

# **TAGARELA 2.0: FRAMEWORK DE COMUNICAÇÃO ALTERNATIVA, MÓDULO DE JOGOS**

**Acadêmico: Elvis Merten Marques**

**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](http://www.inf.furb.br/gcg)



# Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Requisitos
- Especificação
- Implementação e operacionalidade
- Resultados e discussões
- Conclusões e sugestões

# Introdução

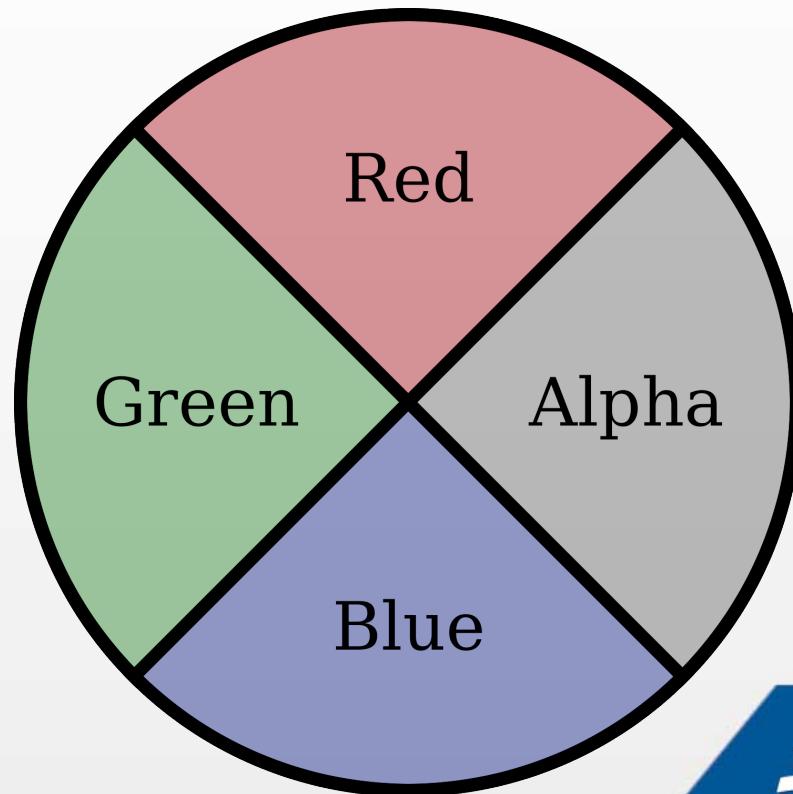
- Comunicação Alternativa
- Tecnologia assistiva
  - Tecnologias que servem para melhorar a qualidade de vida de pacientes
- Jogos educacionais

# Objetivos

- migrar o “Jogo de Letras/Números Voltado para a Tecnologia Assistiva no Android ” (REETZ, 2013) para iOS
- permitir que o usuário escolha o predador, presa, plano de fundo e traçado durante o jogo
- analisar a possibilidade de utilizar um sintetizador de voz para reproduzir o significado das palavras ou frases nos planos customizados

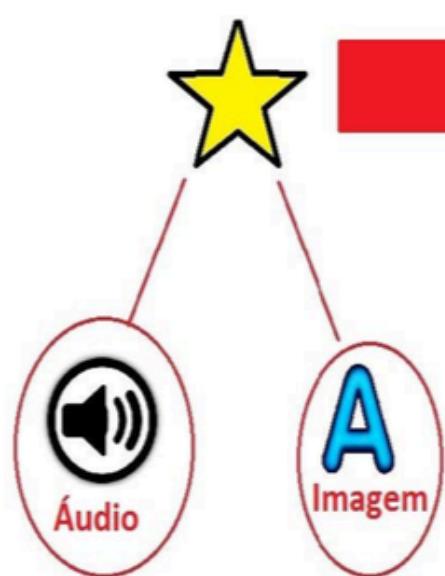
# Fundamentação Teórica

- Jogos 2D
- Sprites
- RGBA



- Tagarela
  - Grupo de Pesquisa em Computação Gráfica, Processamento de Imagens e Entretenimento Digital da FURB
- O Tagarela trabalha com basicamente 3 componentes:
  - Símbolos
  - Pranchas
  - Planos

