# 2D GAMES AS CYBER-PHYSICAL SYSTEMS

VIDYA NARAYANAN

# MOTIVATION

- Physics based games with discrete game controllers are hybrid systems
- Impossible games are no fun
    - Game designs need “correctness” or playability guarantees
- Trivial games are no fun
    - Games must allow “winning strategies” and be non-trivial
- Inherently adversarial — hybrid games
- Opportunity to study  various elements of CPSs  and dL

# RELATED WORK

- Adelhart and Kargov : Mario game solver

  Adelhardt, Kim, and Nedyalko Kargov. "Mario game solver." IT University of Copenhagen (2012).

  - Graph analysis over axioms

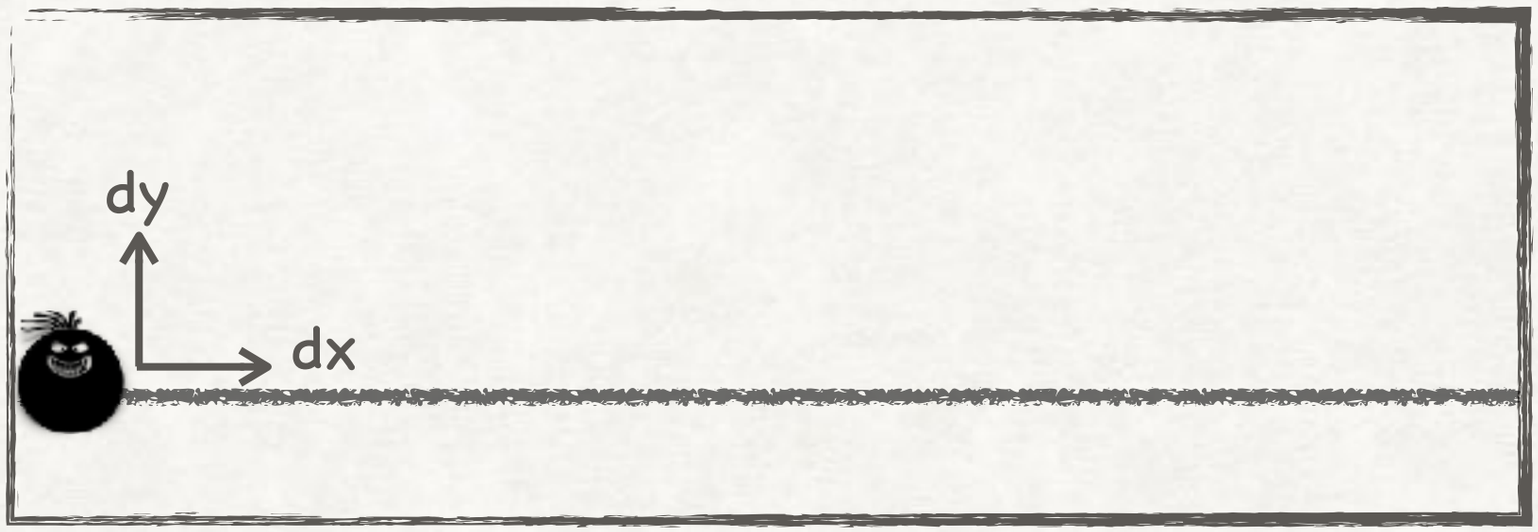
- Aloupis et. al 2015 : Mario is hard

  Aloupis, Greg, et al. "Classic Nintendo games are (computationally) hard." Theoretical Computer Science 586 (2015): 135-160.

- Demaine et.al 2016: Mario is easy

  Demaine, Erik D., Giovanni Viglietta, and Aaron Williams. "Super Mario Bros. is harder/easier than we thought." (2016).

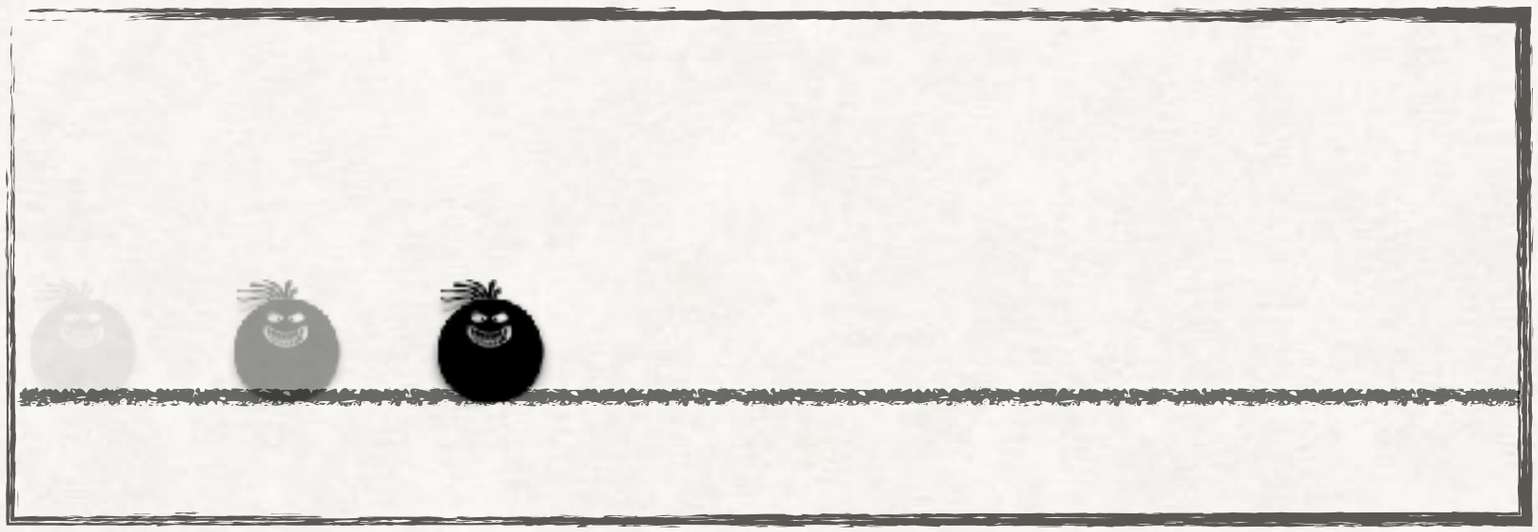
# A (VERY) SIMPLE GAME




PLAYER DYNAMICS:

$x' = vdx, y' = vdy$

# A (VERY) SIMPLE GAME



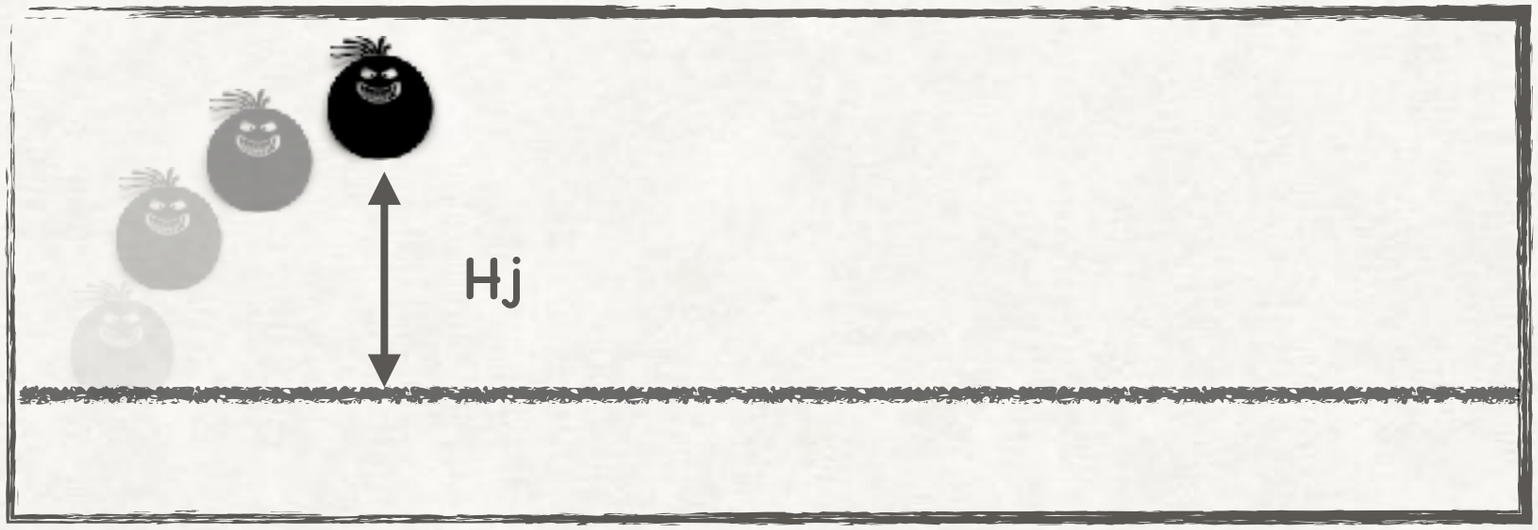
PLAYER DYNAMICS:

