

VISEDU-SIMULA 1.0: VISUALIZADOR DE MATERIAL EDUCACIONAL, MÓDULO DE ANIMAÇÃO COMPORTAMENTAL

Gustavo Rufino Feltrin

Orientador: Dalton Solano dos Reis



FURB - Universidade Regional de Blumenau
DSC - Departamento de Sistemas e Computação
Grupo de Pesquisa em Computação Gráfica, Processamento de
Imagens e Entretenimento Digital
www.inf.furb.br/gcg

Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Requisitos / Especificação
- Implementação
- Resultados e Discussões
- Conclusão / Extensões

Introdução

- Animação Comportamental
 - Pesquisas
 - Entretenimento
 - *Creatures* (Redes Neurais)
 - *The Sims* (Seres Humanos Virtuais)
- Simulador
 - Ambiente
 - Regras
 - Personagens

Objetivos

Criar um simulador 2D para geração de Animações Comportamentais

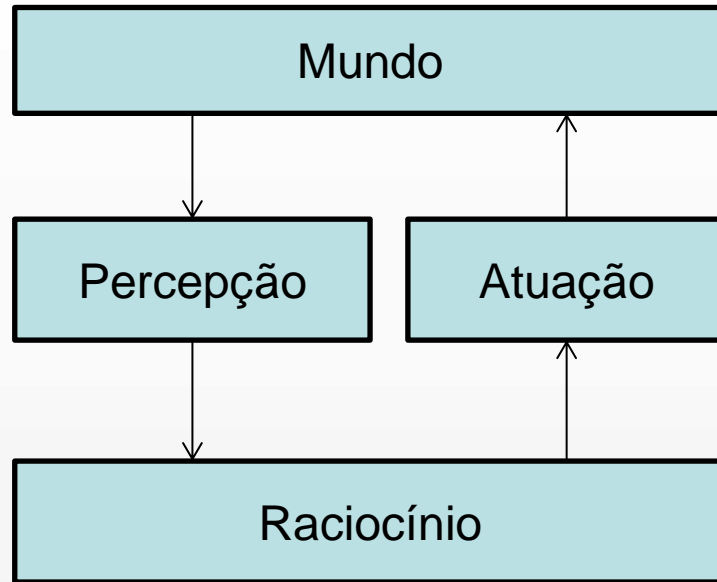
- Estender o projeto VisEdu-Engine para permitir criar Animações Comportamentais
- Proporcionar controle mínimo da percepção, do raciocínio e da atuação do personagem de forma desacoplada da inteligência utilizada
- Utilizar um modelo clássico de Inteligência Artificial (reativo, rede neural, sistema especialista, *Belief–Desire–Intention* (BDI), entre outros) para poder testar o simulador

Fundamentação Teórica

- Animação Comportamental
- Simuladores – Jogos
- HTML5 e Javascript
- Trabalhos Correlatos

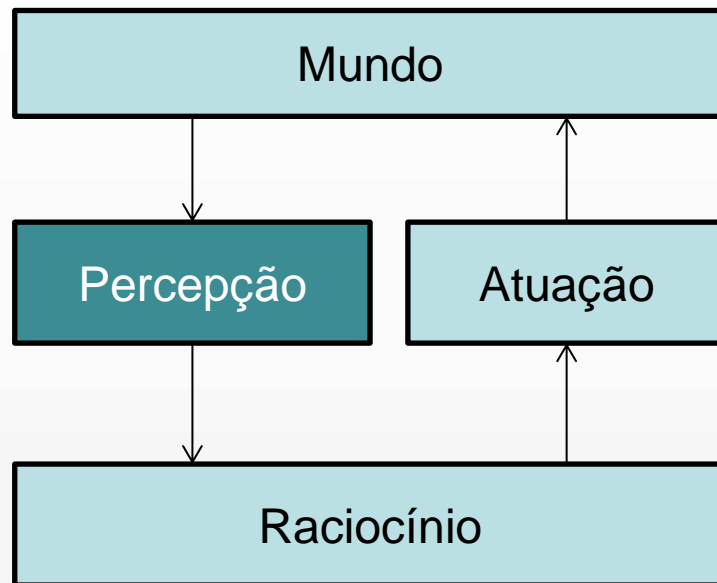
Animação Comportamental

- Percepção
- Raciocínio
- Atuação



Percepção

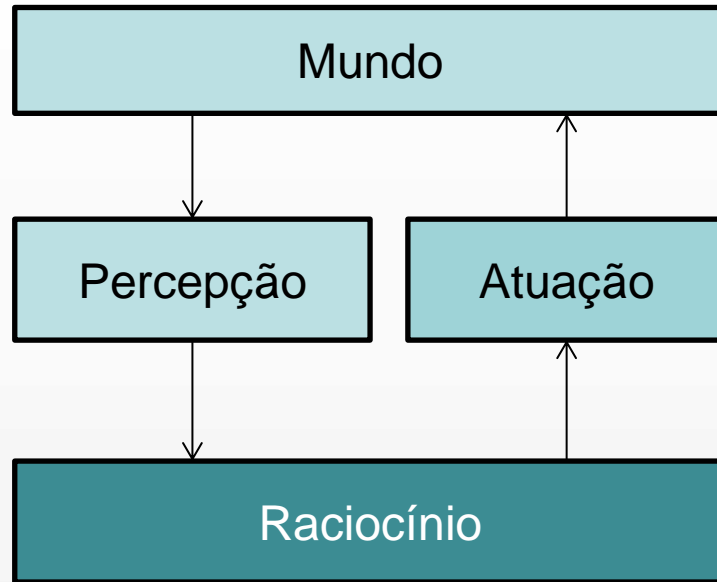
- Reconhecimento
- Informações
- Sensores



Raciocínio

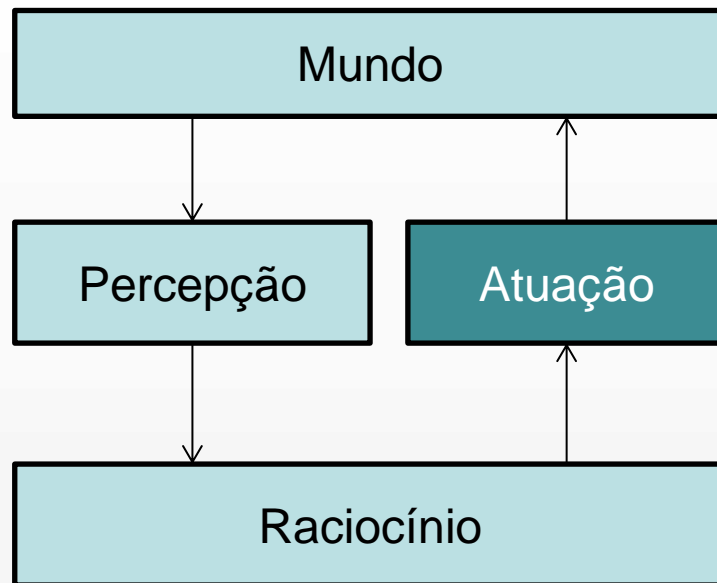
- Crenças
- Fatos
- Aceita verdades

- Aufere conclusões
- Hipóteses
 - Próprias
 - Alheias



Atuação

- Comportamento
- Resposta
- Estimulo externo



Plataforma Jason



- Modelo BDI
- Linguagem AgentSpeak
- Interpretador

Simuladores – Jogos

- Estudos de Von Neumann e Ulan
 - Análise ou técnica de Monte Carlo
- Características de Jogos
 - ambiente aberto/fechado
 - possui personagens
 - qualidades gráficas

Jogos sérios

- Utilizam tecnologias de jogos
- Propósito Profissional
 - treinamentos
 - obtenção de dados
 - simulações de fenômenos
- Não entretenimento
 - *e-learning*
 - simulação militar
 - treinamento médico

HTML5 e Javascript

- HTML5
 - Conteúdo
 - Aplicações complexas para web
 - Especificação finalizada
 - *WebSocket*
- Javascript
 - Linguagem para Web
 - Curva de aprendizagem baixa
 - Comportamento

Trabalhos Correlatos

- Unity 3D
- Mattedi (2007)
- Massive



- Motor de Jogos (2D, 3D)
- Multiplataforma (implantação)
- *Asset Store*
 - Repositório de recursos
 - Com/sem custo
 - Explora uso de Inteligência Artificial

Mattedi (2007)

- Aperfeiçoamento de Reações Comportamentais de Non-Player Character (NPC) no Jogo Doom
- Percepção do ambiente
- Instinto de Autopreservação

MASSIVE

- *Multiple Agent Simulation System In Virtual Environment* (MASSIVE)
- Simulador de Multiagentes
- Número de agentes ilimitados



Batalha dos campos de Pelennor
(O Senhor do Anéis – O Retorno do Rei)

Requisito Funcional

- Adicionar funcionalidades que permitam gerar Animações Comportamentais para os personagens
 - percepção do ambiente
 - raciocínio baseado nas percepções
 - execução da ação determinada pelo raciocínio