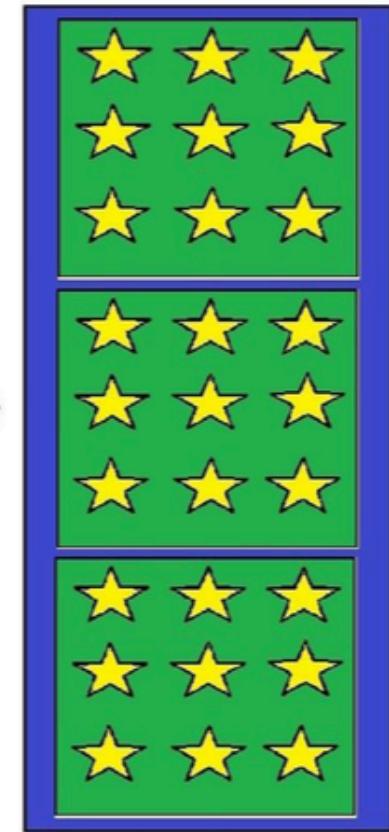
Símbolo



Prancha



Plano



Categorias: verbos, substantivos, emoção, decisão e adicionais

- Predador: imagem que segue o toque do usuário



- Presa: representa os pontos de interesse do jogo



# Trabalhos Correlatos

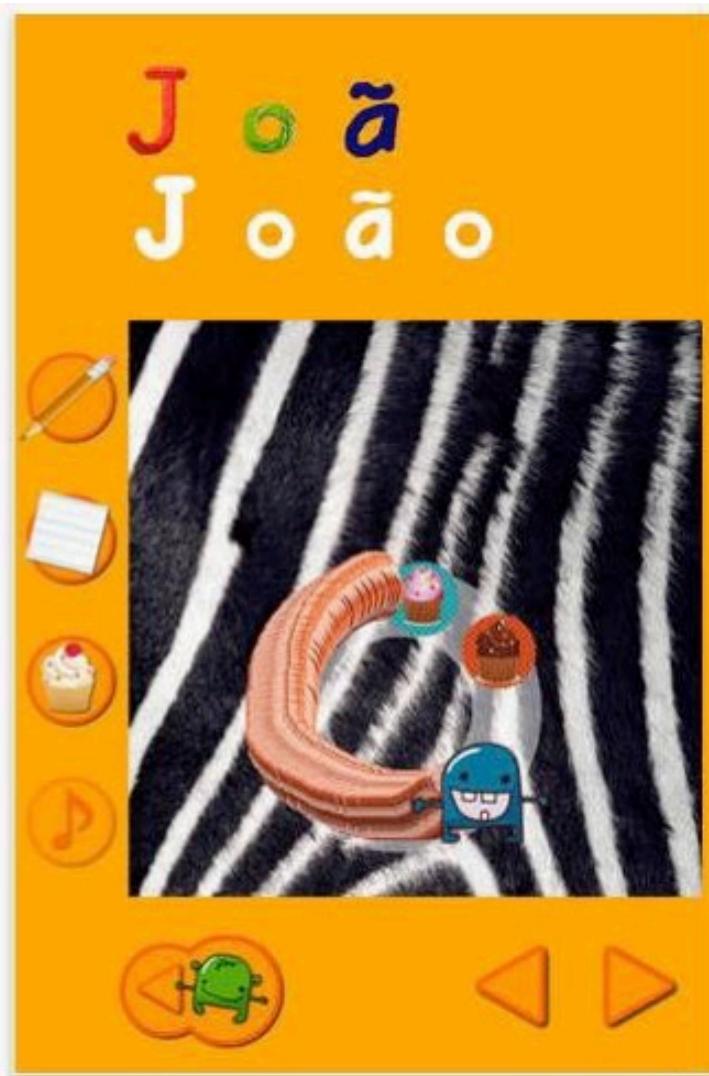
- Livox - Agora Eu Consigo Tecnologias de Inclusão Social Ltda