$x' = vdx, y' = vdy$

PLAYER CONTROL:

{v:=v+1; ++ v:=v-1; ++ ?true}

# A (VERY) SIMPLE GAME




PLAYER DYNAMICS:

$x' = vdx,\ y' = j + vdy,\ j'=-g$

EVOLUTION CONSTRAINTS

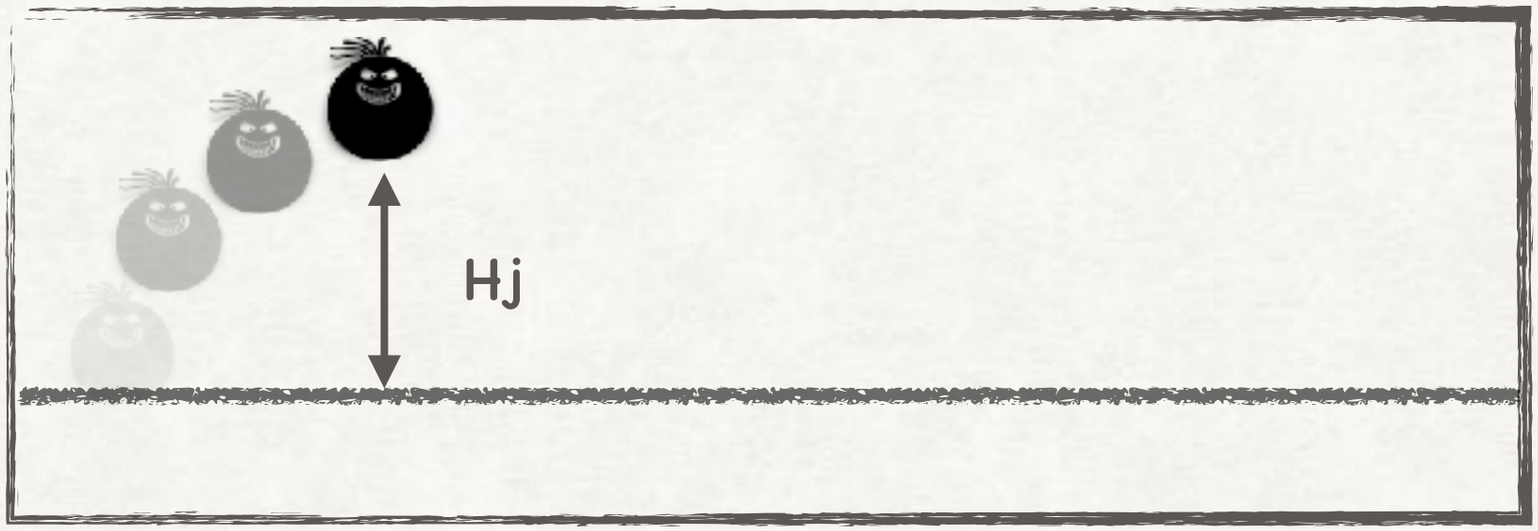
$y >= ground$

PLAYER CONTROL:

{v:=v+1; ++ v:=v-1; ++ ?true}

{{j:=J; g:=G} ++ {j:=0;g:=0} ++ ?true}

# A (VERY) SIMPLE GAME




PLAYER DYNAMICS:

$x' = vdx,\ y' = j + vdy,\ j' = -g$

EVOLUTION CONSTRAINTS

$y >= ground$

PLAYER CONTROL:

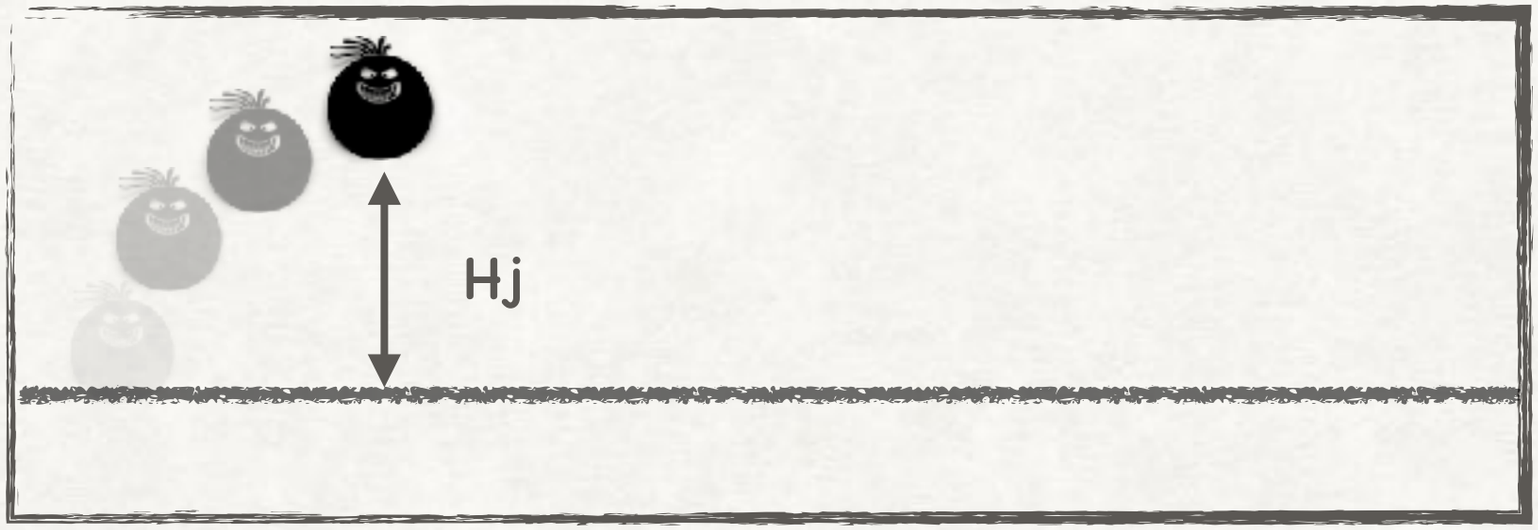
{v:=v+1; ++ v:=v-1; ++ ?true}

{{j:=J; g:=G} ++ {j:=0;g:=0} ++ ?true}

WORLD FIX-UP CONTROL:

if ( y <= ground ) { j:= 0; g:=0 }

# A (VERY) SIMPLE GAME




PLAYER DYNAMICS:

$x' = vdx,\ y' = j + vdy,\ j'=-g,\ t'=1$

EVOLUTION CONSTRAINTS

$y >= ground \quad t <= T$

PLAYER CONTROL:

