Introduction

The objective of this study is to improve Artificial Intelligence Planning systems which are used as decision-making components in video-games.

We here realize this study with data files, henceforth named session files, from the video-game F.E.A.R. (© Monolith, 2005). The decision-making component in F.E.A.R. is an Artificial Intelligence Planning algorithm working from the goal towards the initial situation; the component, looking for the cheapest plan, applies actions in a backward manner according to their costs (see the appendix for these costs) as described in [3,4], hence the heuristic-based backward chaining label for Goal-Oriented Action Planning. The session files are produced thanks to a modified version of the game which we produced from the “F.E.A.R. Public Tools 1.08” Software Development Kit made available to the public by Monolith.

We observe the following facts about the implemented Goal-Oriented Action Planner in F.E.A.R.: (i) plans are never longer than 4 actions, (ii) there are actions which always appear together (in plans of length 2 and of length 3), (iii) the curve of the need for planning (time in milliseconds between two successive calls for planning on the x-axis and number of calls for planning on the x-axis) follows an hyperbola, with exceptions around 2 and 4 seconds (these two peaks are analyzed below), (iv) although the need for planning is more important when attacking than when defending, it is largely due to intermediate actions generated in plans in order to achieve the watching of an area (moving, inspecting a disturbance) or else animation purposes and (v) 80% of the planning activity happens for 20% of all the actions (either offensive, defensive or else intermediate) recorded in the session files.

This study is organized as follows: the next section gathers the Mathematica code developed to process session files; then after loading and explaining their content, a small statistical study of several session files is presented. We end with various plots about the need for planning: How long before the next call for planning? What are the actions in the next plan? That is, respectively, When is planning needed again? What actions planning needs the most? The real numbers implementing the cost of actions used during the heuristic search
for a plan are given in the first appendix as well as the casting of NPCs in the second appendix. 
This file has been produced with *Mathematica* 9.

## Routines

### Constants

We here define several numerical values (and 3 strings) used as constants in the functions of the next section:

1. $\text{SecondTimeRatio} = 60$
2. $\text{TicksPerSecond} = 1000$
3. $\text{TicksPerMilliSecond} = 1$
4. $\text{TicksPerMinute} = \text{N}[\text{TicksPerSecond} \times \text{SecondTimeRatio}]$
5. $\text{AIUpdateFrequency} = \text{TicksPerSecond} / \text{SecondTimeRatio}$
6. $\text{FileNamePattern} = "\text{Session } *.txt"$
7. $\text{EnfOfFileName} = "\text{F.E.A.R. plan lengths.txt}"$
8. $\text{PathToSessionFiles} = \text{"usr\code\Mathematica\AI Planning Analytics\Planning Analytics (Data & Code)\F.E.A.R. Planning Data\Session Files\"}$
9. $\text{FEAROldActionIndex} = 3; \text{KillZone3ActionIndex} = 3$
   $\text{FEARNewActionIndex} = 4; \text{Transformers3ActionIndex} = 4$
10. $\text{FirstActionIndex} = \text{FEAROldActionIndex}$
11. $\text{FEARSessionSymbol} = \text{"s"}$
12. $\text{SessionSymbol} = \text{FEARSessionSymbol}$

### Functions

Here are the *Mathematica* functions defined in order to process F.E.A.R. session files:

13. $\text{UnsortedUnion}[x_\_] := \text{Module}[\{f\}, f[y_] := (f[y] = \text{Sequence}[\{y\}]; f/@x]$  
14. $\text{InsideQ}[x_, \{g_, d\}] := (g \leq x < d)$
15. $\text{GetSessionPlanLengths}[\text{Session}_\text{List}] := \text{Module}[\{l = \text{Map}[\text{Last}, \text{Session}], \text{Map}[\{\#, \text{Count}[\{1, \#\}/\text{Length}[\text{Session}])\&, \text{Union}[\{1\}]\}$
16. $\text{GetAllSessionsPlanLengths}[\text{Sessions}_\text{List}] := \text{Module}[\{11 = \text{Apply}[\text{Join}, \text{Sessions}], 12\}$  
   $\text{12} = \text{Map}[\{\#, \text{Count}[\{12, \#\}/\text{Length}[\text{11}]\&, \text{Union}[\{12\}]\}$
17. $\text{GetTotallyOrderedActions}[\text{Plan}_\text{List}] := \text{Plan}[\{\text{Table}[x, \{x, \text{FirstActionIndex}, \text{Length}[\text{Plan}] - 2, 2)\]\
18. $\text{GetSessionLength}[\text{Session}_\text{List}] := \text{Length}[\text{Session}]$
19. $\text{GetSessionElapsedTime}[\text{Session}_\text{List}, \text{Ratio:_}\text{TicksPerSecond}] := \{\text{Session}[[1, 2]] - \text{Session}[[1, 1]]\}/\text{Ratio}$
20. $\text{GetNPCNames}[\text{Session}_\text{List}, \text{Sorted}_\text{Symbol}:\text{True}] := \text{Union}[\text{Session}[[\text{All}, 2]]]; \text{SameQ}[\text{True}, \text{Sorted}]$
21. $\text{GetNPCNames}[\text{Session}_\text{List}, \text{Sorted}_\text{Symbol}] := \text{UnsortedUnion}[\text{Session}[[\text{All}, 2]]]; \text{SameQ}[\text{False}, \text{Sorted}]$
GetNPCActions[Session_List, NPC_] := {NPC, GetActions[Cases[Session, {_, NPC, ___}]}}

GetAllNPCsActions[Session_List] :=
  Map[(GetNPCActions[Session, #]) &] & GetNPCNames[Session]

GetNPCActionFrequencies[Session_List, NPC_] := Module[{l = Cases[Session, {_, NPC, ___}]},
  {NPC, Sort[Map[{#, Count[Flatten[1], #] / Apply[Plus, Map[Last, 1]]} &,
    Last[GetNPCActions[Session, NPC]]], (Last[#1] < Last[#2]) &]]

GetAllNPCsActionFrequencies[Session_List] :=
  Map[(GetNPCActionFrequencies[Session, #]) &] & GetNPCNames[Session]

GetNPCPlanningRanges[Session_List, NPC_, Ratio_: $TicksPerMilliSecond] :=
  {NPC, Sort[Min[Subtract, Partition[Cases[Session, {_, NPC, ___}]][[All, 1]], 2, 1], 1] / Ratio]

GetAllNPCsPlanningRanges[Session_List, Ratio_: $TicksPerMilliSecond] :=
  Apply[Join, Map[(GetNPCPlanningRanges[Session, #, Ratio] &), GetNPCNames[Session]][[All, 2]]]

GetSessionPlanningRanges[Session_List, Ratio_: $TicksPerMilliSecond] :=
  (Minus[Apply[Subtract, Partition[Session[[All, 1]], 2, 1], 1]] / Ratio)

GetActions[Session_List] :=
  Map[(GetTotallyOrderedActions, Session)]

GetActionFrequencies[Session_List] := Module[{ls = Length[Apply[Join, Map[Function[{d}, DeleteCases[GetTotallyOrderedActions[d], Null]], Session]]],
  Map[({#, Count[Flatten[Session], #] / ls} &), DeleteCases[GetActions[Session], Null]]]

