Titre : Quelle planification dans les first-person shooters ?

Résumé : Pour répondre à la question du titre, je présenterai des données venant de trois jeux : F.E.A.R. (2005), Killzone 3 (2011) et Tranformers 3: Fall of Cybertron (2012).

Après une rapide explication sur le pourquoi et le comment de ces données, je commencerai par détailler des métriques permettant de comparer le composant de planification dans chaque jeu. Puis je proposerai des motifs invariants que l'on peut extraire des données à partir des métriques. Enfin, je résumerai ces résultats par une série de questions dont les réponses sont autant de décisions sur la conception du jeu et des fonctionnalités du composant de planification.

L'approche de ce travail se veut très empirique, un peu comme un biologiste tombant nez à nez avec trois fleurs qu'il n'a jamais vues ; que peut-il en dire ?

Lieu : Université de Québec à Montréal, le jeudi 9 octobre 2014.

La planification dans les First-Person Shooters...

?

La planification dans les First-Person Shooters...

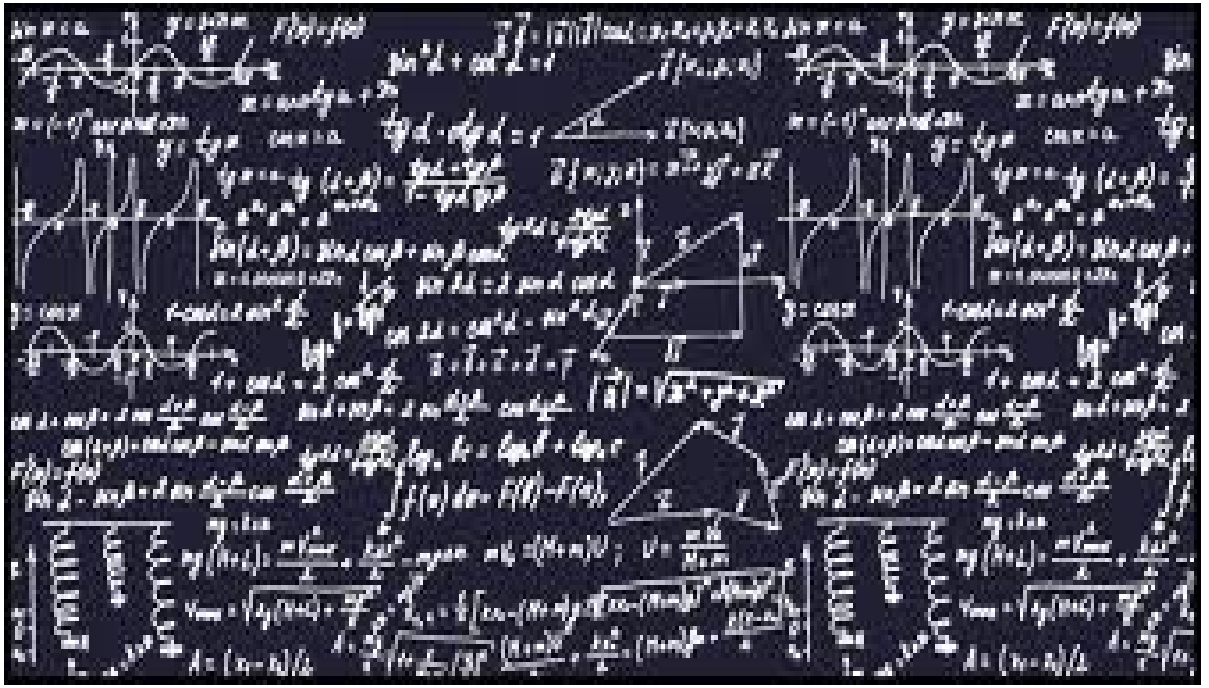
Que peut-on en dire ?

;-)

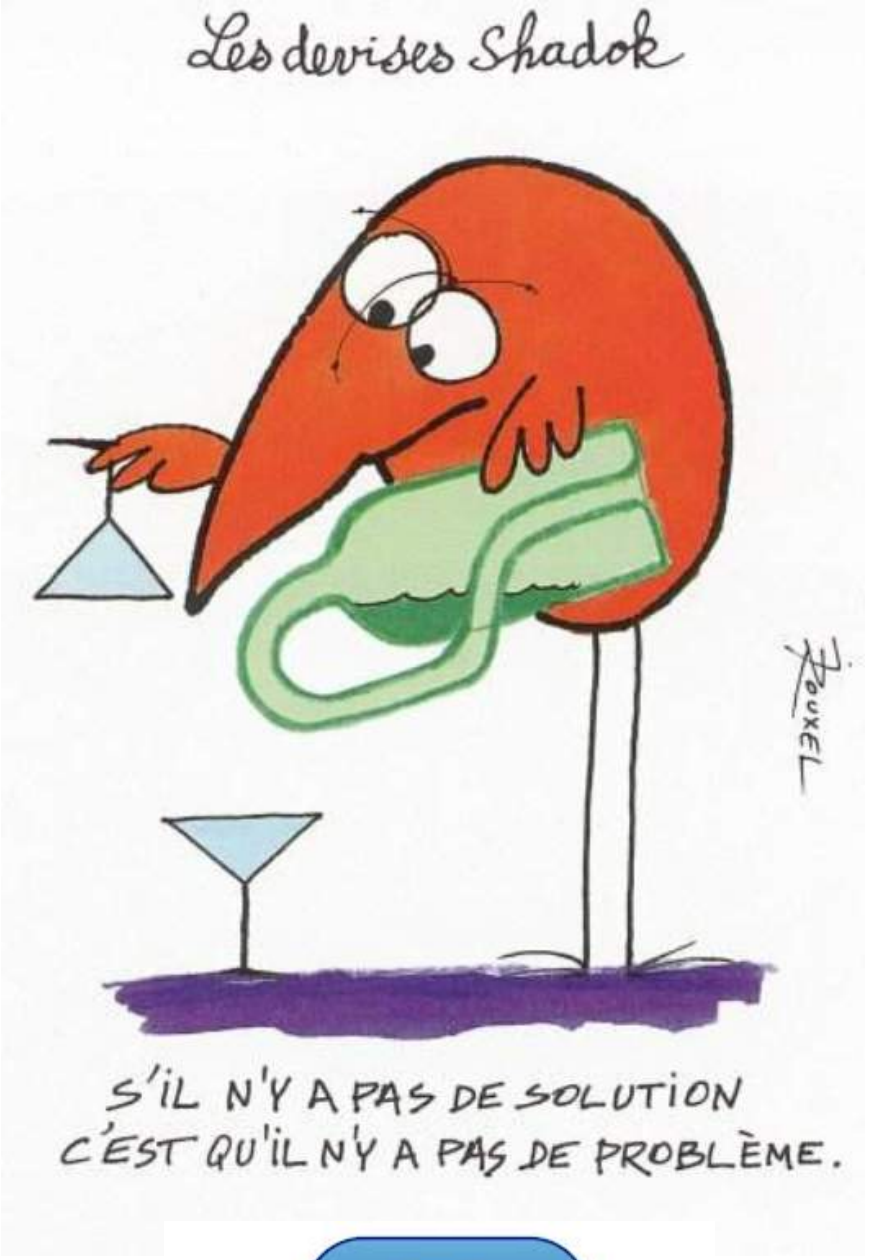

DISCLAIMERS


VIDEO GAMES
experimental gameplay project
WHY WASTE GOOD TECHNOLOGY ON SCIENCE AND MEDICINE?


Terms &
Conditions




Terms &
Conditions



Terms & Conditions

Quelle Planification dans les FPS ?

Éric Jacopin

CREC Saint-Cyr
Écoles de Coëtquidan
F-56381 GUER Cedex

eric.jacopin@st-cyr.terre-net.defense.gouv.fr
✆ +33 (0)2 90 40 40 38
🖷 +33 (0)2 90 40 40 05

Mes remerciements à

Arjen **BEIJ** (Guerilla Games)

Alex **CHAMPANDARD** (AIGameDev.com)

Chris **CONWAY** (Crystal Dynamics)

Carle **COTÉ** (Artifice Studio)

Jean-François **GAUTHIER** (EIDOS Montréal)

Troy **HUMPHREYS** (Turtle Rock Studios)

Jeff **ORKIN** (Giant Otter)

Keith **STAINES** (High Moon Studios)


UBM
AUGUST 2012
gd
GAME DEVELOPER MAGAZINE
THE LEADING GAME INDUSTRY MAGAZINE
VOL 19 NO 8 AUGUST 2012
INSIDE:
SPATIAL ANALYTICS:
GO BEYOND THE HEAT MAP
FIND THE RIGHT AI FOR YOUR GAME

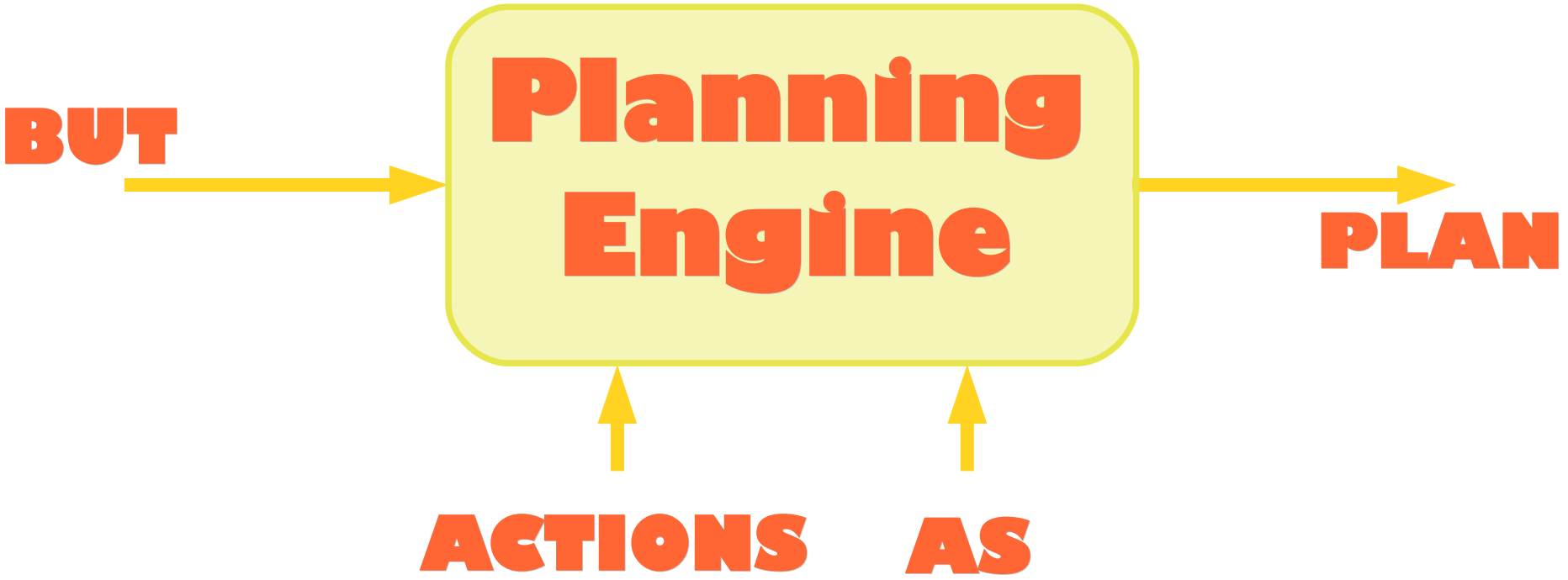
AI ARCHITECTURES

WHAT'S ON THE MENU?

HOW TO CHOOSE
THE RIGHT
ARTIFICIAL
INTELLIGENCE
FOR YOUR
GAME!

ARCHITECTURE	PROS	CONS
AD-HOC RULES	> minimal set-up	> gets unwieldy past the most basic behaviors
FINITE STATE MACHINE (FSM)	> easy to understand, build	> transition between states get hard to manage with more behaviors
HIERARCHICAL FSM	> hierarchy helps cluster behaviors > easy to understand, build	> transitions still can get difficult to manage
BEHAVIOR TREE (BT)	> separates decision logic from state code > easy to understand, build, edit	> hard-coded priorities of behaviors
PLANNER	> ai "discovers" solutions on the fly > handles unique situations better > easily accommodates new action	> some loss of designer control > "re-planning" can be processor-intensive
UTILITY-BASED SYSTEM	> ai constantly weighs all actions > handles unique situations gracefully > allows for variation in behavior	> some loss of designer control > harder to design, edit, and tune
NEURAL NETWORK	> able to "learn" how to play > can be set up relatively quickly	> complete loss of designer control > nearly impossible to edit or tune

