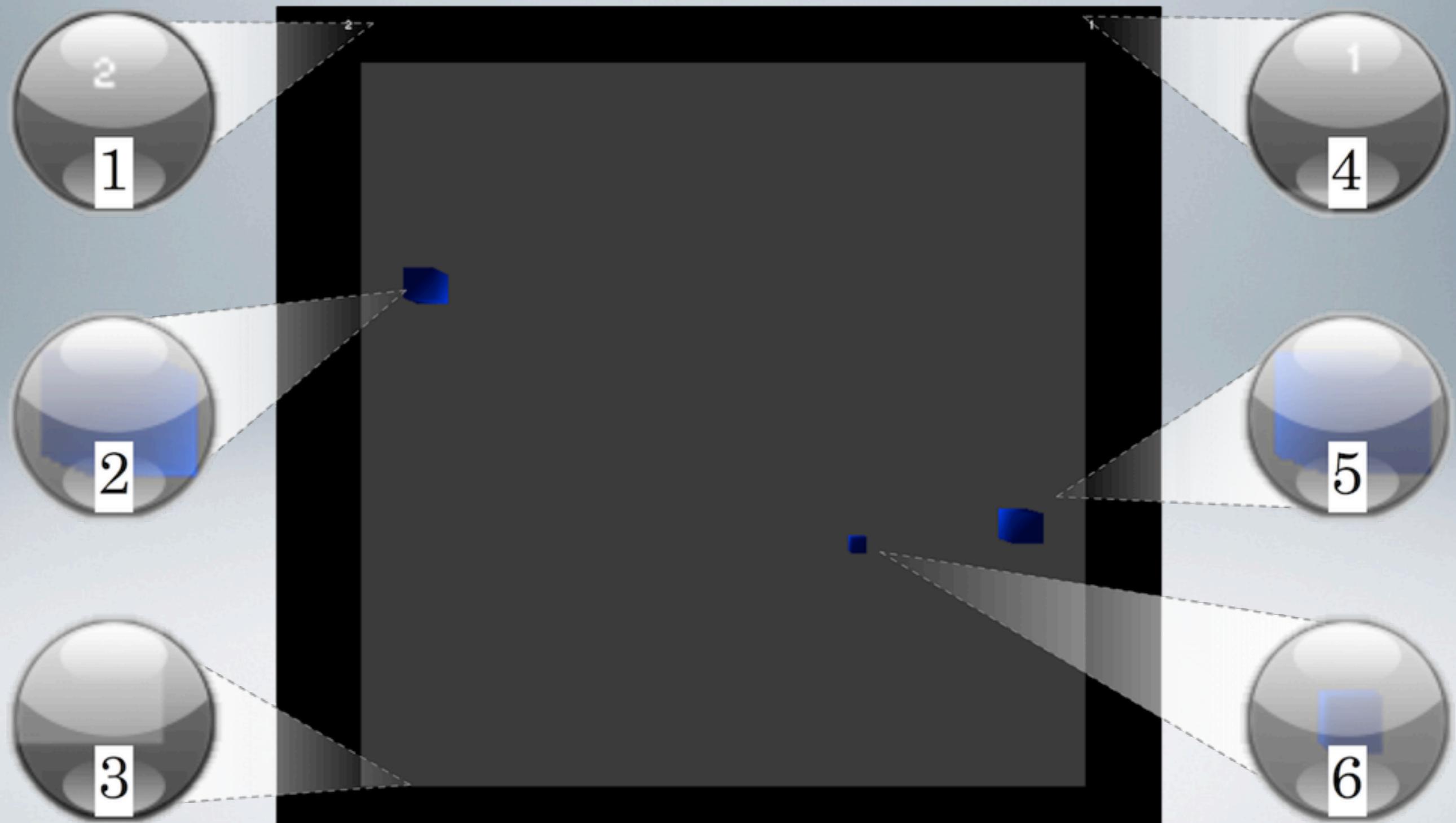
DESENVOLVIMENTO

- Protégé
- C++
- XML
- XCode / Mac
- Código proprietário com fins mercadológicos