GetAssociatedActions[Session_List] :=
  Map[({#, DeleteCases[GetActions[Cases[Session, {___, #, ___}]], #]} &), GetActions[Session]]

GetAssociatedActionsFrequencies[Session_List] :=
  Map[({#, GetActionFrequencies[Replace[Cases[Session, {___, #, ___}], # -> Null, 3]]} &), GetActions[Session]]

GetPrimitivePlans[Session_List] :=
  Module[{l = Union[Map[(Sort[Flatten[{First[#, #][2, All, 1]]}]) &,
    Select[DeleteCases[GetAssociatedActionsFrequencies[Session], {_, {}}],
      (Apply[Equal, Last[#][[All, 2]]]) &]]},
    Union[Flatten[Map[Function[htn, Union[Map[GetTotallyOrderedActions, Select[Session, Function[p, Equal[Length[htn], Last[p]] &
      Apply[And, Map[(MemberQ[Flatten[p, H], htn]]][[1, 1]]]]]]]]], 1, 1]]]

GetPrimitivePlansFrequencies[Session_List] :=
  Module[{p = Map[GetTotallyOrderedActions, Session], b = GetPrimitivePlans[Session]},
    Map[({#, Count[p, #] / Length[p]} &), b]]

GetSessionNumbers[Path_String: $PathToSessionFiles, Pattern_String: $FileNamePattern] :=
  Map[(Module[{p = Take[StringPosition[#, "(Session " ["]"), -2],
    ToExpression[StringTake[#, {p[1, 2] + 1, p[2, 1] - 1]}] &},
    FileNames[$FileNamePattern, $PathToSessionFiles]]}[
    Session_List] :=
  Map[(GetTotallyOrderedActions, Session)]

GetActionFrequencies[Session_List] := Module[{ls = Length[Apply[Join, Map[Function[{d}, DeleteCases[GetTotallyOrderedActions[d], Null]], Session]]],
  Map[({#, Count[Flatten[Session], #] / ls} &), DeleteCases[GetActions[Session], Null]]]

GetAssociatedActions[Session_List] :=
  Map[({#, DeleteCases[GetActions[Cases[Session, {___, #, ___}]], #]} &), GetActions[Session]]

GetAssociatedActionsFrequencies[Session_List] :=
  Map[({#, GetActionFrequencies[Replace[Cases[Session, {___, #, ___}], # -> Null, 3]]} &), GetActions[Session]]

GetPrimitivePlans[Session_List] :=
  Module[{l = Union[Map[(Sort[Flatten[{First[#, #][2, All, 1]]}]) &,
    Select[DeleteCases[GetAssociatedActionsFrequencies[Session], {_, {}}],
      (Apply[Equal, Last[#][[All, 2]]]) &]]},
    Union[Flatten[Map[Function[htn, Union[Map[GetTotallyOrderedActions, Select[Session, Function[p, Equal[Length[htn], Last[p]] &
      Apply[And, Map[(MemberQ[Flatten[p, H], htn]]][[1, 1]]]]]]]]], 1, 1]]]

GetPrimitivePlansFrequencies[Session_List] :=
  Module[{p = Map[GetTotallyOrderedActions, Session], b = GetPrimitivePlans[Session]},
    Map[({#, Count[p, #] / Length[p]} &), b]]

GetSessionNumbers[Path_String: $PathToSessionFiles, Pattern_String: $FileNamePattern] :=
  Map[(Module[{p = Take[StringPosition[#, "(Session " ["]"), -2],
    ToExpression[StringTake[#, {p[1, 2] + 1, p[2, 1] - 1]}] &},
    FileNames[$FileNamePattern, $PathToSessionFiles]]}[
    Session_List] :=
  Map[(GetTotallyOrderedActions, Session)]
In this section, we load the session files, report on the frequency of the generation of plans and gather all the session files in one list for further computations.

Session filenames are created with a number; we first get these numbers:

```
GetSessionNumbers[$PathToSessionFiles, $FileNamePattern]
```

For this study, we'll give a look at the following session numbers:

```
{100, 101, 102, 103, 104, 105, 106, 107, 1082, 108, 1091, 1092, 10, 110, 1112, 1113, 111, 1122, 1123, 112, 1132, 1133, 113, 1142, 114, 1151, 115, 11, 1201, 1202, 120, 1211, 121, 1221, 1222, 122, 1231, 1232, 12331, 123, 1242, 124, 1252, 125, 1262, 126, 1272, 127, 1281, 1282, 128, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

Loading session files

In this section, we load the session files, report on the frequency of the generation of plans and gather all the session files in one list for further computations.

Session filenames are created with a number; we first get these numbers:

```
GetSessionNumbers[$PathToSessionFiles, $FileNamePattern]
```

For this study, we'll give a look at the following session numbers:

```
{100, 101, 102, 103, 104, 105, 106, 107, 1082, 108, 1091, 1092, 10, 110, 1112, 1113, 111, 1122, 1123, 112, 1132, 1133, 113, 1142, 114, 1151, 115, 11, 1201, 1202, 120, 1211, 121, 1221, 1222, 122, 1231, 1232, 12331, 123, 1242, 124, 1252, 125, 1262, 126, 1272, 127, 1281, 1282, 128, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```
Getting the session numbers did not load the corresponding files; we now load these corresponding session files (for instance, 210 plans have been built for the session file numbered 1082 which represents a little more than 3 minutes of play):

```plaintext
AllFEARSessionsInfos = Map[(LoadGameSession[#, $PathToSessionFiles, $TicksPerMinute, $EnfOfFileName]] &
  isn)] // ColumnForm
```

All these session files correspond to about 3 hours and 45 minutes sampled from playing F.E.A.R. between level 2 ("Infiltration") and level 11 ("Sayonara, Sucker") (see [1] for details about these levels):

```
Module[{t = Apply[Plus, AllFEARSessionsInfos[[All, 2]]] / 60, m,
  m = 60 (t - Floor[t]); {Floor[t], Floor[m], Floor[60 (m - Floor[m])]]
}
```

During those 3 hours, 45 minutes and 20 seconds, 6679 plans were generated:

```
Apply[Plus, AllFEARSessionsInfos[[All, 1]]]
```

Thus, a plan was generated (a little more than) every 2 seconds, on average:

```
(60 * Apply[Plus, AllFEARSessionsInfos[[All, 2]]]) / Apply[Plus, AllFEARSessions Infos[[All, 1]]]
```

However, across all the above F.E.A.R. sessions, the frequency of plan generation varies from 1 (≈ 1 plan generated every 0.9 second) to 11 (≈ 1 plan generated every 9.8 seconds):

```
60 * Apply[Divide, AllFEARSessionsInfos[[All, {2, 1}]], 2]
```

What kind of information can be found in these session files? Each session file contains the plans built by the AI Planning component while playing F.E.A.R.; these plans are for the Non-Player Characters (NPCs) to use, so NPCs’ names are recorded together with a time stamp in milli-seconds. Let’s give a look at the content of the first piece of information contained in session file numbered 1123:
The first value, 157484, is a time stamp in milli-seconds. noname1767 is the NPC’s name for which the plan was built. The last piece of information in the above list tells us that this plan is 2-action long. The first action is GotoNode and the second is UseSmartObjectNode. The rest describes predicates about the world of F.E.A.R. which are expected to take a given value. For instance, we expect noname1767 to be AtNode HeliPad[HeliCover[2]] after the GotoNode action is executed; we also expect the predicate WeaponArmed to be true, i.e. equal to 1.