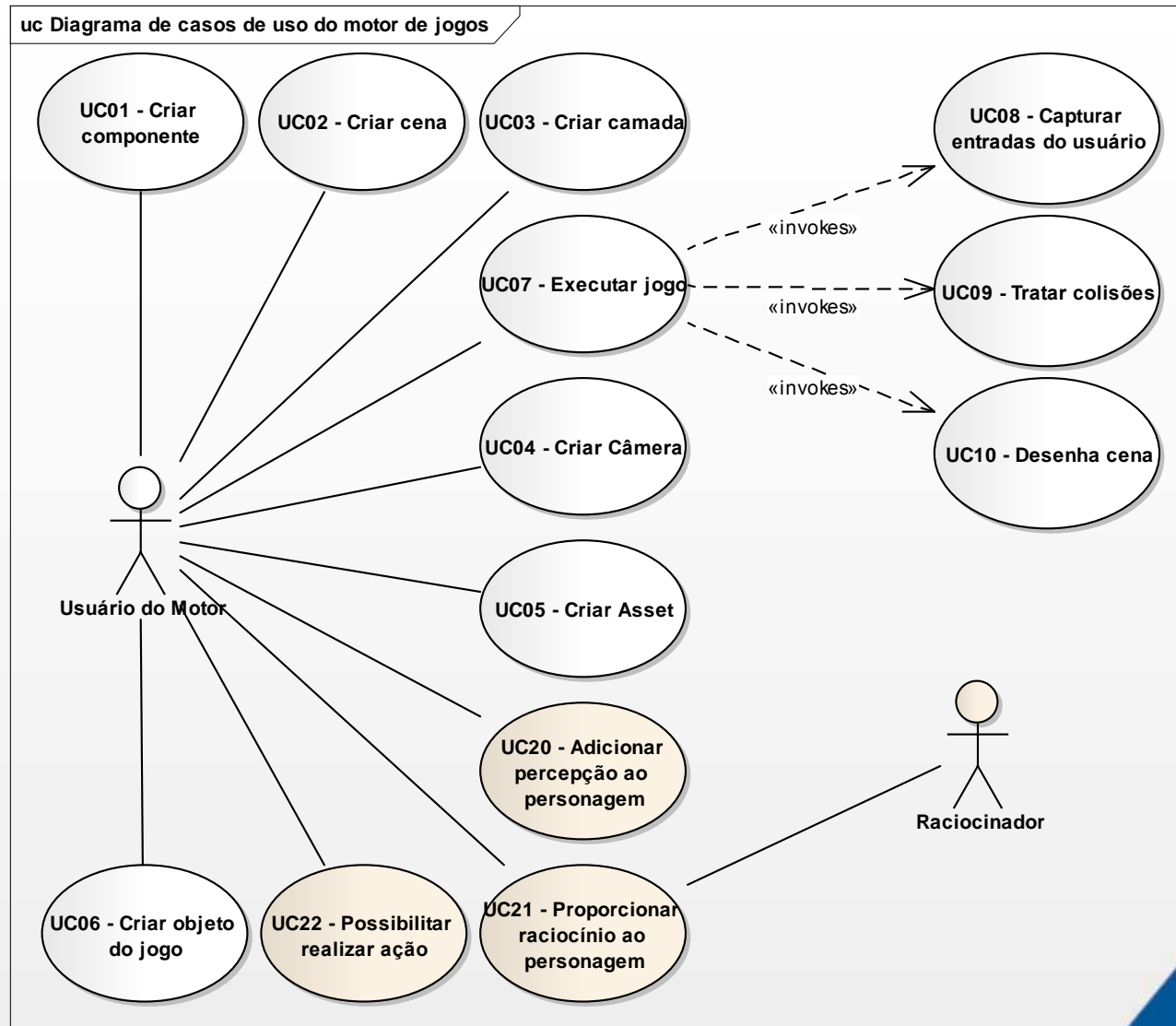
Requisitos Não Funcionais

- Ser desenvolvido de forma fortemente desacoplada
- Proporcionar o controle mínimo da percepção, do raciocínio e da atuação do personagem
- Utilizar a plataforma Jason (modelo BDI) no desenvolvimento de um "Raciocinador" que proporcione raciocínio para os personagens
- Ser compatível com os mesmos navegadores que o Motor de Jogos desenvolvido por Harbs (2013)
- Estender o Motor de Jogos utilizando HTML5 e Javascript
- Desenvolver um "Raciocinador" utilizando a linguagem Java e a plataforma Jason

Especificação

- Diagramas de Caso de Uso
- Diagramas de Pacote
- Diagrama Sequência
- Diagrama da Arquitetura

Diagrama de casos de uso do Motor de Jogos



Classes criadas ou que sofreram alterações dispostas em seus pacotes do Motor de Jogos

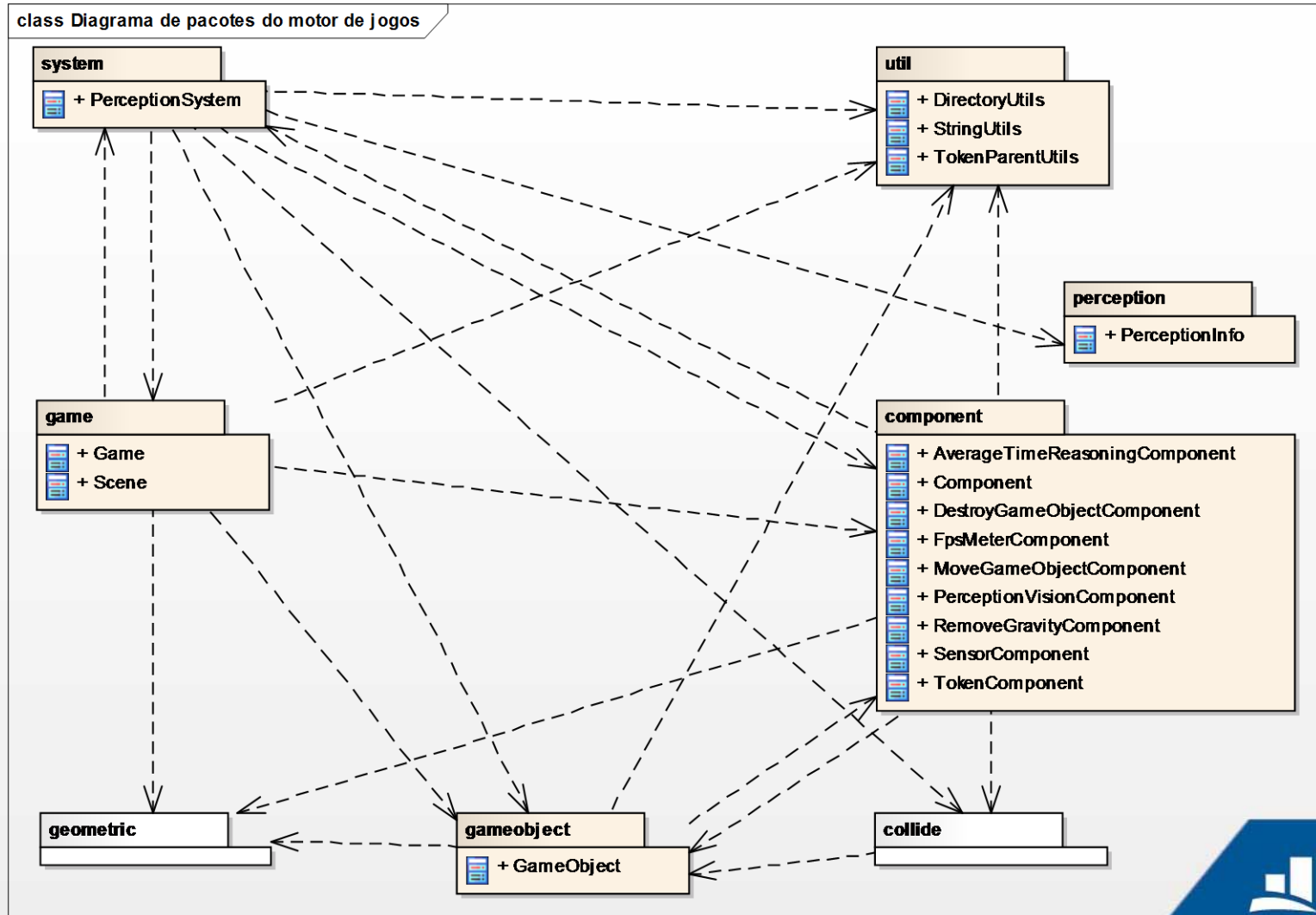
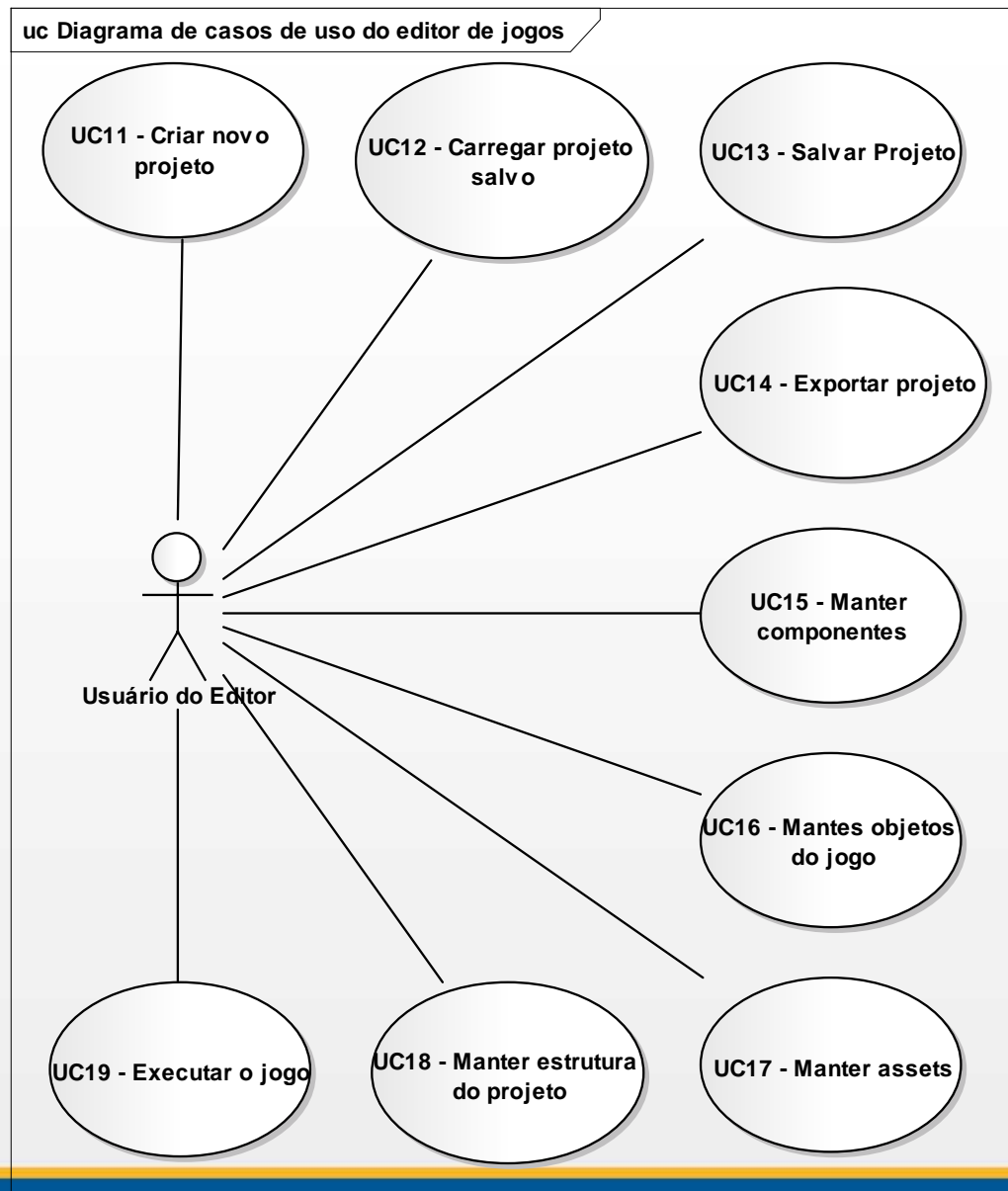