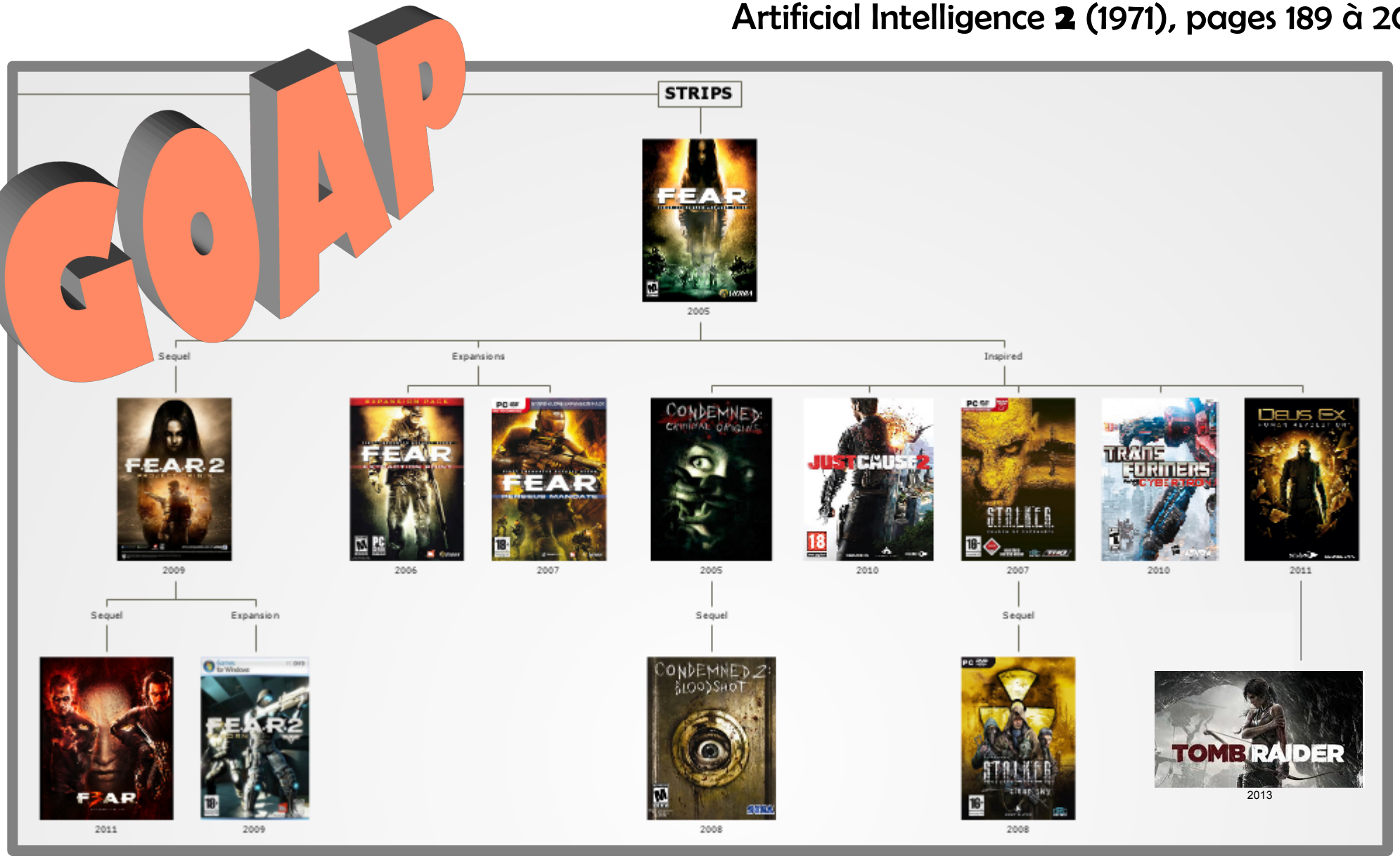
FIGURE 7: THE TYPE OF ARCHITECTURE YOU SELECT NEEDS TO BE BASED ON YOUR NEEDS.


BUT
Planning Engine
PLAN
ACTIONS
AS

Robert **FIKES** & Nils **NILSSON**

STRIPS: A New Approach to the Application of Theorem Proving

Artificial Intelligence **2** (1971), pages 189 à 208



Jeff **ORKIN**, *Applying Goal-Oriented Action Planning to Games*

AI Game Programming Wisdom 2, pages 217 à 228

Hingham, Mass.: Charles River Media (2002).

"Put simply, these are the smartest, most aggressive, most tactically oriented AI opponents that we've ever encountered in a shooter, and they're downright impressive. The AI is incredibly sharp, and they'll do things that you don't expect, like pin you down while one of them flanks you. … These guys move around from cover to cover; they communicate with one another; they'll react to any sound or sight of you."

Jeff Orkin, *The Evolution of Planning in Games*, ICAPS Planning in Games 2010, Toronto

Backward Chaining

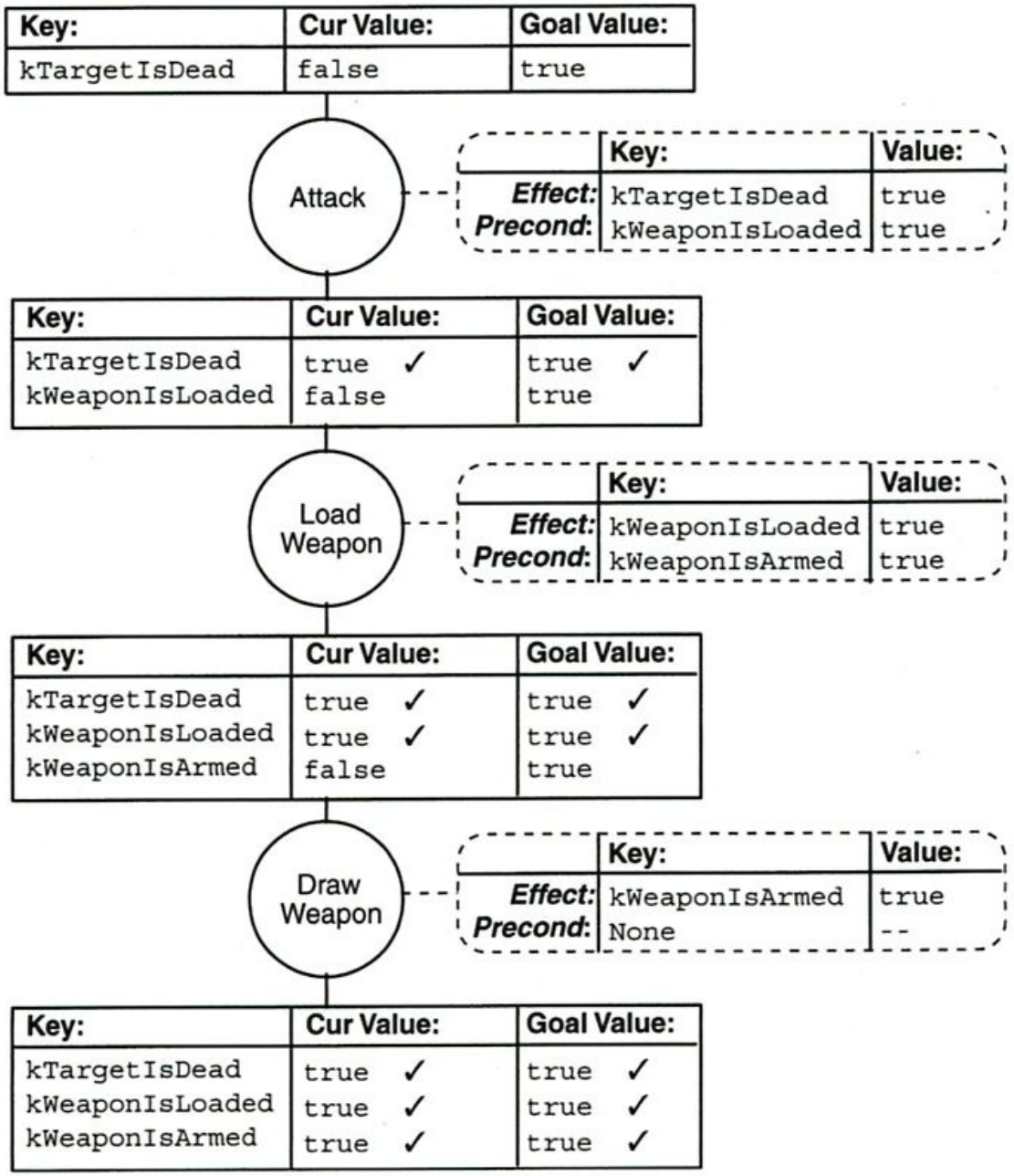


FIGURE 3.4.2 *The planner's regressive search.*

States as Arrays

World Representation

In order to search the space of actions, the planner needs to represent the state of the world in some way that lets it easily apply the preconditions and effects of actions, and recognize when it has reached the goal state. One compact way to represent the state of the world is with a list of world property structures that contain an enumerated attribute key, a value, and a handle to a subject.

```
struct SWorldProperty
{
   GAME_OBJECT_ID hSubjectID;
   WORLD_PROP_KEY eKey;

   union value
   {
      bool   bValue;
      float  fValue;
      int    nValue;
      ...
   };
};
```

Actions as Classes

```
CAIActionAttack::CAIActionAttack()
{
    m_nNumPreconditions = 1;
    m_Preconditions[0].eKey = kWeaponIsLoaded;
    m_Preconditions[0].bValue = true;

    m_nNumEffects = 1;
    m_Effects[0].eKey = kTargetIsDead;
    m_Effects[0].bValue = true;
}
```

55 Actions

Cost per Action

EscapeDanger-**0.5**, UseSmartObjectNodeMounted-**1**, TraverseLinkUncloaked-**1**, TraverseBlockedDoor-**1**, SurveyArea-**1**, SuppressionFireFromCover-**1**, ReloadCovered-**1**, ReactToDanger-**1**, MountVehicle-**1**, MountNodeUncloaked-**1**, IdleOnVehicle-**1**, GotoValidPosition-**1**, GotoNodeOfType-**1**, GotoNode-**1**, FaceNode-**1**, DodgeCovered-**1**, DismountVehicle-**1**, DismountNodeUncloaked-**1**, Charge-**1**, AttackLungeUncloaked-**1**, Animate-**1**, LookAtDisturbance-**1.5**, TraverLink-**2**, SuppressionFire-**2**, KnockDownExplosive-**2**, KnockDownBullet-**2**, InspectDisturbance-**2**, IdleTurret-**2**, Idle-**2**, FollowPlayer-**2**, DodgeRollParanoid-**2**, BlindFireFromCover-**2**, AttackGrenadeFromCover-**2**, UseSmartObjectNode-**3**, LookAtDisturbanceFromView-**3**, LongRecoilHelmetPiercing-**3**, LongRecoilExplosive-**3**, LongRecoilBullet-**3**, Follow-**3**, FlushOutWithGFrenade-**3**, DodgeShuffle-**3**, AttackMeleeUncloaked-**3**, AttackMelee-**3**, GotoTarget-**4**, AttackFromCover-**4**, AttackFromArmoredBounded-**4**, AttackFromArmored-**4**, AttackFromAmbush-**4**, AttackFromView-**4.5**, ReloadCrouch-**5**, AttackCrouch-**5**, AttackTurretCeiling-**6**, Attack-**6**, Attackready-**7**, GotoTargetLost-**8**

Reuse path planning A*

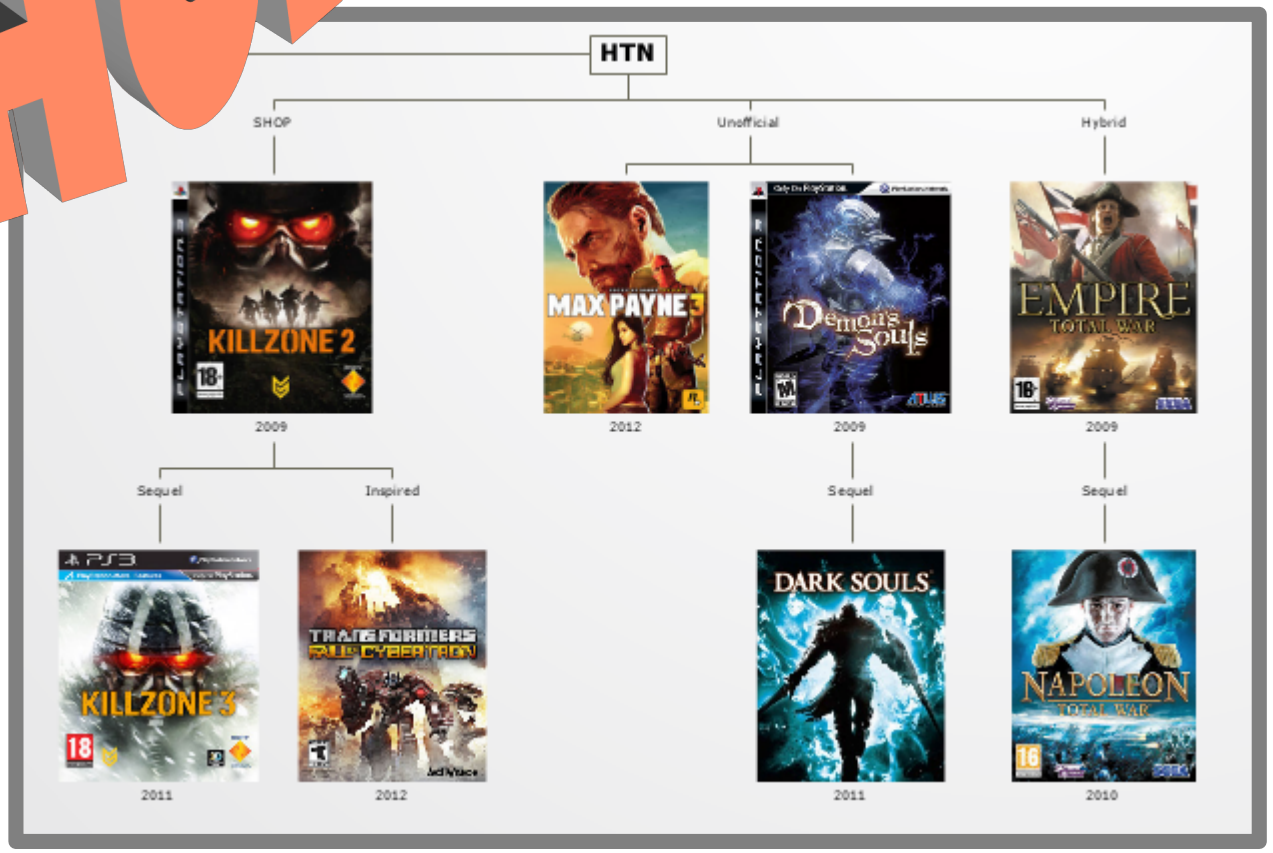
Dana **Nau**, Yue **Cao**, Amnon **Lotem** & Héctor **Muñoz-Avila**