Let’s put all the sessions into one list before going any further:

Let’s begin with the order of appearance of the NPCs in session file numbered 1123:

In the above graphic, the x-axis represents seconds and the y-axis represent NPCs; we thus observe that in this session file, NPCs regularly appear in groups of 3 or 4. The appendix 2 present such graphics for all the session files of AllFEARSessions.

Let’s continue with a look at all the actions reported in session file numbered 1123:

How often do these actions occur (here sorted in increasing order whereas the default printing is alphabetical)?
Out[50]= \{\text{SuppressionFireFromCover,} \frac{1}{448}\}, \{\text{LookAtDisturbanceFromView,} \frac{1}{448}\}, \\
\{\text{LongRecoilExplosive,} \frac{1}{448}\}, \{\text{GotoNodeOfType,} \frac{1}{448}\}, \{\text{BlindFireFromCover,} \frac{1}{448}\}, \\
\{\text{AttackGrenadeFromCover,} \frac{1}{448}\}, \{\text{FlushOutWithGrenade,} \frac{1}{224}\}, \{\text{AttackMelee,} \frac{1}{224}\}, \\
\{\text{GotoTargetLost,} \frac{3}{448}\}, \{\text{AttackFromAmbush,} \frac{1}{112}\}, \{\text{SurveyArea,} \frac{5}{448}\}, \\
\{\text{ReactToDanger,} \frac{5}{448}\}, \{\text{Attack,} \frac{5}{448}\}, \{\text{TraverseBlockedDoor,} \frac{3}{224}\}, \{\text{Follow,} \frac{3}{224}\}, \\
\{\text{Animate,} \frac{3}{224}\}, \{\text{LongRecoilBullet,} \frac{1}{448}\}, \{\text{InspectDisturbance,} \frac{9}{448}\}, \\
\{\text{TraverseLink,} \frac{13}{448}\}, \{\text{GotoTarget,} \frac{13}{448}\}, \{\text{EscapeDanger,} \frac{13}{448}\}, \{\text{Idle,} \frac{27}{448}\}, \\
\{\text{AttackReady,} \frac{29}{448}\}, \{\text{AttackFromCover,} \frac{19}{224}\}, \{\text{UseSmartObjectNode,} \frac{61}{224}\}, \{\text{GotoNode,} \frac{9}{32}\}\}

We observe that the action \text{AttackFromCover} is the offensive action which happens the most (about \frac{19}{224} \approx 8.5\%) while \text{UseSmartObjectNode} and \text{GotoNode} are by far the most frequent actions (respectively \frac{61}{224} \approx 27.2\% and \frac{9}{32} = 28, 125\%).

What are the names of the NPCs involved in that session?

Out[51]= \text{GetNPCNames[s1123]}\}

\{\text{noname1753, noname1756, noname1767, noname1770, noname1782, noname1784,} \\
\text{noname1786, noname1813, noname1815, noname1817, noname1946, CAI[Dead03],} \\
\text{Dead[Guy], HeliGun[BackHall[S01]].CAI, HeliGun[BackHall[S02]].CAI,} \\
\text{HeliGun[CitoC2[S01]].CAI, HeliGun[CitoC2[S02]].CAI, HeliGun[CitoC2[Ambush[S01]]].CAI,} \\
\text{HeliGun[CitoC2[Ambush[S02]]].CAI, HeliGun[CitoC2[Ambush[S03]]].CAI,} \\
\text{HeliGun[CYard[S01]].CAI, HeliGun[CYard[S02]].CAI, HeliGun[CYard[S03]].CAI,} \\
\text{HeliGun[CYard[S04]].CAI, HeliGun[CYard[S07]].CAI, HeliGun[CYard[S01]].CAI,} \\
\text{HeliGun[CYard[S02]].CAI, HeliGun[CYard[S03]].CAI, HeliGun[LowerHall[S02]].CAI,} \\
\text{HeliGun[LowerHall[S03]].CAI, HeliGun[MRoom[S01]].CAI, HeliGun[MRoom[S02]].CAI}\}

We observe that HeliGun[CYard[S03]].CAI appears as the 23rd name in the previous list of names of NPCs from session numbered 1123:

Out[52]= \text{Position[GetNPCNames[s1123], HeliGun[CYard[S03]].CAI]}\}

\{\{23\}\}

What are the actions of HeliGun[CYard[S03]].CAI?

Out[53]= \text{GetNPCActions[s1123, HeliGun[CYard[S03]].CAI]}\}

\{\text{HeliGun[CYard[S03]].CAI,} \\
\text{\{Attack, AttackFromCover, GotoNode, Idle, TraverseLink, UseSmartObjectNode\}\}}

... and how often do HeliGun[CYard[S03]].CAI's actions occur (automatically sorted in increasing order)?

Out[54]= \text{GetNPCActionFrequencies[s1123, HeliGun[CYard[S03]].CAI]}\}

\{\text{HeliGun[CYard[S03]].CAI,} \\
\text{\{\{Attack,} \frac{1}{18}\}, \{\text{TraverseLink,} \frac{1}{9}\},} \\
\text{\{AttackFromCover,} \frac{1}{9}\}, \{\text{Idle,} \frac{1}{6}\}, \{\text{UseSmartObjectNode,} \frac{5}{18}\}, \{\text{GotoNode,} \frac{5}{18}\}\}}

What about all the NPCs in that session?

Out[55]= \text{GetAllNPCsActionFrequencies[s1123]}\}
{AttackReady, 1/5}, {UseSmartObjectNode, 3/10}, {GotoNode, 3/10})

{HeliGun[BackHall[S02]].CAI, \{LongRecoilBullet, 1/11}, {Idle, 1/11}, {EscapeDanger, 1/11},
{AttackFromCover, 2/11}, {UseSmartObjectNode, 3/11}, {GotoNode, 3/11}}

{HeliGun[CitoC2[S01]].CAI, \{ReactToDanger, 1/14}, {LongRecoilBullet, 1/14},
{Idle, 1/14}, {GotoTargetLost, 1/14}, {AttackReady, 1/14}, {Attack, 1/14}, {Animate, 1/14},
{UseSmartObjectNode, 3/14}, {GotoNode, 2/7}}, {HeliGun[CitoC2[S02]].CAI},
{{TraverseLink, 1/7}, {EscapeDanger, 1/7}, {UseSmartObjectNode, 2/7}, {GotoNode, 3/7}}

{HeliGun[CitoC2[Ambush[S01]]].CAI, {{UseSmartObjectNode, 1/4}, {LongRecoilBullet, 1/4},
{GotoNode, 1/4}, {EscapeDanger, 1/4}}}, {HeliGun[CitoC2[Ambush[S02]]].CAI},
{{EscapeDanger, 1/5}, {UseSmartObjectNode, 2/5}, {GotoNode, 2/5}}},

