

# The SimpleFPS Planning Domain: A PDDL Benchmark for Proactive NPCs

Stavros Vassos, Michail Papakonstantinou  
University of Athens, Greece

# Very brief introduction

2

- Focus on first-person shooter (FPS) games and the non-player characters (NPCs) that act against the human player.
- Focus on goal-oriented action planning (GOAP) for NPC behavior.

# Very brief introduction

3

- Focus on first-person shooter (FPS) games and the non-player characters (NPCs) that act against the human player.
- Focus on goal-oriented action planning (GOAP) for NPC behavior.
- SimpleFPS is a first step towards evaluating how the existing academic approaches for planning would perform if directly applied in an FPS setting.

# Overview

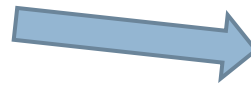
4

- STRIPS planning and PDDL
- The SimpleFPS planning domain
- Preliminary results using award-winning planners
- Conclusions and future work

# STRIPS Planning

5

- Given:
  - Initial State



# STRIPS Planning

6

- Given:
  - Initial State
  - Goal



# STRIPS Planning

7

- Given:
  - Initial State
  - Goal
  - Available actions



# STRIPS Planning

8

## □ Given:

- Initial State

- Goal

- Available actions

## □ Find:

- A **sequence of actions** that satisfy the goal

- E.g.: [**Left, Down, Left, Up, ...**]



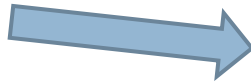


# STRIPS Planning

9

## □ Given:

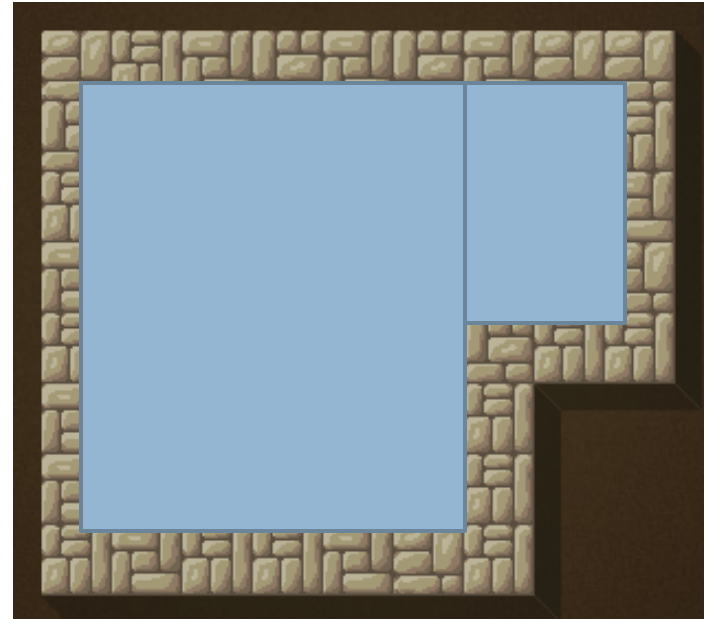
□ Initial State



□ Goal



□ Available actions



## □ Find:

□ A **sequence of actions** that satisfy the goal



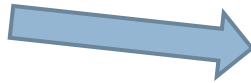
□ **Planner: Works for every initial state/goal**

# STRIPS Planning

10

## □ Given:

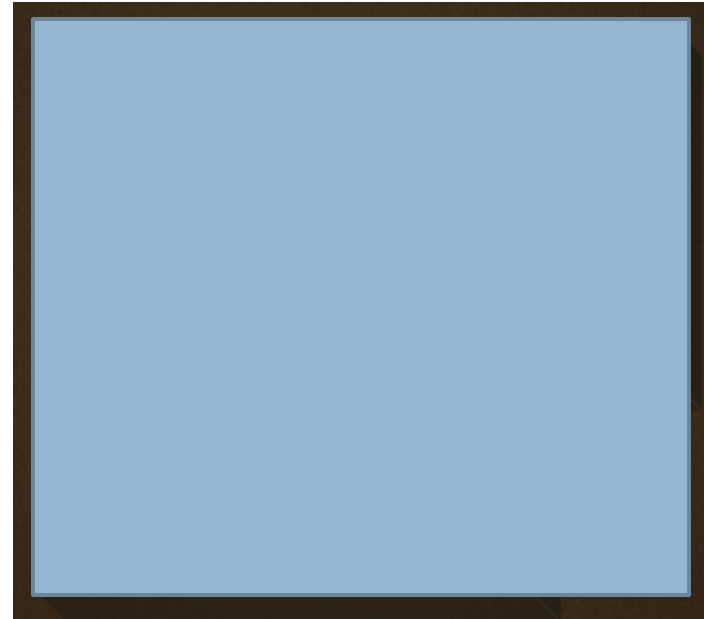
□ Initial State



□ Goal



□ Available actions



## □ Find:

□ A **sequence of actions** that satisfy the goal



□ **Planner: Works for every initial state/goal, domain**

# Planning Domain Description Language

11

- Language for specifying STRIPS planning problems
- Formal syntax like a programming language
  - ▣ Initial State        (:init ...)
  - ▣ Goal                (:goal ...)
  - ▣ Actions             (:action name  
                          :parameters (?from ?to ?dir)  
                          :preconditions (...)  
                          :effects (...)  
                          )

# Planning Domain Description Language

12

- Language for specifying STRIPS planning problems
- Formal syntax like a programming language

- Initial State        (:init ...)
- Goal                (:goal ...)
- Actions             (:action name  
                          :parameters (?from ?to ?dir)  
                          :preconditions (...)  
                          :effects (...)  
                          )

- Based on **literals**, e.g., bot-at(area1)    (bot-at area1)

# Planning Domain Description Language

13

- Language for specifying STRIPS planning problems
- Formal syntax like a programming language

- Predicates (:predicates ...)
- Actions (:action name  
:parameters (?from ?to ?dir)  
:preconditions (...)  
:effects (...)  
)

Planning  
Domain

- Objects (:objects ...)
- Initial State (:init ...)
- Goal (:goal ...)

Planning  
Problem

# Planning Domain Description Language

14

## □ International Planning Competition 1998 – today

- SAT Plan
- TL Plan
- FF
- BlackBox
- SHOP2
- TALPlanner
- ...

Planning Domains in PDDL:  
Blocks world,  
Storage, Trucks, ...

Planning Problems in PDDL  
for these domains

Comparisons  
Evaluation  
Conclusions



# Planning Domain Description Language

15

## □ International Planning Competition 1998 – today

- SAT Plan
- TL Plan
- FF
- BlackBox
- SHOP2
- TALPlanner
- ...

Planning Domains in PDDL:  
**SimpleFPS Domain**

Planning Problems in PDDL  
**for SimpleFPS domain**



Comparisons  
Evaluation  
Conclusions



# Motivation for SimpleFPS

16

- Planning in academia: extensively tested
  - ▣ Many PDDL planning domains and problems available
  - ▣ Many off-the-shelf PDDL planners available
- Planning in FPS video games: not extensively tested
  - ▣ A few success stories but not clear if the same works under different assumptions or what is the best approach
- SimpleFPS: A PDDL domain for evaluating academic planning techniques for NPCs in First-Person Shooters



# SimpleFPS

17

- SimpleFPS\_PDDL\_Domain.txt
- SimpleFPS\_PDDL\_ProblemGenerator.c

# SimpleFPS

18

- SimpleFPS\_PDDL\_Domain.txt:
  - ▣ Specifies the **predicates** that can be used to describe the initial state of the game-world and the goal condition for the NPC.
  - ▣ Specifies a list of **available actions** that the NPC can perform, along with their **preconditions** and **effects** in terms of the predicates of the domain.
- SimpleFPS\_PDDL\_ProblemGenerator.c:
  - ▣ A tool to generate problem instances, i.e., an **initial state** of the game-world and a **goal** for the NPC.