Diagrama de casos de uso do Editor de Jogos



Classes criadas ou que sofreram alteração dispostas em seus pacotes do Editor de Jogos

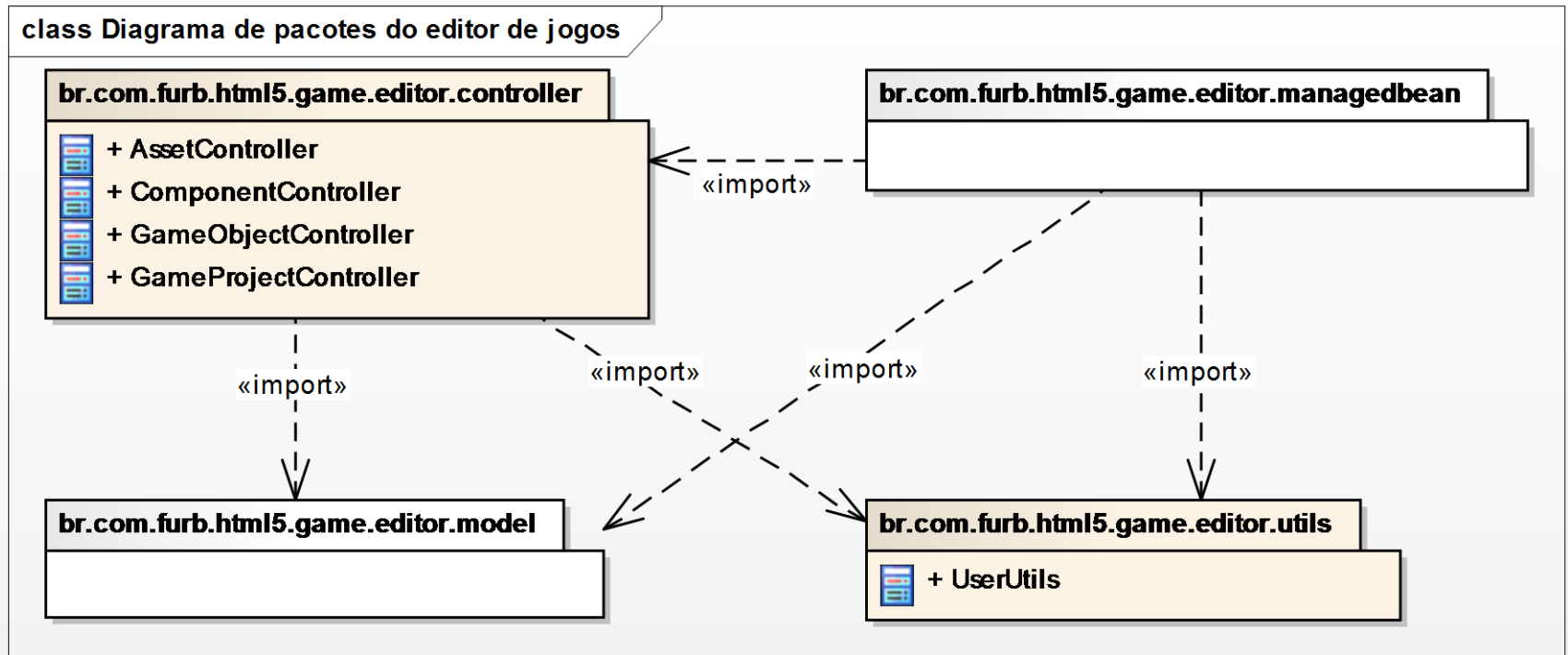


Diagrama de casos de uso do Raciocinador

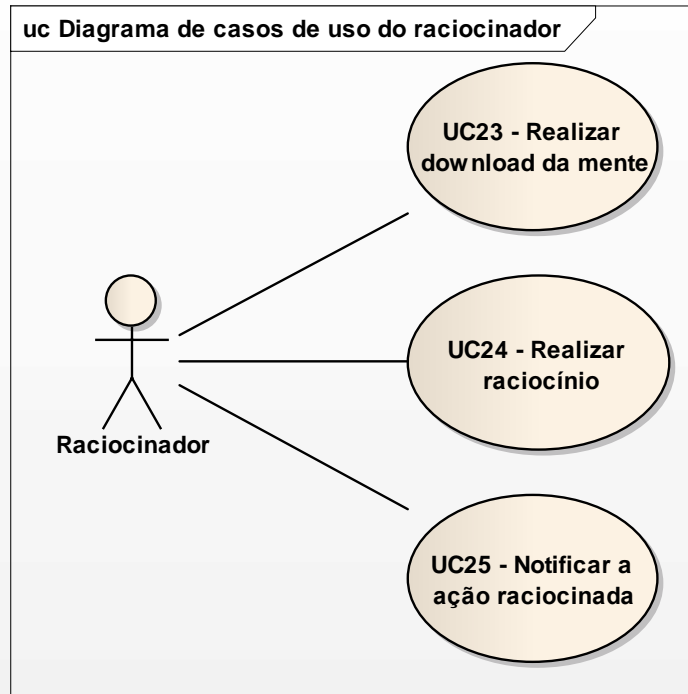


Diagrama de pacotes do Raciocinador

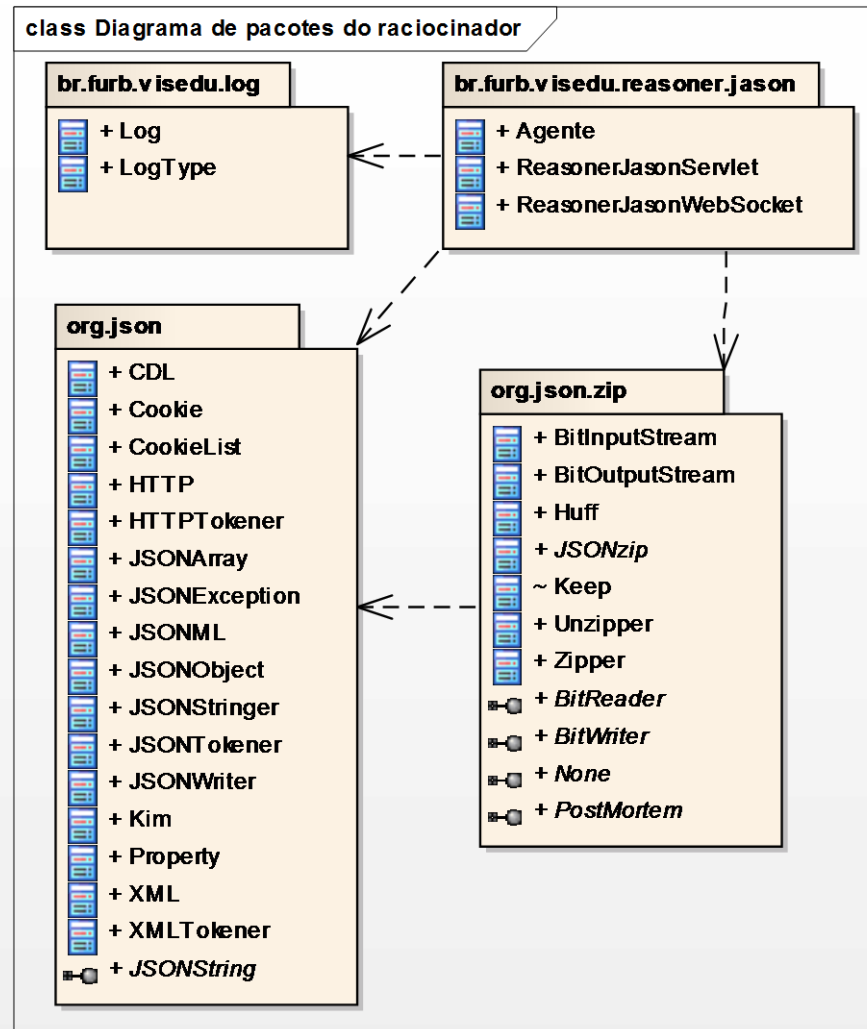


Diagrama de sequência do loop principal do motor de jogos

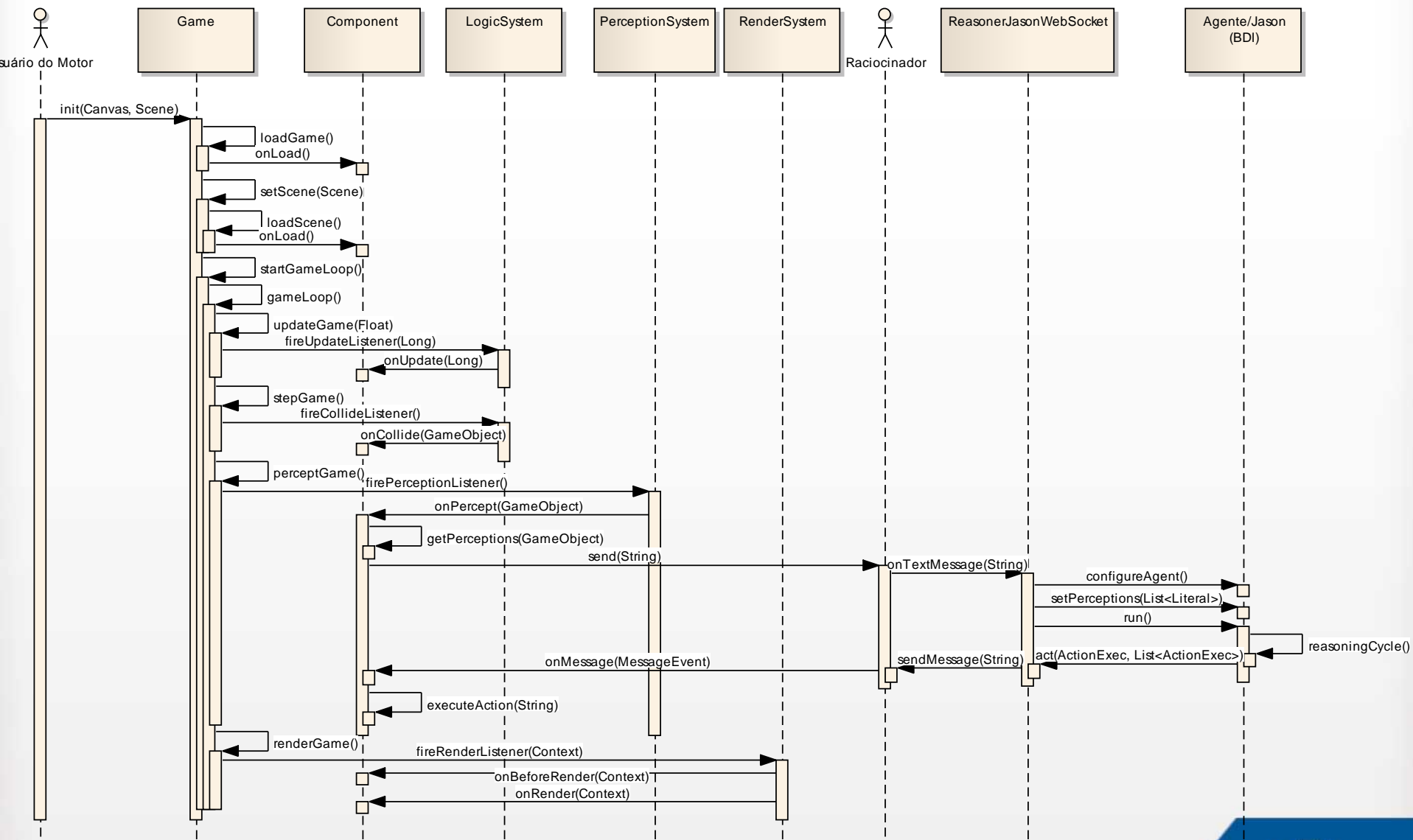
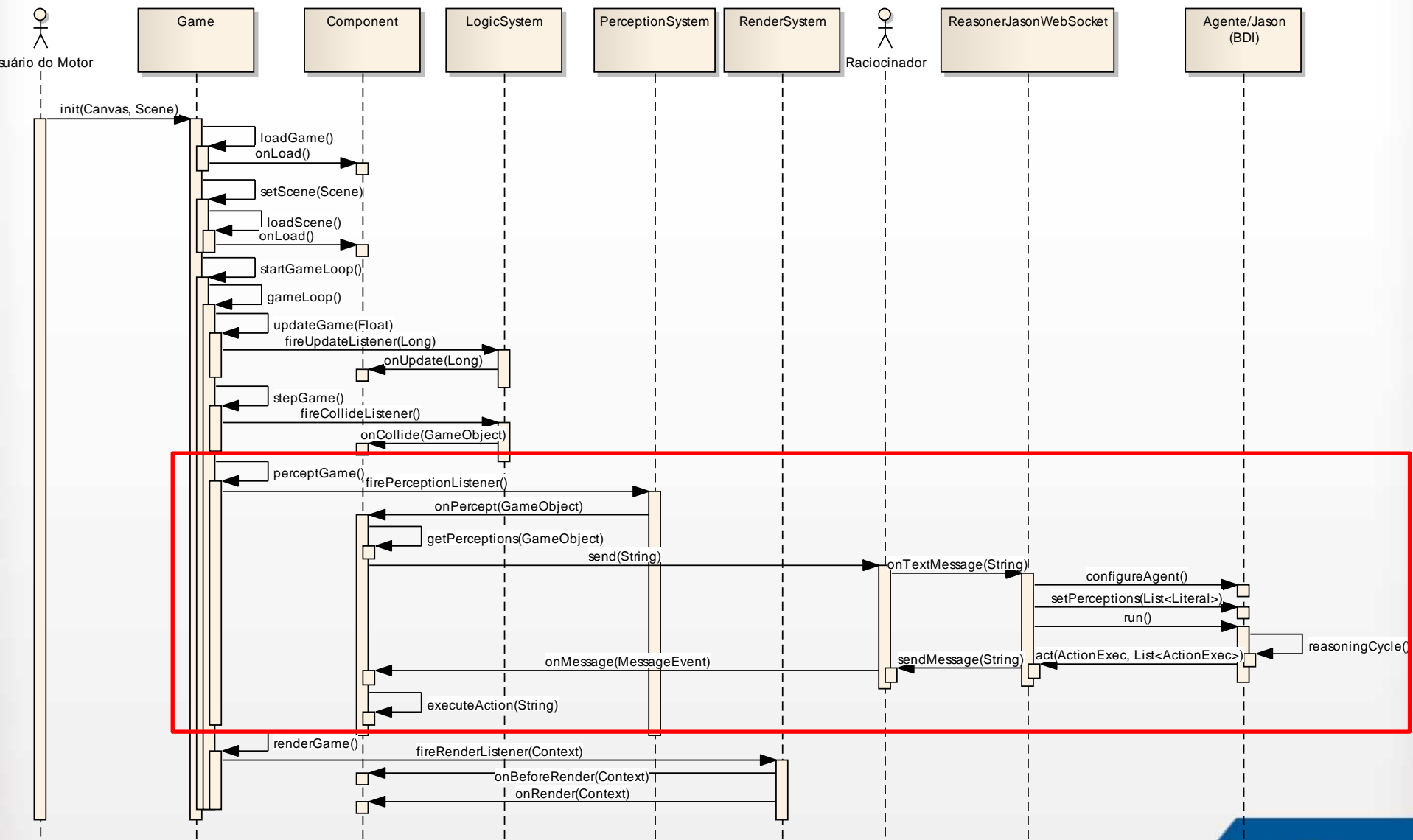


Diagrama de sequência do loop principal do motor de jogos



Editor de Jogos

Managed Beans

Controllers

Models

Raciocinador

Jason



Motor de Jogos

Game

Scene

Layer

Game
Object

Components

Logic
System

Render
System

Mouse
System

Key System

Kinect
System

Gamepad
System

Perception
System

Editor de Jogos

Managed Beans

Controllers

Models

Raciocinador

Jason



Motor de Jogos

Game

Scene

Layer

Game
Object

Components

Logic
System

Render
System

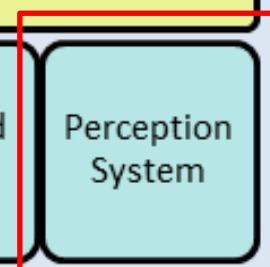
Mouse
System

Key System

Kinect
System

Gamepad
System

Perception
System



Implementação

- Técnicas
 - Orientação a Agentes
 - Orientação a Componentes
 - Orientação a Objetos
- Linguagens
 - AgentSpeak
 - HTML5
 - Java 7
 - Javascript
- Biblioteca
 - Jason
- Servidores de aplicação
 - Apache Tomcat
 - JBoss
- Ferramentas
 - Eclipse IDE for Java EE Developers
 - Sublime Text 2
- Plugins
 - Jasonide
 - JBoss Tools
- Navegadores
 - Google Chrome 38
 - Internet Explorer 11
 - Mozilla Firefox 33
 - Opera 25