{HeliGun[CitoC2[Ambush[S03]]].CAI, {{TraverseLink, 1/11}, {EscapeDanger, 1/11},
{AttackFromCover, 1/11}, {UseSmartObjectNode, 4/11}, {GotoNode, 4/11}}},

{HeliGun[CYard[S01]].CAI, {{EscapeDanger, 1/21}, {BlindFireFromCover, 1/21},
{TraverseLink, 2/21}, {AttackFromCover, 2/21}, {Idle, 1/7},
{UseSmartObjectNode, 2/7}, {GotoNode, 2/7}}}, {HeliGun[CYard[S02]].CAI},
{{GotoTargetLost, 1/14}, {AttackFromCover, 1/14}, {AttackFromAmbush, 1/14}, {Animate, 1/14},
{Idle, 1/7}, {AttackReady, 1/7}, {UseSmartObjectNode, 3/14}, {GotoNode, 3/14}}},

{HeliGun[CYard[S03]].CAI, {{Attack, 1/18}, {TraverseLink, 1/9}, {AttackFromCover, 1/9},
{Idle, 1/6}, {UseSmartObjectNode, 5/18}, {GotoNode, 5/18}}}, {HeliGun[CYard[S04]].CAI},
{{UseSmartObjectNode, 1/8}, {SurveyArea, 1/8}, {InspectDisturbance, 1/8}, {Idle, 1/8},
{GotoTarget, 1/8}, {GotoNode, 1/8}, {EscapeDanger, 1/8}, {AttackMelee, 1/8}}},

{HeliGun[CYard[S07]].CAI, {{TraverseBlockedDoor, 1/7}, {Idle, 1/7},
{EscapeDanger, 1/7}, {UseSmartObjectNode, 2/7}, {GotoNode, 2/7}}},

{HeliGun[CYard2[S01]].CAI, {{TraverseLink, 1/22}, {ReactToDanger, 1/22},
{LongRecoilBullet, 1/22}, {GotoTarget, 1/22}, {AttackFromCover, 1/22}, {Idle, 1/11},
{EscapeDanger, 1/11}, {UseSmartObjectNode, 2/11}, {GotoNode, 2/11}, {AttackReady, 5/22}}},
Let's end with the Whisker boxes of the number of plans per NPC for all sessions:

```plaintext
BoxWhiskerChart[Map[
    Function[{s}, Map[(Length[Cases[s, {_, #, __}]] & , GetNPCNames[s]])], AllFEARSessions]]
```

And here are the NPCs responsible for the maximum number of plans in each session, together with that number of plans:
Determining fixed plans

"Fixed" plans are plans which are generated with the same, fixed, totally ordered sequence of actions; the purpose of this section is to look for such plans in the session files.

We begin with the plan lengths of session s1123 and their proportions:

```
In[58]:= GetSessionPlanLengths[s1123]

Out[58]= {{1, \(\frac{179}{311}\)}, {2, \(\frac{127}{311}\)}, {3, \(\frac{5}{311}\)}}
```

Note that it is immediate to plot the same information from all the sessions:

```
In[59]:= GetSessionPlanLengths[Apply[Join, AllFEARSessions]]

Out[59]= {{1, \(\frac{3664}{6679}\)}, {2, \(\frac{2934}{6679}\)}, {3, \(\frac{71}{6679}\)}, {4, \(\frac{10}{6679}\)}}
```

```
ListPlot[GetSessionPlanLengths[Apply[Join, AllFEARSessions]] /.
{x_, y_} \rightarrow {x, 100 * y},
Frame \rightarrow True, GridLines \rightarrow (Range[5], Range[0, 50, 10]),
PlotRange \rightarrow \{\{0, 5\}, \{\{-2, 60\}\}}, PlotStyle \rightarrow \{RGBColor[1, 0, 0], PointSize[0.02]},
PlotLabel \rightarrow "Plan length frequencies in F.E.A.R."}
```

We first observe from the above plot that the plans recorded in all the sessions are never more than 4-action long; moreover 3-action plans and 4-action plans are very rare in comparison with 1-action plans (the most frequent: \(\frac{3664}{6679} \approx 54.86\%) and 2-action plans (\(\frac{2934}{6679} \approx 43.93\)) which both represent \(\frac{3664 + 2934}{6679} \approx 98.8\% of all the plans generated by the AI Planning component while playing F.E.A.R.

We can observe that the actions composing the 3-action plans and the 4-action plans do not involve fighting with the player but surveillance (\(\frac{17}{71} \approx 24\% and \frac{20}{213} \approx 9.4\%\)) and (probably) animation (\(\frac{19}{40} = 47.5\% and \frac{9}{20} = 45\%\)):

```
In[60]:= GetActionFrequencies[Cases[Apply[Join, AllFEARSessions], {___, 3}]]

Out[60]= {{GotoNode, \(\frac{20}{213}\)}, {GotoTarget, \(\frac{17}{71}\)}, {InspectDisturbance, \(\frac{17}{71}\)},
{MountNodeUncloaked, \(\frac{20}{213}\)}, {SurveyArea, \(\frac{17}{71}\)}, {UseSmartObjectNodeMounted, \(\frac{20}{213}\)}}
```
Let's now turn to looking for "fixed" plans by looking at the actions which appear with a given action:

In[63]:= GetAssociatedActions[s1123] // ColumnForm

Out[63]= {Animate, {}}
{Attack, {}}
{AttackFromAmbush, {}}
{AttackFromCover, {}}
{AttackGrenadeFromCover, {}}
{AttackMelee, {}}
{AttackReady, {}}
{BlindFireFromCover, {}}
{EscapeDanger, {}}
{FlushOutWithGrenade, {}}
{Follow, {}}
{GotoNode, {UseSmartObjectNode}}
{GotoNodeOfType, {LookAtDisturbanceFromView}}
{GotoTarget, {InspectDisturbance, SurveyArea}}
{GotoTargetLost, {}}
{Idle, {}}
{InspectDisturbance, {GotoTarget, SurveyArea}}
{LongRecoilBullet, {}}
{LongRecoilExplosive, {}}
{LookAtDisturbanceFromView, {GotoNodeOfType}}
{ReactToDanger, {}}
{SuppressionFireFromCover, {}}
{SurveyArea, {GotoTarget, InspectDisturbance}}
{TraverseBlockedDoor, {}}
{TraverseLink, {}}
{UseSmartObjectNode, {GotoNode}}

In the previous list computed from session file s1123, we observe that some actions, e.g. Animate, Attack, AttackFromAmbush, ..., are never associated with other actions.

When actions are associated with others, we can ask for their frequencies (in the following, the actions never associated with other actions are deleted):

In[64]:= DeleteCases[GetAssociatedActionsFrequencies[s1123], __, 4] // ColumnForm

Out[64]= {GotoNode, {UseSmartObjectNode, 1}}
{GotoNodeOfType, {LookAtDisturbanceFromView, 1}}
{GotoTarget, {InspectDisturbance, SurveyArea}}
{InspectDisturbance, {GotoTarget, SurveyArea}}
{LookAtDisturbanceFromView, {GotoNodeOfType, 1}}
{SurveyArea, {GotoTarget, InspectDisturbance}}
{UseSmartObjectNode, {GotoNode, 1}}

We note that the frequencies of the actions associated with action SurveyArea are equal to $\frac{1}{2}$.