# SimpleFPS domain

# SimpleFPS domain

20

- Represent only very simple features:
  - ▣ Game-world consists of interconnected **areas**, each of which has a number of **points of interest** (POIs).
  - ▣ A few types of **items** located at POIs (weapons, ammo, med-kits, keycards).
  - ▣ NPC can perform **basic actions** (move between areas or POIs, pick-up/use items, attack, take cover).

# A SimpleFPS problem instance

21

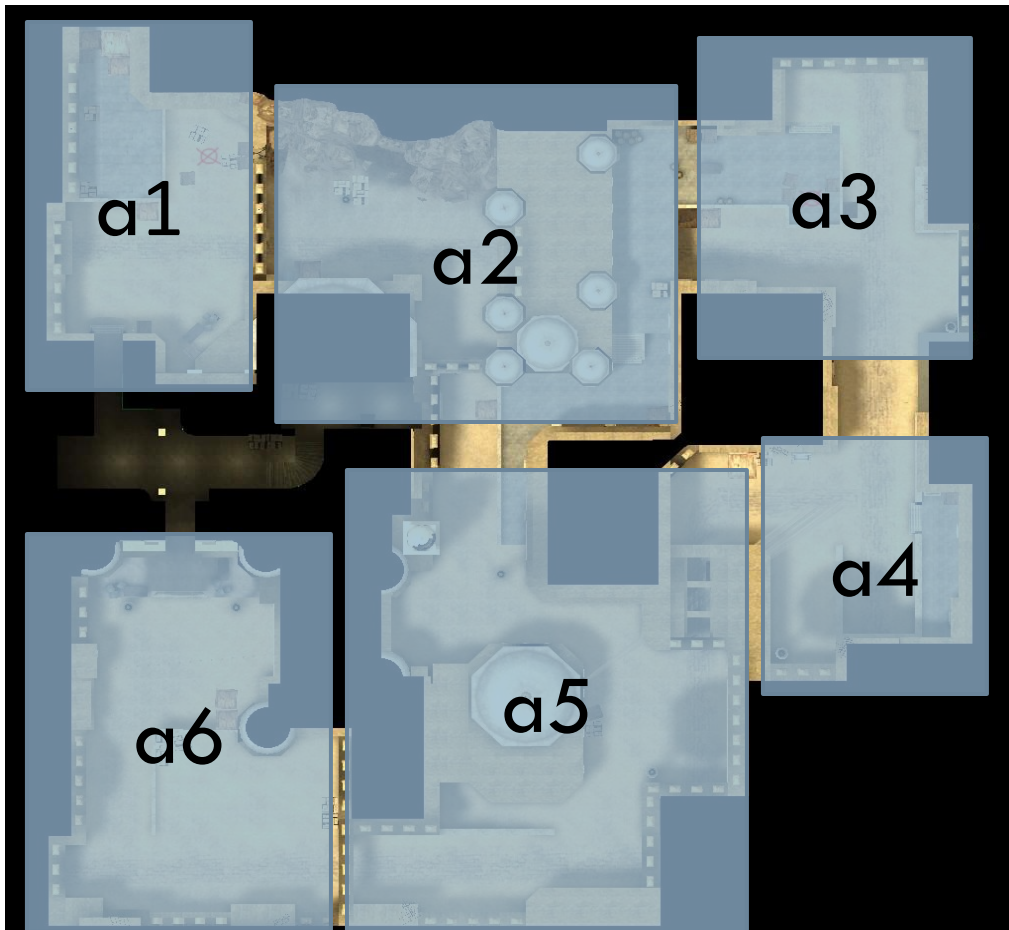
□ (:init ...)



# A SimpleFPS problem instance

22

□ (:init ...)



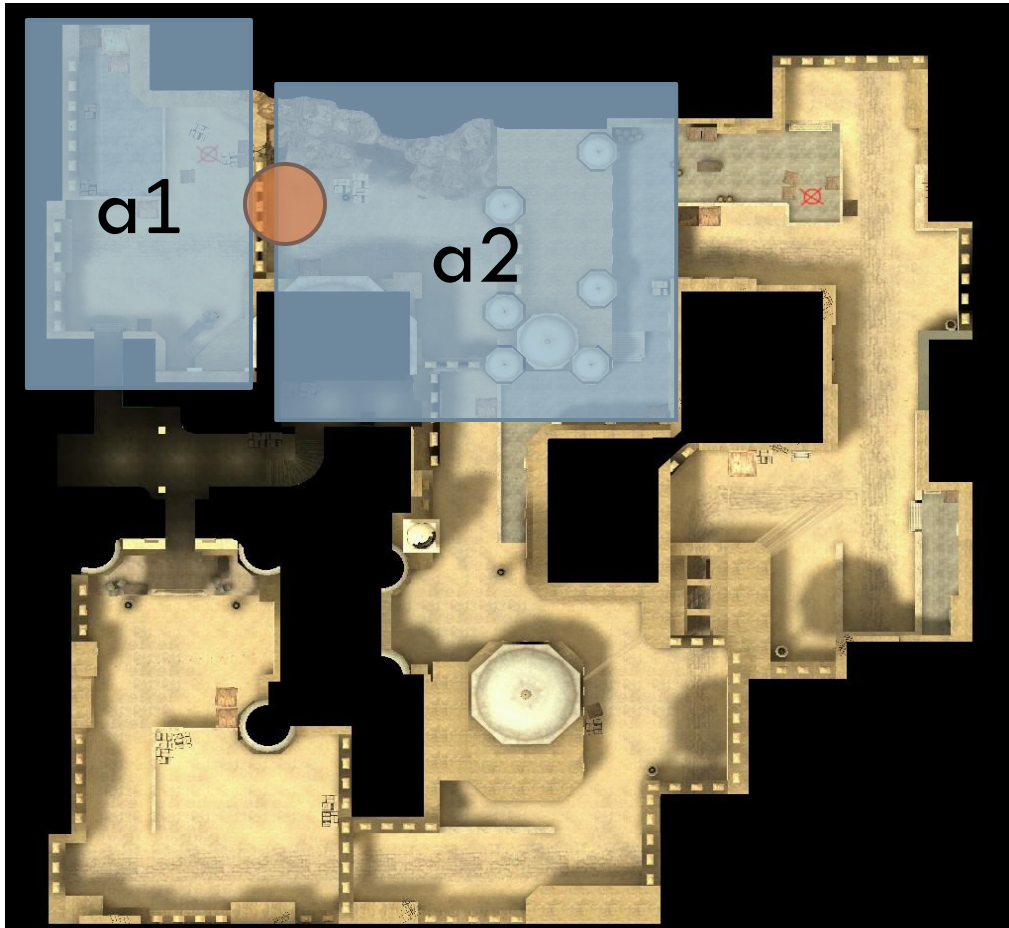
□ 6 areas, some of them connected through doors and corridors:

- (area a1)
- (area a2)
- (area a3)
- (area a4)
- (area a5)
- (area a6)

# A SimpleFPS problem instance

23

□ (:init ...)