Implementação

- Módulo de Animação Comportamental
 - Raciocinador
 - Motor de Jogos
 - Editor de Jogos


```

1 @WebServlet("/jason")
2 public class ReasonerJasonServlet extends WebSocketServlet {
3
4     private static final long serialVersionUID = 1L;
5
6     protected StreamInbound createWebSocketInbound(String
7         subProtocol, HttpServletRequest request) {
8         String agent = request.getParameter("agent");
9         String assetsDir = request.getParameter("assetsDir");
10        String aslDir = null;
11        if ( !assetsDir.contains( "html5-2d-game-editor" ) ) {
12            try {
13                Log.info( String.format("Start download of"
14                    + " \"%s\" @ %s", agent, assetsDir) );
15                aslDir = downloadAgentMind(agent, assetsDir);
16                Log.info( String.format("Download complete of"
17                    + " \"%s\" @ %s", agent, assetsDir) );
18            } catch (IOException e) {
19                Log.err( String.format("Download error. File:"
20                    + " \"%s\" @ %s", agent, assetsDir) );
21                e.printStackTrace();
22            }
23        }
24        return new ReasonerJasonWebSocket( agent, aslDir );
25    }
26
27    private static String downloadAgentMind(String name, String dir)
28        throws IOException {
29        File tmpAsl = File.createTempFile(name, ".asl");
30        URL urlFile = new URL( String.format("%s%s.asl", dir,
31            name) );
32        ReadableByteChannel rbc =
33            Channels.newChannel(urlFile.openStream());
34        FileOutputStream fos = new FileOutputStream( tmpAsl );
35        fos.getChannel().transferFrom(rbc, 0, Long.MAX_VALUE);
36        return tmpAsl.getAbsolutePath();
37    }
38 }
39

```

```

1 @WebServlet("/jason")
2 public class ReasonerJasonServlet extends WebSocketServlet {
3
4     private static final long serialVersionUID = 1L;
5
6     protected StreamInbound createWebSocketInbound(String
7         subProtocol, HttpServletRequest request) {
8         String agent = request.getParameter("agent");
9         String assetsDir = request.getParameter("assetsDir");
10        String aslDir = null;
11        if ( !assetsDir.contains( "html5-2d-game-editor" ) ) {
12            try {
13                Log.info( String.format("Start download of"
14                    + " \"%s\" @ %s", agent, assetsDir) );
15                aslDir = downloadAgentMind(agent, assetsDir);
16                Log.info( String.format("Download complete of"
17                    + " \"%s\" @ %s", agent, assetsDir) );
18            } catch (IOException e) {
19                Log.err( String.format("Download error. File:"
20                    + " \"%s\" @ %s", agent, assetsDir) );
21                e.printStackTrace();
22            }
23        }
24        return new ReasonerJasonWebSocket( agent, aslDir );
25    }
26
27    private static String downloadAgentMind(String name, String dir)
28        throws IOException {
29        File tmpAsl = File.createTempFile(name, ".asl");
30        URL urlFile = new URL( String.format("%s%s.asl", dir,
31            name) );
32        ReadableByteChannel rbc =
33            Channels.newChannel(urlFile.openStream());
34        FileOutputStream fos = new FileOutputStream( tmpAsl );
35        fos.getChannel().transferFrom(rbc, 0, Long.MAX_VALUE);
36        return tmpAsl.getAbsolutePath();
37    }
38 }
39

```

```
1 @WebServlet("/jason")
2 public class ReasonerJasonServlet extends WebSocketServlet {
3
4     private static final long serialVersionUID = 1L;
5
6     protected StreamInbound createWebSocketInbound(String
7         subProtocol, HttpServletRequest request) {
8         String agent = request.getParameter("agent");
9         String assetsDir = request.getParameter("assetsDir");
10        String aslDir = null;
11        if ( !assetsDir.contains( "html5-2d-game-editor" ) ) {
12            try {
13                Log.info( String.format("Start download of"
14                    + " \"%s\" @ %s", agent, assetsDir) );
15                aslDir = downloadAgentMind(agent, assetsDir);
16                Log.info( String.format("Download complete of"
17                    + " \"%s\" @ %s", agent, assetsDir) );
18            } catch (IOException e) {
19                Log.err( String.format("Download error. File:"
20                    + " \"%s\" @ %s", agent, assetsDir) );
21                e.printStackTrace();
22            }
23        }
24        return new ReasonerJasonWebSocket( agent, aslDir );
25    }
26
27    private static String downloadAgentMind(String name, String dir)
28        throws IOException {
29        File tmpAsl = File.createTempFile(name, ".asl");
30        URL urlFile = new URL( String.format("%s%s.asl", dir,
31            name) );
32        ReadableByteChannel rbc =
33            Channels.newChannel(urlFile.openStream());
34        FileOutputStream fos = new FileOutputStream( tmpAsl );
35        fos.getChannel().transferFrom(rbc, 0, Long.MAX_VALUE);
36        return tmpAsl.getAbsolutePath();
37    }
38 }
39
```

```

5 private JSONObject json;
6 private Agente ag;
7
8 public ReasonerJasonWebSocket(String agentName, String aslDir) {
9     Log.info("new Agent: " + agentName + " @ " + aslDir);
10    ag = new Agente(agentName, aslDir, this);
11 }
12 ...
13 @Override
14 protected void onTextMessage(CharBuffer msg) throws IOException {
15     Log.info("onTextMessage: " + msg);
16
17     if ( HAND_SHAKE.equalsIgnoreCase(msg.toString()) ) {
18         sendMessage(HAND_SHAKE);
19         return;
20     }
21
22     json = new JSONObject(msg.toString());
23     String sPerceptions = (String) json.get("perceptions");
24     JSONArray ja = new JSONArray(sPerceptions);
25
26     List<Literal> perceptions = new ArrayList<Literal>();
27     for (int i = 0; i < ja.length(); i++) {
28         perceptions.add( Literal.parseLiteral(
29             ja.getString(i) ) );
30     }
31     ag.configureAgent();
32     ag.setPerceptions(perceptions);
33     ag.run();
34 }

```



```
5 private JSONObject json;
6 private Agente ag;
7
8 public ReasonerJasonWebSocket(String agentName, String aslDir) {
9     Log.info("new Agent: " + agentName + " @ " + aslDir);
10    ag = new Agente(agentName, aslDir, this);
11 }
12 ...
13 @Override
14 protected void onTextMessage(CharBuffer msg) throws IOException {
15     Log.info("onTextMessage: " + msg);
16
17     if ( HAND_SHAKE.equalsIgnoreCase(msg.toString()) ) {
18         sendMessage(HAND_SHAKE);
19         return;
20     }
21
22     json = new JSONObject(msg.toString());
23     String sPerceptions = (String) json.get("perceptions");
24     JSONArray ja = new JSONArray(sPerceptions);
25
26     List<Literal> perceptions = new ArrayList<Literal>();
27     for (int i = 0; i < ja.length(); i++) {
28         perceptions.add( Literal.parseLiteral(
29             ja.getString(i) ) );
30     }
31     ag.configureAgent();
32     ag.setPerceptions(perceptions);
33     ag.run();
34 }
```

```

5     private JSONObject json;
6     private Agente ag;
7
8     public ReasonerJasonWebSocket(String agentName, String aslDir) {
9         Log.info("new Agent: " + agentName + " @ " + aslDir);
10        ag = new Agente(agentName, aslDir, this);
11    }
12
13    @Override
14    protected void onTextMessage(CharBuffer msg) throws IOException {
15        Log.info("onTextMessage: " + msg);
16
17        if ( HAND_SHAKE.equalsIgnoreCase(msg.toString()) ) {
18            sendMessage(HAND_SHAKE);
19            return;
20        }
21
22        json = new JSONObject(msg.toString());
23        String sPerceptions = (String) json.get("perceptions");
24        JSONArray ja = new JSONArray(sPerceptions);
25
26        List<Literal> perceptions = new ArrayList<Literal>();
27        for (int i = 0; i < ja.length(); i++) {
28            perceptions.add( Literal.parseLiteral(
29                ja.getString(i) ) );
30        }
31        ag.configureAgent();
32        ag.setPerceptions(perceptions);
33        ag.run();
34    }

```

```

5     private JSONObject json;
6     private Agente ag;
7
8     public ReasonerJasonWebSocket(String agentName, String aslDir) {
9         Log.info("new Agent: " + agentName + " @ " + aslDir);
10        ag = new Agente(agentName, aslDir, this);
11    }
12    ...
13    @Override
14    protected void onTextMessage(CharBuffer msg) throws IOException {
15        Log.info("onTextMessage: " + msg);
16
17        if ( HAND_SHAKE.equalsIgnoreCase(msg.toString()) ) {
18            sendMessage(HAND_SHAKE);
19            return;
20        }
21
22        json = new JSONObject(msg.toString());
23        String sPerceptions = (String) json.get("perceptions");
24        JSONArray ja = new JSONArray(sPerceptions);
25
26        List<Literal> perceptions = new ArrayList<Literal>();
27        for (int i = 0; i < ja.length(); i++) {
28            perceptions.add( Literal.parseLiteral(
29                ja.getString(i) ) );
30        }
31        ag.configureAgent();
32        ag.setPerceptions(perceptions);
33        ag.run();
34    }