In[65]:= Cases[DeleteCases[GetAssociatedActionsFrequencies[s1123], __, 4], {SurveyArea, __}]

Out[65]= {SurveyArea, {GotoTarget, 1, InspectDisturbance, 1}}

This value of $\frac{1}{2}$ suggests that the actions SurveyArea, GotoTarget and InspectDisturbance always appear together in the same plan, in the session numbered 1123. Are there any other such plans in this session?
GetPrimitivePlans[s1123]  
Out[66]= { {GotoNode, UseSmartObjectNode}, {GotoNodeOfType, LookAtDisturbanceFromView}, {SurveyArea, GotoTarget, InspectDisturbance}}  

Thus, there are 3 plans which are always composed with the same "fixed" sequence of actions; how often do these "fixed" plans appear in the session (sorted in increasing order)?

Sort[GetPrimitivePlansFrequencies[s1123], (Last[#1] < Last[#2]) &] // ColumnForm

<table>
<thead>
<tr>
<th>FixedPlans</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>{GotoNode, UseSmartObjectNode}</td>
<td>(\frac{1}{311})</td>
</tr>
<tr>
<td>{SurveyArea, GotoTarget, InspectDisturbance}</td>
<td>(\frac{5}{311})</td>
</tr>
<tr>
<td>{GotoNode, UseSmartObjectNode}</td>
<td>(\frac{122}{311})</td>
</tr>
</tbody>
</table>

With a frequency of \(\frac{122}{311} \approx 39.2\%\), we observe that the "fixed" plan \(\text{GotoNode, UseSmartObjectNode}\) appears much more than the other two "fixed" plans. Is this specific to this session file or does this stand over all session files?

First, let's look at the "fixed" plans for each of the sessions we study:

FixedPlans = Apply[Union, Map[GetPrimitivePlans, AllFEARSessions]]

<table>
<thead>
<tr>
<th>DismountNodeUncloaked, GotoTarget</th>
<th>{DismountNodeUncloaked, Idle}</th>
<th>{GotoNode, MountVehicle}</th>
<th>{GotoNode, UseSmartObjectNode}</th>
<th>{GotoNodeOfType, LookAtDisturbanceFromView}</th>
<th>{GotoTarget, InspectDisturbance}</th>
<th>{ReloadCovered, AttackFromCover}</th>
<th>{ReloadCovered, BlindFireFromCover}</th>
<th>{ReloadCrouch, Attack}</th>
<th>{ReloadCrouch, AttackFromArmed}</th>
<th>{GotoNode, MountNodeUncloaked, UseSmartObjectNodeMounted}</th>
<th>{SurveyArea, GotoTarget, InspectDisturbance}</th>
</tr>
</thead>
</table>

Each of these "fixed" plans could have been, for instance, implemented as a primitive Hierarchical Task Network (HTN) [2]. A primitive HTN is an HTN only containing primitive actions: a primitive HTN does not contain actions which can be expanded (i.e. actions which are themselves HTNs).

What are the NPCs using these "fixed" plans (sorted in alphabetical order of the first action of a "fixed" plan)?

Map[Function[{fp}, (fp, Union[DeleteCases[Map[(Module[{p = GetTotallyOrderedActions[p], f}, If[SameQ[p, f], #[[2]]]) &, Apply[{Join, AllFEARSessions[]}, Null]]], FixedPlans]]}, Assassin, Corridor], Entry[Assassin[Guard[Cover01]]], Entry[Assassin[Guard[Cover02]]], Entry[Assassin[Guard[Cover03]]]}, {DismountNodeUncloaked, Idle}, kitty[scripted]], {GotoNode, MountVehicle}, {Guard, FixedPlans}, {BackupSoldier01, Delta00, Delta01, -2-CAI00-Enemiespawned, -2-CAI01-Enemiespawned, ExitHeavyGd00, ExitSoldierGd00, ExitSoldierGd01, ExitSoldierGd02, ExitSoldierGd03, ExitSoldierGd04, ExitSoldierGd05, ExitSoldierPt00, GarageHeavy, GarageS01, GarageS02, GarageS03, GarageS04, GarageS05, H50101, I03P03TunnelLead, Jin, Jumpy, Jumpy03, Krig, Krigel, L1, L2, MonitorBkUpS01, MonitorS01, MonitorS02, MonitorS03, noname1225, noname1250, noname1584, noname1753, noname1756, noname1767, noname1770, noname1782, noname1784, noname1786, noname1813, noname1815, noname1817, noname1946, noname2194, noname2220, noname2236, noname2238, noname2240, noname2272, noname2279, noname2281, noname2292, noname2293, noname2294, noname2295, noname2305, noname2310, noname2311, noname2312, noname2313, noname2318, noname2319, noname2320, noname2321, noname2332, noname2334, noname2344, noname2351, noname2353, noname2356, noname2367, noname2377, noname2383, noname2385, noname2403, noname2410, noname2412, noname2422, noname2424, noname2435, noname2442, noname2444, noname2454, noname2456, noname2478, noname2485, noname2487, noname2499, noname2501, noname2511, noname2517, noname2519, noname2529, noname2536, noname2538, noname2548, noname2550, noname2562, noname2569, noname2570, noname2572, noname2582, noname2584, noname2588, noname2590, noname2601, noname2603, noname2605, noname2614, noname2620,
How often do these HTNs appear across all session files (sorted in increasing order)?
As the first three "fixed" plans only appear once across all sessions, we probably cannot say much about them; however, for the last two "fixed" plans, which respectively appear $\frac{529}{6679} \approx 8\%$ and $\frac{2355}{6679} \approx 35\%$ of the time, it is reasonable to think of them as HTNs [2].

We eventually note that the "fixed" plan \{GotoNode, UseSmartObjectNode\} is very common, as it represents a (little more than a) third of all the plans built for the NPCs; are there any NPCs for which this plan is not built?

Yes, there are! Then, what are the actions they use?

```
In[70]:= Map[GetNPCActionFrequencies[Apply[Join, AllFEARSessions], #]] & , npc1
```

```
Map[GetNPCActionFrequencies[Apply[Join, AllFEARSessions], #]] & , npc1
```
[SpookAssassin, \{[\text{Idle, }\frac{1}{6}], \{\text{TraverseLinkUncloaked, }\frac{1}{3}\}, \{\text{GotoNode, }\frac{1}{2}\}\}],

[WTFExfilAssCAI, \{[\text{TraverseLinkUncloaked, }\frac{2}{9}], \{\text{Idle, }\frac{1}{3}\}, \{\text{GotoNode, }\frac{4}{9}\}\}],

\{132[crow01], \{[\text{Idle, }\frac{1}{10}]\}\}, \{\text{Arrival[Assassin00]},

\{[\text{Idle, }\frac{1}{10}], \{\text{TraverseLinkUncloaked, }\frac{1}{5}\}, \{\text{AttackLungeUncloaked, }\frac{1}{5}\}, \{\text{GotoTarget, }\frac{1}{2}\}\}],

