

# ***BEN 10 - RESGATE DO VÔ MAX***



***JOÃO VITOR MENDES - RA: 237881***

***TIAGO PERRUPATO ANTUNES - RA: 194058***

# INTRODUÇÃO

- **O JOGO É SINGLE PLAYER**
- **MODO AVENTURA**
- **USUÁRIO CONTROLA O BEN10 PELO MAPA**
- **O OBJETIVO DO JOGO É MATAR TODOS OS INIMIGOS E CHEGAR DO OUTRO LADO DA SALA**
- **ASSIM ELE CONSEGUE RESGATAR O VÔ MAX**



# MECÂNICA DO JOGO

- INTERFACE GRÁFICA



## ● INTERFACE GRÁFICA





# MECÂNICA DO JOGO

- **INTERFACE GRÁFICA**



## MOVIMENTAÇÃO



A tropa do Vilgax capturou seu avô Max, derrote todos os inimigos e chegue ao outro lado da sala para salvá-lo. Caso precise de ajuda preione o botão HELP

# MECÂNICA DO JOGO

## • MOVIMENTAÇÃO



# MECÂNICA DO JOGO

- **TROCA DE ALIENÍGENAS**





# MECÂNICA DO JOGO

- **TROCA DE ALIENÍGENAS**



Forward			Life	 <p>A tropa do Vilgax capturou seu avô Max, derrote todos os inimigos e chegue ao outro lado da sala para salvá-lo. Caso precise de ajuda precione o botão HELP</p>
Left...	Attack	Righ...	Four Arms	
			Flames	
Backward			Diamond	
				HELP

## TROCA DE ALIENÍGENAS



## TROCA DE ALIENÍGENAS



● **CHAMAS**





**● CHAMAS**



# MECÂNICA DO JOGO

## ● CHAMAS



Forward		Life	
Left... <input type="button" value="Attack"/> Right...		Four Arms	
Backward		Flames	
		Diamond	
<p>A tropa do Vilgax capturou seu avô Max, derrote todos os inimigos e chegue ao outro lado da sala para salvá-lo. Caso precise de ajuda precione o botão HELP</p>			
HELP			