```



```

5 private JSONObject json;
6 private Agente ag;
7
8 public ReasonerJasonWebSocket(String agentName, String aslDir) {
9     Log.info("new Agent: " + agentName + " @ " + aslDir);
10    ag = new Agente(agentName, aslDir, this);
11 }
12 ...
13 @Override
14 protected void onTextMessage(CharBuffer msg) throws IOException {
15     Log.info("onTextMessage: " + msg);
16
17     if ( HAND_SHAKE.equalsIgnoreCase(msg.toString()) ) {
18         sendMessage(HAND_SHAKE);
19         return;
20     }
21
22     json = new JSONObject(msg.toString());
23     String sPerceptions = (String) json.get("perceptions");
24     JSONArray ja = new JSONArray(sPerceptions);
25
26     List<Literal> perceptions = new ArrayList<Literal>();
27     for (int i = 0; i < ja.length(); i++) {
28         perceptions.add( Literal.parseLiteral(
29             ja.getString(i) ) );
30     }
31
32     ag.configureAgent();
33     ag.setPerceptions(perceptions);
34     ag.run();
35 }

```



```
5 private JSONObject json;
6 private Agente ag;
7
8 public ReasonerJasonWebSocket(String agentName, String aslDir) {
9     Log.info("new Agent: " + agentName + " @ " + aslDir);
10    ag = new Agente(agentName, aslDir, this);
11 }
12 ...
13 @Override
14 protected void onTextMessage(CharBuffer msg) throws IOException {
15     Log.info("onTextMessage: " + msg);
16
17     if ( HAND_SHAKE.equalsIgnoreCase(msg.toString()) ) {
18         sendMessage(HAND_SHAKE);
19         return;
20     }
21
22     json = new JSONObject(msg.toString());
23     String sPerceptions = (String) json.get("perceptions");
24     JSONArray ja = new JSONArray(sPerceptions);
25
26     List<Literal> perceptions = new ArrayList<Literal>();
27     for (int i = 0; i < ja.length(); i++) {
28         perceptions.add( Literal.parseLiteral(
29             ja.getString(i) ) );
30     }
31     ag.configureAgent();
32     ag.setPerceptions(perceptions);
33     ag.run();
34 }
```

Jason determinando ação para a simulação

```
1 @Override
2 public void act(ActionExec action, List<ActionExec> feedback) {
3     showInfo("Agent " + getAgName() + ": doing: " +
4         action.getActionTerm());
5     getWs().sendMessage( action.getActionTerm().toString() );
6     action.setResult(true);
7     feedback.add(action);
8 }
```

Mente reativa implementada em AgentSpeak

```
1 +onPercept (C)
2     <- changeMyFillStyle (C) .
```

Evento para disparo da percepção

```
Component.js  x
226  ~/
227  Component.prototype.onPercept = function(gameObject){}
```

Loop do jogo com o *pipeline* de percepção

```
Game.js  x
116  this.gameloop = function(){
117      var deltaTime = (Date.now() - this.lastUpdateTime) / 1000;
118      if(!Game.paused){
119          this.updateGame(deltaTime);
120          this.stepGame();
121          this.perceptGame();
122          this.renderGame();
123      }
124      this.lastUpdateTime = Date.now();
125  }
```

Evento para disparo da percepção

```
Component.js  x
226  ~/
227  Component.prototype.onPercept = function(gameObject){}
```

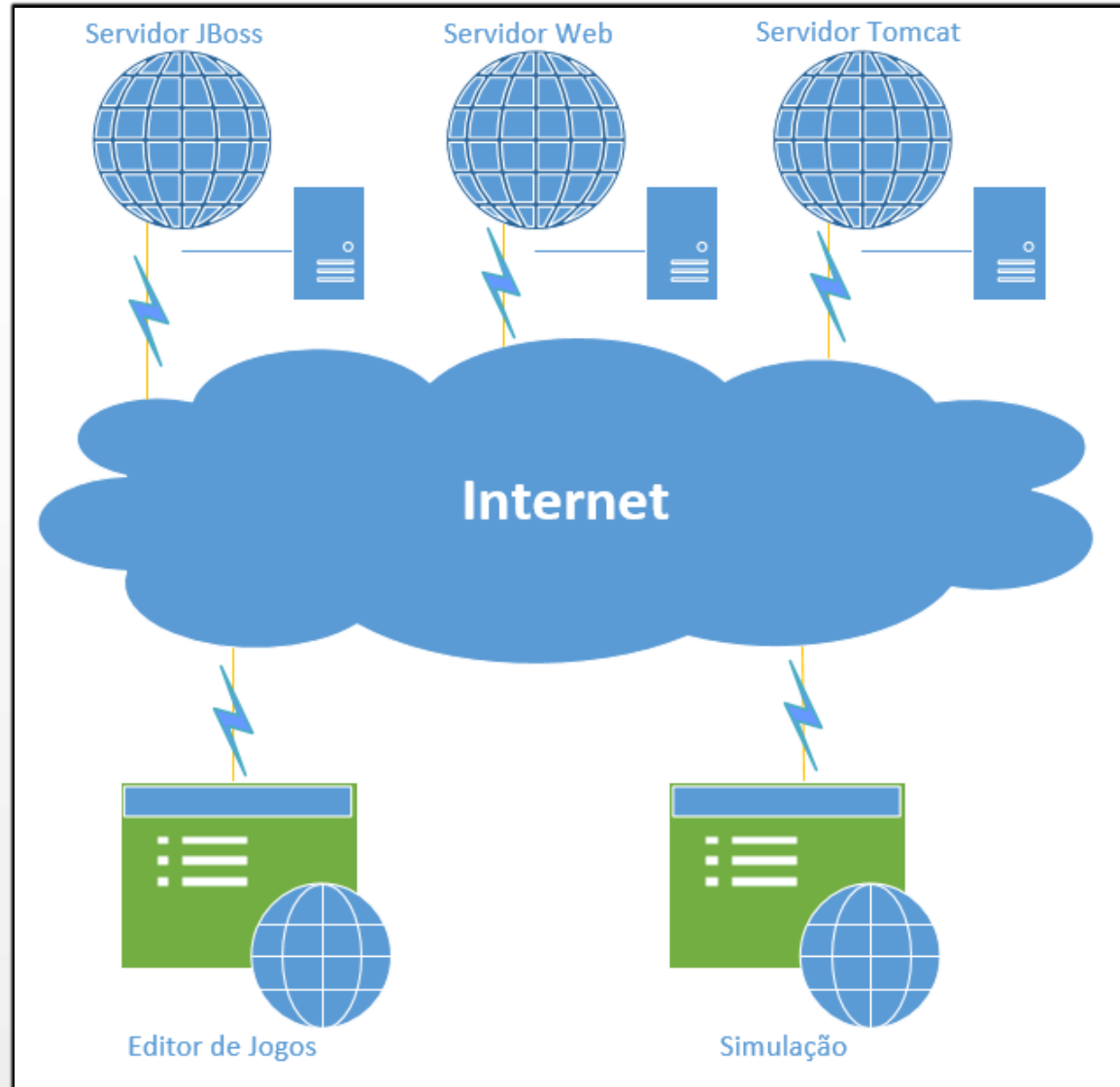
Loop do jogo com o *pipeline* de percepção

```
Game.js  x
116  this.gameloop = function(){
117      var deltaTime = (Date.now() - this.lastUpdateTime) / 1000;
118      if(!Game.paused){
119          this.updateGame(deltaTime);
120          this.stepGame();
121          this.perceptGame();
122          this.renderGame();
123      }
124      this.lastUpdateTime = Date.now();
125  }
```

Resultados e Discussões

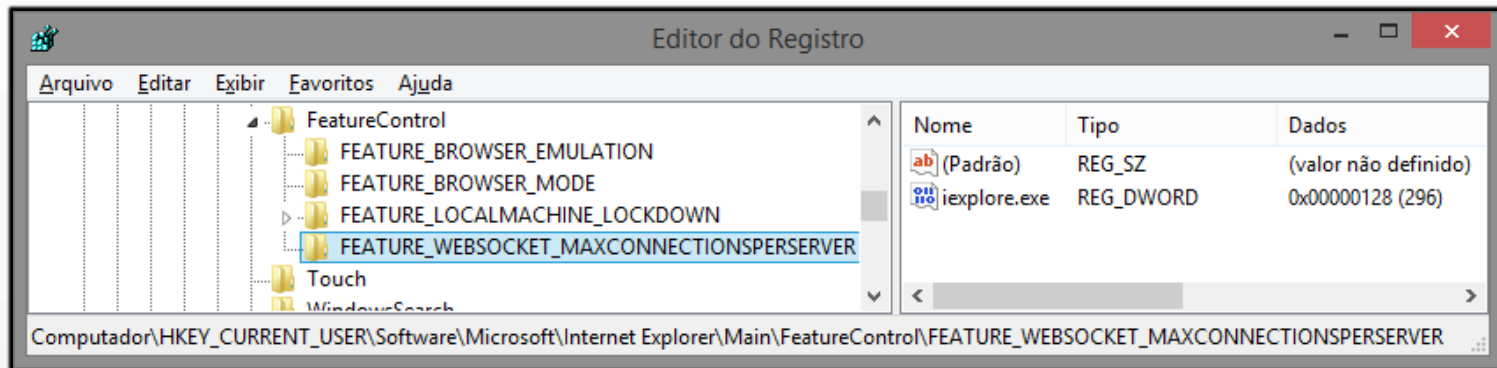
- Arquitetura desenvolvida
- Limitação
- Testes de desempenho
- Trabalhos correlatos