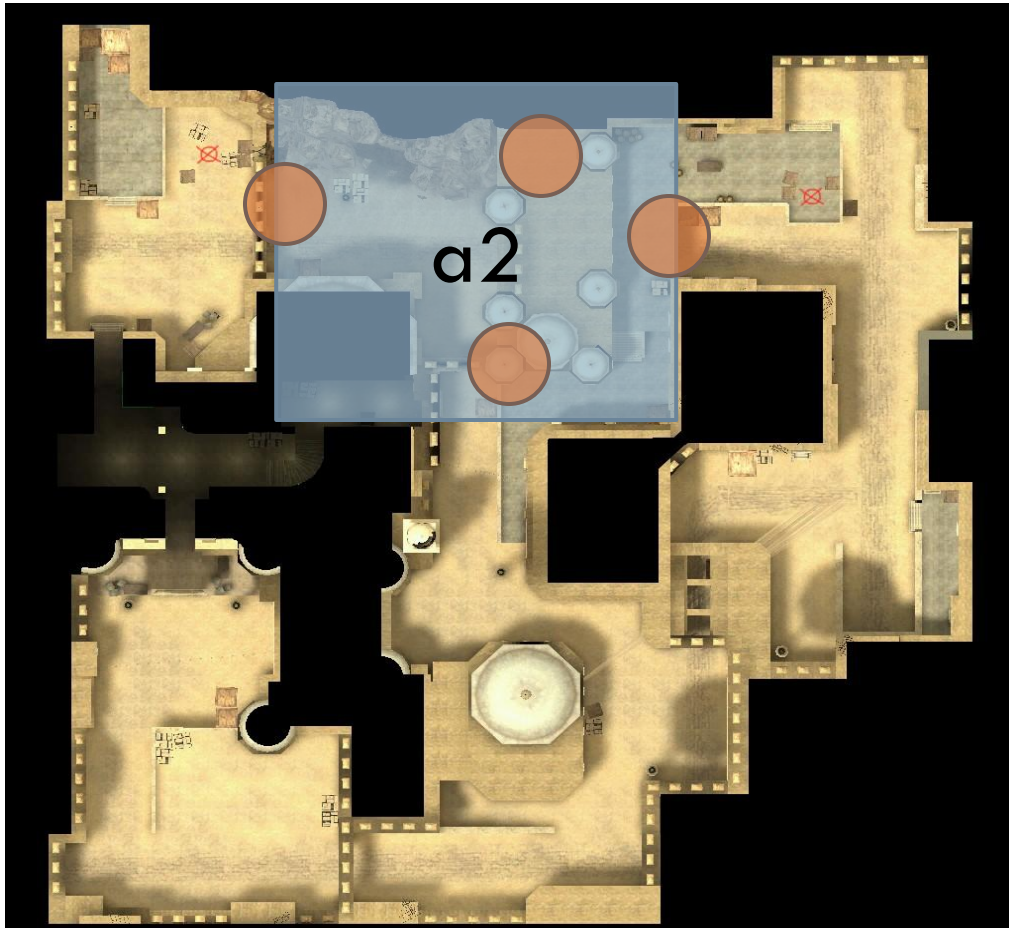


- 6 areas, some of them connected through doors and corridors:
- (poi door1 a1)
- (waypoint door1)
- (connected a1 a2 door1)
- (closed door1)
- (opens door1 keycard1)

# A SimpleFPS problem instance

24

□ (:init ...)



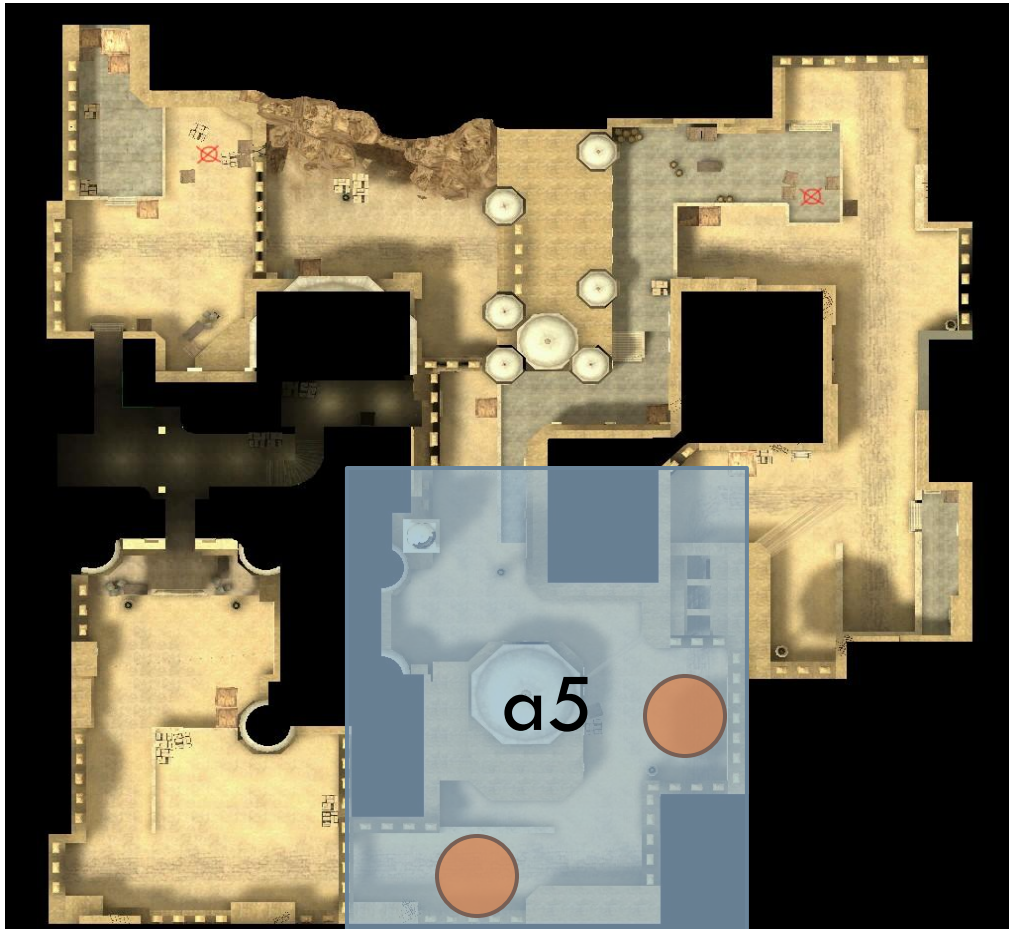
- For each area a number of POIs are listed along with their properties:
- (poi door1 a2)
- (poi c1 a2)
- (poi c2 a2)
- (connected a2 a1 door1)
- (cover-point c1)
- (cover-point c2)



# A SimpleFPS problem instance

25

□ (:init ...)