## Mais acessados

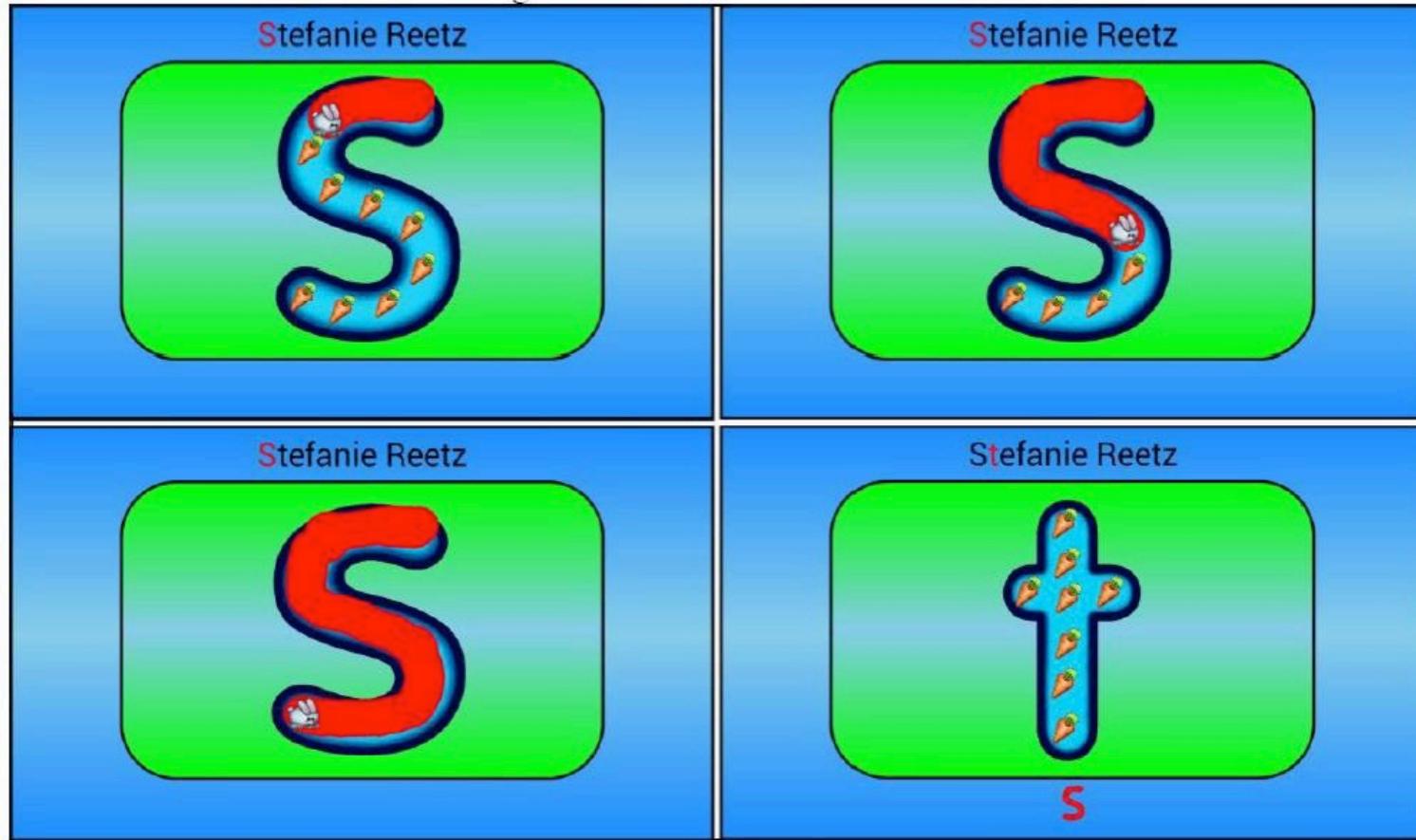
	<b>suco</b> Eu quero suco	6 
	<b>patati patata</b> patati papata	4 
	<b>Quem é quem é você?</b> quem é você?	3 
	<b>massinha</b> Eu quero brincar de massinha	2 
	<b>mingau</b> Eu quero mingau	2 
	<b>Xuxa</b> xuxa 11	2 
	<b>Pedidos...</b> pedidos	1 
	<b>esconde-esconde</b> Eu quero brincar de esconde esconde	1 