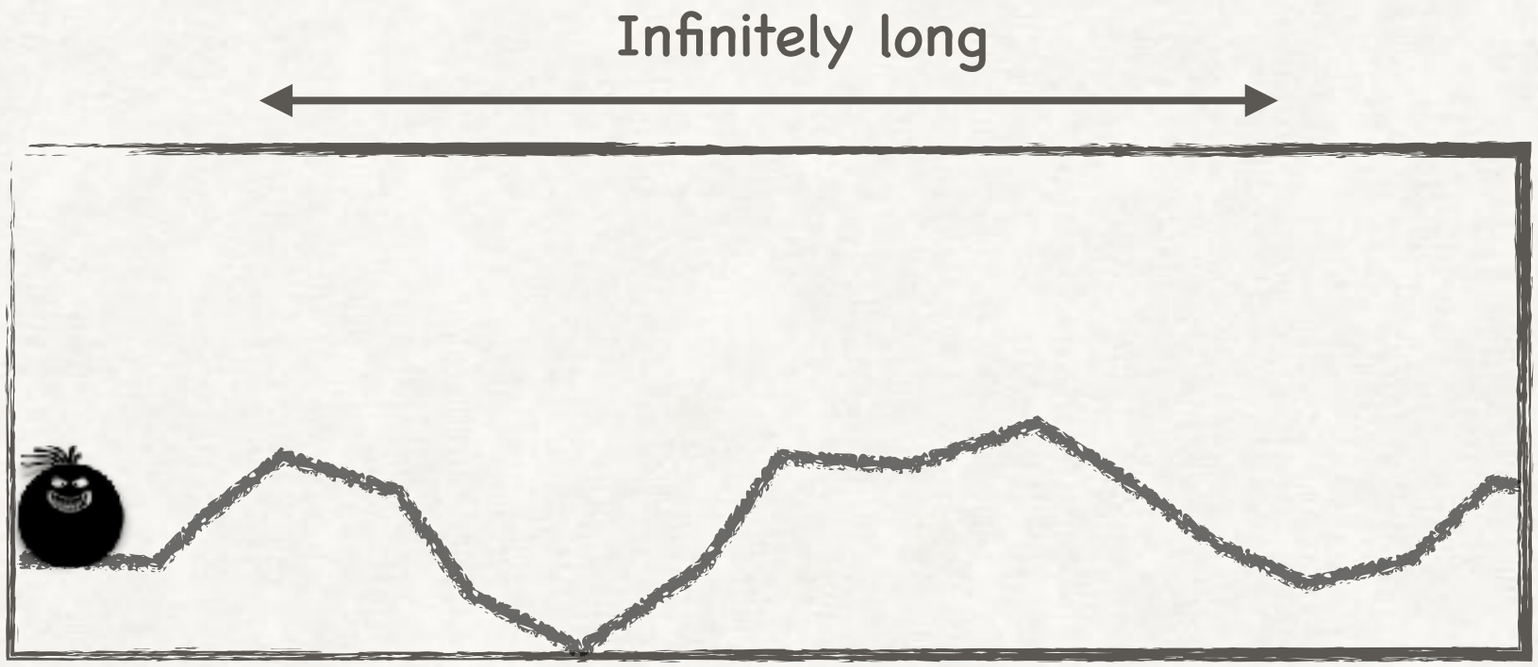
{v:=v+1; ++ v:=v-1; ++ ?true}

{{j:=J; g:=G} ++ {j:=0;g:=0} ++ ?true}

WORLD FIX-UP CONTROL:

if ( y <= ground ) { j:= 0; g:=0 }

# LEVEL DESIGN




VIRTUAL PLAYER/WORLD DYNAMICS:

$x' = vdx, l' = vdy$

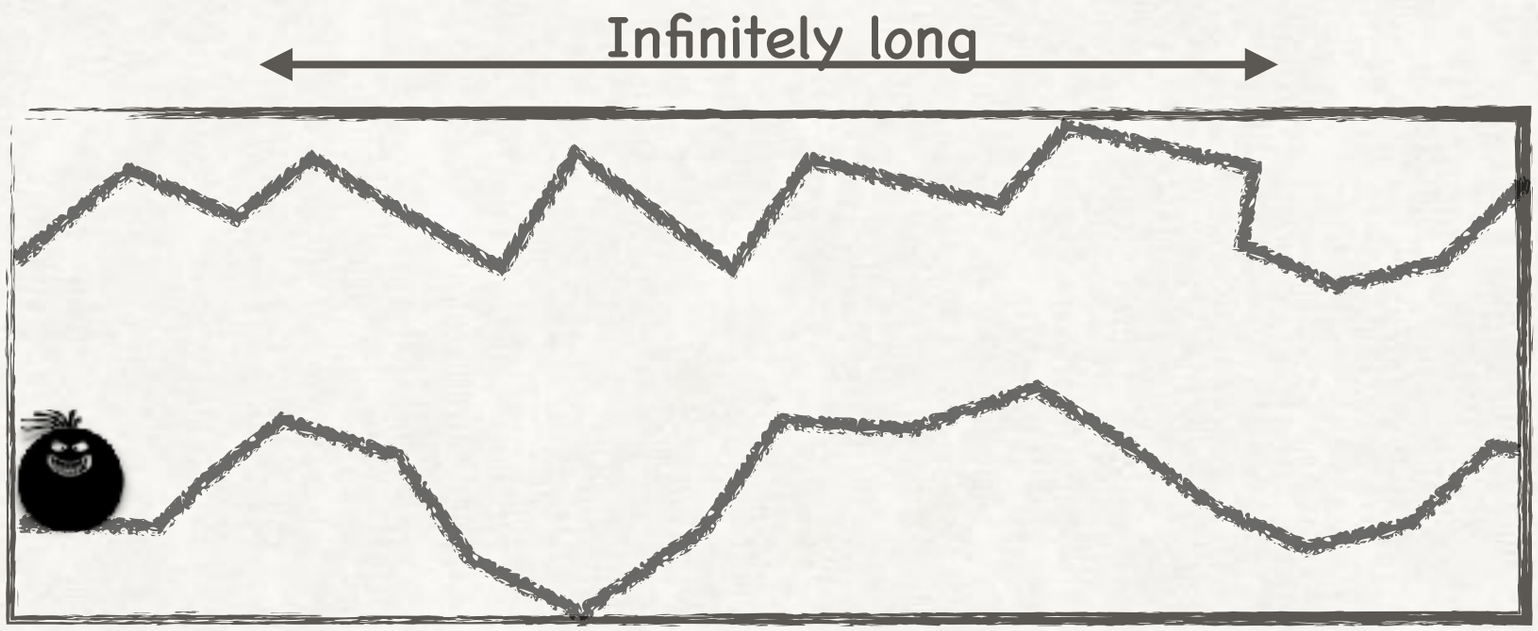
WORLD CONTROL:

dx:=*; dy:=*

?dx^2+dy^2=1;

**? dx > 0**

# LEVEL DESIGN




VIRTUAL PLAYER/WORLD DYNAMICS:

$ux' = v\ udx,\ h' = v\ udy$

WORLD CONTROL:

dx:=*; dy:=*

?dx^2+dy^2=1;