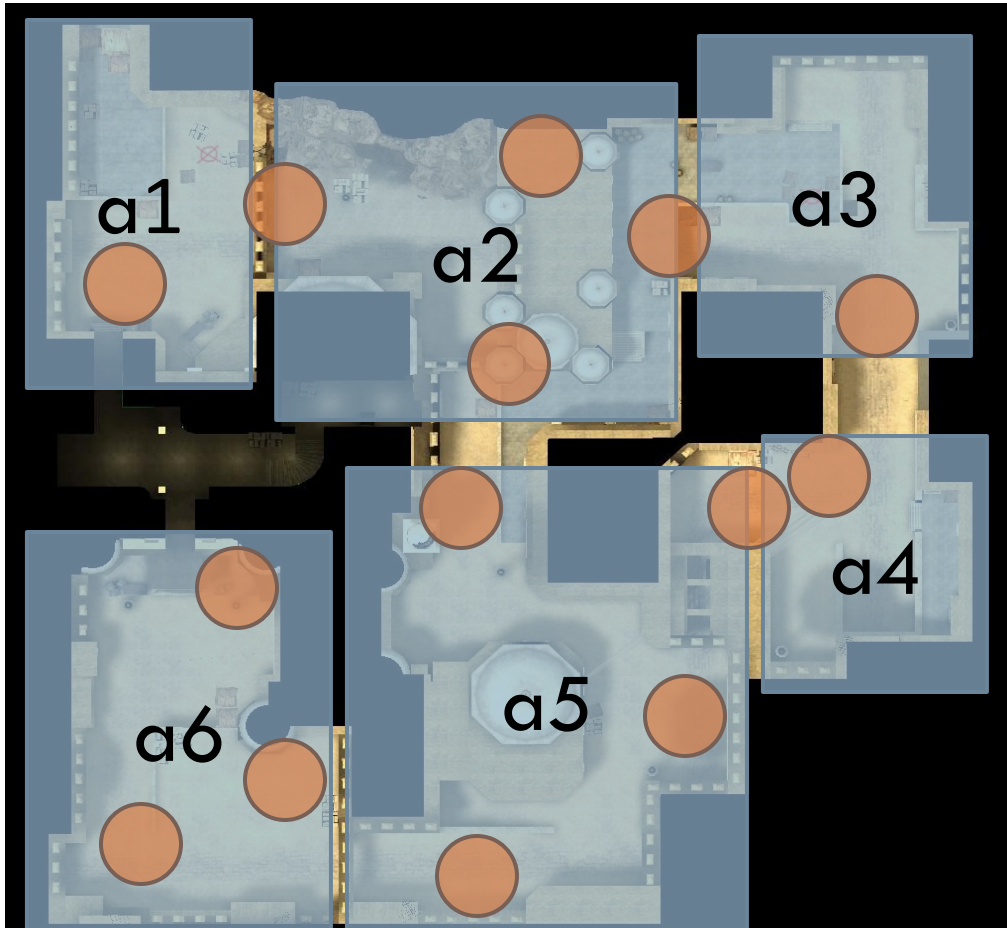


- For each area a number of POIs are listed along with their properties:
- (poi g1 a5)
- (poi amm1 a5)
- (gun g1)
- (unloaded g1)
- (ammo amm1 g1)

# A SimpleFPS problem instance

26

□ (:init ...)



□ For each area a number of POIs are listed along with their properties:

- knife
- med-kit
- control-box
- night-vision-gun
- ...

# SimpleFPS domain: Predicates

27

## □ NPC-related

- (npc-at ?a)
- (npc-close-to ?p)
- (npc-covered)
- (npc-uncovered)
- (npc-holding ?o)
- (npc-injured)
- (npc-full-health)
- (npc-aware)
- (npc-unaware)

## □ Area-related

- (area ?a)
- (conn ?a1 ?a2 ?w)
- (waypoint ?w)
- (lighted ?area)
- (dark ?area)
- (poi ?p ?a)
- (control-box ?p)
- (cover-point ?p)
- (item ?p)

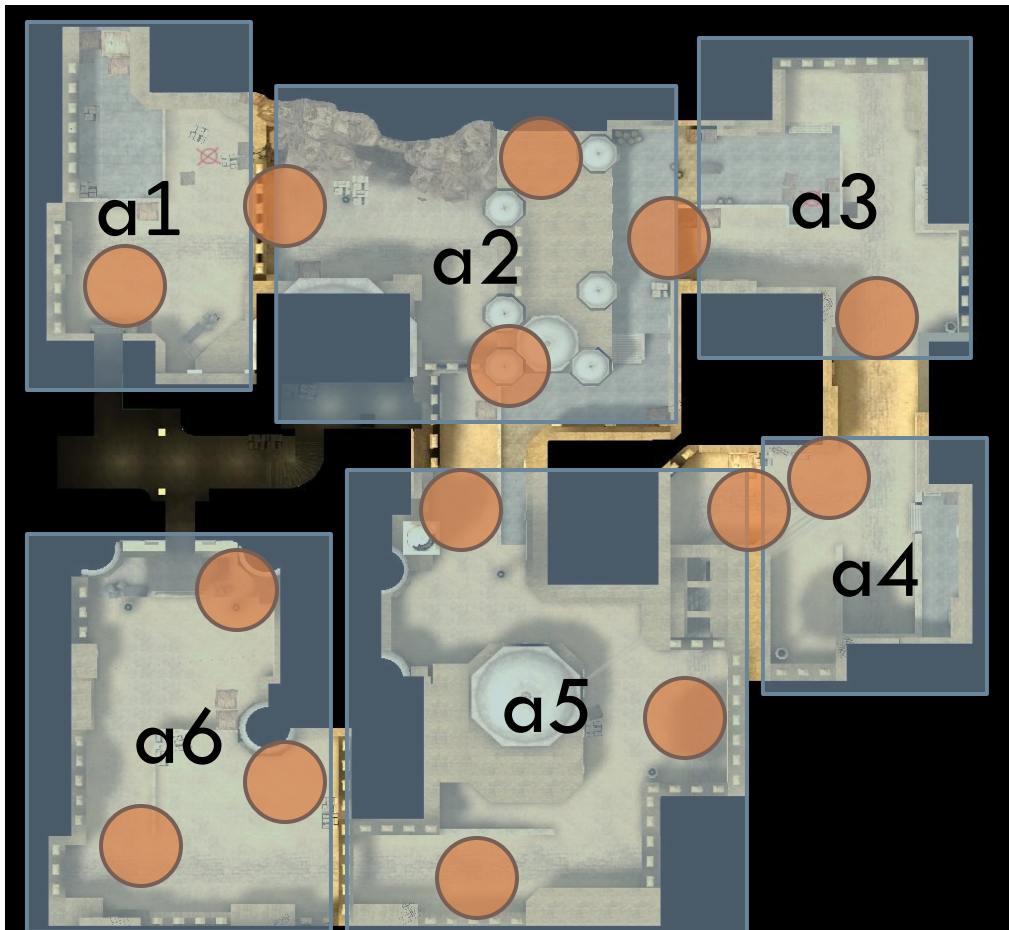
## □ Item-related:

- (med-kit ?m)
- (knife ?k)
- (gun ?g)
- (loaded ?g)
- (unloaded ?g)
- (ammo ?i ?g)
- (night-vision ?g)

# SimpleFPS domain: Actions

28

□ (: action ...)