14h55

- Desenhe e Aprenda a Escrever - FizzBrain



- Jogo de Letras/Números Voltado para a Tecnologia Assistiva no Android

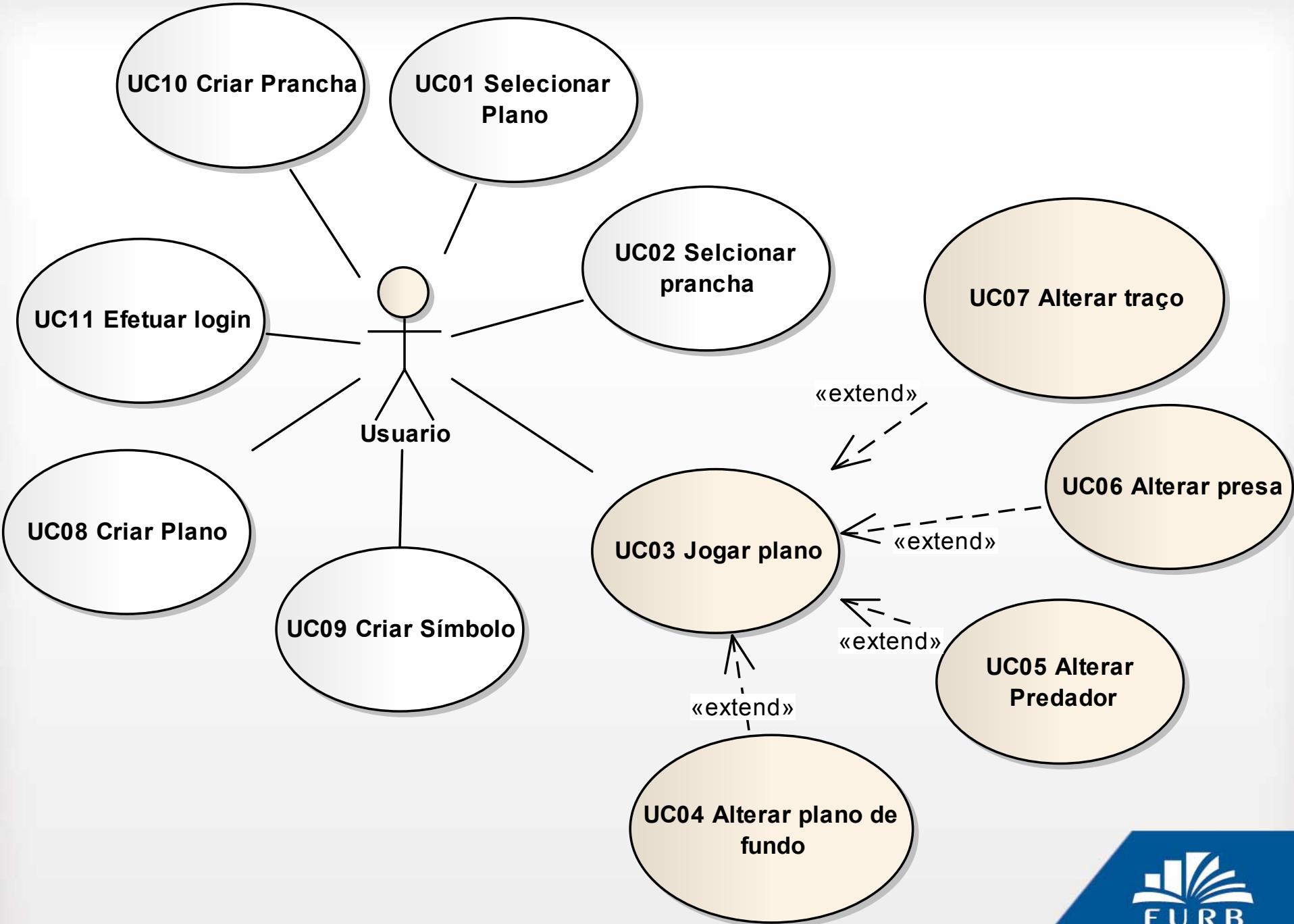


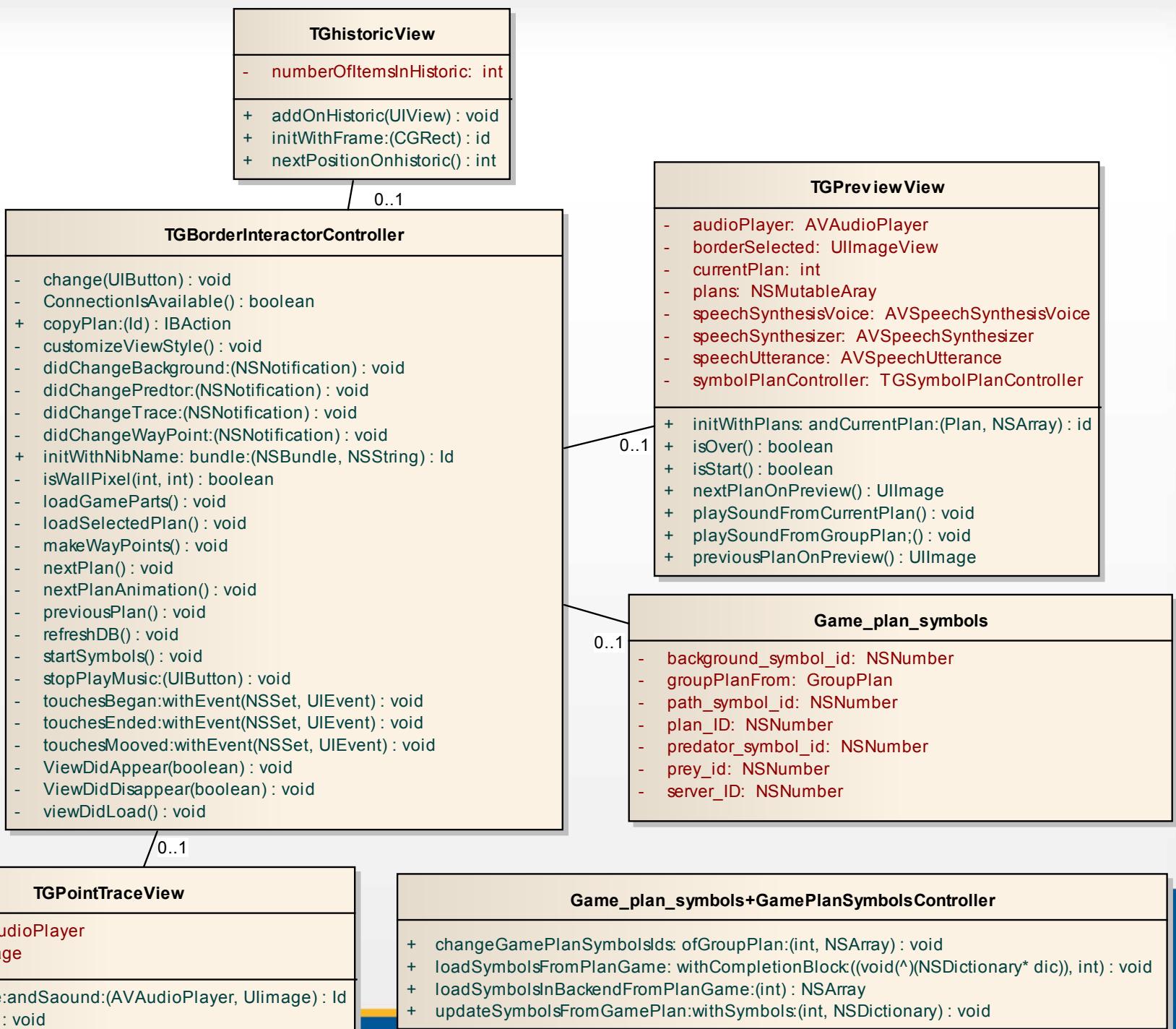
# **Requisitos Funcionais**

- permitir a escolha do símbolo de plano de fundo;
- ter melodia de fundo definido pelo símbolo do plano de fundo
- permitir a escolha de um símbolo para definir o traçado, predador e presa