**? dx > 0**

# LEVEL DESIGN

PLAYER CONTROLLER

if( t>=T) { t:=0, …}

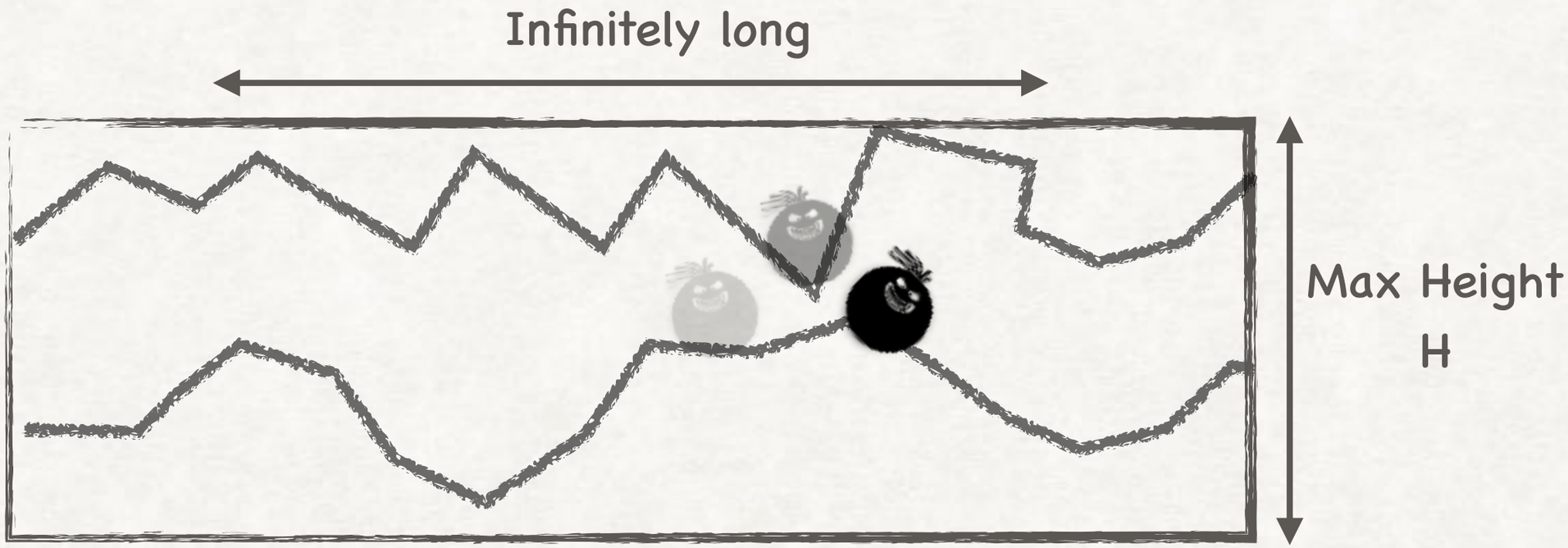
DYNAMICS

….

WORLD CONTROLLER

….

# SAFETY AND PLAYABILITY




PLAYER CONTROL:

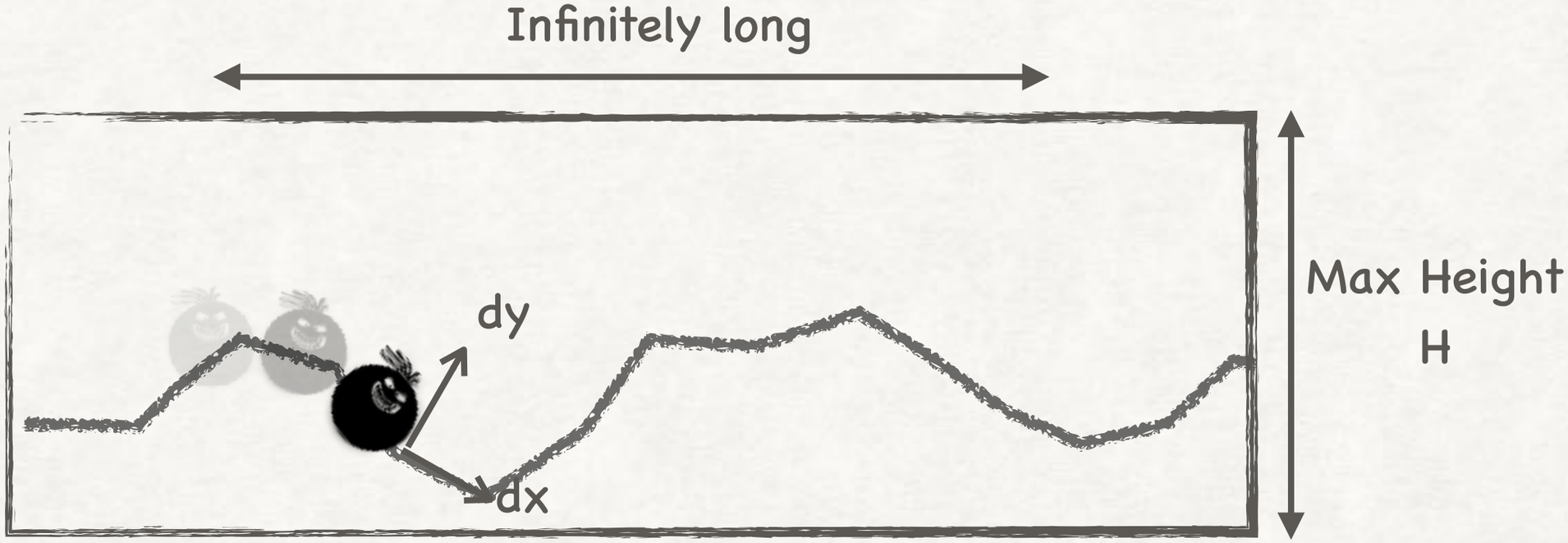
if( t >= T & y = l) { j….}

WORLD CONTROL:

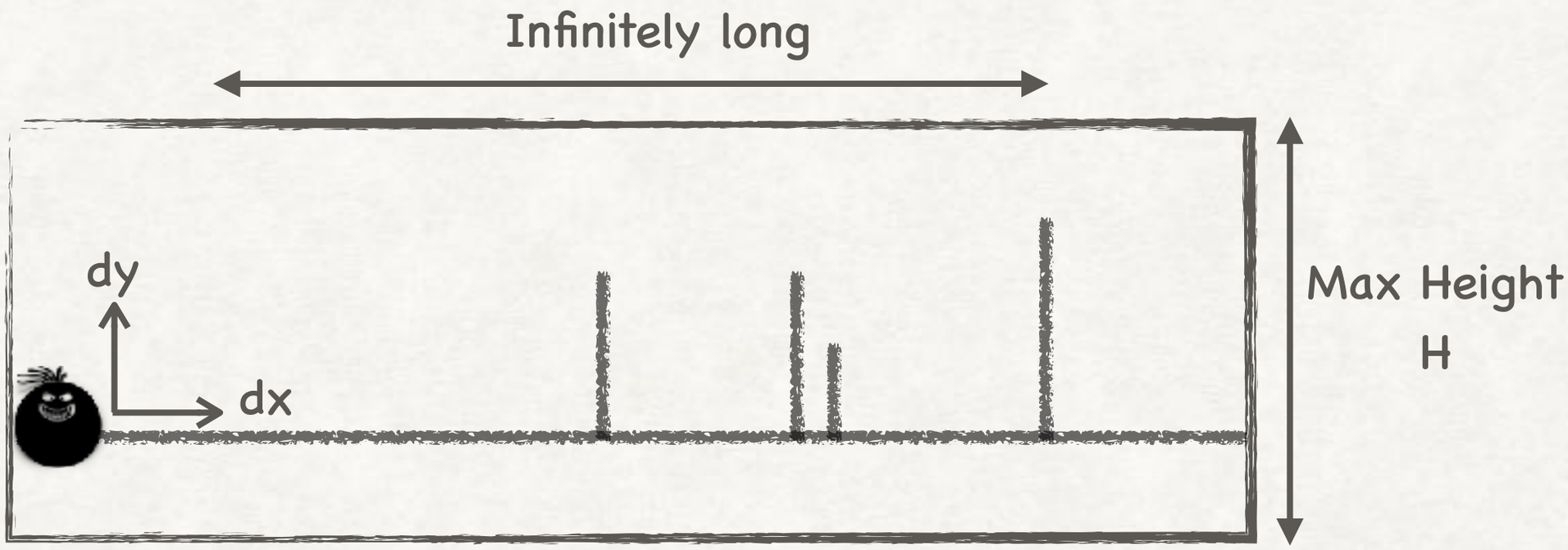
dx:=*; dy:=*

?l+dy*T + clearance < H

# LEVEL MODELING



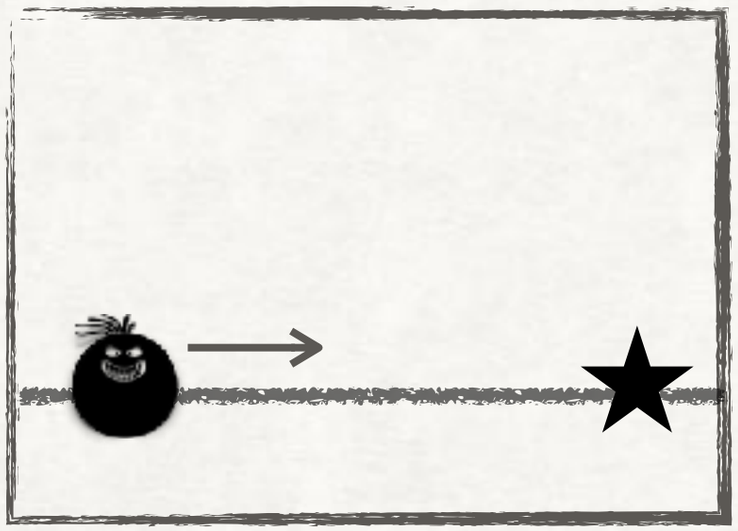
# SAFETY AND *PLAYABILITY*



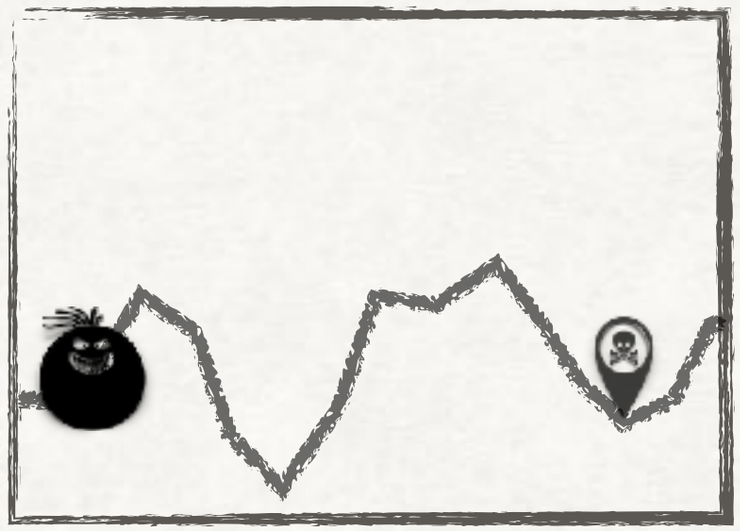

OBSTACLES:

hb + clearance < H

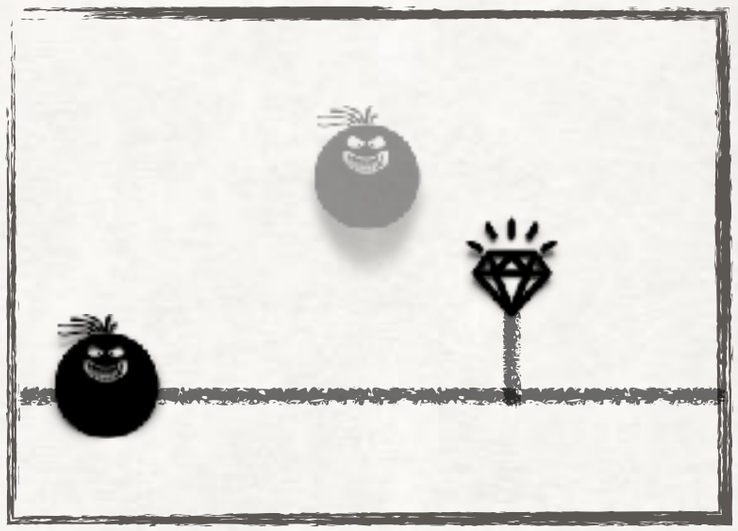
# GAME CHALLENGES



PROGRESS

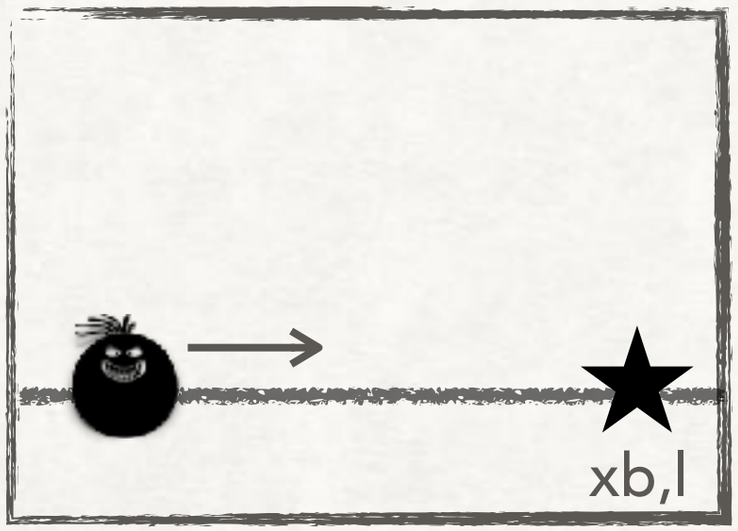


AVOID



ATTACK

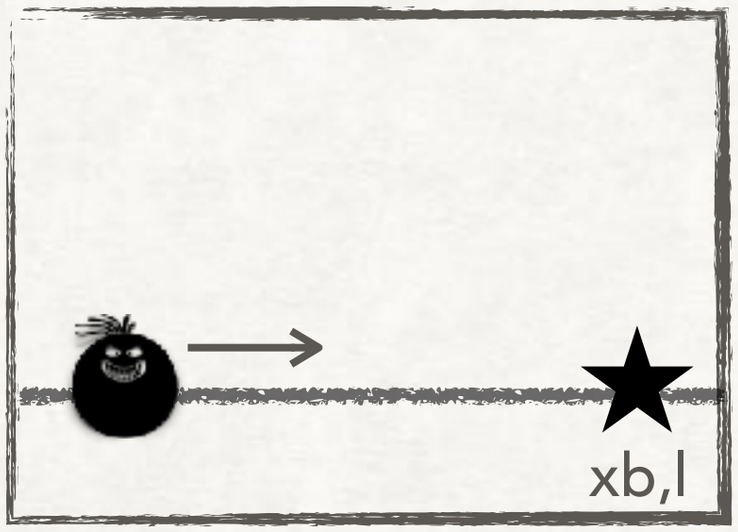
# GAME CHALLENGES




```
{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }

{ WORLD CONTROLLER}
```