- Available NPC actions:
- move-to-area
- move-to-poi
- pick-up-item
- use-item
- take-cover
- un-cover

# SimpleFPS domain: move-to-point

29

```
(:action move-to-point
  :parameters (?area ?point)
  :precondition (and
    (npc-at ?area)
    (point-of-interest ?point ?area)
  )
  :effect (and
    (npc-close-to ?point)
  )
)
```

# SimpleFPS domain: reload

30

**(:action reload**

**:parameters** (?gun ?item)

**:precondition** (and

(npc-holding ?gun) (gun ?gun) (unloaded ?gun)

(npc-holding ?item) (ammo ?item ?gun)

)

**:effect** (and

(not (unloaded ?gun))

(loaded ?gun)

(not (npc-holding ?item))

)

)

# The SimpleFPS domain: Actions

31

## □ Location-related:

- moving-to-patrol
- moving-to-take-position
- move-away-from-point
- move-to-point
- move-to-point-from-point
- make-accessible
- place-in-inventory
- turn-on-lights
- turn-off-lights

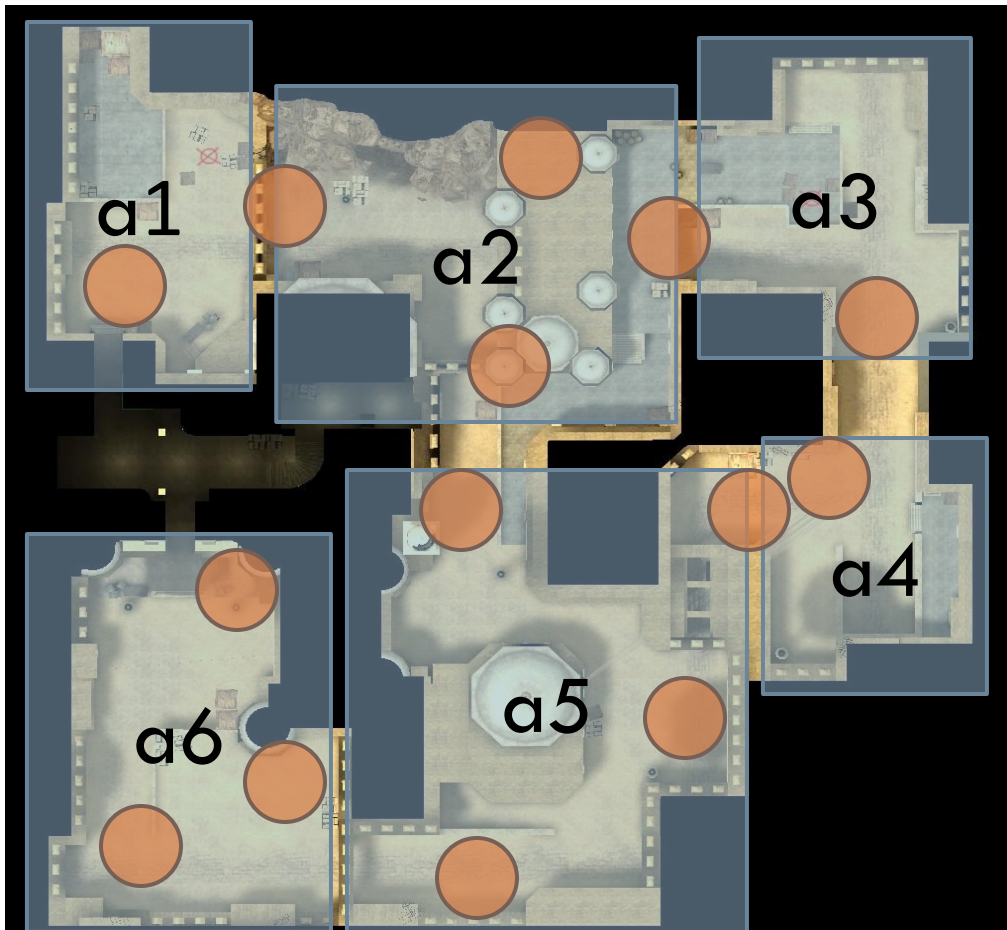
## □ Attack-related:

- make-contact
- take-cover
- uncover
- use-med-kit
- reload
- attack-melee
- attack-ranged
- sneak-kill

# A SimpleFPS problem instance

32

□ (:goal ...)



- NPC goals:
- **g1**: (player-wounded)
- **g2**: (npc-covered)
- **g3**: (npc-full-health)
- **g4**: (and g1 g2 g3)



# A SimpleFPS problem instance

33

- 1. (move-to-point area0 door3-0)
- 2. (moving-to-patrol area0 area3 door3-0)
- 3. (move-to-point area3 control-box3)
- 4. (turn-on-lights area3 control-box3)
- 5. (move-to-point-from-point area3 ammogun3 control-box3)
- 6. (make-contact area3 p)
- 7. (place-in-inventory area3 ammogun3)
- 8. (move-to-point area3 gun3)
- 9. (place-in-inventory area3 gun3)
- 10. (move-to-point area3 firstaid1)
- 11. (reload gun3 ammogun3)
- 12. (attack-ranged area3 gun3 p)
- 13. (place-in-inventory area3 firstaid1)
- 14. (move-to-point area3 coverpoint1)
- 15. (use-medikit firstaid1)
- 16. (take-cover area3 coverpoint1)

# SimpleFPS problem generator

# SimpleFPS problem generator

35

- Takes as input:
  - -a number of areas,
  - -c the probability that two areas are connected
  - -n total number of points of interest
  - -g the goal condition as one of g1, g2, g3, g4
  - -l the number of instances to be generated
- Generates problem instances also using some rules:
  - Card-keys are added for locked doors
  - Ammo is added for guns that are unloaded
  - ...

# Preliminary results with SimpleFPS

36

- Used the tool to generate 3 datasets:
  - 5 areas
  - 7 areas
  - 10 areas
- For each dataset we generated 10 instances with:
  - 10 POIs
  - ...
  - 100 POIs
- For each of the 4 goals:
  - g1, g2, g3, g4

# Preliminary results with SimpleFPS

37

- Used three award-winning planners in these datasets:
  - BlackBox (Kauts and Selman 1999)
  - FastForward (Hoffman 2001)
  - FastDownward (Helmert 2006)
- Laptop: Ubuntu 11.04, Core Duo 2GHz, 4GB RAM

# Preliminary results with SimpleFPS

38

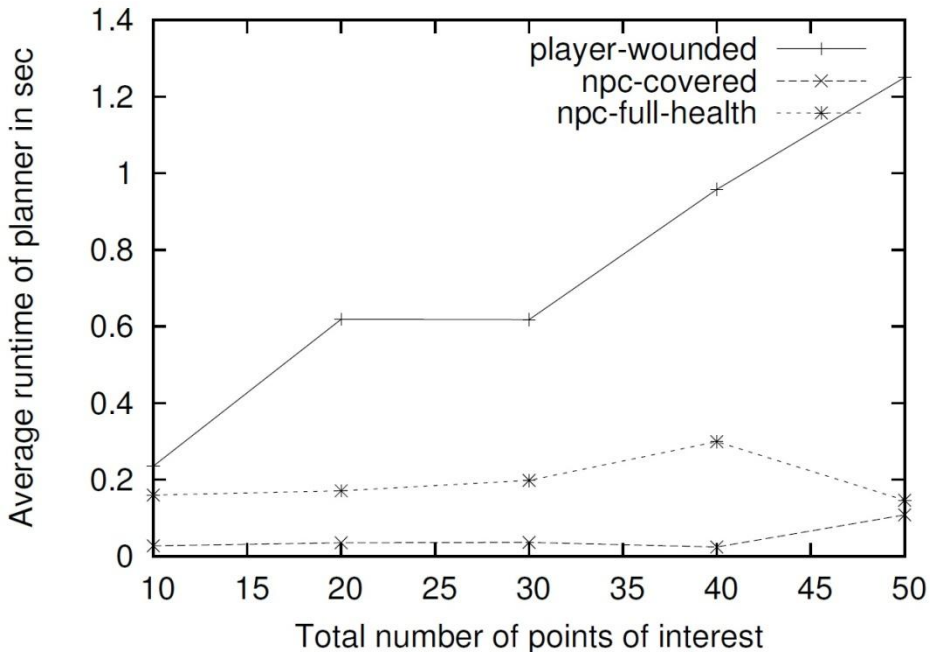
- BlackBox, FastForward: Two graphs in the paper
  - ▣ Problems planners always return an answer within 1.5sec