# Requisitos Não Funcionais

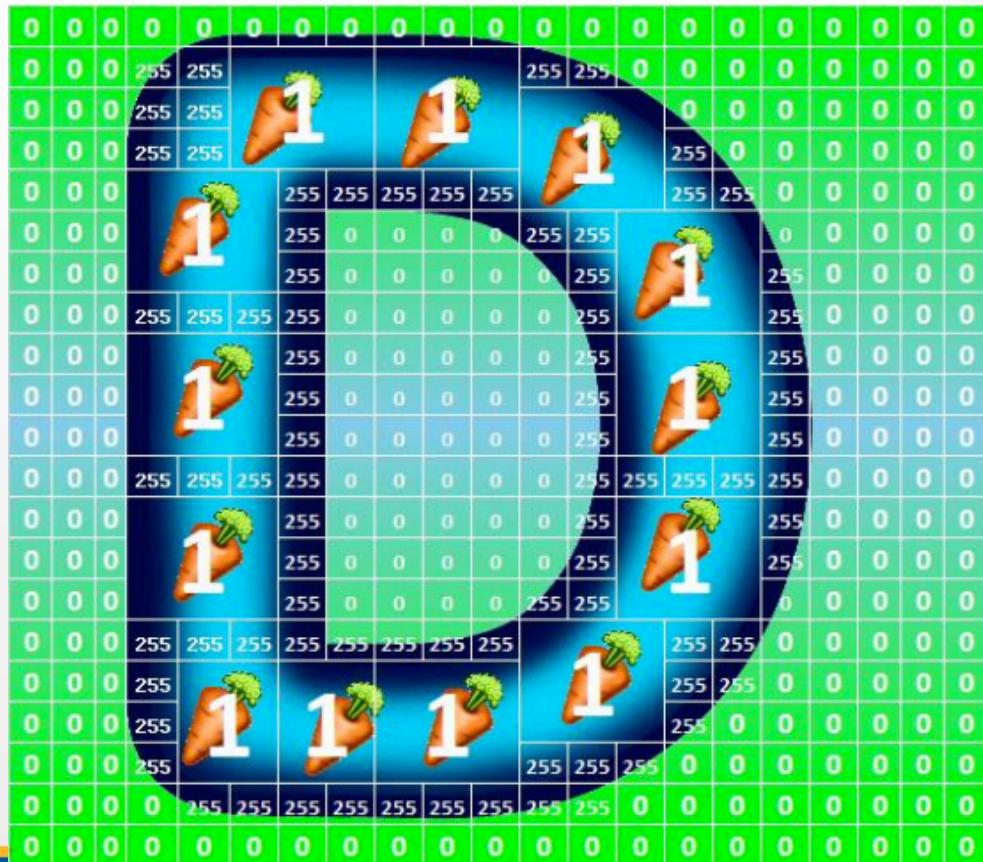
- ser implementado utilizando o ambiente Xcode 6.0
- utilizar a linguagem Objective-c
- permitir executar no sistema operacional iOS 7 ou superior
- continuar mantendo a sincronização dos planos com o servidor web.





# Implementação

- Classe TGborderInteractorController
    - O método makeWayPoints
    - Método touchesMoved
    - Método isWallPixel



# Implementação

- makeWayPoints

```
643 -(void)makeWayPoints{  
644     [_wayPoints removeAllObjects];  
645     for (int x = 0; x< imageView1.image.size.width; x++) {  
646         for (int y = 0; y< imageView1.image.size.height; y++){  
647             const UInt8* data = CFDataGetBytePtr(_pixelData);  
648             int pixelInfo = ((imageView1.image.size.width * y) + x) * 4;  
649             int alpha = data[pixelInfo + 3];  
650             if (alpha!=0 && alpha!=255) {  
651                 UIImageView *point2 = [[UIImageView alloc] initWithImage:  
652                                 _wayPointImageView.image];  
653                 point2.frame =CGRectMake(x/_scale-20, y/_scale-20, 40, 40);  
654                 [point2 setContentMode:UIViewContentModeScaleAspectFit];  
655                 point2.clipsToBounds = YES;  
656                 [_wayPoints addObject:point2];  
657             }  
658     }  
}
```

# Implementação

- isWallPixel

```
626 - (BOOL)isWallPixel: (int) x :(int) y {
627     if (x<0||x>imageView1.frame.size.width|| y<0||y>imageView1.frame.size.height)
628     {
629         return false;
630     }
631     const UInt8* data = CFDataGetBytePtr(_pixelData);
632     int pixelInfo = ((imageView1.image.size.width * (y*_scale)) + (x*_scale) ) *
633     4; // The image is png
634     int alpha = data[pixelInfo + 3];
635
636     if (alpha==0){
637         return NO;
638     }else{
639         return YES;
640     }
641 }
```