PROGRESS

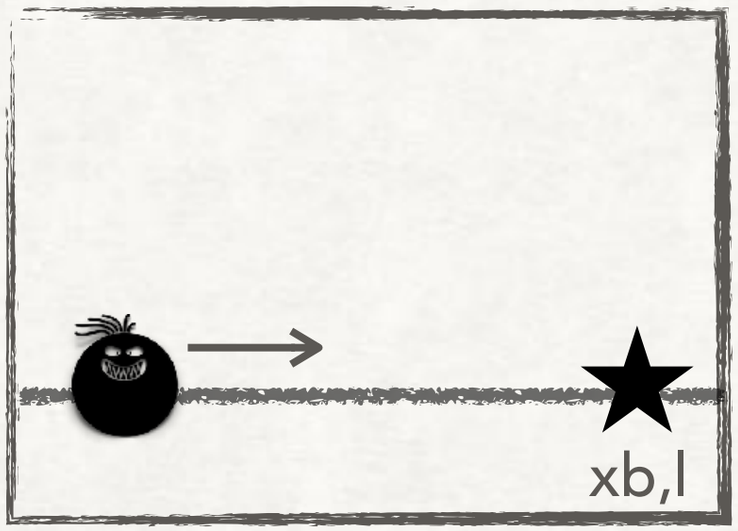
# GAME CHALLENGES




PROGRESS

```
INITIAL CONDITIONS ->
<{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }

{ WORLD CONTROLLER}
}*
> PROGRESS?
```

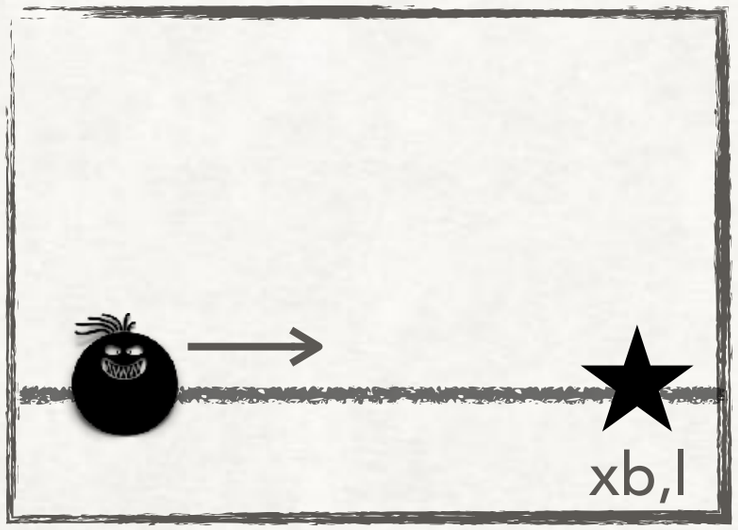
# GAME CHALLENGES




PROGRESS

```
INITIAL CONDITIONS ->
<{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }

{ WORLD CONTROLLER}
> PROGRESS?
```

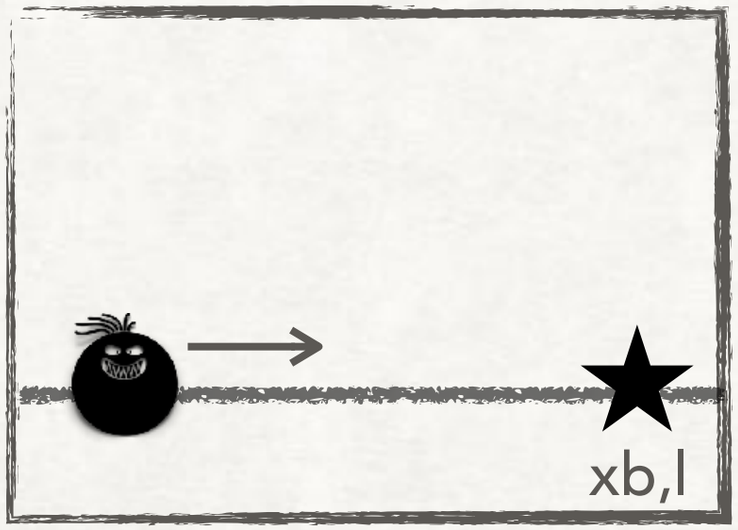
# GAME CHALLENGES




PROGRESS

```
INITIAL CONDITIONS ->
<{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }

{ WORLD CONTROLLER}^@
> PROGRESS?
```

# GAME CHALLENGES

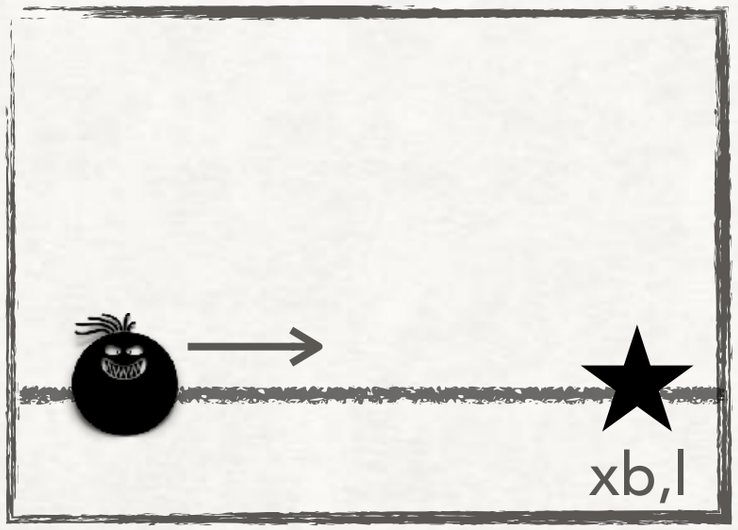



PROGRESS

```
INITIAL CONDITIONS ->
<{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }^@

{ WORLD CONTROLLER}^@
> PROGRESS?
```

ENVIRONMENT IS AN ADVERSARY BY DESIGN

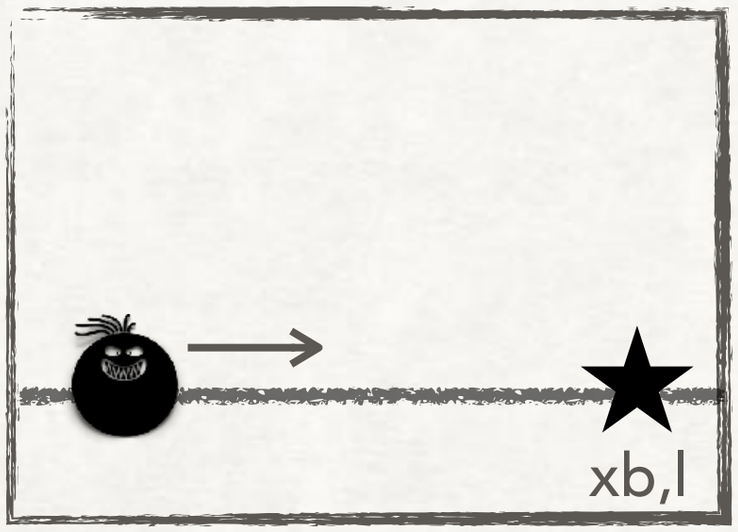
# GAME CHALLENGES




PROGRESS

```
INITIAL CONDITIONS ->
<{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }^@

{ WORLD CONTROLLER}^@
> PROGRESS?
```

# GAME CHALLENGES

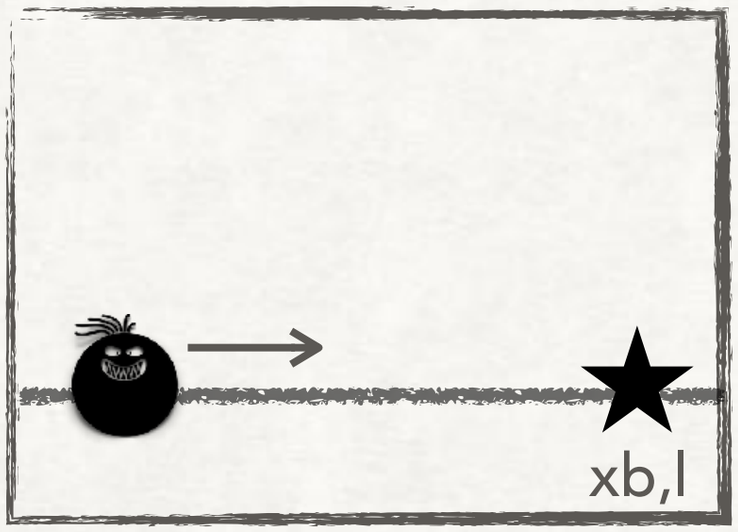



PROGRESS

```
INITIAL CONDITIONS ->
<{
{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }^@   ?(T > 0)^@

{ WORLD CONTROLLER}^@ ?(DX > 0)^@
}*
> PROGRESS?
```