# Preliminary results with SimpleFPS

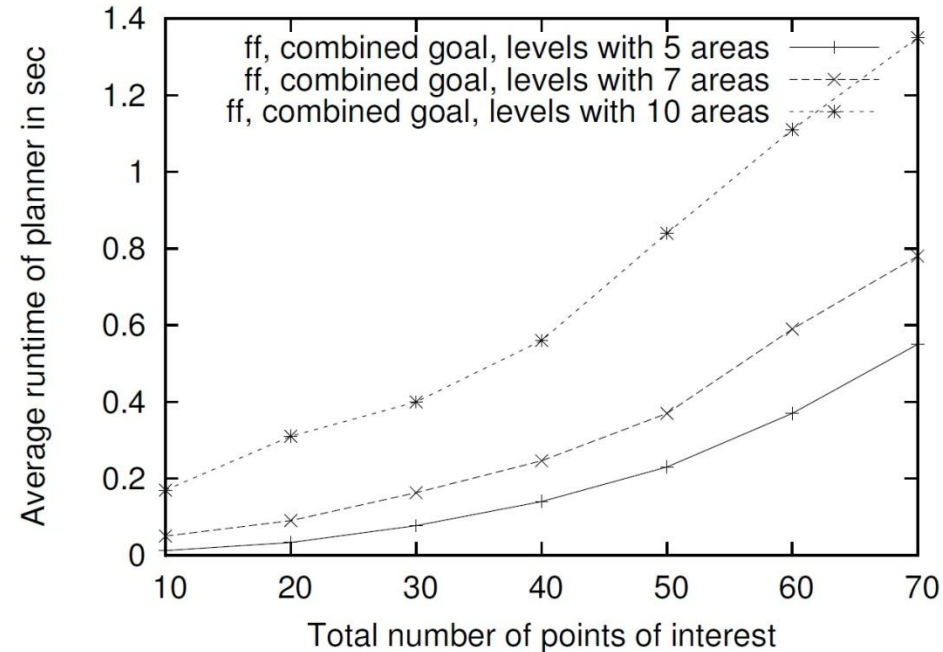
39

- BlackBox, FastForward: Two graphs in the paper
  - Problems planners always return an answer within 1.5sec