The SHOP Planning System

AI Magazine **22** (3), AAAI Press (Fall 2001)

pages 91-94

SHOP



© Alex Champandard, AIGameDev.com

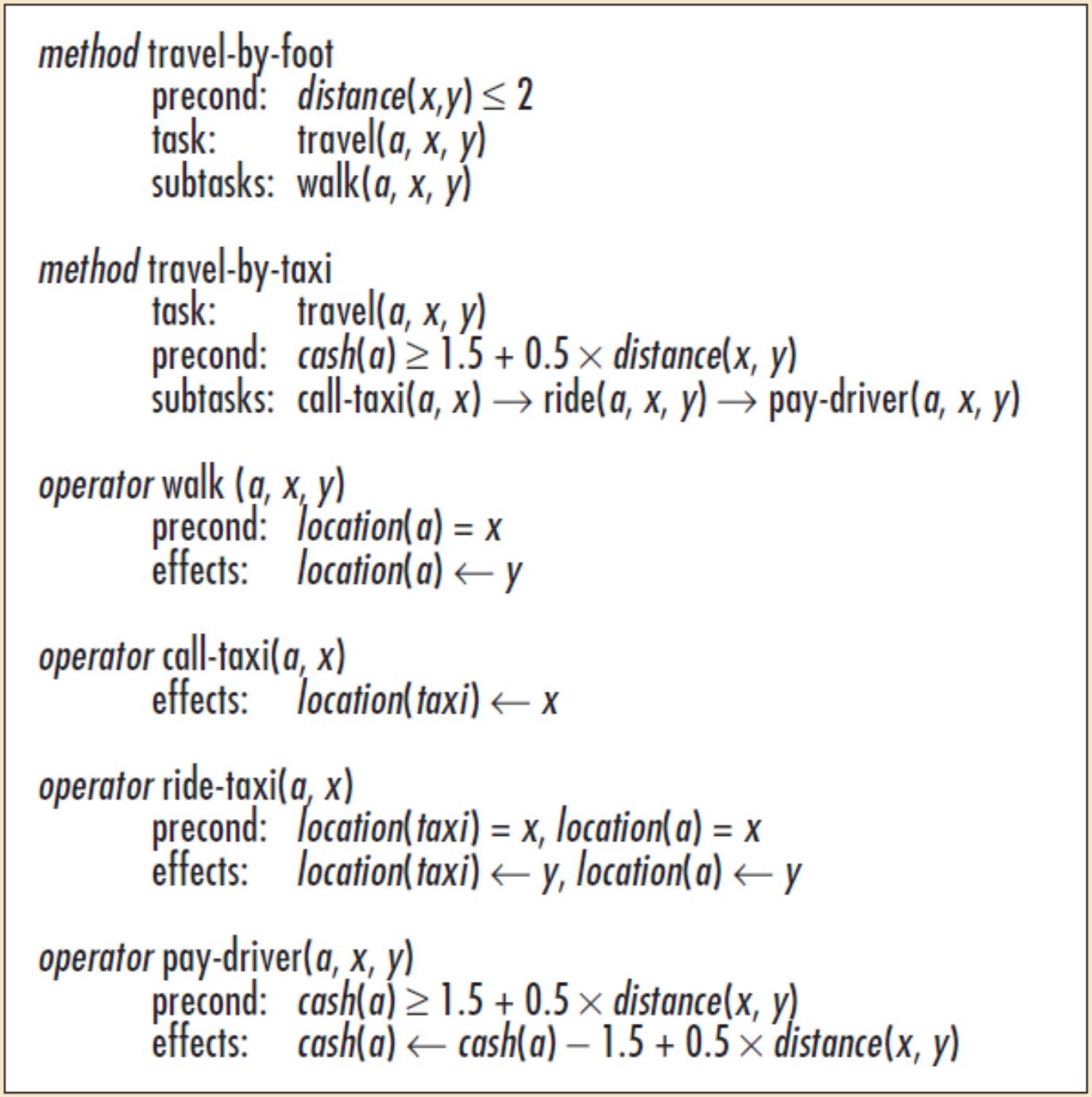


Figure A. Pseudocode representation of a simple travel-planning domain. Left-arrows denote assignments of values to state variables; right-arrows are ordering constraints.

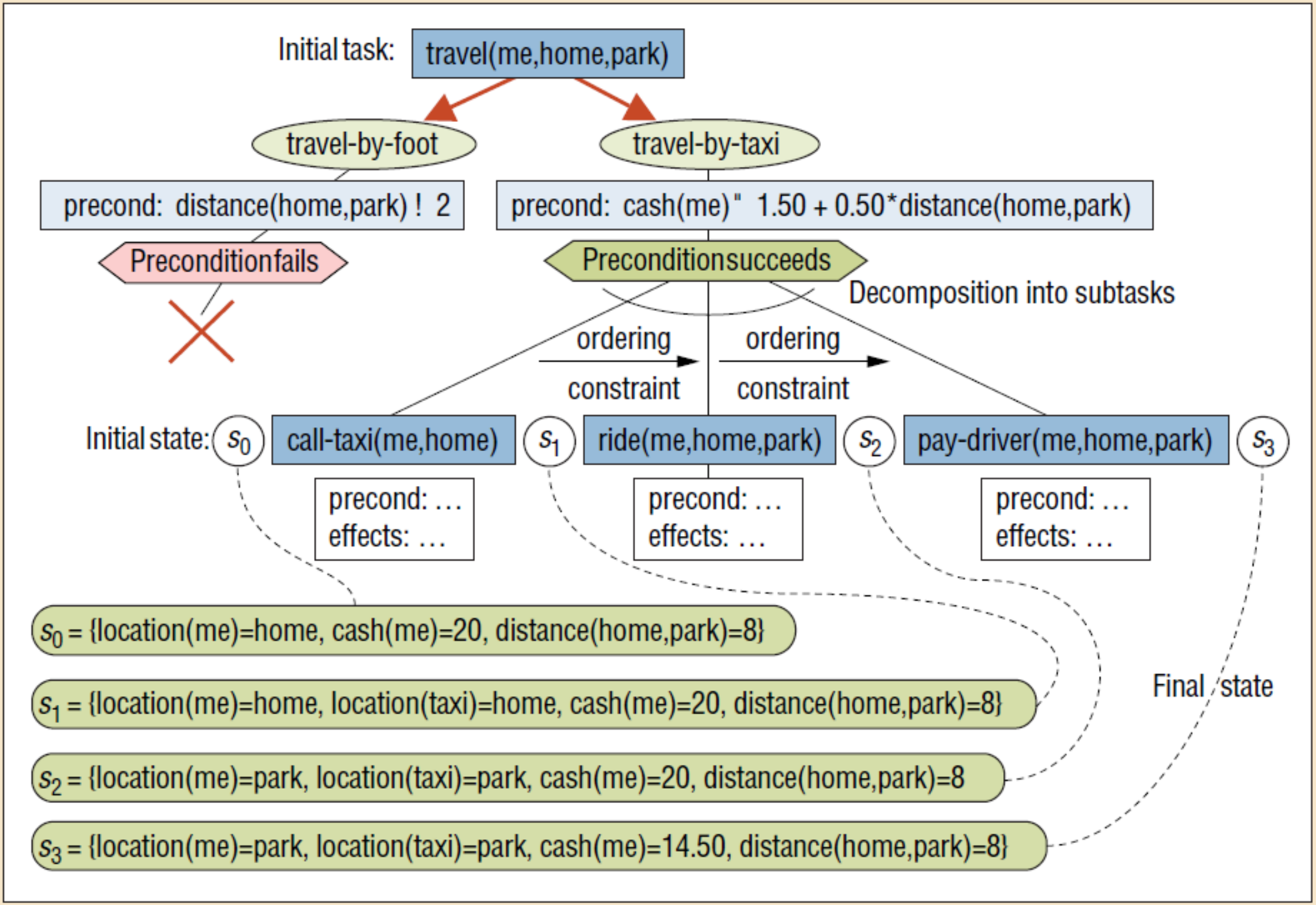
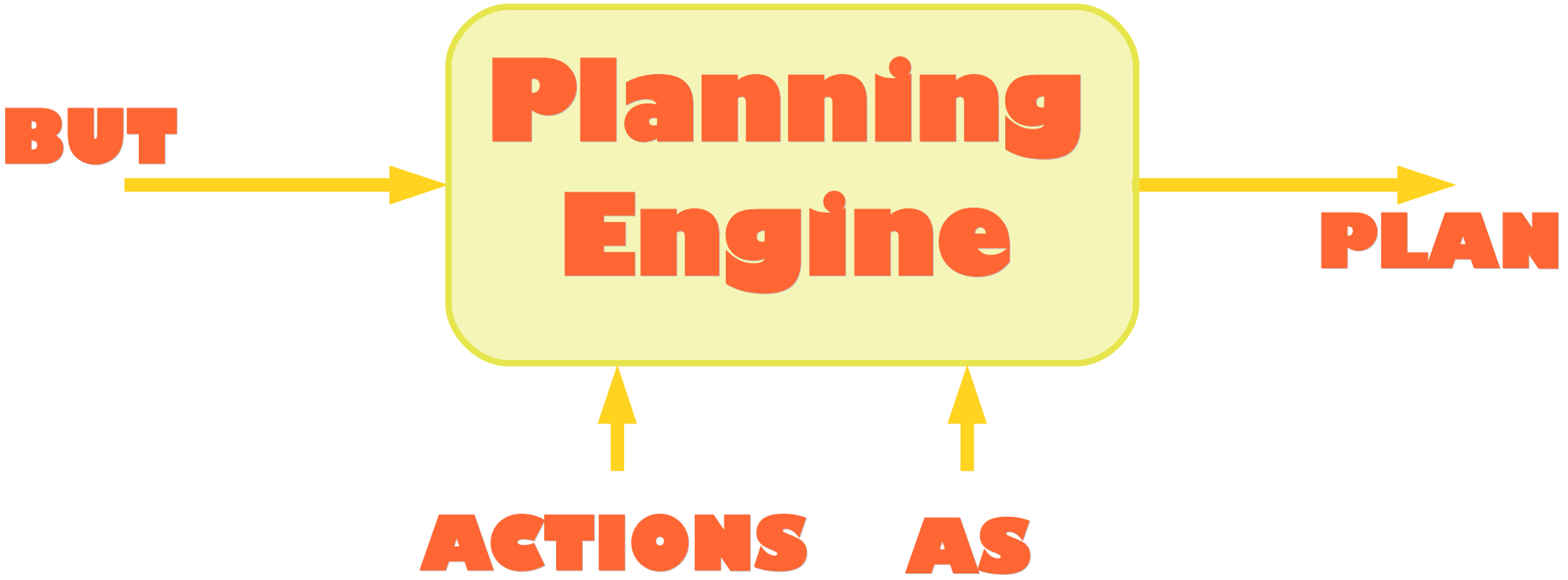
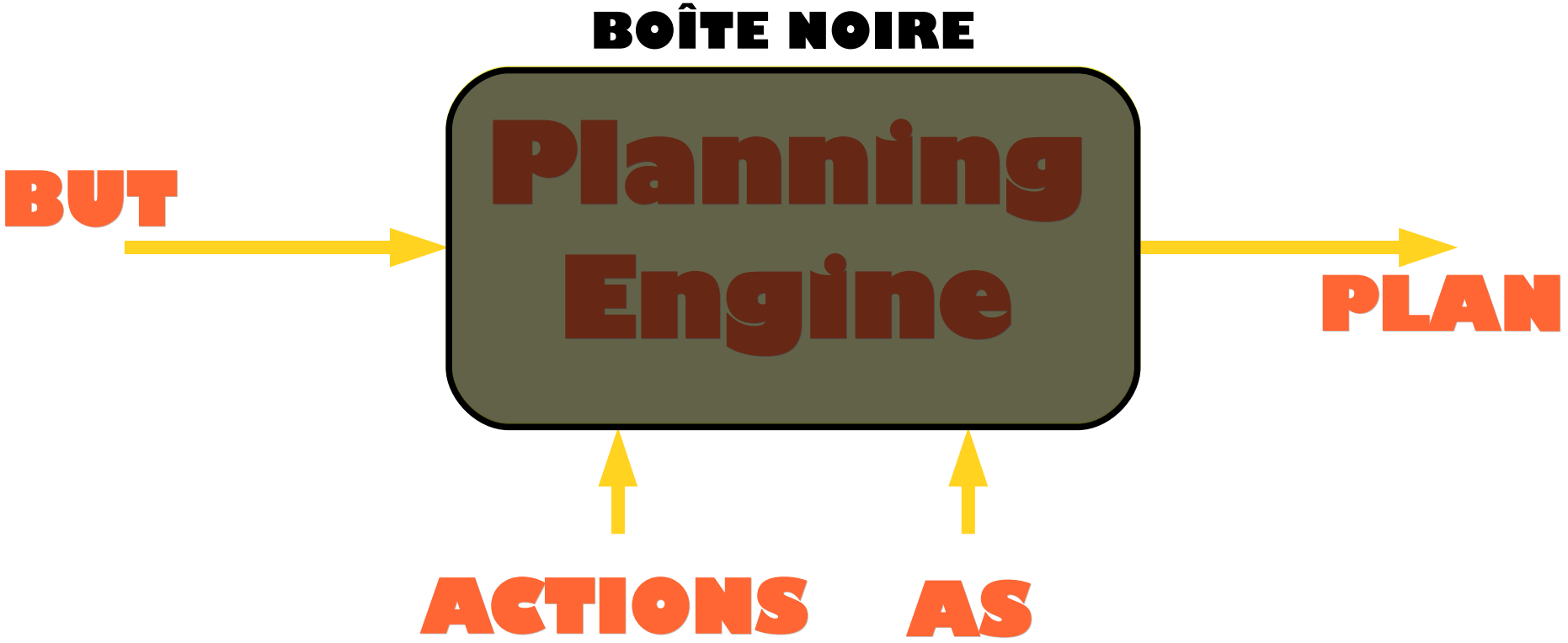
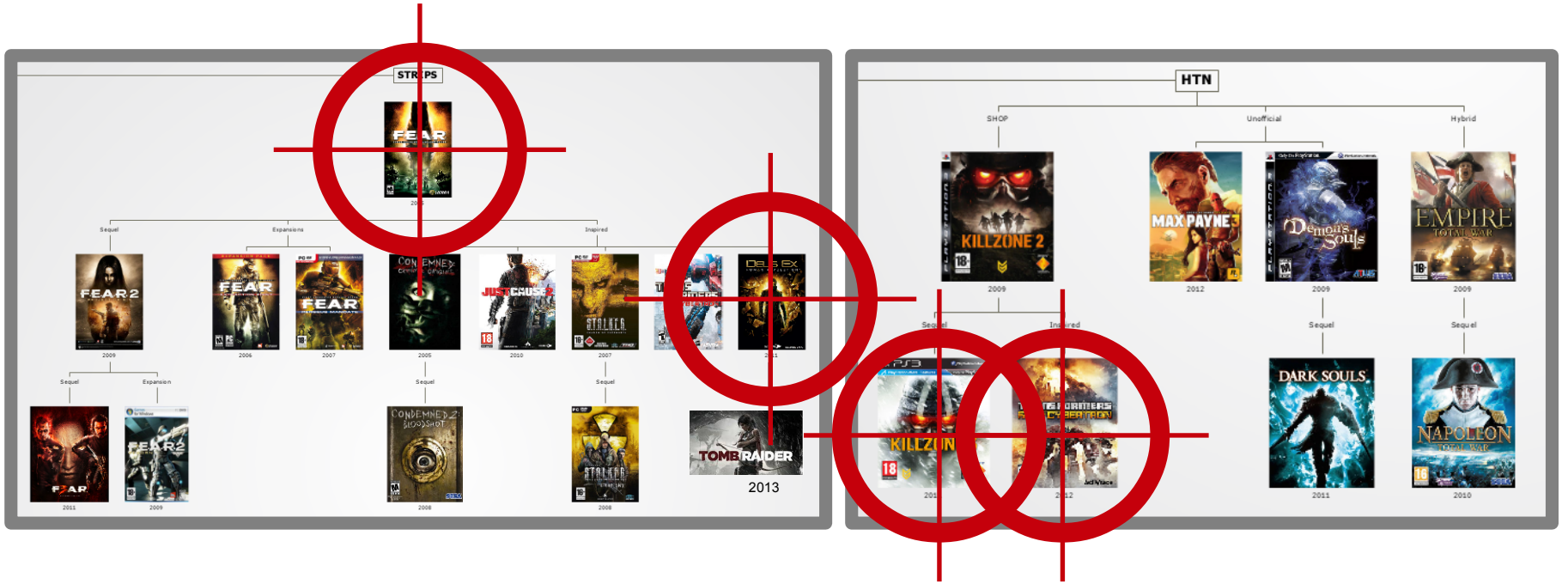