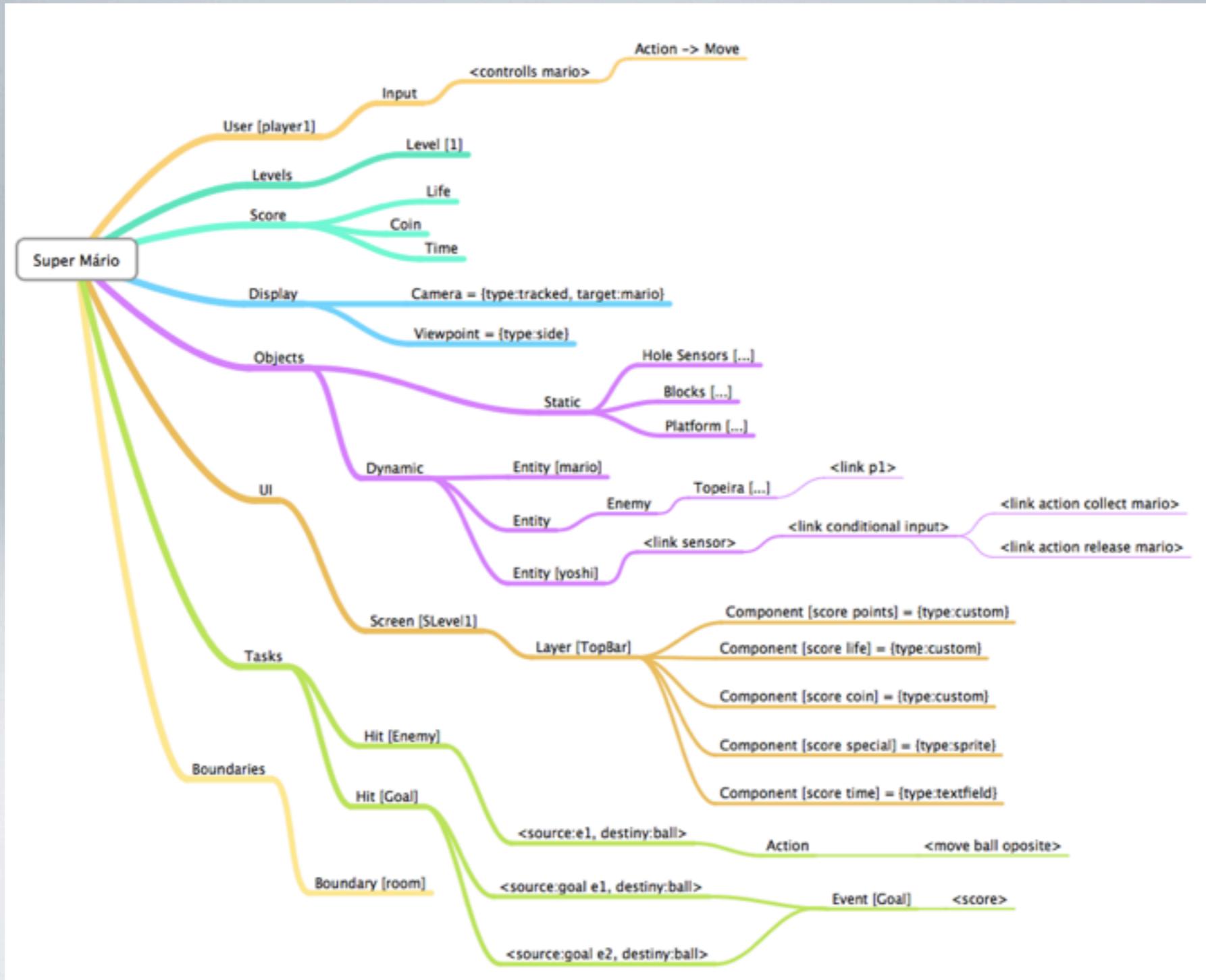
ESTUDO DE CASO



ESTUDO DE CASO



ESTUDO DE CASO



CONCLUSÃO

- DIGEO
- DIGEO Languague
- MuGE

TRABALHOS FUTUROS

- Continuar com a definição de classes e propriedades da DIGEO através de mais exemplos práticos.
- Propor a especificação formal da DIGEO Language.
- Lançar o motor proprietário.
- Criar editor visual.

OBRIGADO.

Monografia, ontologia e apresentação disponíveis em
www.filipesilvestrim.com/digeo