BB: up to 5 areas/50 POIs



FF: up to 10 areas/70 POIs

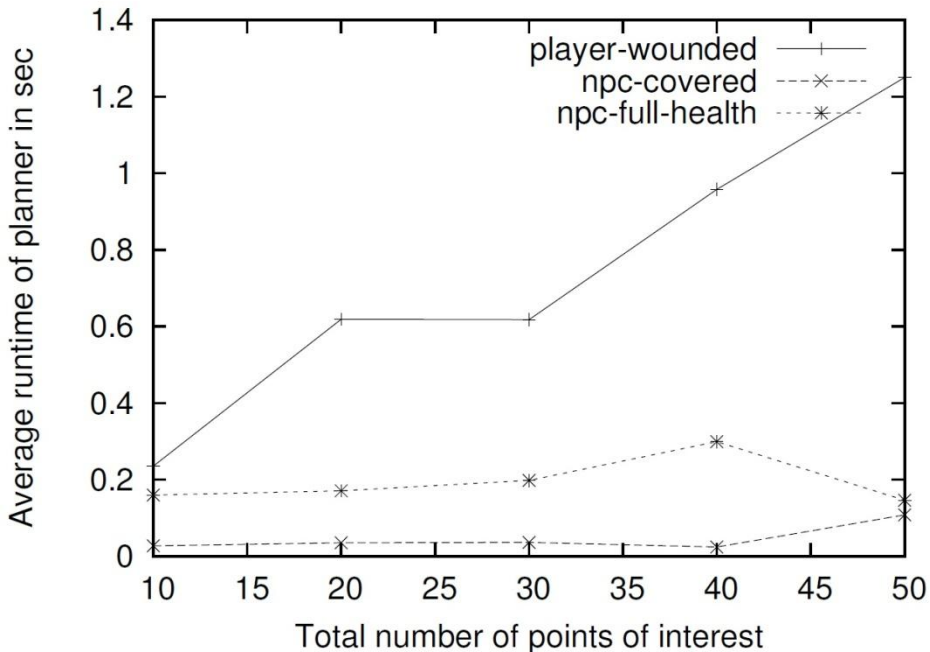


# Preliminary results with SimpleFPS

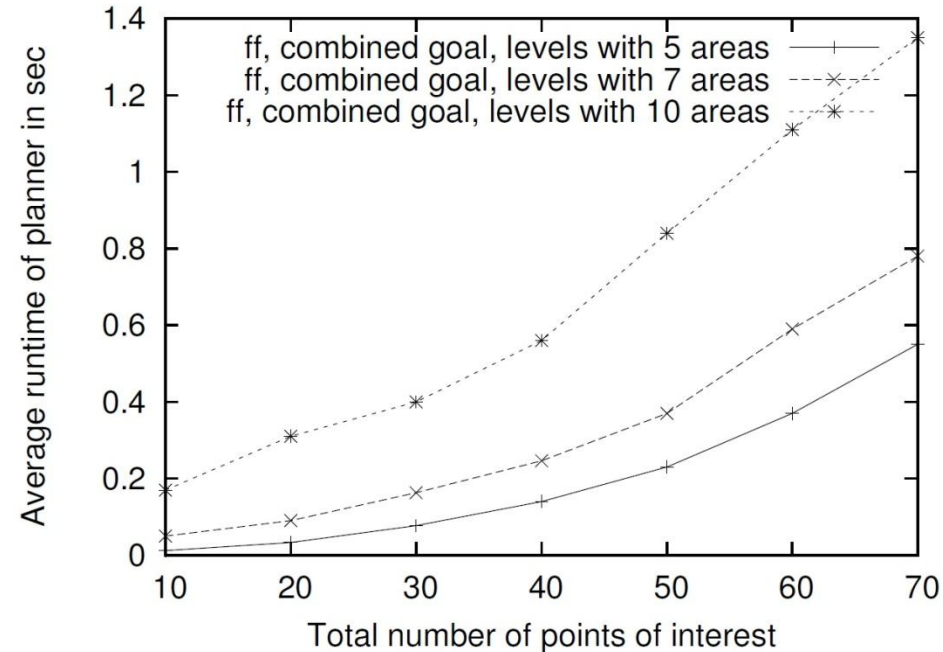
40

Different planning techniques  
make a lot of difference

BB: up to 5 areas/50 POIs



FF: up to 10 areas/70 POIs





# Preliminary results with SimpleFPS

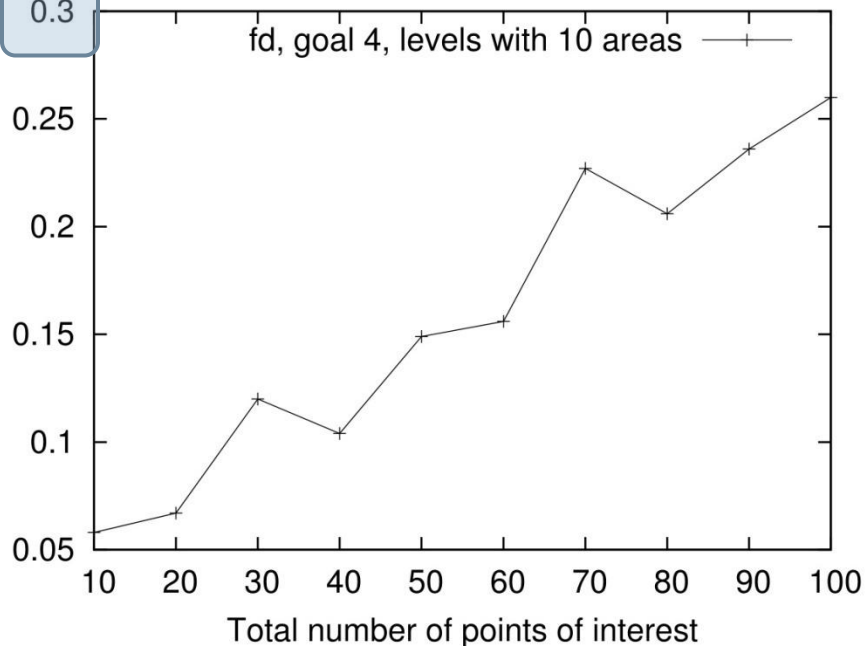
41

## FastDownward:

- Always returns an answer within  $\sim 0.25$ sec

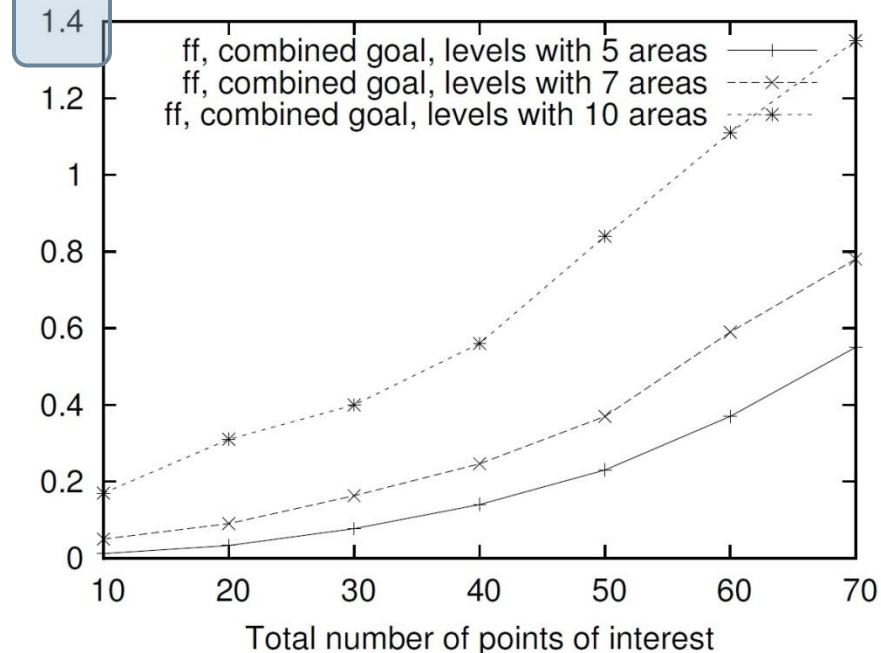
FD: 10 areas / 100 POIs / g4

Average time of planner in sec



FF: up to 10 areas / 70 POIs

Average runtime of planner in sec



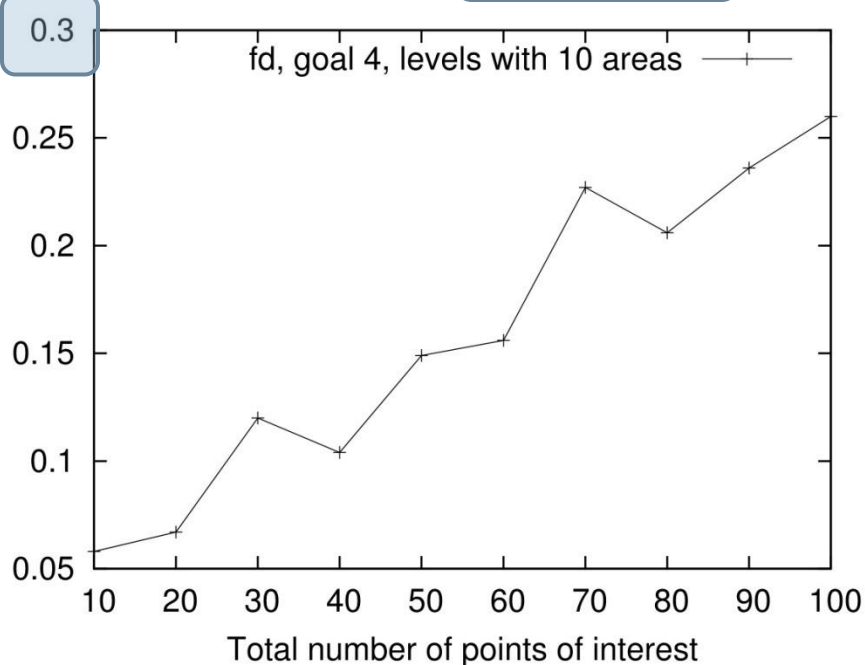
# Preliminary results with SimpleFPS

42

Planning techniques and planners  
get better over time

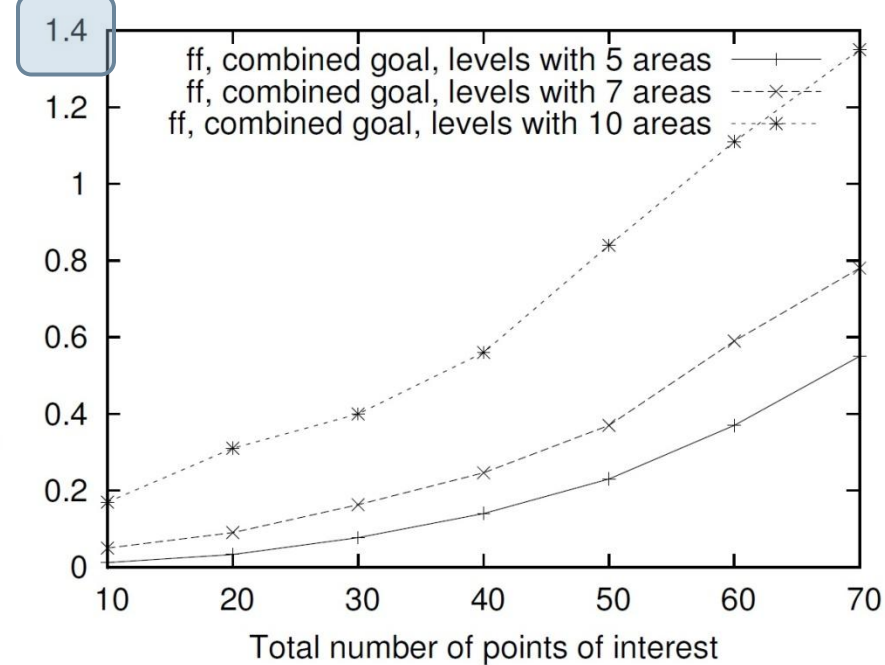
FD: 10 areas / 100 POIs / g4

Average time of planner in sec



FF: up to 10 areas / 70 POIs

Average runtime of planner in sec

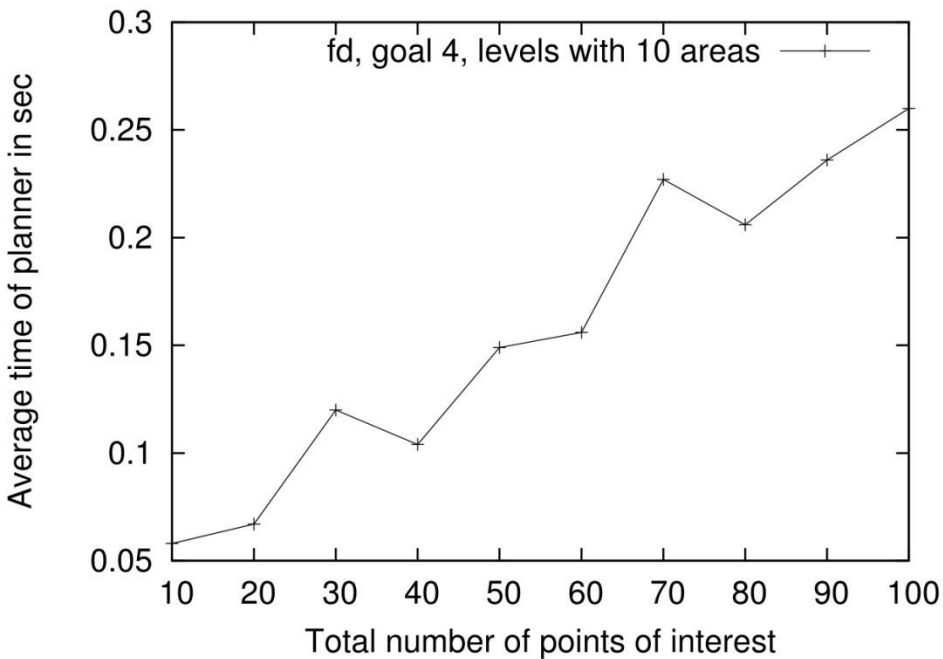


# Preliminary results with SimpleFPS

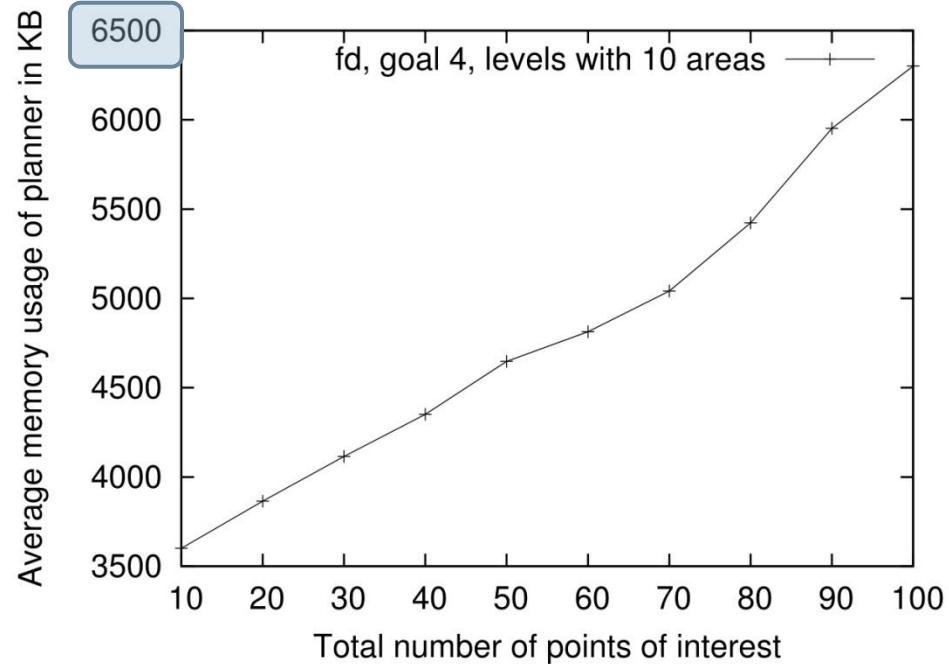
43

- FastDownward:
  - ▣ Uses up to  $\sim 6.5\text{MBs}$

## FD: Running time



## FD: Memory



# Preliminary results with SimpleFPS

44

- Relatively small-sized problems wrt FPS games:
  - ▣ 7 areas/100 POIs
- The planner takes a lot of resources:
  - ▣ A lots of **time**: ~7 frames to respond using 100% of the CPU resources of a laptop
  - ▣ A lot of **memory**: ~6.5MB

# Preliminary results with SimpleFPS

45

- Relatively small-sized problems wrt FPS games:
  - ▣ 7 areas/100 POIs
- The planner takes a lot of resources:
  - ▣ A lots of **time**: ~7 frames to respond using 100% of the CPU resources of a laptop
  - ▣ A lot of **memory**: ~6.5MB