Arquitetura desenvolvida



Limitações

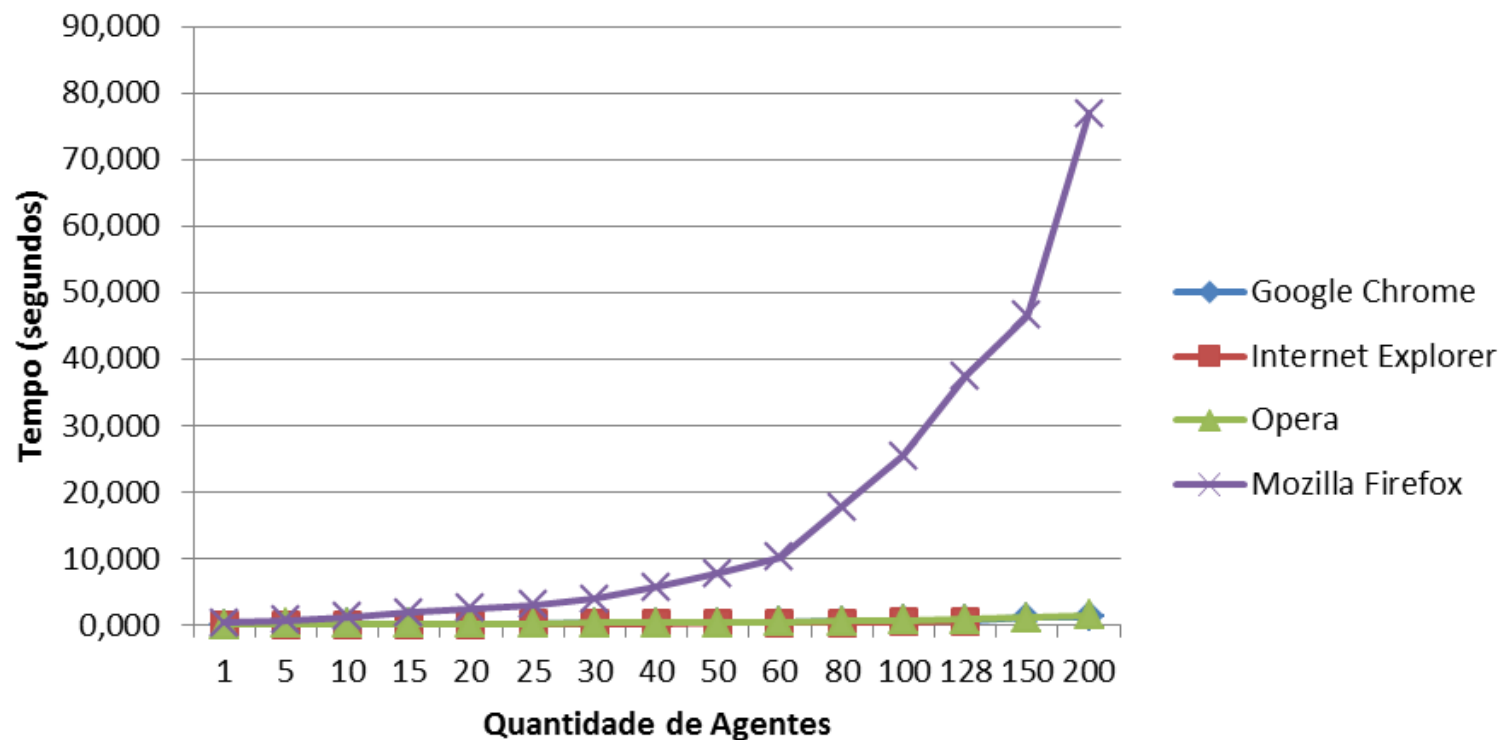
- Internet Explorer – Seis conexões



- Editor de Jogos - *download asset* .as1.

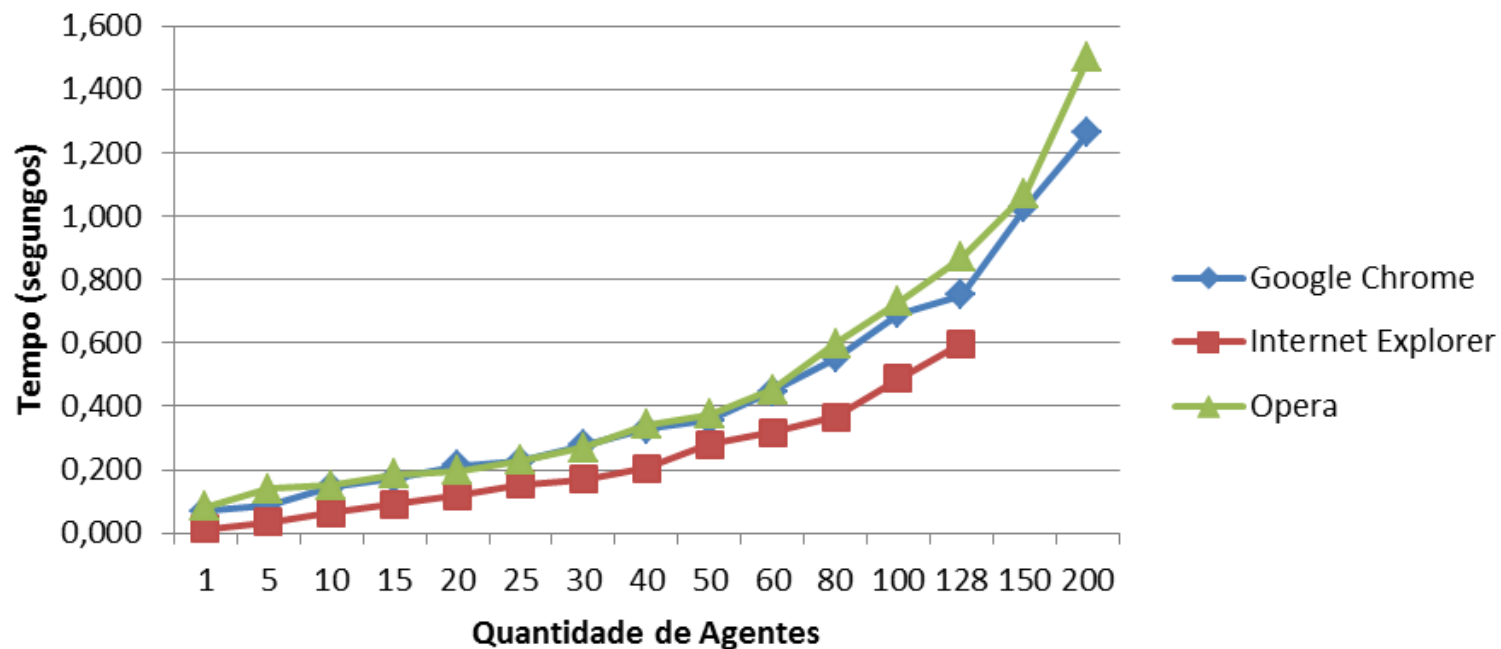
Testes de desempenho

Comparação de navegadores entre conexão do cliente com o servidor



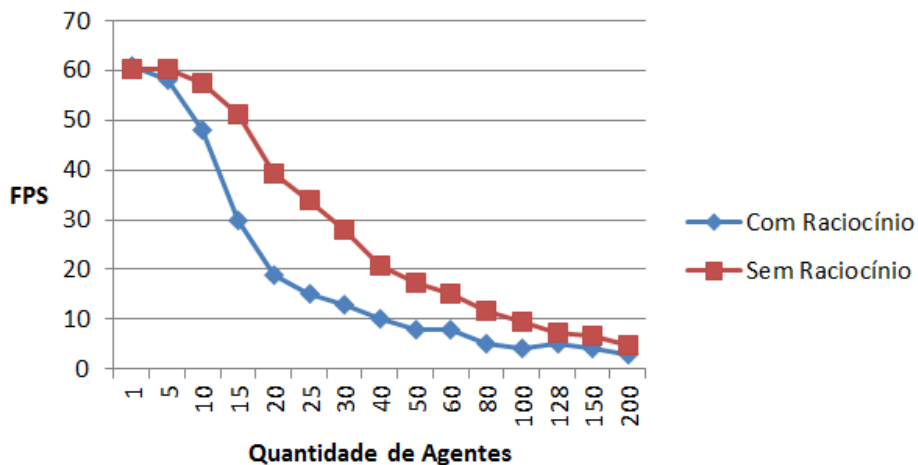
Testes de desempenho

Comparação de navegadores entre conexão do cliente com o servidor (sem Mozilla Firefox)

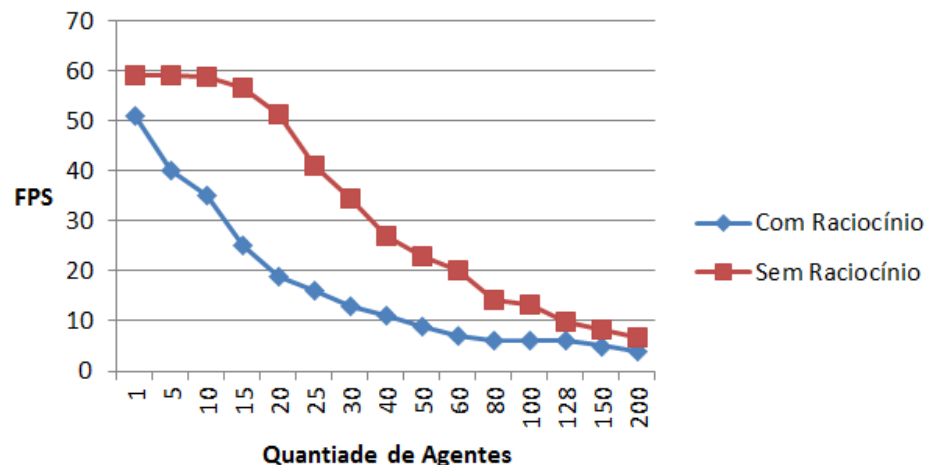


Quantidade de quadros por segundo (FPS) de uma mesma simulação com e sem a presença de raciocínio