Figure B. Solving a planning problem in the travel-planning domain.


BUT
Planning Engine
PLAN
ACTIONS
AS

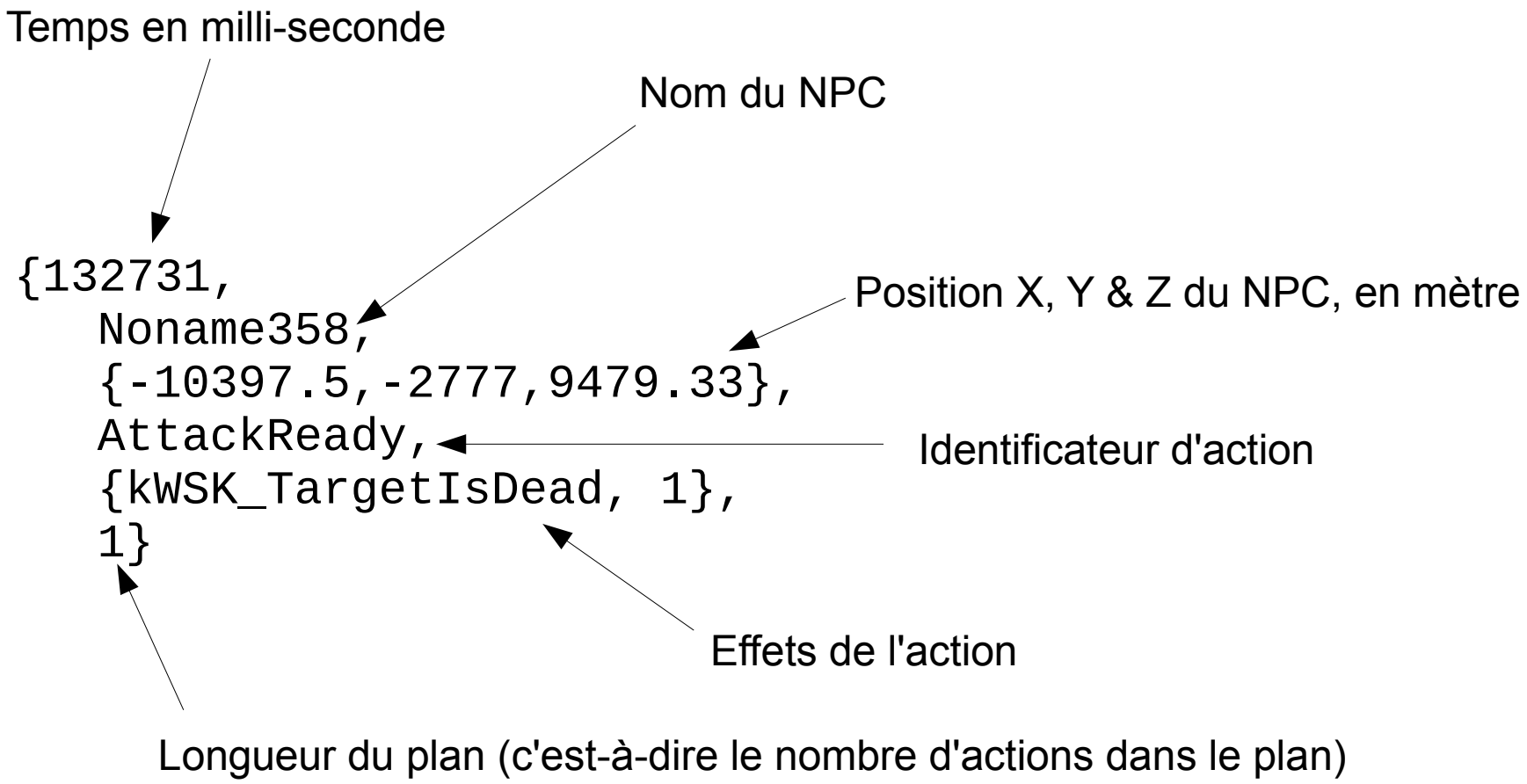

BOÎTE NOIRE
BUT
Planning Engine
PLAN
ACTIONS
AS


STRIPS
Sequel
Expansions
Inspired
Sequel
Expansion
Sequel
Sequel
2013
HTN
SHOP
Unofficial
Hybrid
2009
2012
2009
2009
Sequel
Sequel
2011
2010

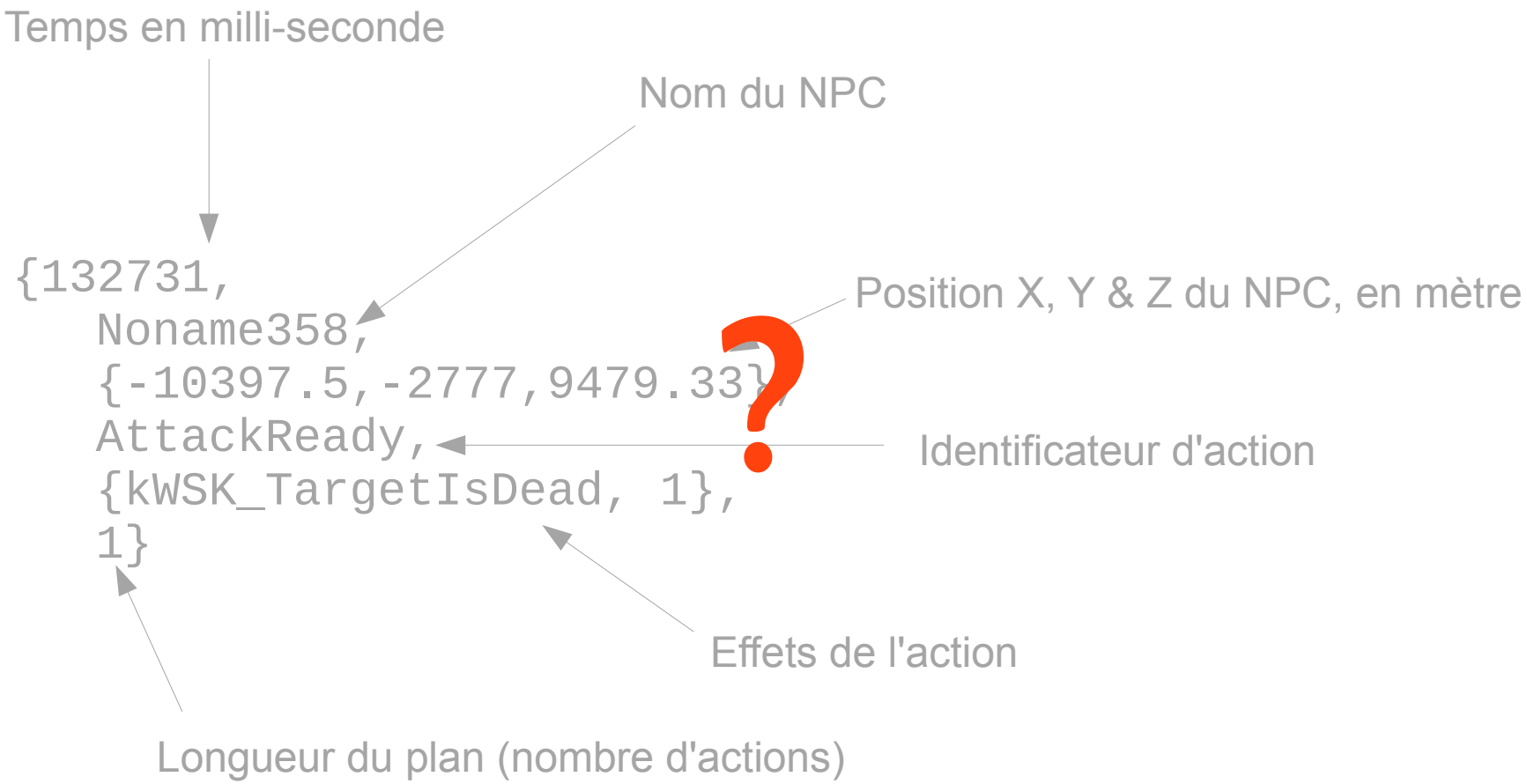
[Monolith, 2005]

F.E.A.R. SDK [Monolith, 2006]
- VC++ 2003

Format des données exportées



Format des données exportées



Format des données exportées

```
{132731,
   Noname358,
   {-10397.5,-2777,9479.33},
   AttackReady,
   {kWSK_TargetIsDead, 1},
   1}
```

BUT oublié !

10^{-3} sec pas assez précis !

[Monolith, 2005]

[Guerrilla Games, 2011]

[High Moon Studios, 2012]

Format des données exportées

```
{132731,
   Noname358,
   {-10397.5,-2777,9479.33},
   AttackReady,
   {kWSK_TargetIsDead, 1},
   1}
```

Toujours pas de buts...

[Killzone 3, Guerrilla Games, 2011]

```
{24717610150,
   BOTBaseTOW6.BOTBase6TOW,
   remember.activeplan,{orderpatrol,18},
   hoversegment,{{1032,1033,1034,1036,1035},any,auto,{},{},{}},
   2}
```

79 800 000 ticks par seconde (12,5313 10^{-9})

Position oubliée !

[Transformers 3, High Moon Studios, 2012]

```
{76681,
   TnHtnAiPawn13944[AICTSoldier],
   {50488.47, 102197.94, 24349.72},
   IgnorantToPrepared,
   {SpecificStanceToClear
      SOMEONE DIDN'T MAKE A DESCRIPTION FOR THIS VALUE},
   1}
```

1/10000 seconde

Données sans valeur...

Que Peut-On Apprendre De Ces Plans ?

VOLUME

FEAR

17 fichiers, 3h45 de jeu, niveau 3 à 11

KZ3

1 fichier, 13min de jeu, deux premières sections du niveau 7

T3

4 fichiers, 35 min de jeu, niveaux inconnus

VOLUME

FEAR

17 fichiers, 3h45 de jeu, niveau 3 à 11
6679 plans – une moyenne d'un plan toutes les 2023 milli-secondes

KZ3

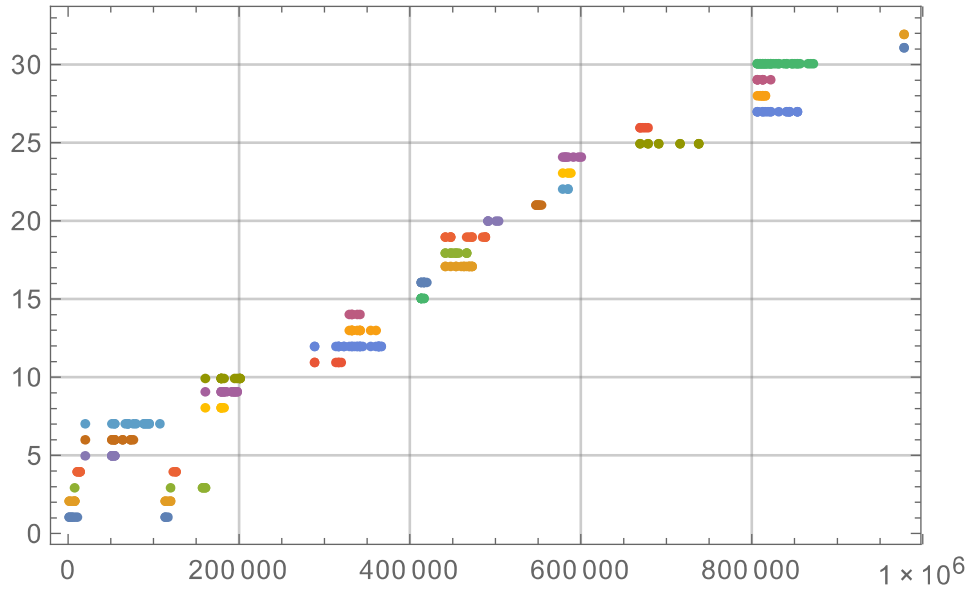
1 fichier, 13min de jeu, deux premières sections du niveau 7
2349 plans – 1 plan toutes les 333 milli-secondes