- Different assumptions in academia and FPS games

# Preliminary results with SimpleFPS

46

- Relatively small-sized problems wrt FPS games:
  - ▣ 7 areas/100 POIs
- The planner takes a lot of resources:
  - ▣ A lots of **time**: ~7 frames to respond using 100% of the CPU resources of a laptop
  - ▣ A lot of **memory**: ~6.5MB
- Different assumptions in academia and FPS games
  - ▣ Plan length up to 20 actions

# Conclusions

47

- BB, FF, FD more or less perform as expected:
  - One of them performs poorly
  - One of them is slow but not too slow
  - One of them is good but not good enough for real-time planning

# Conclusions

48

- BB, FF, FD more or less perform as expected:
  - One of them performs poorly
  - One of them is slow but not too slow
  - One of them is good but not good enough for real-time planning
- But this is not necessarily bad news as these planners are designed to fulfill different needs:
  - Find complete solutions as long lists of actions
  - Optimal
  - Aim for speed using memory at will



# Future work

49

- Search for plans that are:
  - ▣ not optimal
  - ▣ not complete
  
- Take advantage of “propositionalization”
  - ▣ 6.5 MBs is too much for 1 character but how about 100?
  
- Combine with non-planning approaches for NPCs along with minimal planning capabilities
  - ▣ The Golog family of languages

# The SimpleFPS benchmark

50

- Available at <http://stavros.lostre.org/sFPS> :
  - SimpleFPS\_PDDL\_Domain.txt
  - SimpleFPS\_PDDL\_ProblemGenerator.c
  - Datasets with 5 areas, 7 areas, 10 areas
- Feel free to download and experiment!
- Any feedback is mostly appreciated!
- Email: [stavros@cs.toronto.edu](mailto:stavros@cs.toronto.edu)