\{\text{Arrival[Search[S02]]}, \{[\text{Idle, }\frac{1}{6}], \{\text{EscapeDanger, }\frac{1}{6}\}, \{\text{Animate, }\frac{1}{6}\}, \{\text{Follow, }\frac{1}{2}\}\}\}],

\{\text{CAI[Dead03]}, \{[\text{Animate, }1]\}\}, \{\text{Dead[Guy]}, \{[\text{Animate, }1]\}\},

\{\text{Admin[Cons[DA01]].CAI}, \{[\text{GotoNode, }1]\}\},

\{\text{Admin[Sec[MapesGate]].CAI}, \{[\text{Animate, }\frac{1}{3}], \{\text{GotoNode, }\frac{2}{3}\}\}\}],

\{\text{Badge[Turret01].Weapon}, \{[\text{IdleTurret, }\frac{1}{41}], \{\text{AttackTurretCeiling, }\frac{16}{41}\}, \{\text{FaceNode, }\frac{24}{41}\}\}\}],

\{\text{Badge[Turret03].Weapon}, \{[\text{IdleTurret, }\frac{2}{27}], \{\text{FaceNode, }\frac{4}{27}\}, \{\text{AttackTurretCeiling, }\frac{7}{9}\}\}\}],

\{\text{Trouble[Turret01].Weapon}, \{[\text{FaceNode, }\frac{1}{9}], \{\text{IdleTurret, }\frac{2}{9}\}, \{\text{AttackTurretCeiling, }\frac{2}{3}\}\}\}],

\{\text{Trouble[Turret02].Weapon},

\{[\text{AttackTurretCeiling, }\frac{18}{77}], \{\text{IdleTurret, }\frac{20}{77}\}, \{\text{FaceNode, }\frac{39}{77}\}\}\}],

\{\text{Trouble[Turret03].Weapon}, \{[\text{AttackTurretCeiling, }\frac{10}{41}],

\{\text{IdleTurret, }\frac{21}{82}\}, \{\text{FaceNode, }\frac{1}{2}\}\\}, \{\text{Trouble[Turret04].Weapon},

\{[\text{IdleTurret, }\frac{11}{60}], \{\text{FaceNode, }\frac{7}{30}\}, \{\text{AttackTurretCeiling, }\frac{7}{12}\}\}\}],

\{\text{Trouble[Turret05].Weapon}, \{[\text{IdleTurret, }\frac{1}{10}], \{\text{AttackTurretCeiling, }\frac{9}{10}\}\}\}],

\{\text{Entry[Assassin[Guard[Cover01]]]},

\{[\text{UseSmartObjectNodeMounted, }\frac{1}{6}], \{\text{MountNodeUncloaked, }\frac{1}{6}\}, \{\text{GotoTarget, }\frac{1}{6}\}\}],

\{\text{GotoNode, }\frac{1}{6}\}, \{\text{DismountNodeUncloaked, }\frac{1}{6}\}, \{\text{AttackMeleeUncloaked, }\frac{1}{6}\}\}],

\{\text{Entry[Assassin[Guard[Cover03]]]}, \{[\text{Idle, }\frac{1}{42}], \{\text{AttackMeleeUncloaked, }\frac{1}{42}\}\}],

\{\text{DismountNodeUncloaked, }\frac{1}{21}\}, \{\text{TraverseLinkUncloaked, }\frac{1}{14}\}, \{\text{GotoTarget, }\frac{1}{7}\}\}],

\{\text{MountNodeUncloaked, }\frac{3}{14}\}, \{\text{GotoNode, }\frac{3}{14}\}, \{\text{UseSmartObjectNodeMounted, }\frac{11}{42}\}\}],

\{\text{kitty scripted}}, \{[\text{DismountNodeUncloaked, }\frac{1}{5}], \{\text{UseSmartObjectNodeMounted, }\frac{1}{5}\}\}],

\{\text{MountNodeUncloaked, }\frac{1}{5}\}, \{\text{Idle, }\frac{1}{5}\}, \{\text{GotoNode, }\frac{3}{10}\}\}],

\{\text{Mapes[2]}, \{[\text{GotoNode, }1]\}\}, \{\text{NA[Ghost01]}, \{[\text{AttackMelee, }\frac{1}{8}], \{\text{Idle, }\frac{1}{4}\}\}],

\{[\text{AttackLungeUncloaked, }\frac{1}{4}], \{\text{GotoTarget, }\frac{3}{8}\}\}\}, \{\text{NA[Ghost02]},

\{[\text{AttackLungeUncloaked, }\frac{1}{7}], \{\text{Idle, }\frac{2}{7}\}, \{\text{GotoTarget, }\frac{2}{7}\}, \{\text{AttackMelee, }\frac{2}{7}\}\}\}],
Very specific actions for very specific NPCs (animated characters such as Bishop, Fettel and Mapes, cloakable Assassins and ceiling Turrets): we here run into actions sets, as reported in [3]. We later report on three sets of actions: offensive, defensive and animation-oriented.

The need for planning

This section deals with the need for planning: how often is planning needed, overall sessions? For which specific NPCs? Is planning called more often when defending or when attacking? This section provides plots as elements to answering these questions.

How often is a plan built for the NPC named HeliGun[CYard[S03]].CAI? Or, put differently, what is HeliGun[CYard[S03]].CAI’s need for planning? The following numbers represent the number of milli-seconds between 2 successive calls for a plan for the NPC HeliGun[CYard[S03]].CAI. These numbers are not sorted so as to inform about the variation between successive calls for a plan while playing the game (and recorded in session 1123):

```
In[73]:= GetNPCPlanningRanges[s1123, HeliGun[CYard[S03]].CAI]
Out[73]= {HeliGun[CYard[S03]].CAI,
          {547, 563, 766, 781, 1172, 1234, 1250, 2172, 3438, 5031, 13640, 17469}}
```

Now, let us give a look at the need for planning for all NPCs in the same session:
Get and then sort the need for planning across all NPCs of the session 1123 (we refer to these measures as the "NPC" need for planning as these numbers are computed from time-stamps of one NPC):
In[75]:= Sort[GetAllNPCsPlanningRanges[s1123]]

Out[75]= 

In[76]:= Sort @ Map @ H Max @ GetAllNPCsPlanningRanges @ AllFEARSessions D D L &, AllFEARSessions D D D 

Out[76]= 

So the longest need for planning for this session is 113.391 seconds (about 1 minute and 53 seconds); but what is the longest need for planning of each session (sorted in increasing order)?