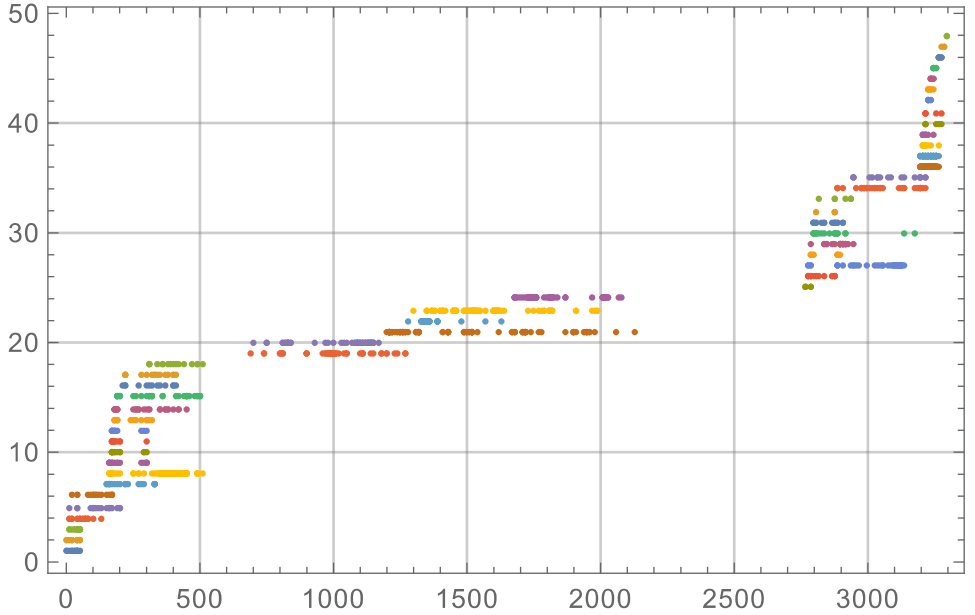
T3

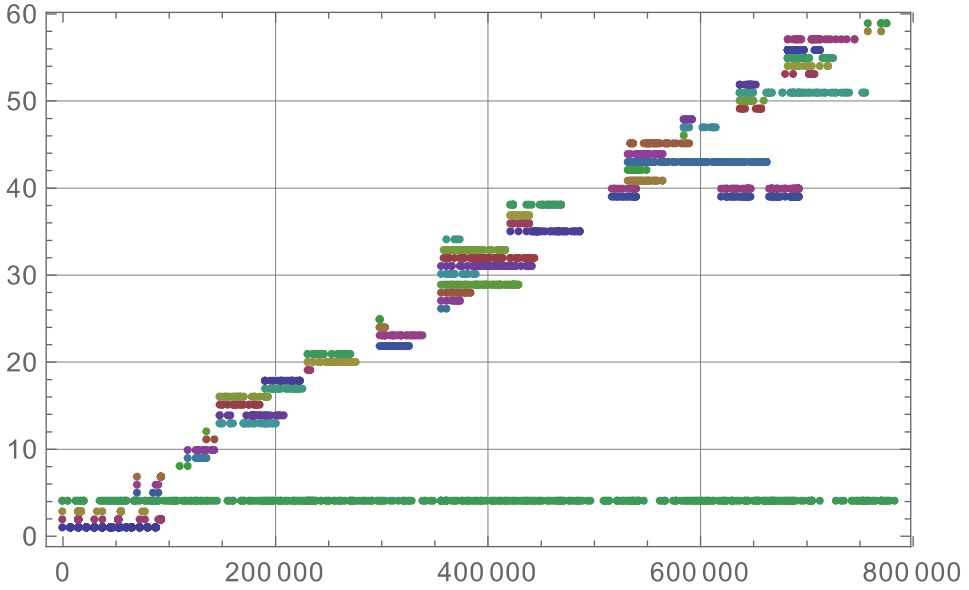
4 fichiers, 35 min de jeu, niveaux inconnus
8751 plans – 1 plan toutes les 243 milli-secondes

Métriques

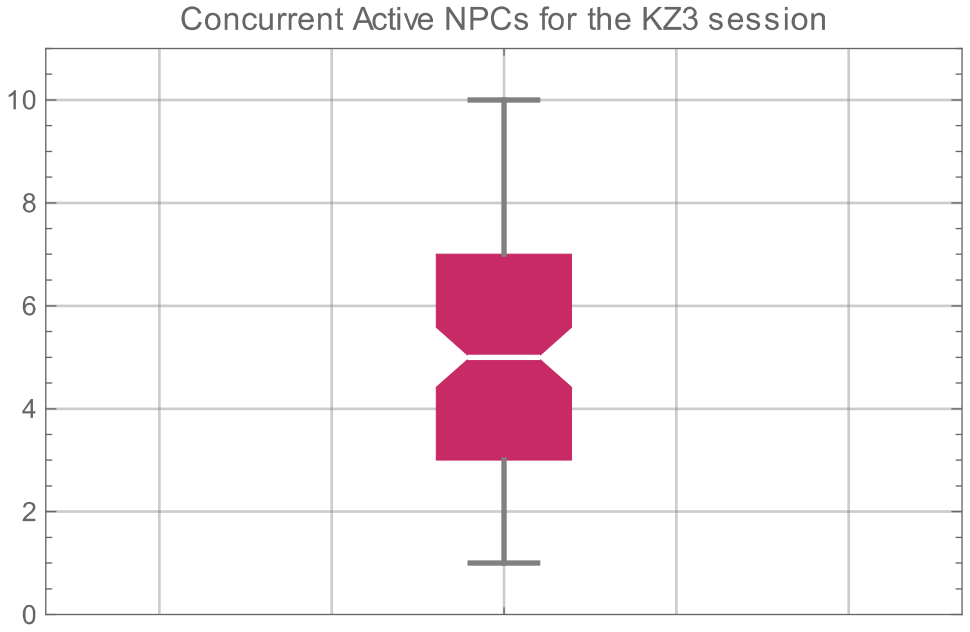
CASTING

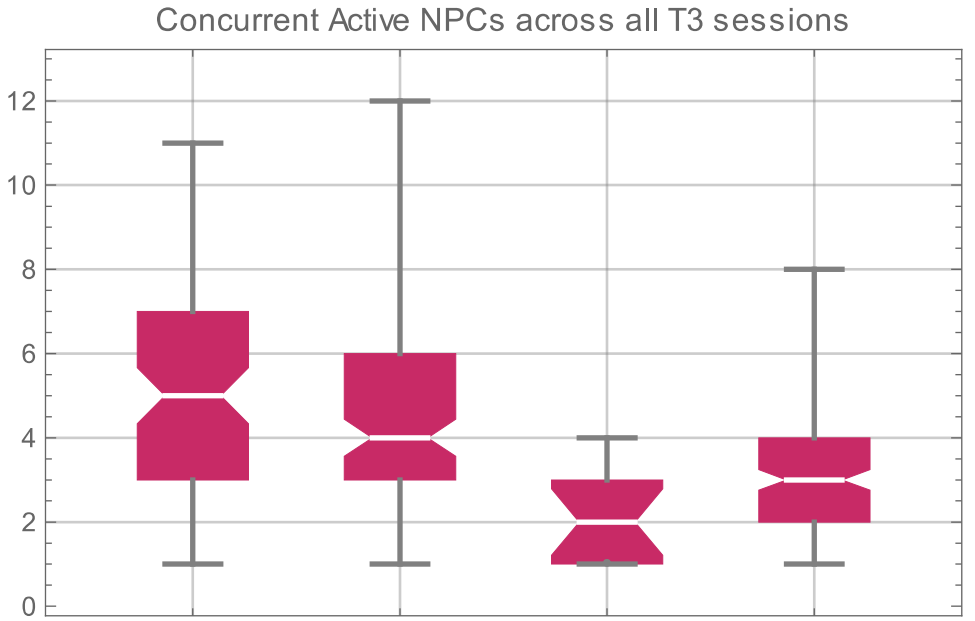

30
25
20
15
10
5
0
0
200 000
400 000
600 000
800 000
1 × 10^6

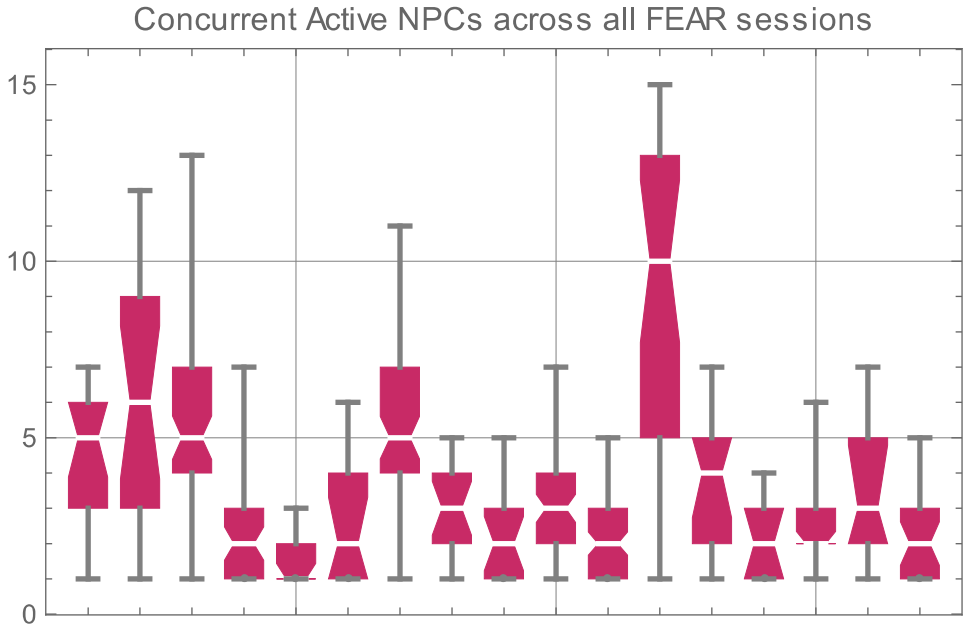

50
40
30
20
10
0
0
500
1000
1500
2000
2500
3000

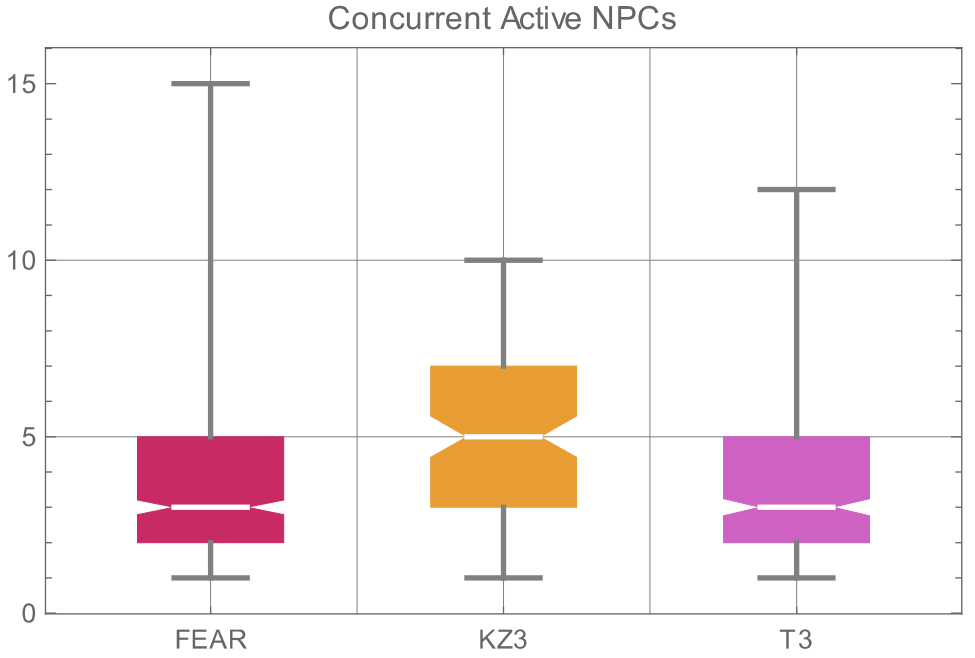

60
50
40
30
20
10
0
0
200 000
400 000
600 000
800 000

Concurrent Active NPCs



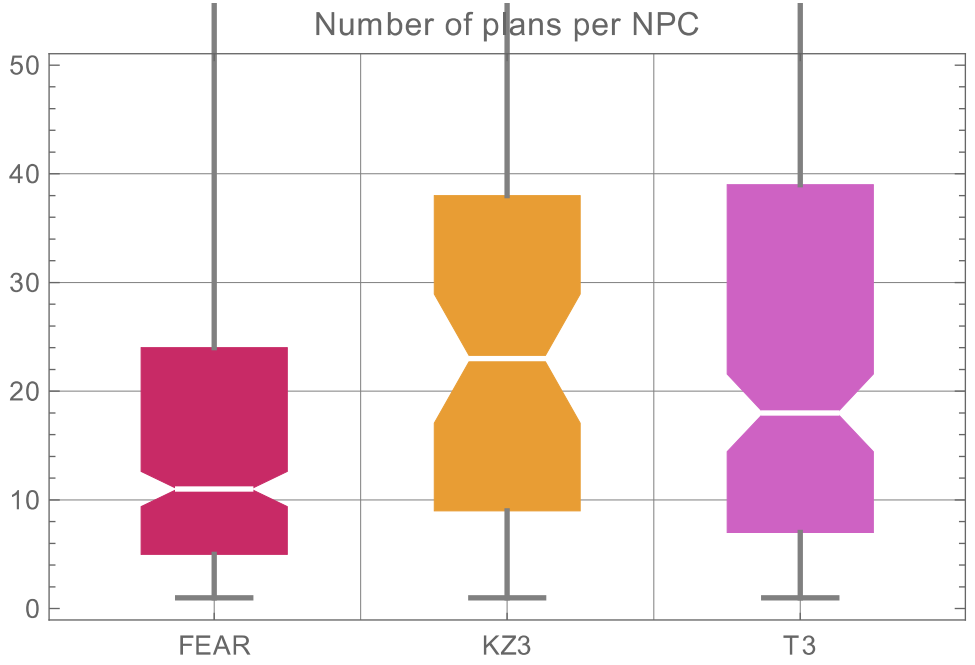






Number of Plans per NPC

```
{{1,5,11,24,393},
 {1,9,23,38,463},
 {1,7,18,39,1985}}
```