# MECÂNICA DO JOGO

## • QUATRO BRAÇOS



# MECÂNICA DO JOGO

## • QUATRO BRAÇOS





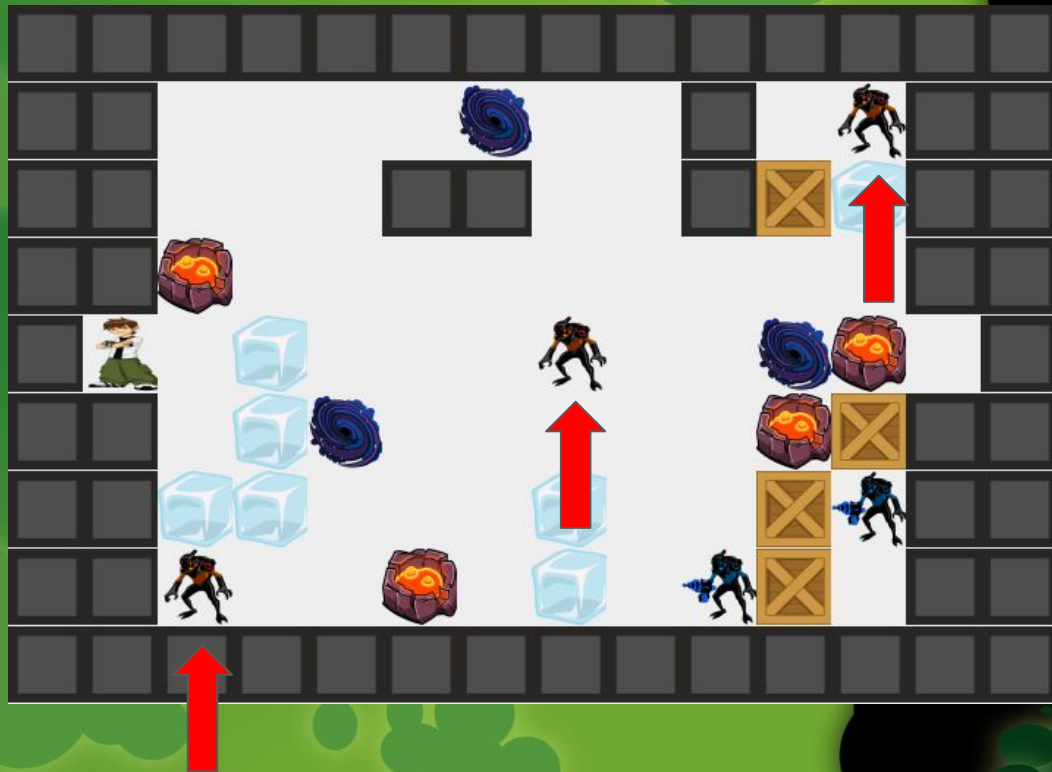
# MECÂNICA DO JOGO

## DIAMANTE



Forward		<b>Life</b> 	
Left...	Attack	Righ...	<b>Four Arms</b> 
Backward		<b>Flames</b> 	
		<b>Diamond</b> 	
<p>A tropa do Vilgax capturou seu avô Max, derrote todos os inimigos e chegue ao outro lado da sala para salvá-lo. Caso precise de ajuda pressione o botão HELP</p>			
HELP			

## ● INIMIGOS



# MECÂNICA DO JOGO

## • INIMIGOS



## ● INIMIGOS





# MECÂNICA DO JOGO

## ● GAMEOVER



Infelizmente você fracassou =( Não desista de sua jornada Ben Tennyson. Reinicie o jogo e tente novamente

HELP

[Grid of game elements: Ben Tennyson, enemies, obstacles, and power-ups]									
Forward		Life		 <p>Infelizmente você fracassou =( Não desista de sua jornada Ben Tennyson. Reinicie o jogo e tente novamente</p>					
Left...	Attack	Right...	Four Arms						
Backward			Flames						
			Diamond						
				HELP					

# MECÂNICA DO JOGO

● VITÓRIA



# MECÂNICA DO JOGO

● VITÓRIA



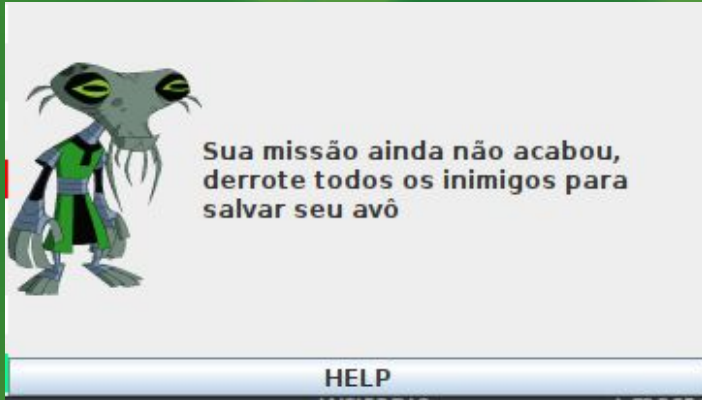
Parabéns Ben, seu avô ficará muito orgulhoso quando salvarmos ele