- **TouchesMoved**

```
589     if(CGRectIntersectsRect(fingerRect, _drawView.frame) && [self isWallPixel:  
590         location.x-258 :location.y-120]) {  
591         _predatorView.frame = CGRectMake(location.x-288 ,location.y-150, 60,60);  
592         _predatorView.alpha = 1;  
593         _predatorView.layer.zPosition=99;  
594  
595         UIImageView* Imageview = [[UIImageView alloc] initWithImage:_pointTrace.trace]  
596             ;  
597         Imageview.frame = CGRectMake(location.x-288 ,location.y-150, 60,60);  
598         [_backgroundAudio setVolume:0.2];  
599         [_pointTrace playSound];  
600         [self.drawView addSubview:Imageview];  
601  
602         NSMutableArray *ToDelete = [[NSMutableArray alloc] init];  
603         location = [[touches anyObject] locationInView:imageView1];  
604         fingerRect = CGRectMake(location.x-10, location.y-10, 20, 20);  
605         for(UIView *point in self.wayPoints){  
606             CGRect subviewFrame = point.frame;  
607             if(CGRectIntersectsRect(fingerRect, subviewFrame)){  
608                 [ToDelete addObject:point];  
609                 [point removeFromSuperview];  
610             }  
611         }  
612         [self.wayPoints removeObjectsInArray:ToDelete];  
613  
614         if([self.wayPoints count]==0){  
615             [self nextPlanAnimation];  
616         }  
617     }
```

# Operacionalidade da Implementação



Olá caneta,  
bem vindo ao Tagarela !

## Tutores

caneta - TUTOR

Tagarela

[www.inf.furb.br/gcg/tagarela](http://www.inf.furb.br/gcg/tagarela)

[Criar Símbolos](#) [Criar Prancha](#) [Vincular tutor](#) [Visualizar Histórico](#) [Visualizar Símbolos](#) [Visualizar Observações](#)



Olá caneta,  
bem vindo ao Tagarela !

Tutores

1

Cancelar

caneta - TUTOR

Tagarela

[www.inf.furb.br/gcg/tagarela](http://www.inf.furb.br/gcg/tagarela)

**Tagarela**  
Qual categoria voce quer criar o novo Plano?

- Prancha
- Jogo
- Quebra - Cabeça

[Cancelar](#)

2

# Imagen



[Criar Símbolos](#) [Criar Prancha](#) [Vincular tutor](#) [Visualizar Histórico](#) [Visualizar Símbolos](#) [Visualizar Observações](#)

[Cancelar](#)

[Voltar](#) Selecionar uma categoria

- Verbos
- Substantivos
- Emoção
- Decisão
- Adicionais

3

OK

[Cancelar](#)

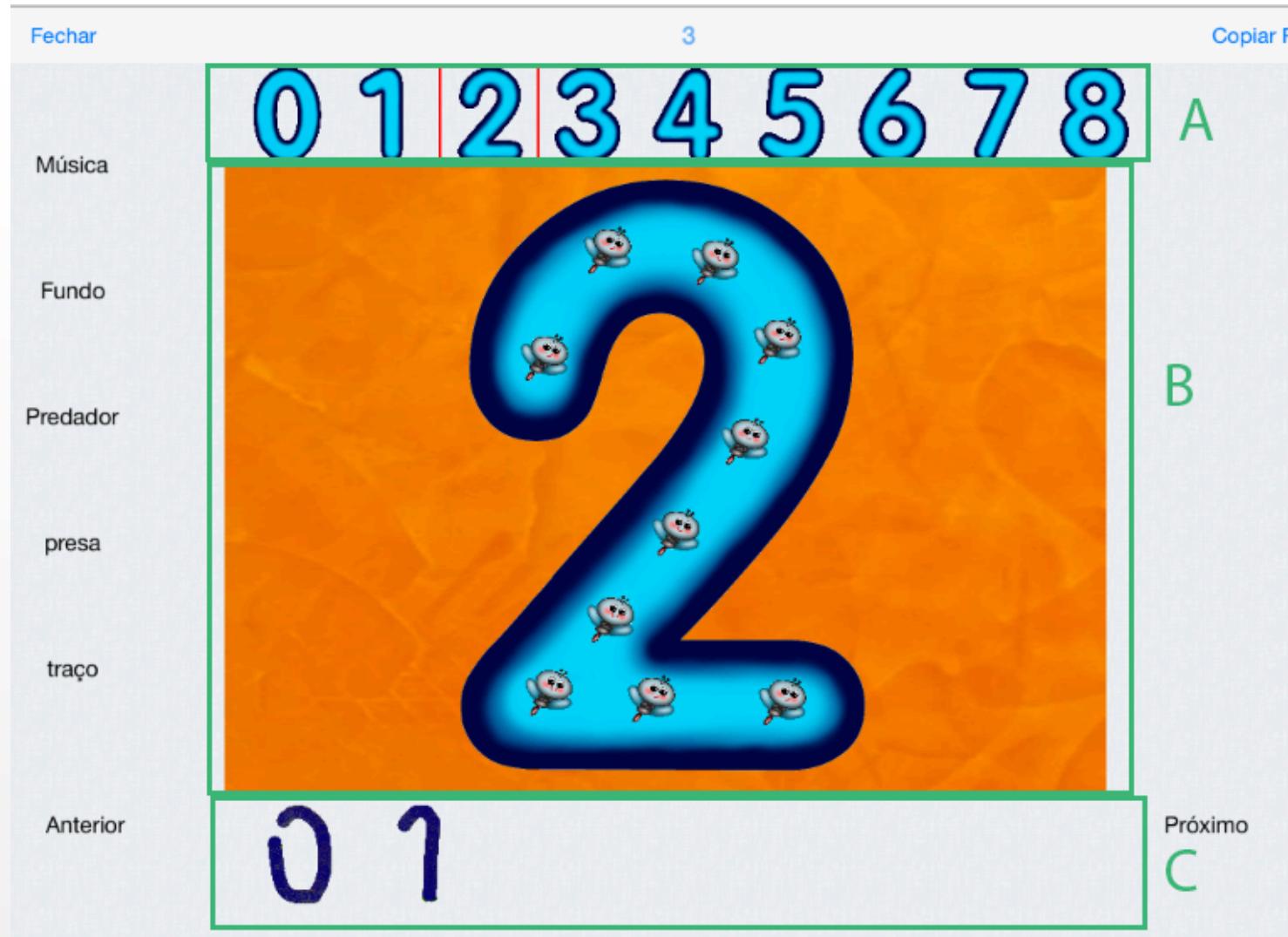
[Cancelar](#) Selecionar um grupo para o plano [Criar novo](#)