The largest value for the need for planning is 902 563 milli-seconds, which is about 15 minutes! Let us draw the Whisker box from the previous list; from the Mathematica help: a Whisker box takes the form of a box that spans the distance between two quantiles surrounding the median, typically the 25% quantile to the 75% quantile. Commonly, "whiskers," lines that extend to span either the full data set or the data set excluding outliers, are added. In other words, a Whisker box tells us where most of the values are (in the — red — box!):

According to this Whisker box, the first interesting maximum value for the need of planning is bit less than 50 seconds; but as we are about to see, this value is in fact too big for plotting interesting Whisker boxes. Now, we compute the time between every two successive plans in the session file, independently of the NPCs. These measures refer to what we call the "absolute" (because independent of any NPC) need for planning. Some values are 0 because the time stamp of two successive plans is the same number of milli-seconds; this happens when 2 NPCs call for planning at (about) the same time, that is, when the time stamps are the same when measured in milli-second (we should expect them to be different if measured with a greater precision, e.g. micro-seconds):
In[78]:= \text{Sort[GetSessionPlanningRanges[s1123]]}

Out[78]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 15, 16, 16, 31, 31, 31, 31, 31, 31, 31, 32, 32, 46, 46, 47, 47, 47, 47, 47, 47, 47, 47, 47, 62, 63, 63, 63, 78, 78, 78, 78, 78, 78, 78, 78, 93, 94, 94, 94, 109, 109, 109, 110, 110, 110, 110, 125, 125, 125, 125, 125, 125, 140, 156, 156, 156, 157, 171, 171, 171, 172, 172, 187, 188, 188, 188, 188, 203, 203, 203, 203, 204, 219, 234, 235, 235, 250, 250, 250, 250, 265, 266, 281, 281, 313, 328, 328, 328, 328, 328, 343, 343, 343, 344, 344, 359, 359, 359, 360, 375, 375, 375, 391, 406, 406, 406, 407, 422, 422, 438, 438, 515, 515, 516, 531, 547, 547, 563, 563, 563, 563, 578, 578, 594, 594, 609, 625, 625, 640, 641, 656, 657, 671, 672, 672, 672, 688, 688, 688, 704, 734, 734, 735, 750, 750, 766, 766, 781, 797, 797, 797, 844, 844, 859, 860, 875, 875, 891, 891, 922, 953, 1000, 1015, 1016, 1031, 1078, 1078, 1078, 1094, 1094, 1140, 1156, 1156, 1156, 1172, 1172, 1172, 1188, 1218, 1219, 1235, 1250, 1250, 1250, 1250, 1250, 1250, 1250, 1250, 1250, 1250, 1250, 1250, 1250, 1281, 1282, 1328, 1328, 1359, 1359, 1438, 1453, 1516, 1532, 1610, 1625, 1640, 1703, 1718, 1765, 1765, 1765, 1765, 1788, 1843, 1860, 1922, 1984, 2031, 2062, 2093, 2219, 2250, 2250, 2250, 2250, 2250, 2250, 2250, 2282, 2359, 2375, 2422, 2469, 2562, 2562, 2625, 2672, 2687, 2703, 2797, 2828, 3015, 3219, 3281, 3297, 3406, 3547, 3688, 3766, 3843, 3891, 3953, 4110, 4125, 4218, 4360, 4625, 4906, 5563, 5766, 5875, 5922, 6843, 7500, 8109, 8969, 9250, 10312, 11187, 11328, 12421, 12593, 17563, 19984, 21031, 25140, 25422, 26157, 32703, 32828, 43109, 46250, 69203, 69265, 86328, 106328}

So the longest need for planning for this session is 106.328 seconds (about 1 minute and 46 seconds); but what is the longest need for planning of each session (sorted in increasing order)?

In[79]:= \text{Sort[Map[HMax[GetSessionPlanningRanges[D]], AllFEARSessions]]}

Out[79]= {5469, 9094, 9110, 9313, 41562, 88391, 104594, 106328, 109703, 111204, 135719, 137688, 144719, 163969, 246281, 299234, 529750}

The largest value for the need for planning is 529.750 milli-seconds, which is about 8 minutes and 50 seconds. Let us draw the Whisker box from the previous list:

In[80]:= \text{BoxWhiskerChart[Map[HMax[GetSessionPlanningRanges[D]], AllFEARSessions]]}

According to this Whisker box, the lowest interesting maximum value for the need of planning is 41.562 seconds; but it is in fact a too big value for plotting interesting Whisker boxes (in the two plots below, the y-axis represents milli-seconds while the x-axis is the session number, in order of appearance in AllFEARSessions):
In fact, we have to restrict ourselves to a much smaller value of about 2 seconds if we want to learn something from these Whisker boxes:

\[
\text{Out[82]= BoxWhiskerChart[Map[(Select[#,(InsideQ[#,{0,2200}])&)],Map[GetSessionPlanningRanges,AllFEARSessions]]}
\]

We observe that the "NPC" need for planning is rather similar to the above "absolute" need for planning:
Visualization of the need for planning

Get the "NPC" need for planning (for each NPC, we build a list with its calls to the planning component and thus, the time between two successive calls for planning is only related to one NPC), across several session files:

```math
In[84]:= Length[NPCsPR = Sort[Apply[Join, Map[GetAllNPCsPlanningRanges, AllFEARSessions]]]]
Out[84]= 6339
```

Let us look at the distribution of the time between two successive calls for planning (related to one NPC):

```math
In[85]:= ListPlot[MapIndexed[((First[#2] / Length[NPCsPR]) &) &, NPCsPR], PlotStyle -> RGBColor[0, 0, 1], PlotRange -> All, Frame -> True, GridLines -> Automatic]
```

More than 95% of these values are below 15 seconds (the x-axis is time in milli-seconds and the y-axis is the corresponding %), while 55% are below 2 seconds:

```math
In[86]:= Table[{i, 100. * Length[Select[NPCsPR, (# <= i) &]] / Length[NPCsPR]}, {i, 1000, 15000, 1000}]
Out[86]= {{1000, 39.4384}, {2000, 55.7028}, {3000, 61.8867}, {4000, 67.487}, {5000, 77.9934}, {6000, 84.4613}, {7000, 87.4113}, {8000, 89.3201}, {9000, 91.0869}, {10000, 92.4278}, {11000, 93.4532}, {12000, 93.8634}, {13000, 94.5575}, {14000, 95.1254}, {15000, 95.5513}}
```

Get the "absolute" need for planning (this time, we consider the time between two successive calls for planning, independently of NPCs) across several session files:
Let us look at the distribution of the time between two successive calls to the planning component, whatever the NPC calling for planning:

```
In[88]:=
ListPlot[MapIndexed[(#1, First[#2] / Length[APR]) &, APR],
PlotStyle -> RGBColor[1, 0, 0], PlotRange -> All, Frame -> True, GridLines -> Automatic]
```

More than 95% of these values are below 5 seconds (the x-axis is time in milli-seconds and the y-axis is the corresponding %), while about 83% are below 2 seconds:

```
In[89]:=
Table[i, 100. * Length[Select[APR, (H ≤ i) &]] / Length[APR]], {i, 1000, 15000, 1000}
```

As successive (i.e. in the order they appear in the session file) time stamps may not belong to the same NPC, the previous number of measures, 6662, related to the "absolute" need for planning is expected to be greater than the above number of measures, 6339, related to the "NPC" need for planning.