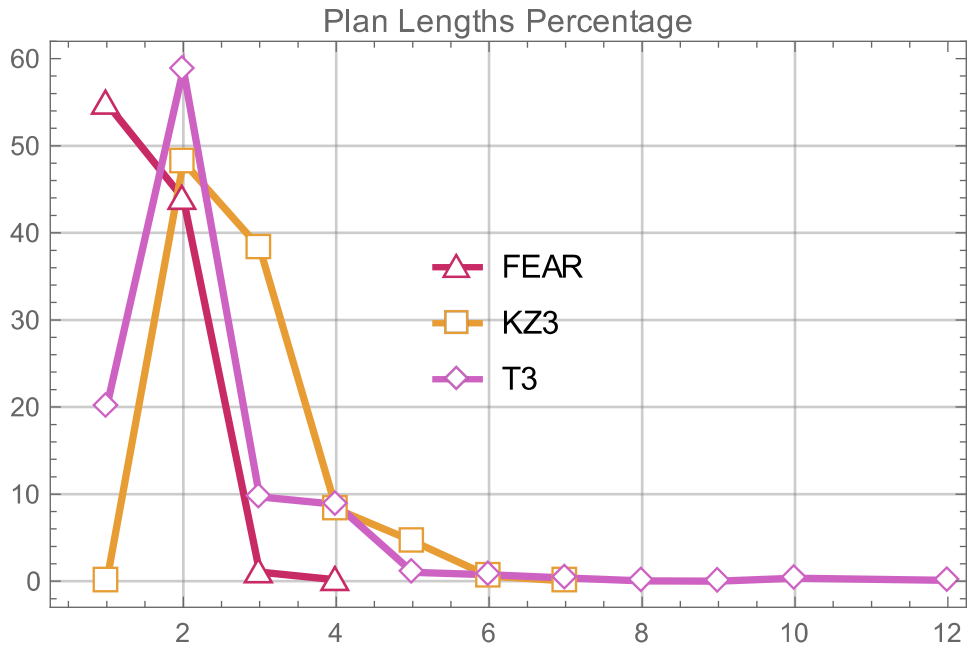



```
{{{ExitFettel,1},{Rat02,41}}},
{{{ExitFettel,2},{Rat02,67}}},
{{{HSol01,2},{Rat02,393}}},
{{{Dead[Guy],1},{noname1946,33}}},
{{{runningMAN,1},{Assassin[Corridor[Ass00]],23}}},
{{{Admin[Sec[MapesGate]].CAI,3},{Arrival[Lounge[S04]].CAI,48}}},
{{{132[crow01],1},{noname2311,96}}},
{{{noname2272,4},{noname2279,40}}},
{{{noname2281,3},{noname2311,47}}},
{{{Holiday,2},{Delta01,380}}},
{{{kitty[scripted],1},{noname2620,80}}},
{{{ATC[Rush[2]],2},{4[S02],96}}},
{{{mapes,2},{noname2929,35}}},
{{{Loading[S02],2},{Loading[S01],29}}},
{{{Lab[2[S01]],2},{TurnOn[Hall[S06]].CAI,30}}},
{{{Trouble[Turret05].Weapon,1},{Trouble[Turret02].Weapon,77}}},
{{{Mapes[2],2},{6[Backup[S02]],49}}}

{{{HGHwave2SMG.HGHslopeReinforcements33,1},{Rico,463}}}

{{{TnHtnAiPawn7704[AICTShotgunner],2},{TnHtnAiPawn10954[AICTSoldier],119}}},
{{{TnHtnAiPawn3612[AICTSoldier],1},{TnHtnAiPawn9938[AICTLaserbeak],1985}}},
{{{TnHtnAiPawn12027[AICTSoldier],24},{TnHtnAiPawn16368[AICTMegatron],316}}},
{{{TnHtnAiPawn9513[AICTSniper],2},{TnHtnAiPawn4236[AICTSwarmer],276}}}
```

Longueur des Plans



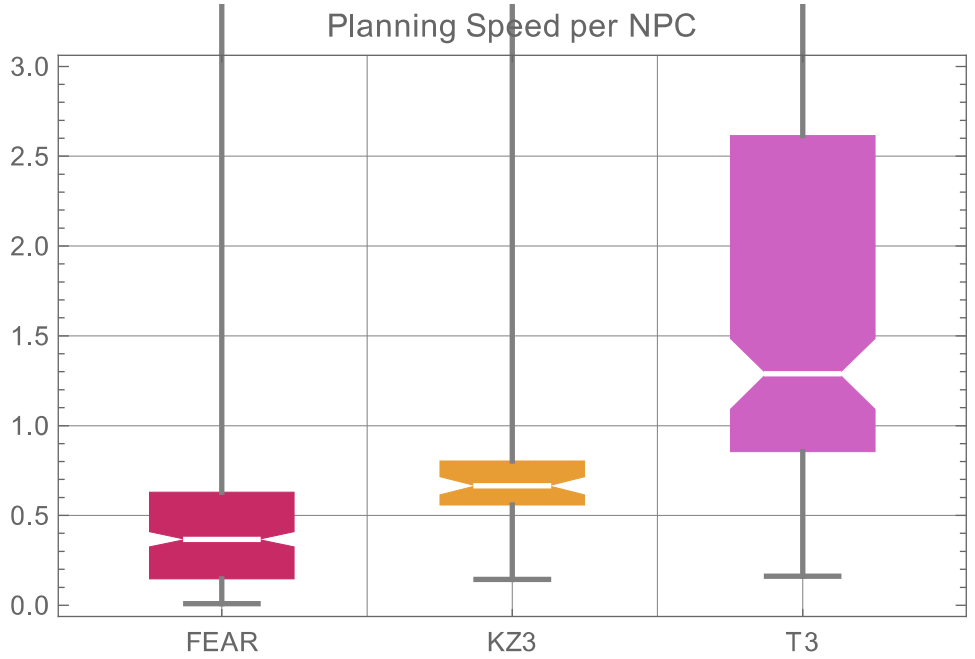
Les plans les plus...

...COURTS sont pour les animaux, les drônes et les tourelles

...LONGS sont pour les snipers.

Vitesse de Planification

(nombre de plans par NPC par seconde pendant que ce NPC est actif)

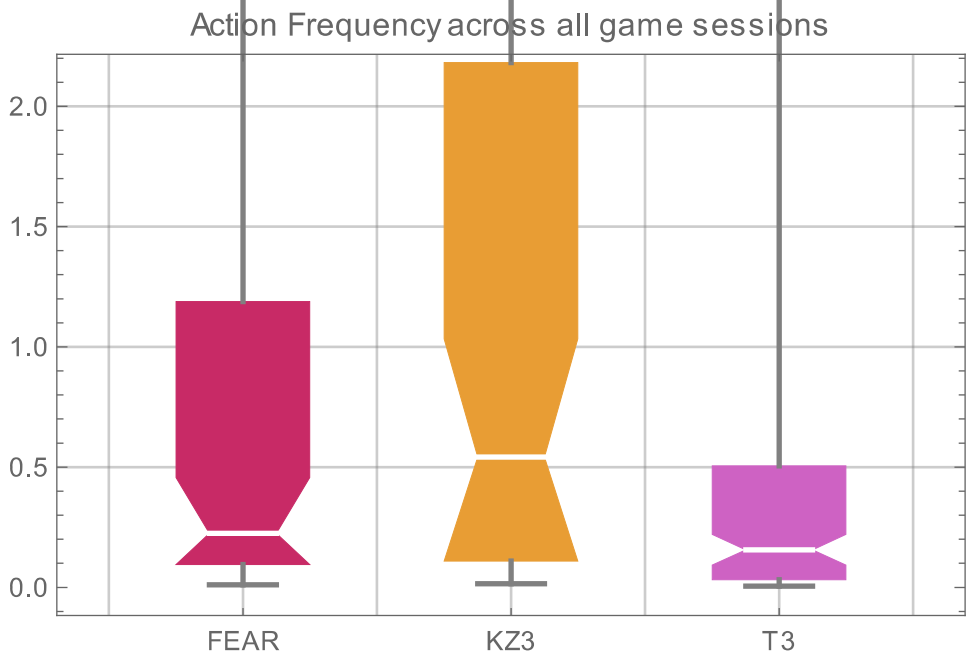


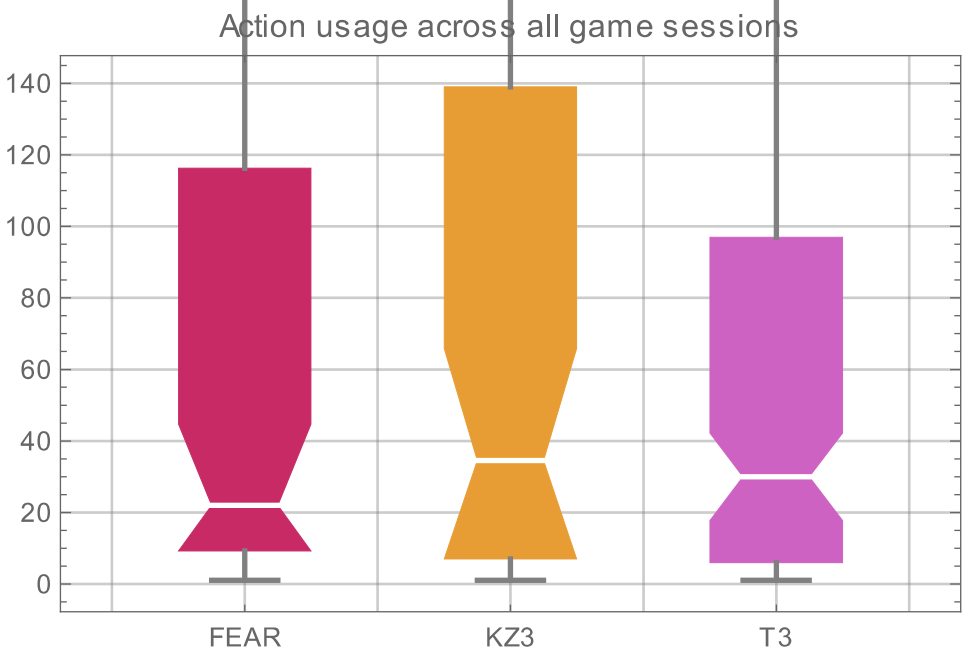

```
{{0.009, 0.147, 0.367, 0.630,  8.547},
 {0.144, 0.559, 0.665, 0.803,  3.380},
 {0.161, 0.854, 1.288, 2.614, 59.880}}
```

MIN	25%	50%	75%	MAX

Utilisation des actions

{{55,9785,6679},{44,6371,2349},{137,19187,8751}}
{nb actions, nb occurrences, nb plans}