HELP

Forward		Life	
Left...	Attack	Four Arms	
Backward		Flames	
		Diamond	
			HELP

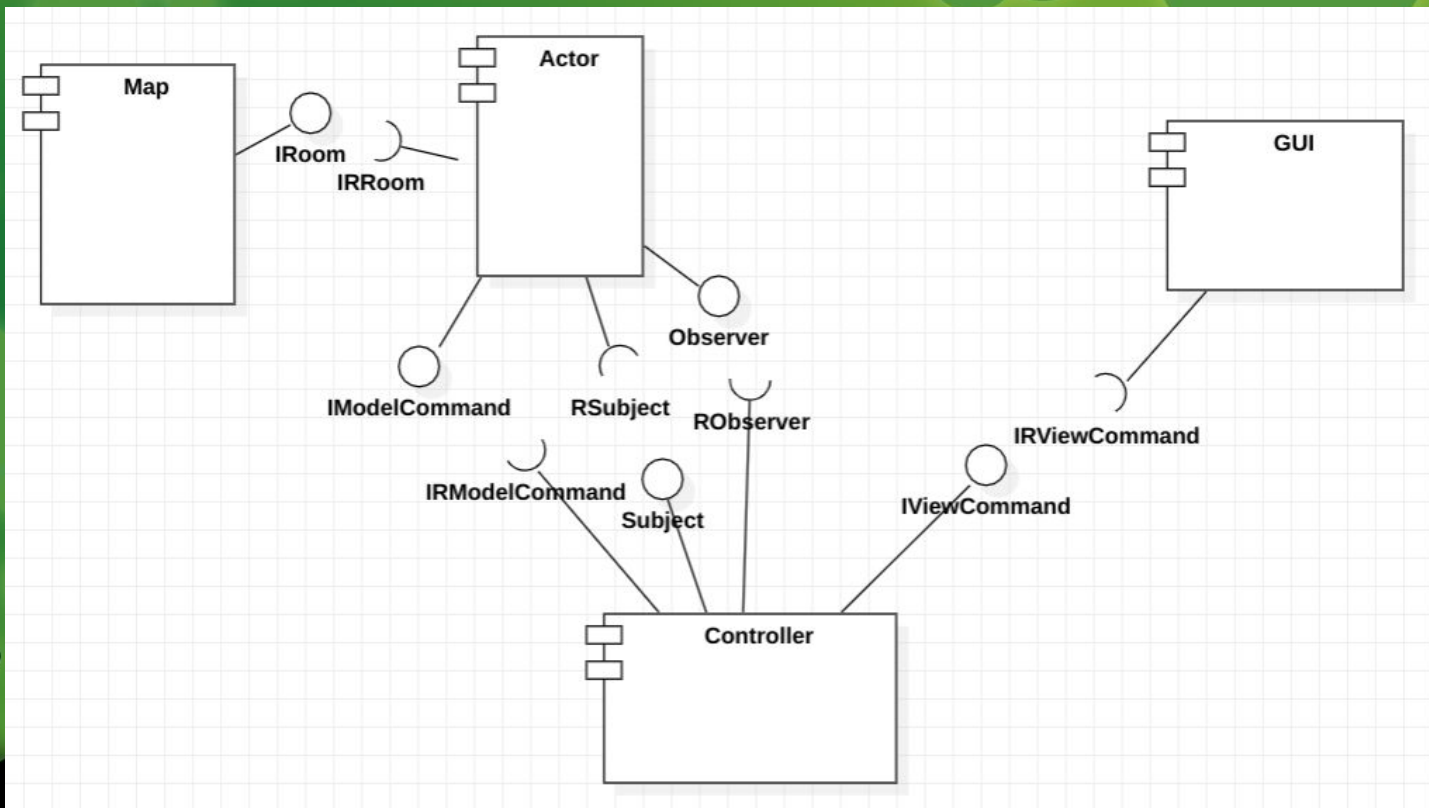
# MECÂNICA DO JOGO

## VITÓRIA





# ARQUITETURA GERAL (COMPONENTES)



# DESTAQUES

- **CLOCK**

```
public Clock(int rate) {  
    this.observers = new ArrayList<Observer>();  
    this.rate=rate;  
    this.timer=new Timer();  
}  
public void start() {  
    this.timer.scheduleAtFixedRate(new TimerTask() {  
        public void run() {notifyObservers();}  
    }, this.rate, this.rate);  
}
```

# DESTAQUES

- **TROCAS DE HERÓIS CONTROLADOS**

```
// verifica se o comando foi de mudar de Personagem
String changeHero[] = {"B10", "FA", "FL", "DI"};
for(String i: changeHero)
    if(i.equals(command))
        if (this.verifyChangeHero(command))
            //retorna o novo personagem selecionado
            return this.changeHero(command);
```