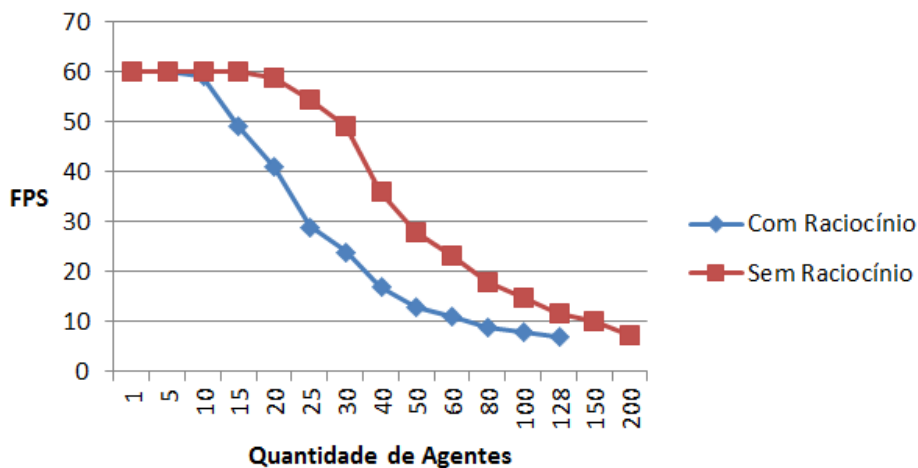
Google Chrome



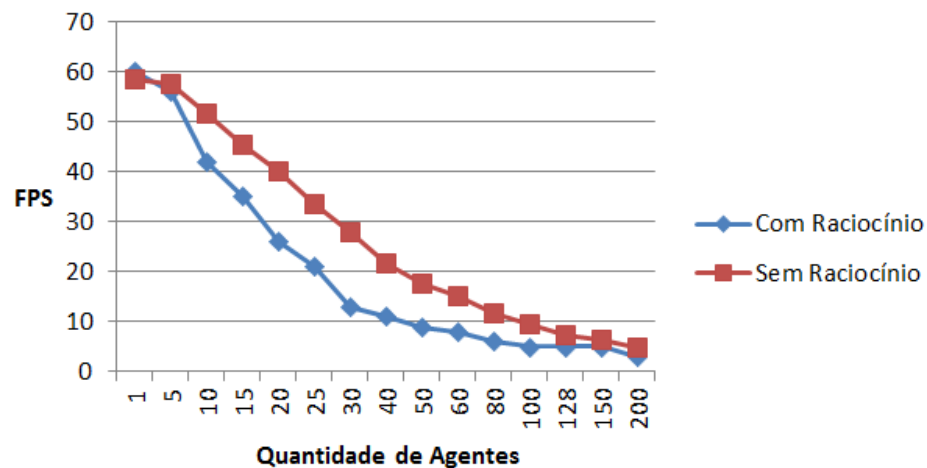
Mozilla Firefox



Internet Explorer



Opera



Comparativo com correlatos

Característica	Unity 3D	Mattedi	Massive	VisEdu-Simula
Proporciona Animação Comportamental	X*	X	X	X
Desacoplado do modelo de IA	X	-	-	X
Controle da percepção, raciocínio e ação	-	X	X	X
Utiliza Modelo Clássico de IA	X	X	X	X
Desenvolvido com HTML5 e JavaScript	-	-	-	X
Disponível para acesso Web	-	-	-	X
Permite criar comportamento imprevisível	X*	-	X*	X
Eficiência com grandes quantidades de agentes	-	-	X	-
Interação com o ambiente	X*	X	X	X
Tipos variados de mente	X*	-	X	X

* Características não propriamente identificadas, porém acredita-se que os correlatos as tenham.

Conclusões

- Atendendo os requisitos o Módulo de Animação Comportamental cumpre os objetivos propostos
- O Motor de Jogos possui dois pontos positivos
 - Orientação a componentes
 - Coesão da estrutura
- O Editor de Jogos possui um ponto negativo
 - Usabilidade
- Plataforma Jason flexível e escalável
- Modelo mental desacoplado ao módulo desenvolvido

Extensões

- Criar mentes Jason mais elaboradas e de capacidades distintas
- Utilizar outras técnicas de IA
- Resolver problemática do Internet Explorer (subdomínios)
- Evitar cálculos de colisão dos sensores
- Utilizar Web Workers para propagação das percepções
- Identificar término de colisão (para uso da percepção) através da Box2DJS
- Estudar formas de campos de visão para novos sentidos
- Evitar colisão do agente com o próprio sensor
- Viabilizar *download* de *assets* quando executados pelo Editor de Jogos

Demonstração

Obrigado !

Mensagem enviada pela simulação para o Raciocinador

```
1 {  
2   "origin": "1d252436-04b8-4756-b1fd-510d11ea15fe",  
3   "target": "467de1a7-a811-4bb2-9ccd-c86003b9408a",  
4   "perceptions": "[\"onPercept(\\\\"#673dd7\\\\"")\\"]"  
5 }
```

Mensagem enviada pelo Raciocinador para a simulação

```
1 {  
2   "origin": "1d252436-04b8-4756-b1fd-510d11ea15fe",  
3   "target": "467de1a7-a811-4bb2-9ccd-c86003b9408a",  
4   "perceptions": "[\"onPercept(\\\\"#673dd7\\\\"")\\"]",  
5   "action": "changeMyFillStyle(\\\\"#673dd7\\") "  
6 }
```

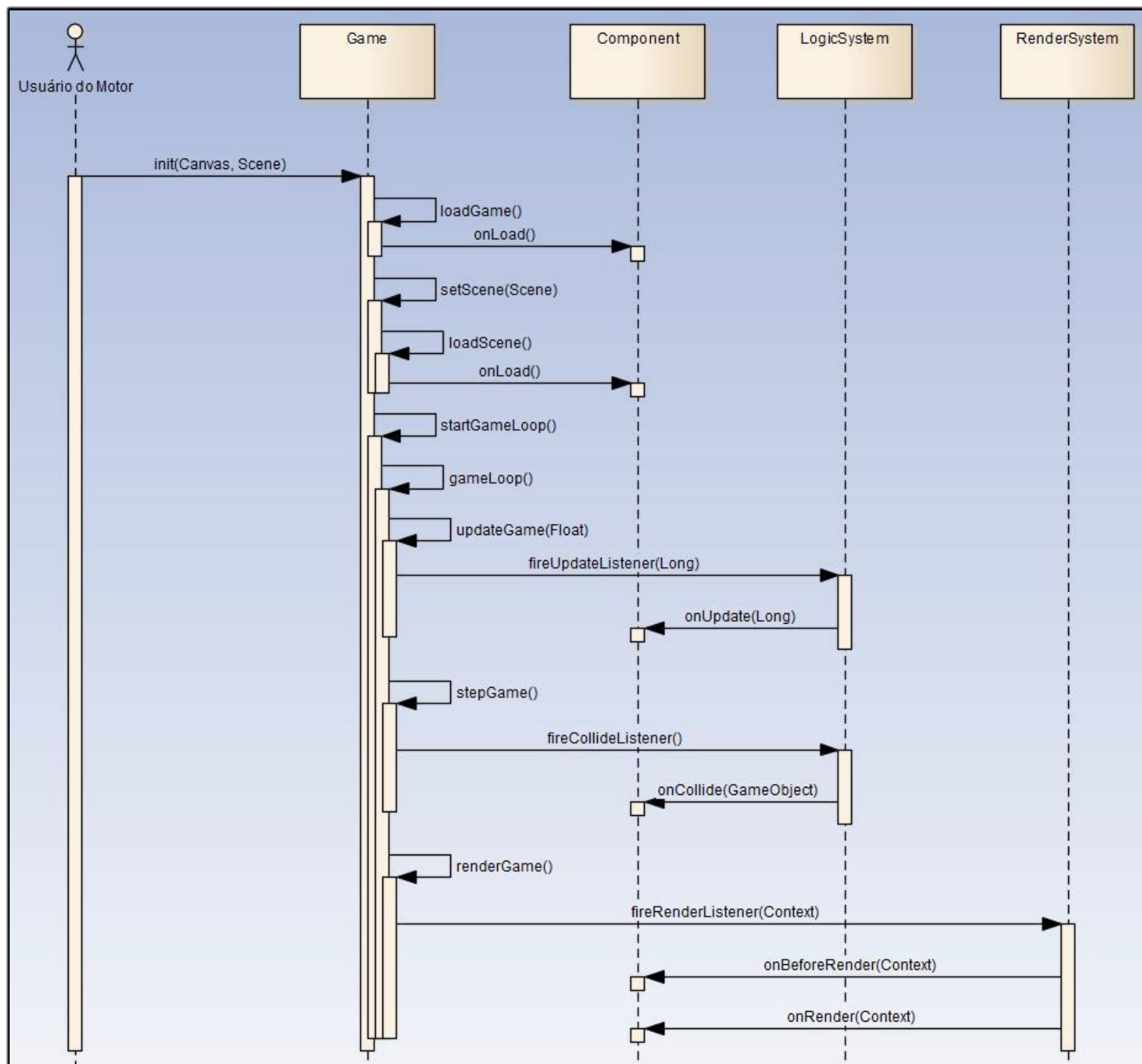



Diagrama de sequência do loop principal do motor de jogos