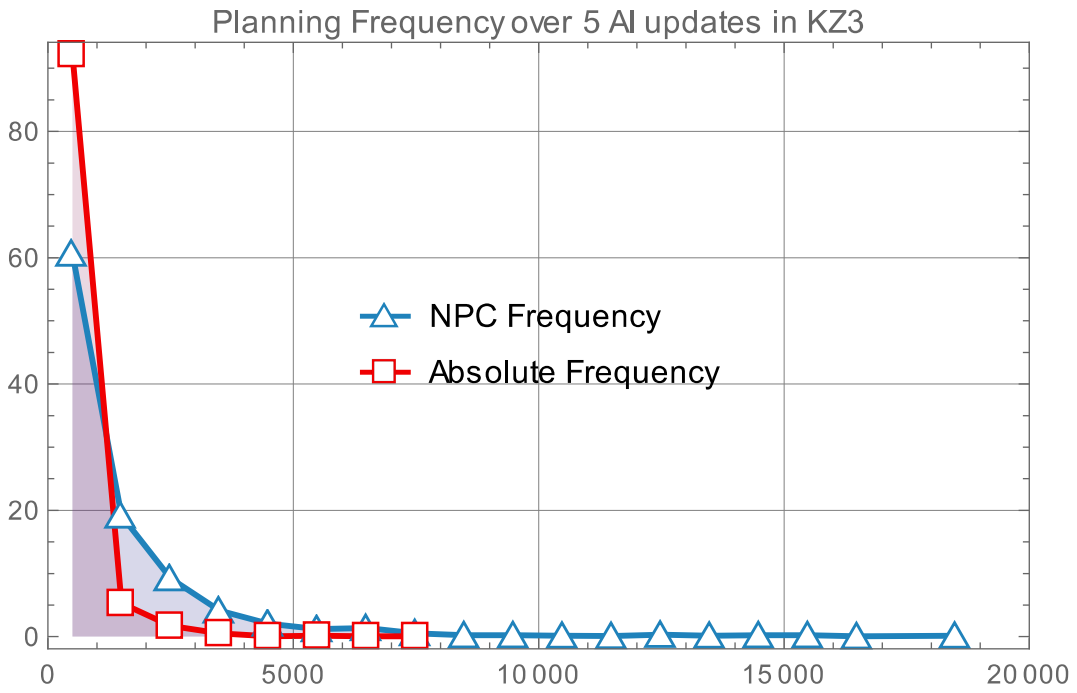




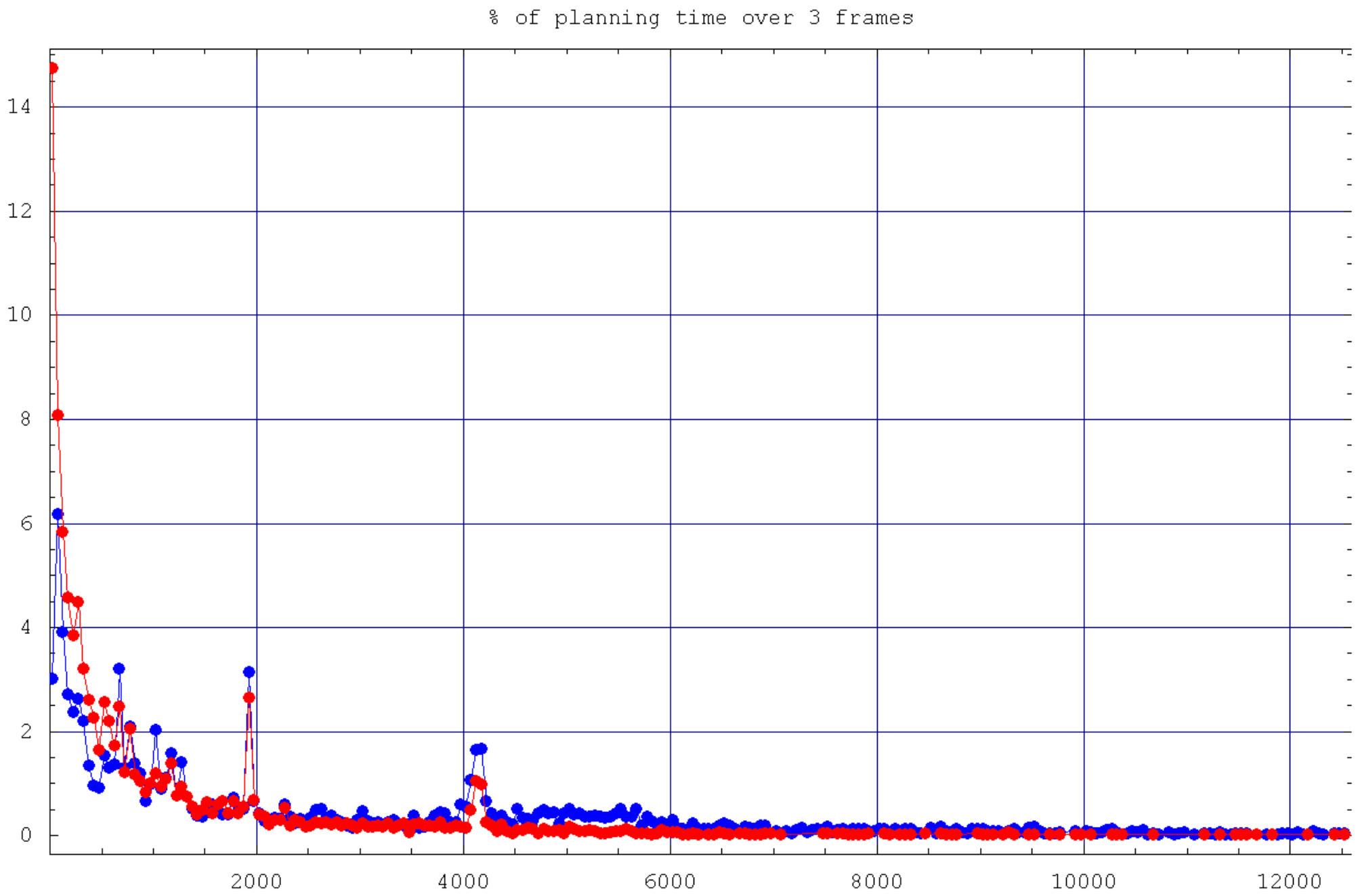
{{1,9,22,119,2566},{1,6,33,132,2355},{1,6,30,97,2305}}

Invariants

Fréquence de planification
(allure hyperbolique)

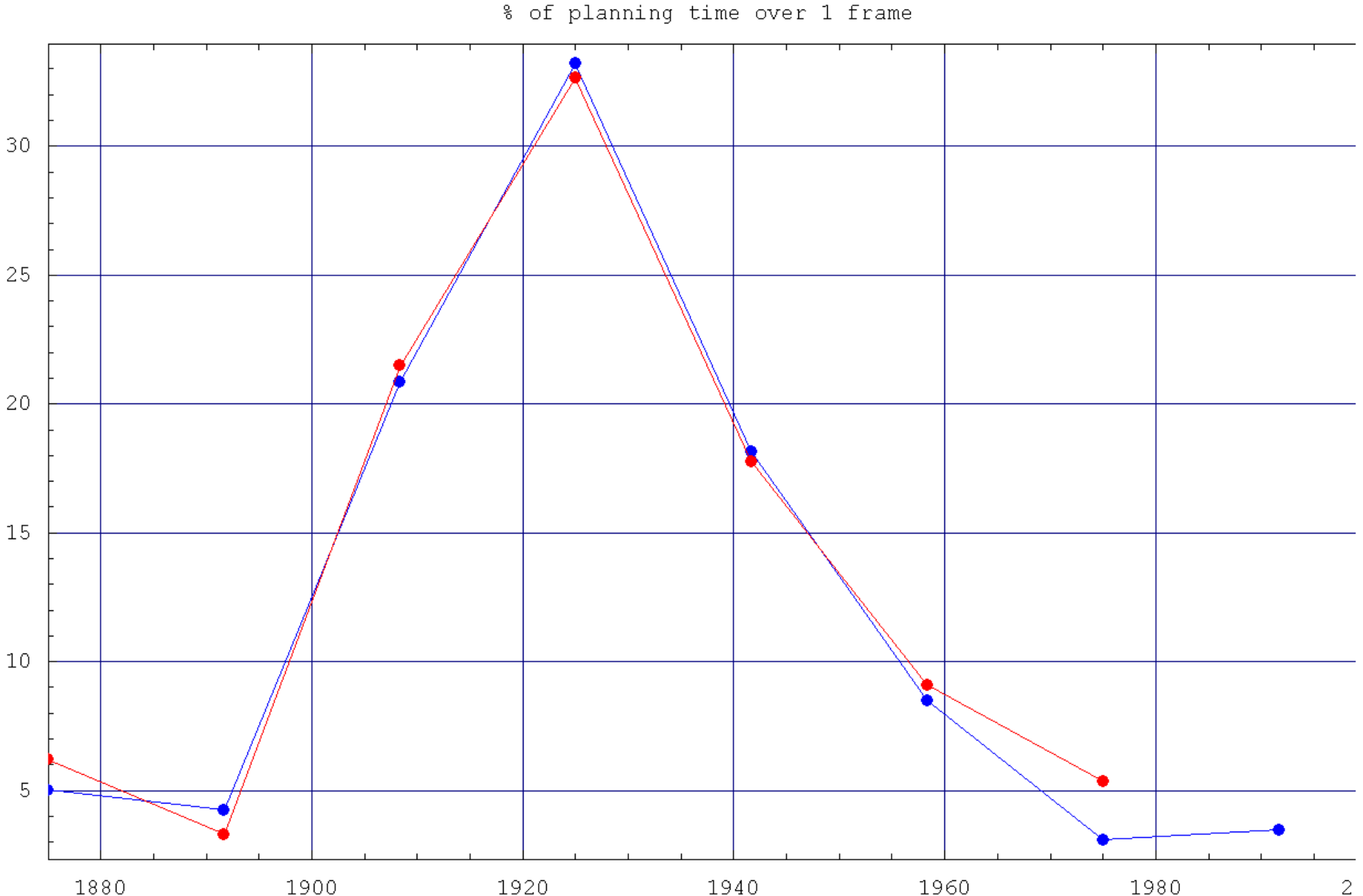


F.E.A.R. Planning Frequency



1er Pic

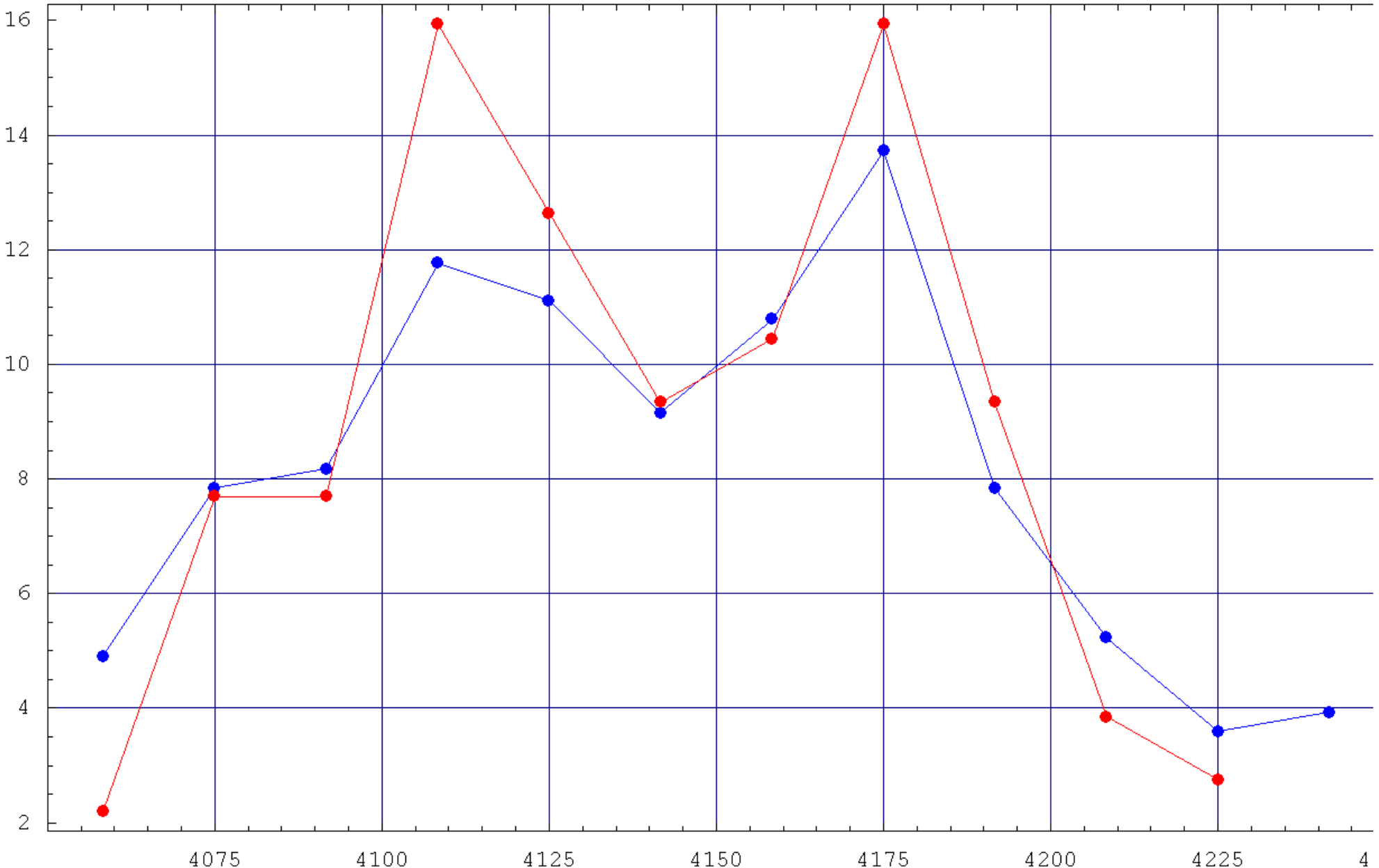
$\{\{\text{Delta01}, \{\text{UseSmartObjectNode}\}\}, \frac{73}{87}\}$



2ème Pic

$$\left\{\{\text{Delta01}, \{\text{Animate}\}\}, \frac{139}{162}\right\}$$

% of planning time over 1 frame



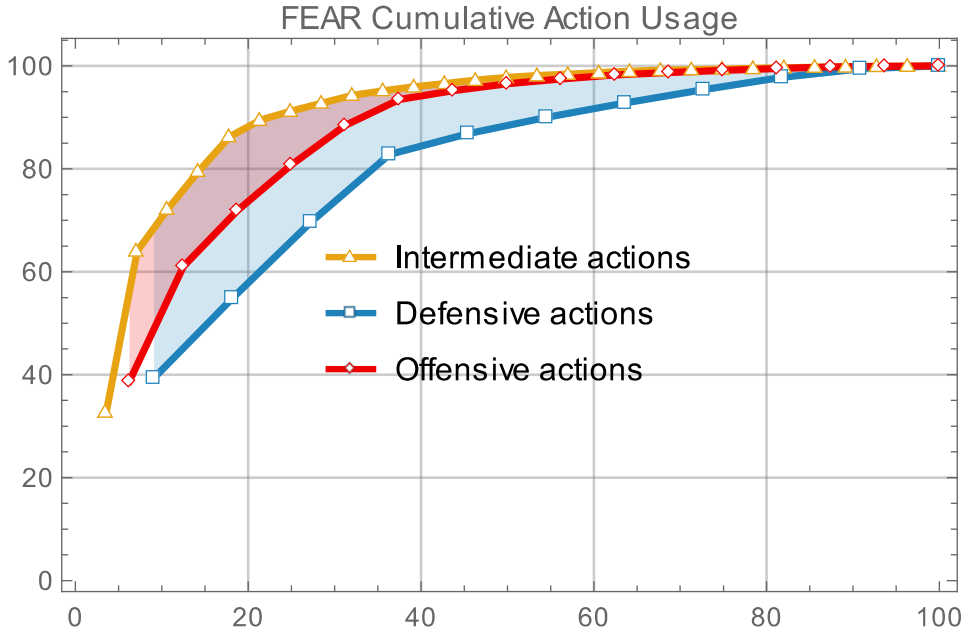

Delta01
Holiday
Delta01
KWSK_AnimLooped: -1
KWSK_AnimPlayed: -1
KWSK_AtNode: None
KWSK_AtNodeType: Invalid
KWSK_AtTargetPos: 0
KWSK_CoverStatus: None
KWSK_DisturbanceExists: 0
KWSK_Idling: 0
KWSK_MountedObject: None
KWSK_PositionIsValid: 1
KWSK_RidingVehicle: None
KWSK_ReactedToWorldStateEvent: 0
KWSK_SurveyedArea: 0
KWSK_TargetIsAimingAtMe: 0
KWSK_TargetIsLookingAtMe: 0
KWSK_TargetIsDead: 0
KWSK_TargetIsFlushedOut: 0
KWSK_TargetIsSuppressed: 0
KWSK_TraversedLink: None
KWSK_UsingObject: None
KWSK_WeaponArmed: 1
KWSK_WeaponLoaded: 1
105
0
12/88