# DESTAQUES





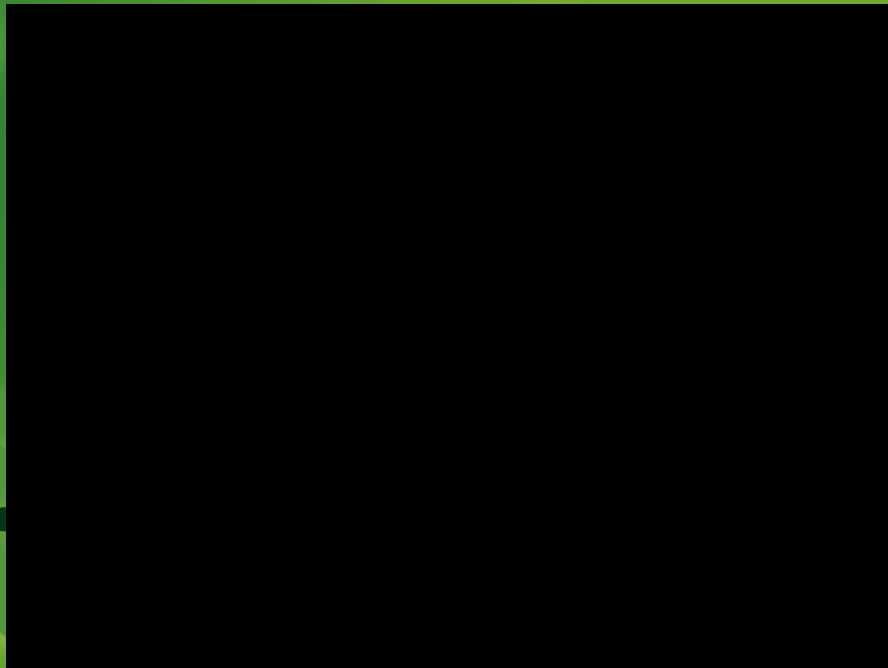
# DESTAQUES

- ***INIMIGOS INTELIGENTES QUE PERSEGUEM O JOGADOR***

```
public void update() {  
    Hero hero = searchHero();
```

```
if(verifyMovement("forward")) {  
    move("forward");
```

***FAZ A MOVIMENTAÇÃO A PARTIR  
DA POSIÇÃO DO HERÓI***



# DESTAQUES POO

## ● HERANÇA NOS HERÓIS DO JOGO

```
public Actor(int posRow, int posColumn, String typeActor) {
    this.posRow = posRow;
    this.posColumn = posColumn;
    this.typeActor = typeActor;
    this.alive = true;
}

public class Hero extends Actor implements IHero {

    private static int life = 10;
    private int stamina = 10;
    // ...
    private static Hero heros[];
    private boolean isTransformed=false;
    private String aim;

    public Hero(int posRow, int posColumn, String typeActor) {

        super(posRow, posColumn, typeActor);
        this.aim = "right";
        // ...
    }
}
```

# DESTAQUES POO

## ● HERANÇA NOS HERÓIS DO JOGO

```
public Ben10(int posRow, int posColumn, String typeActor) {  
    super(posRow, posColumn, typeActor);  
    this.setTransformed(true);  
}
```

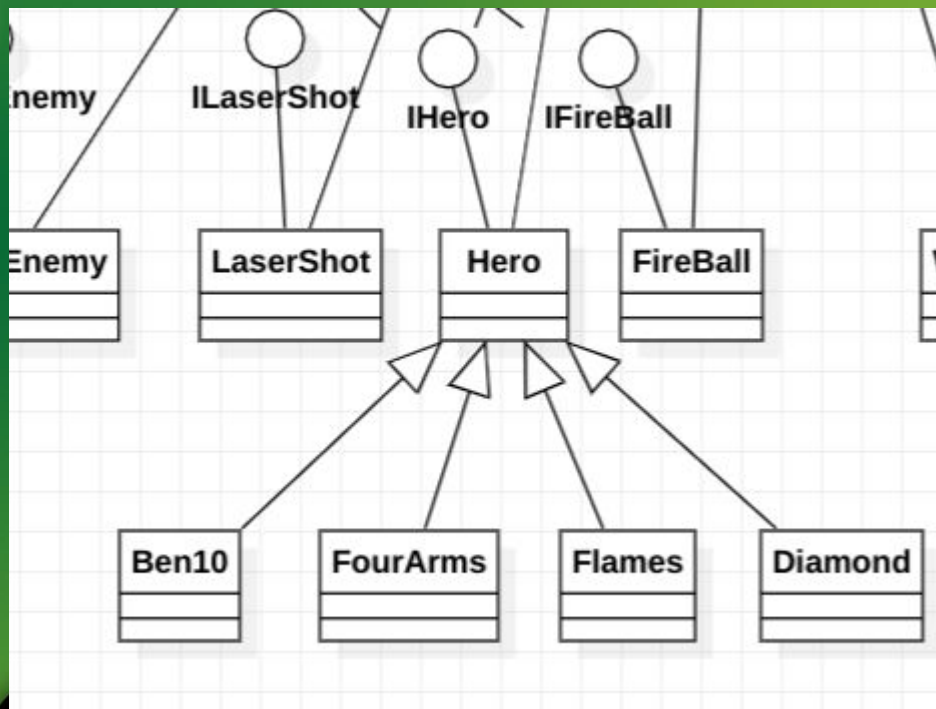
```
public FourArms(int posRow, int posColumn, String typeActor) {  
    super(posRow, posColumn, typeActor);  
}
```

```
public Flames(int posRow, int posColumn, String typeActor) {  
    super(posRow, posColumn, typeActor);  
}
```

```
public Diamond(int posRow, int posColumn, String typeActor) {  
    super(posRow, posColumn, typeActor);  
}
```