ENVIRONMENT IS REASONABLE

# GAME CHALLENGES



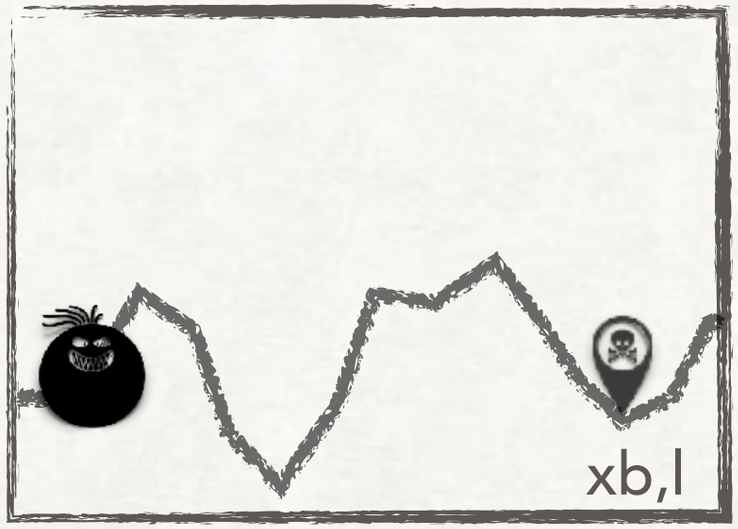

PROGRESS

```
INITIAL CONDITIONS ->
<{
{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }^@   ?(T > EPS)^@

{ WORLD CONTROLLER}^@ ?(DX > EPS)^@
}*
> PROGRESS?
```

IF V = 1, EPS^2 PROGRESS IN EACH ITERATION

ENVIRONMENT IS “EPSILON” REASONABLE

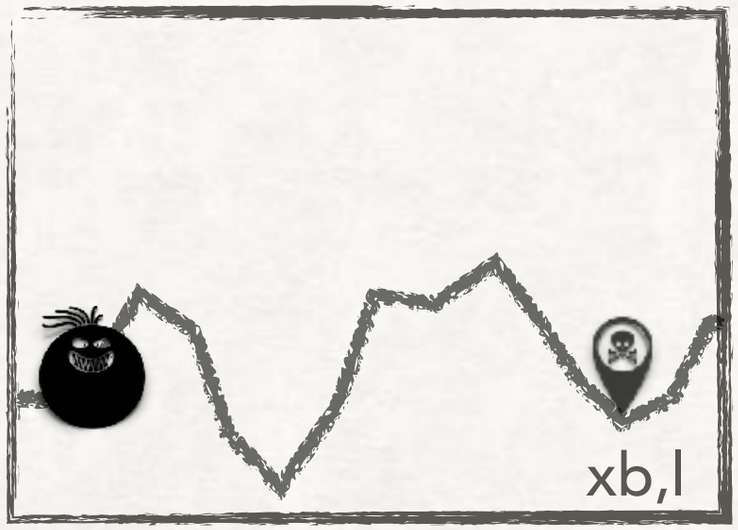
# GAME CHALLENGES




AVOID

```
INITIAL CONDITIONS ->
<{
{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }^@   ?(T > EPS)^@

{ WORLD CONTROLLER}^@ ?(DX > EPS)^@
}*
> AVOID?
```

# GAME CHALLENGES

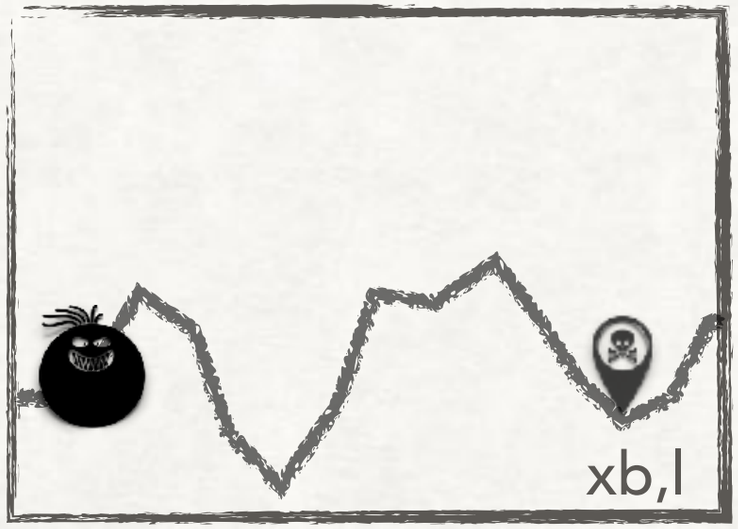



AVOID

```
INITIAL CONDITIONS ->
<{
{
IF ( T = T)
      T := 0
      PLAYER CONTROLLER
}

{ ODES  & T < T }^@   ?(T > EPS)^@

{ WORLD CONTROLLER}^@ ?(DX > EPS)^@
}
> AVOID?
```

SINGLE STEP

# GAME CHALLENGES




INITIAL CONDITIONS:
XB = X + TJ * V , T = 2*TJ, DX > 0

POST CONDITION:
X = XB -> Y > L

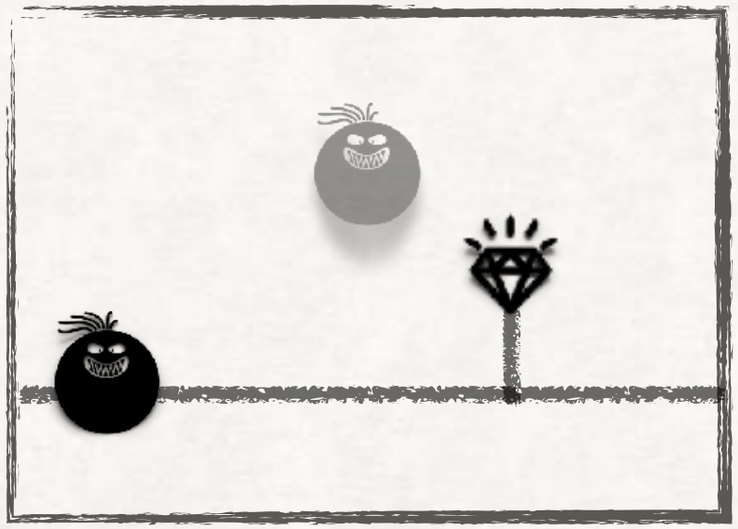
## AVOID

TJ = (J+V)^2/2G

CAN MAKE "PROGRESS" ON THE LEVEL

CAN AVOID OBSTACLE BY JUMPING IF CONDITIONS HOLD

# GAME CHALLENGES



## ATTACK

INITIAL CONDITIONS:

XB = X + (TJ + TB)*V , TJ + TB < T

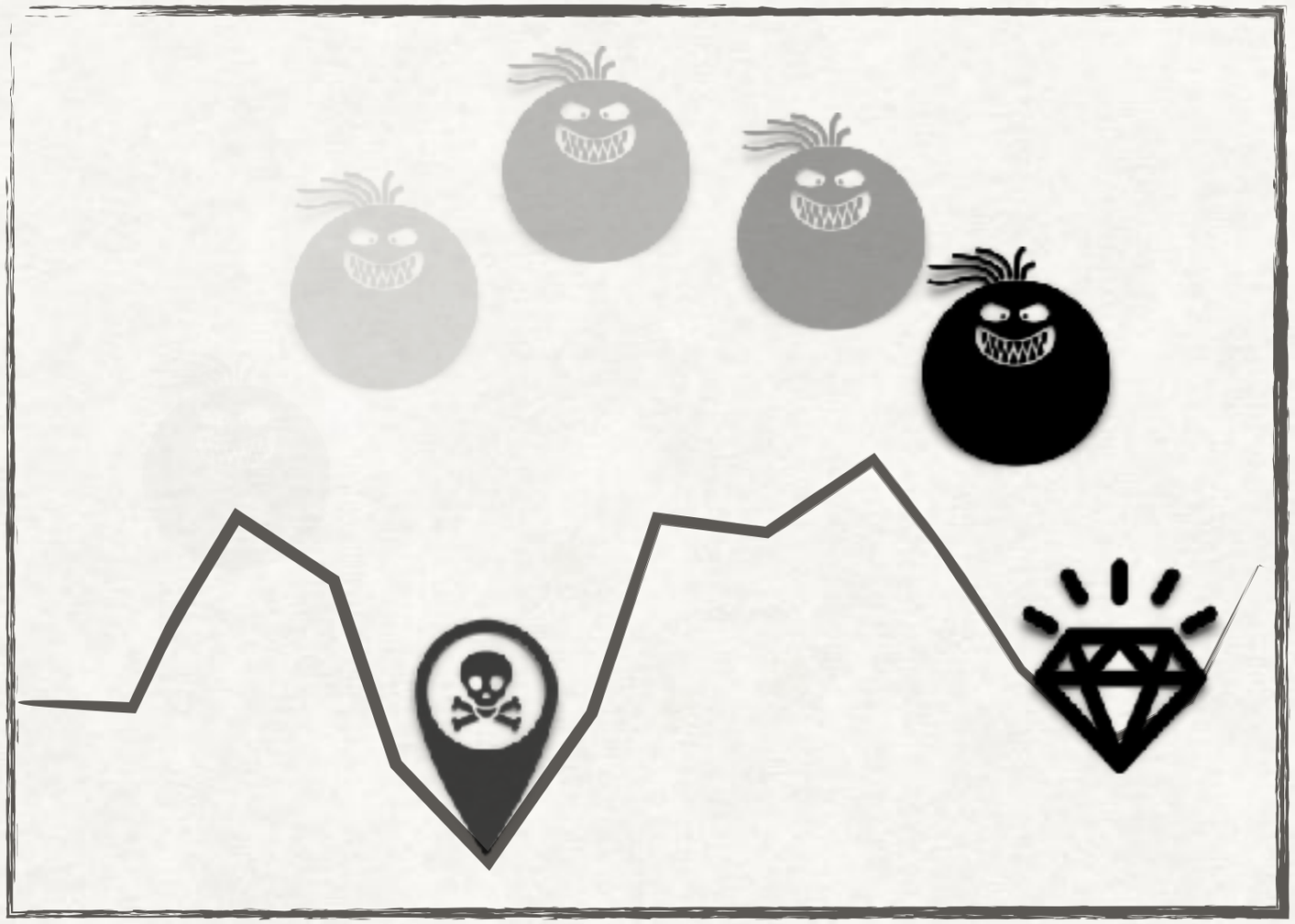
POST CONDITION:
X = XB -> Y = YB

HJ = (J+V)^2/2G

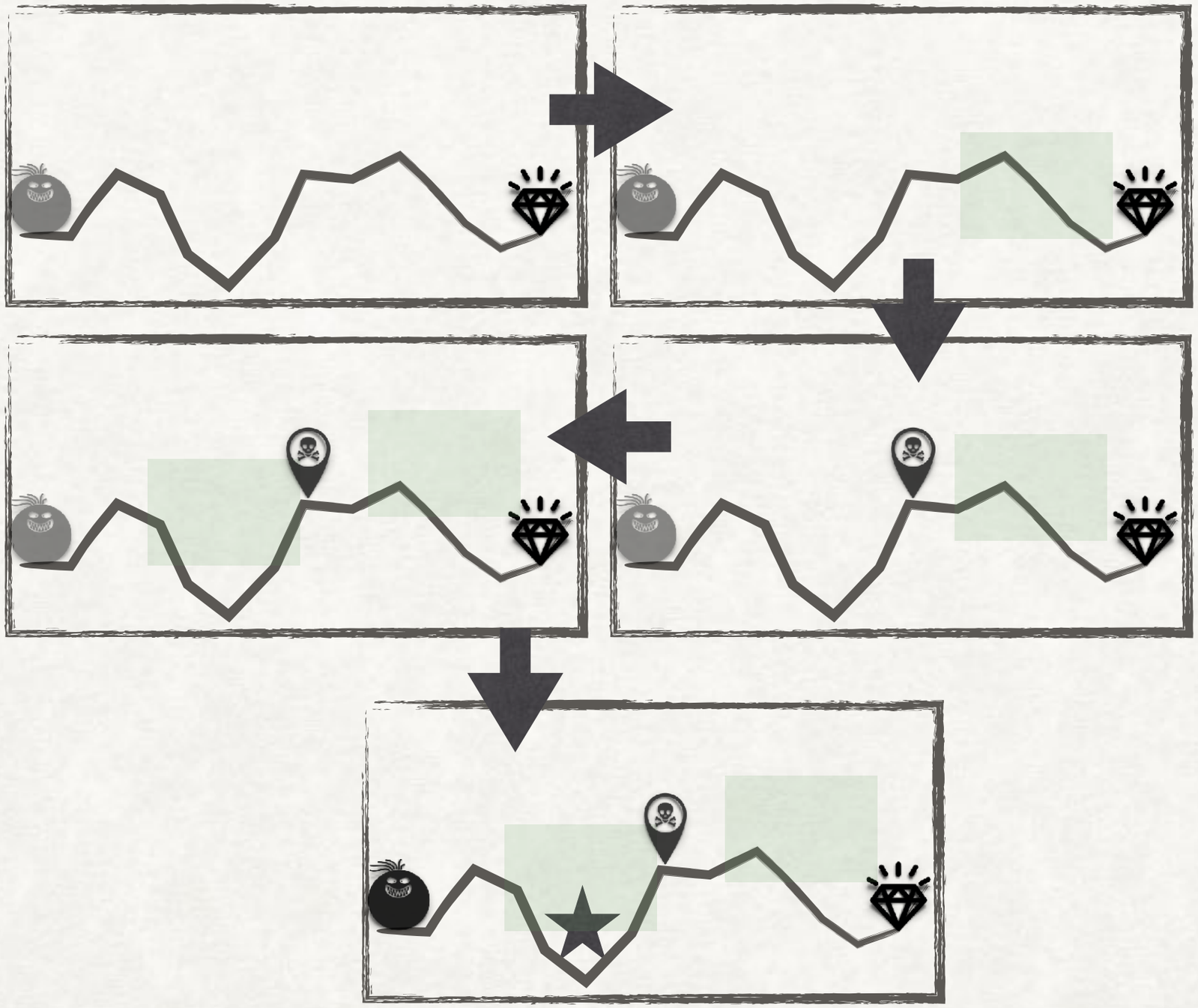
TJ = (J+V)/G

TB^2 = 2(HB-HJ)/G , TB > 0

# COMPOSITION?



# INTERACTIVE DESIGN?



# THANK YOU!

ICON CREDITS: THE NOUN PROJECT
JOEL MCKINNEY, ICONSPHERE, CORPUS DELICTI