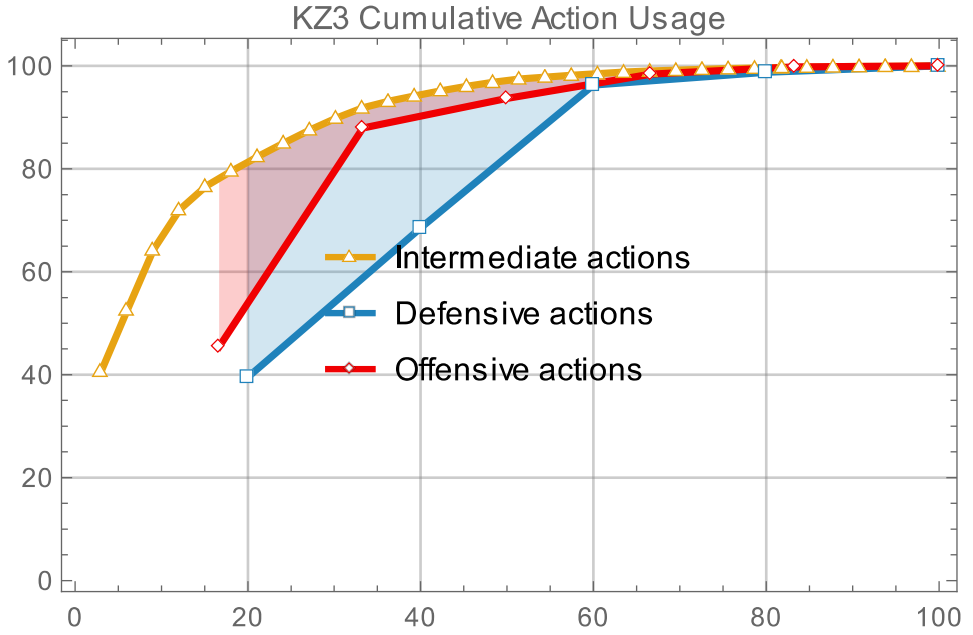
Plans Fixes

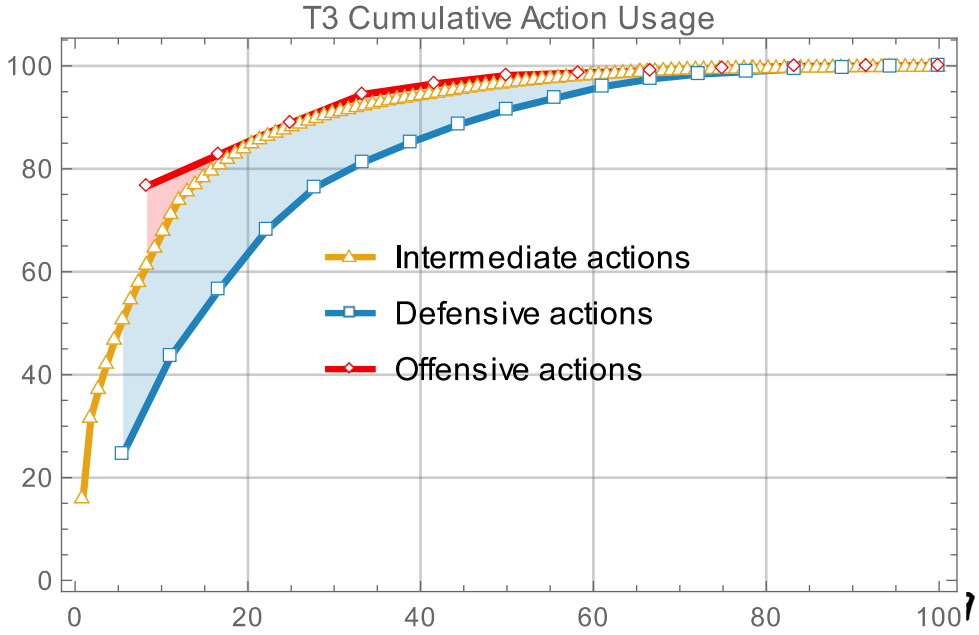
	Séquence fixe d'actions	Fréquence
FEAR	{GotoNodeOfType, AttackFromView}	13/6679
	{SurveyArea, GotoTarget, InspectDisturbance}	51/6679
	{GotoNode, UseSmartObjectNode}	2355/6679
KZ3	{remember.activeplan, staggerfire}	1/87
	{remember.activeplan, lapapeekatentity}	4/261
	{remember.activeplan, scanwaypointlistlapa}	115/2349
T3	{StartStrafing, StrafingIdle}	199/8751
	{Asleep, Idle}	626/8751
	{ChooseNewAttackPoint, FlyTo}	1838/8751

Principe de Pareto

(20% des causes expliquent 80% des résultats)







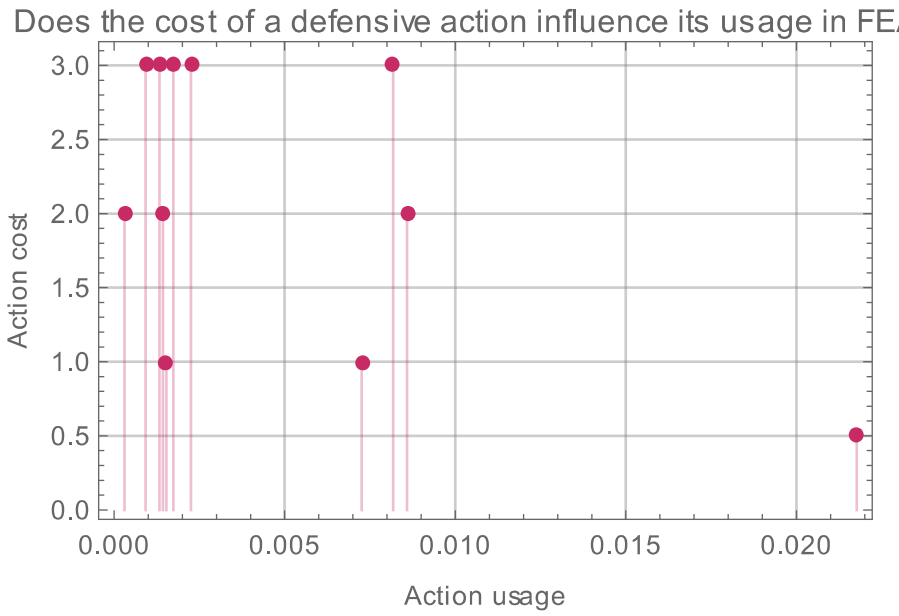
	Defensive	Offensive	Intermediate
FEAR	11	16	28
KZ3	5	6	33
T3	18	12	107

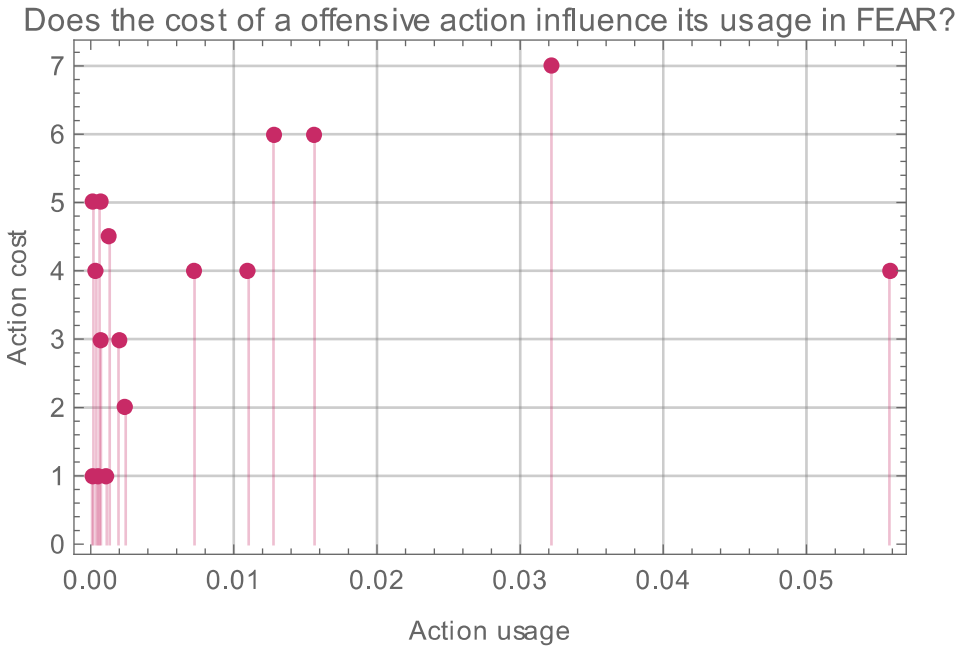
Le composant de planification de F.E.A.R. utilise un coût pour accélérer la recherche d'un plan.

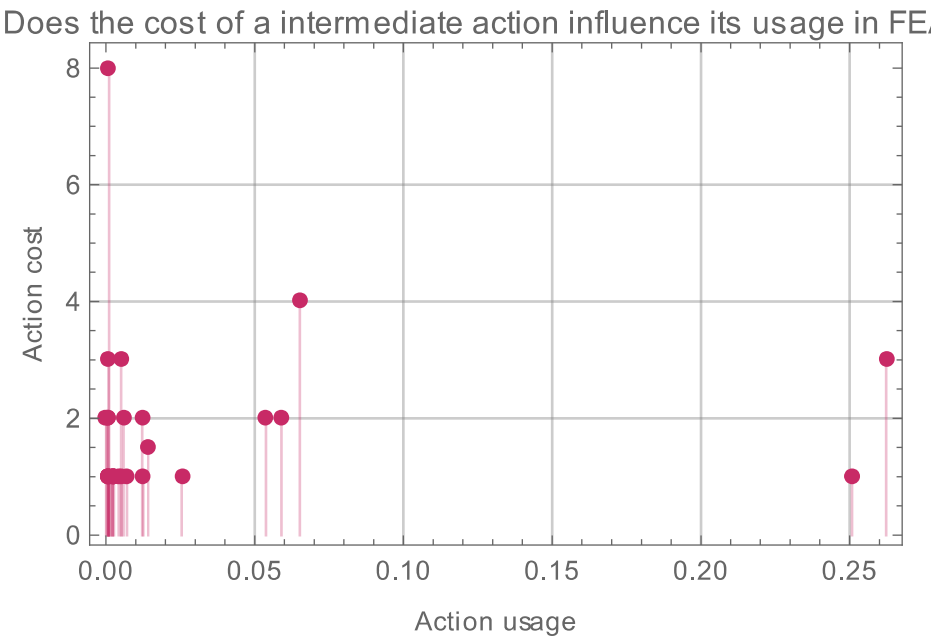
Ce coût a-t-il des effets sur l'usage des actions ?

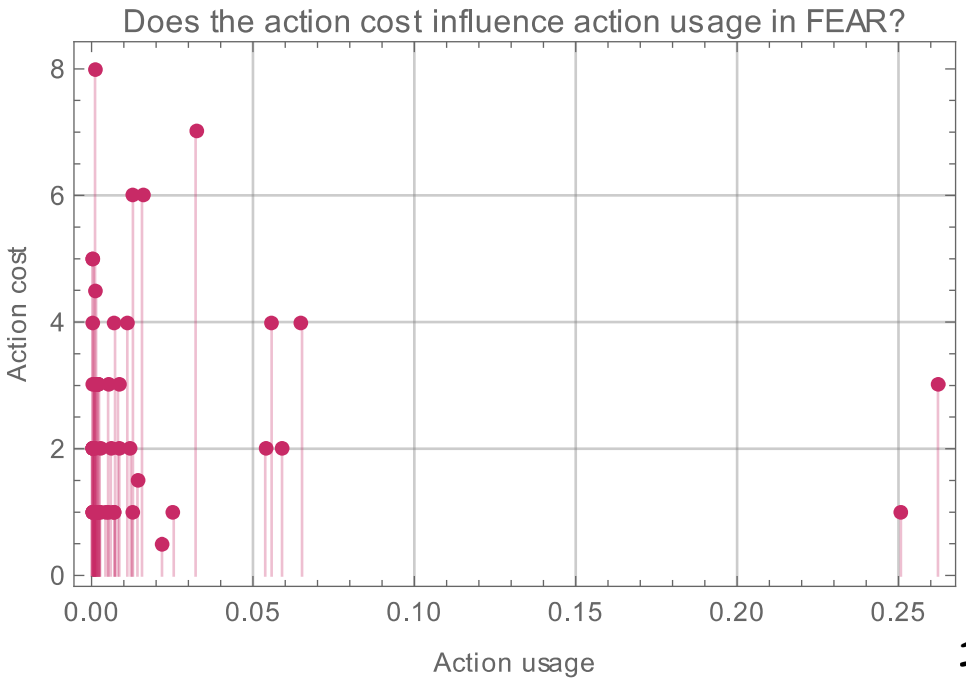
C'est-à-dire,
les actions les plus coûteuses
sont-elles moins utilisées que
les actions les moins coûteuses ?

Usages et coûts des 55 actions dans F.E.A.R.









Usages et coûts des 55 actions dans F.E.A.R.

S'il y a des actions plus utilisées que d'autres
(principe de Pareto)
C'est un effet du game design
Ce n'est pas un effet des coûts fixes des actions

Rétro-Ingénierie

Quelques questions en guise de conclusion...

1. À quelle fréquence le planificateur change-t-il de NPC ?
2. La planification est-elle la bonne solution pour si peu / tant de NPCs ?
3. Que fait-on des NPCs qui
   - n'utilisent presque pas la planification ?
   - monopolisent la planification ?
4. Quels NPCs doivent profiter de la vitesse de planification ?
5. Ne devrait-on pas mémoriser les plans fixes dans une table de hashing, et les y classer selon leur situation initiale et leur situation finale ?
6. Quel est le niveau d'utilisation du planificateur ? (c'est-à-dire, trop utilisé ou bien pas assez ?)
7. Lorsqu'une grande vitesse de planification est disponible,
   - plus de NPCs à l'écran ?
   - plus de plans par NPC ?
8. Que fait-on des actions qui
   - n'apparaissent presque pas dans les plans ?
   - monopolisent la planification ?

MERCI !