# DESTAQUES POO

## HERANÇA NOS HERÓIS DO JOGO





# DESTAQUES POO

- **POLIMORFISMO**
  - **ATORES DENTRO DE CADA CÉLULA**

```
public class Cell implements ICell {  
    private ArrayList<IActor> actors;
```

```
Hero ben10 = new Ben10(posRow, posColumn, "B10");  
Hero fourArms = new FourArms(posRow, posColumn, "FA");  
Hero flames = new Flames(posRow, posColumn, "FL");  
Hero diamond = new Diamond(posRow, posColumn, "DI");
```

# DESTAQUES POO

## ● POLIMORFISMO

### ATORES DENTRO DE UMA CÉLULA

```
IActor obj;  
  
switch (actorType) {  
case "B10":  
    this.createHeros(posRow, posColumn);  
    return;  
  
case "NE":  
    obj = new NearEnemy(posRow, posColumn, actorType);  
    break;  
  
case "DE":  
    obj = new DistantEnemy(posRow, posColumn, actorType);  
    break;  
  
case "LP":  
    obj = new LavaPool(posRow, posColumn, actorType);  
    break;  
  
case "BH":  
    obj = new BlackHole(posRow, posColumn, actorType);  
    break;
```

### VETOR DE HEROS

```
private static Hero heros[];
```

# DESTAQUES DE PATTERNS

- **STRATEGY**

```
public interface AttackStrategy {  
    public void attack();  
}
```

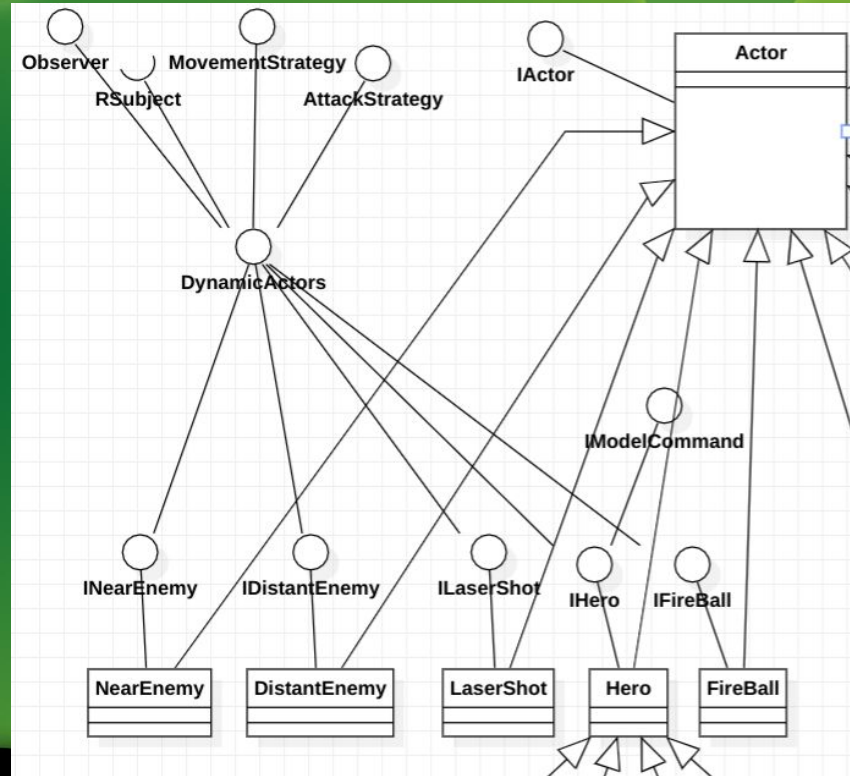
```
public interface MovementStrategy {  
    public void move(String direction);  
    public boolean verifyMovement(String direction);  
}
```

```
public interface DynamicActor extends Observer, RSubject, AttackStrategy, MovementStrategy {
```

```
    public interface IHero extends DynamicActor, IModelCommand{
```

# DESTAQUES DE PATTERNS

## ● STRATEGY





# DESTAQUES DE PATTERNS

- **OBSERVER**

```
public interface Subject {  
  
    public void register(Observer obj);  
    public void remove(Observer obj);  
    public void notifyObservers();  
}
```

```
public interface Observer {  
  
    public void update();  
    public void setSubject(Subject obj);  
    public Subject getSubject();  
}
```

# DESTAQUES DE PATTERNS

## ● OBSERVER