- Numeros
- Letras minusculas
- novo

4

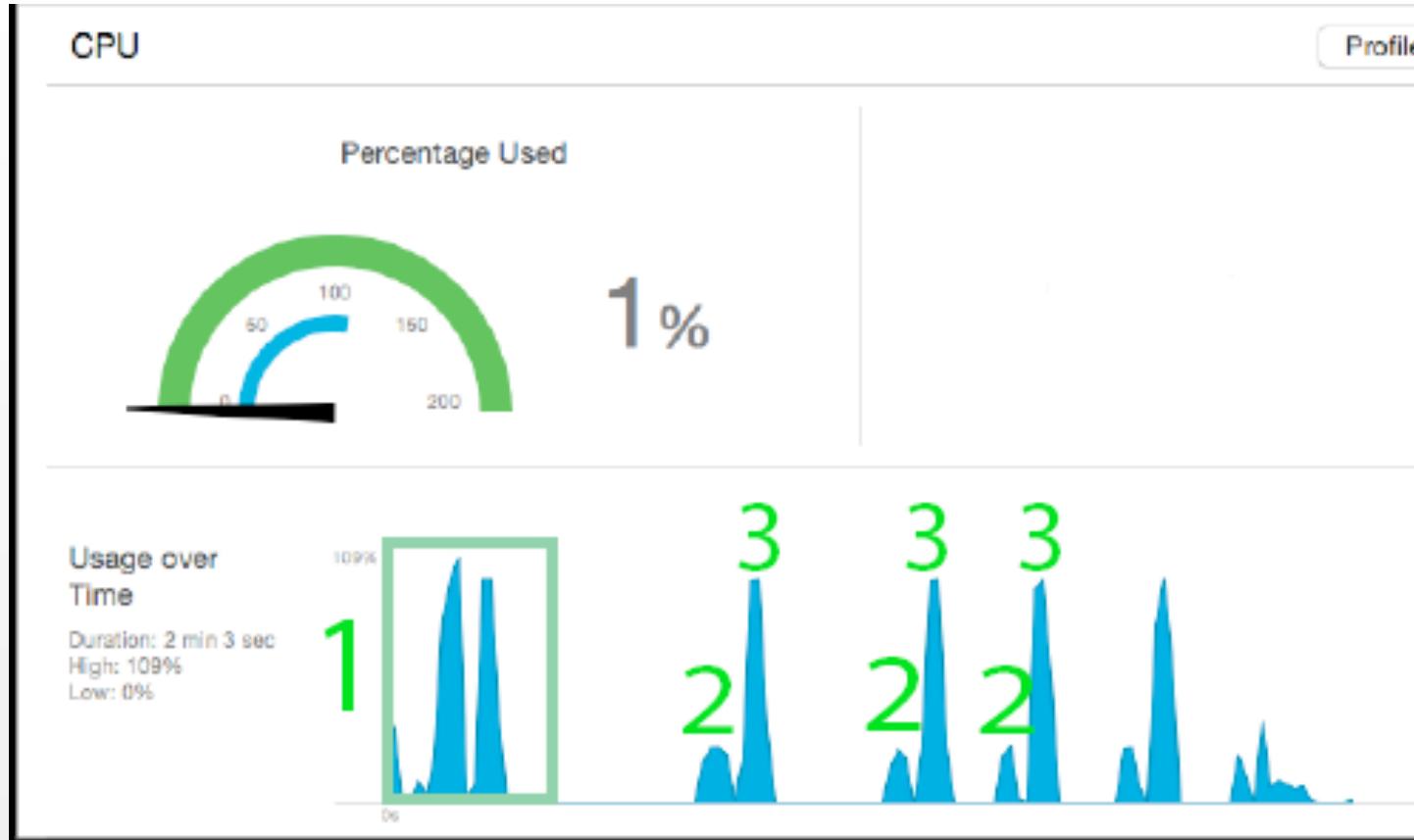


- Tela principal do jogo



# Resultados e Discussões

- Processamento no Ipad 2

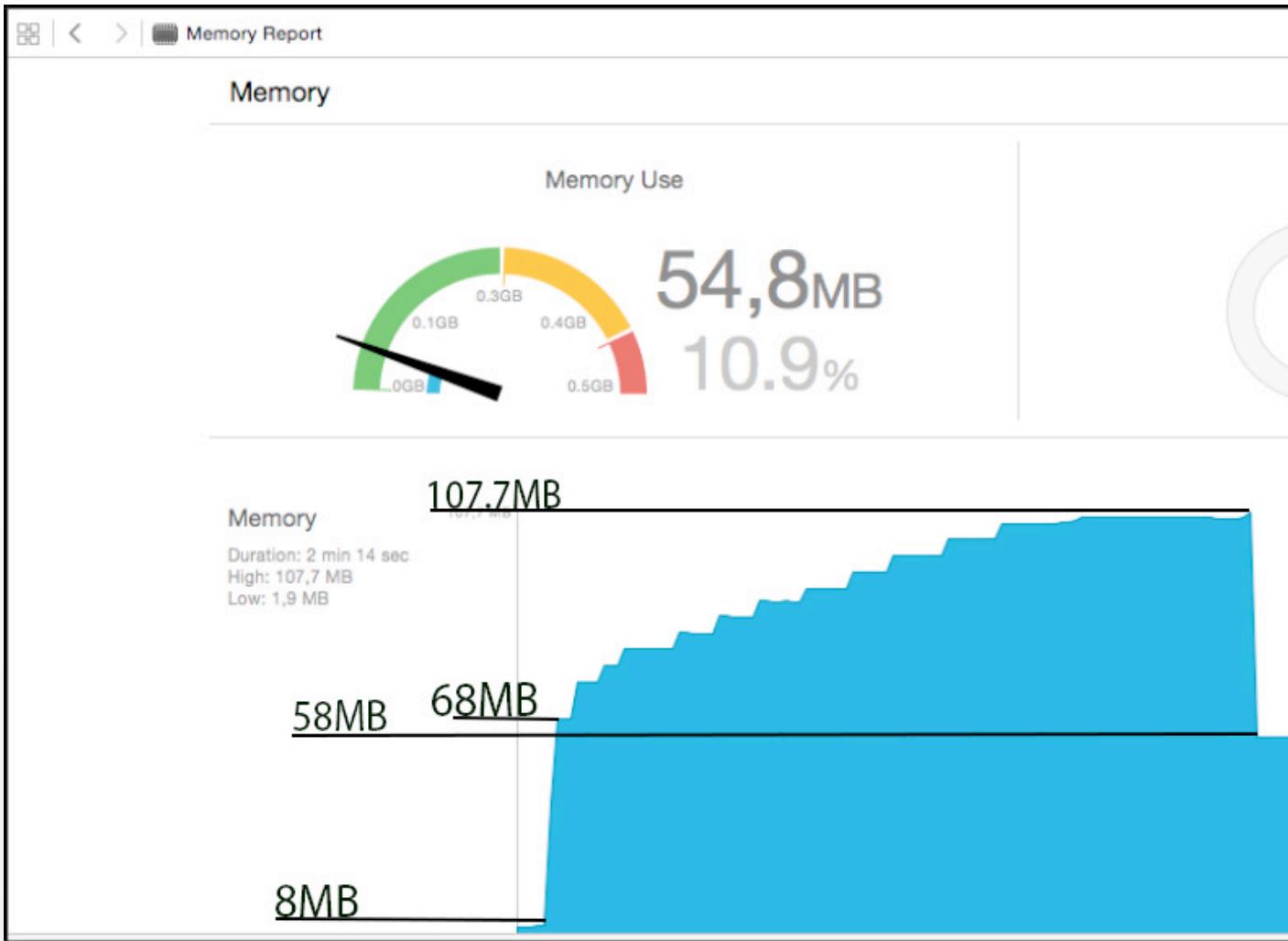


# Resultados e Discussões

- Comparação entre funções durante o jogo

Função	Nível de processamento	Tempo para a finalização (em segundos)
Posicionamento dos waypoints	>101%	2.503
Carregamento inicial	>27%	0.070
Durante o traçado	>26%	0.035

- Memória no iPad 2



# Resultados e Discussões

- Comparação entre consumo de memória e número de pranchas no plano

Quantidade de pranchas	Quantidade de memória alocada	Memória consumida após o plano concluído
1	31MB	32MB
5	44.6MB	61MB
10	68MB	107.7MB
20	108MB	181MB

# Conclusões e Sugestões

- Migração do módulo jogos do Android
- Sintetizador
- Símbolos para personalizar o jogo
- Lentidão entre transições de pranchas

# Conclusões e Sugestões

- Sugestões
  - adicionar ao histórico geral do projeto Tagarela.
  - pedir ao usuário os pontos de interesse onde ficarão as presas;
  - permitir selecionar a prancha de maneira intuitiva na pré-visualização;

# Demonstração