As the successive calls to the planning component can correspond to distinct NPCs, the absolute need for planning is greater than the NPC need for planning (that is, the time pressure on the planning component is greater than the time pressure on an NPC). In the following plot, we observe that from 0.5 to 4 seconds, it is from 20% to 28% greater:

```
In[90]:=
ListPlot[Table[{i, 100. * (Length[Select[APR, (H ≤ i) &]] / Length[APR]) -
(Length[Select[NPCsPR, (H ≤ i) &]] / Length[NPCsPR])},
{i, 50 / 3, 15000, 50 / 3}], PlotRange -> All, Frame -> True, GridLines -> Automatic]
```

In the plots below, the x-axis represents the time measured in milli-seconds (up to 12600 milli-seconds in the first series of plots and up to 1000 milli-seconds in the second series) and the y-axis represents the percentage of calls for planning. The red curve represents the absolute need for planning and the blue curve represents the "NPC" need for planning.
For instance, the first red dot of the first plot tells us that about 9.5% of the "absolute" calls for planning happen at the same time. In practice, this means that the time-stamp should be measured more precisely than milli-seconds in order to make all time-stamps have different values.

Although the x-axis represents the time in milli-seconds, percentages are gathered over a time-window of multiple of $\frac{1}{60}$th of a second. That is, for the first red dot of the first plot, we gather all the calls for planning which happen in the range $[0, 1 \times \frac{1}{60}]$; then, for the second red dot, we gather all the calls for planning which happen in the range $[1 \times \frac{1}{60}, 2 \times \frac{1}{60}]$; the third corresponds to $[2 \times \frac{1}{60}, 3 \times \frac{1}{60}]$, and so on.

In the successive plots below, the time-window is increased from 1 frame ($\frac{1}{60}$th of a second in the first plot) to $\frac{2}{60}$, $\frac{3}{60}$, ..., $\frac{12}{60}$th of a second in the last plot (which is 200 milli-seconds). Thus, the first red dot of the last plot represents the percentage of call for planning on the range $[0, 1 \times \frac{12}{60}]$, which is about a third of the time (as the planning ranges are in milli-seconds the last parameter is $\frac{1}{60} \times 1000 = \frac{50}{3}$ and not $\frac{1}{60}$):

\begin{verbatim}
In[91]:= PlotTheNeedForPlanning[NPCsPR, APR, 0, 12 600, 200, 50 / 3]
\end{verbatim}
% of planning time over 4 frames

% of planning time over 5 frames

% of planning time over 6 frames
Two peaks, around 2 and 4 seconds, break the (relative) hyperbolic look of the curves. What are the plans involved in these peaks?

**The peak just before 2 seconds**

Let's first locate this peak more precisely:

```plaintext
In[32]:= PlotTheNeedForPlanning[NPCsPR, APR, 1875, 2000, 30, 50 / 3]
```

We observe from the above plot that the peak is located around 1925 milli-seconds.

We can now look, around 1925 milli-seconds, for the plans, the NPCs, and their frequencies:
\textbf{Module}\{\text{dd} = \text{Map[Rest, } \text{Apply[Join, } \text{Map[Function}[\{s\}, \text{Select[}}\end{equation}
\begin{equation}
\text{Map[([2, 1] - [1, 1], [2, 2], GetTotallyOrderedActions[#[2]]]) \&, \text{Partition}[s, 2, 1] \&, \text{First}[[#, \{1900, 1950\}] \&, \text{AllFEARSessions}]]], b], b = \text{Union[dd]}; \text{Sort[Map[([#, Count[dd, #] / Length[dd]] \&, b],}
\begin{equation}
\text{Last}[\#1 < \text{Last}[\#2]] \&;]}\} \text{// ColumnForm}
\end{equation}
\begin{equation}
\text{Out[93]} = \\
\begin{array}{ll}
\{\text{sec[soldier01], [AttackReady], } & \frac{1}{174} \\
\{\text{FrLI[PA00], [AttackFromArmoredBounded]}, & \frac{1}{174} \\
\{\text{Trouble[Turret03].Weapon, [FaceNode]}, & \frac{1}{174} \\
\{\text{HeliGun[MRoom[S02]].CAI, [GotoTarget]}, & \frac{1}{174} \\
\{\text{Arrival[Security[S02]].CAI, [Idle]}, & \frac{1}{174} \\
\{\text{Arrival[Lounge[S03]].CAI, [GotoNode, UseSmartObjectNode]}, & \frac{1}{174} \\
\{\text{ATC[Rush[1]].[AttackFromCover]}, & \frac{1}{174} \\
\{\text{7[TK[S03]], [SuppressionFireFromCover]}, & \frac{1}{174} \\
\{\text{4[S02], [AttackFromCover]}, & \frac{1}{174} \\
\{\text{WTFFRBlead, [Idle]}, & \frac{1}{174} \\
\{\text{noname2929, [GotoNode, UseSmartObjectNode]}, & \frac{1}{174} \\
\{\text{noname2675, [Attack]}, & \frac{1}{174} \\
\{\text{noname2412, [AttackFromAmbush]}, & \frac{1}{174} \\
\{\text{noname2295, [BlindFireFromCover]}, & \frac{1}{174} \\
\{\text{noname2238, [GotoNode, UseSmartObjectNode, GotoNode, UseSmartObjectNode]}, & \frac{1}{174} \\
\{\text{Holiday, [Idle]}, & \frac{1}{174} \\
\{\text{GarageS05, [GotoNodeOfType, AttackFromView]}, & \frac{1}{174} \\
\{\text{ExitSoldierGd03, [AttackFromCover]}, & \frac{1}{174} \\
\{\text{ExitHeavyGd00, [Idle]}, & \frac{1}{174} \\
\{\text{ExitSoldierGd02, [AttackFromCover]}, & \frac{1}{87} \\
\{\text{Rat02, [GotoNode, UseSmartObjectNode]}, & \frac{7}{174} \\
\{\text{Delta01, [UseSmartObjectNode]}, & \frac{73}{87} \\
\end{array}
\end{equation}
\end{equation}
\end{equation}
\end{equation}
\begin{equation}
\begin{align*}
&\text{With } \frac{73}{87} = 84\% \text{ of the occurrences between 1900 and 1950 milli-seconds, the NPC named Delta01 is responsible for the peak around 1925 milli-seconds, with a 1-action plan: UseSmartObjectNode.} \\
&\text{The peaks just after 4 seconds} \\
&\text{Let's locate this second peak more precisely:}
\end{align*}
\end{equation}
From the above plot, we see there are two peaks respectively located around 4110 milliseconds and around 4175 milliseconds.

In one call, let's look at the two peaks for the plans, the NPCs, and their frequencies:

With \( \frac{139}{162} = 86\% \) of the occurrences between 4075 and 4200 milliseconds, the NPC named Delta01 is responsible for the two peaks located around 4140 milliseconds, with a 1-action plan: Animate.

UseSmartObjectNode and Animate thus are important actions for Delta01; let's look at Delta01's other actions:
Map[(GetNPCActionFrequencies[#, Delta01]) & AllFARSessions]

Out:

{{Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}},
  {Delta01, {}}, {Delta01, {}}, {Delta01, {GotoValidPosition, 1/6}},
  {Follow, 1/6}, {UseSmartObjectNode, 1/3}, {GotoNode, 1/3},
  {GotoValidPosition, 1/6}, {Follow, 1/6}, {UseSmartObjectNode, 1/3}, {GotoNode, 1/3},
  {Delta01, {GotoValidPosition, 1/5}, {UseSmartObjectNode, 2/5}, {GotoNode, 2/5}},
  {Delta01, {UseSmartObjectNode, 1/2}, {Animate, 1/2}},
  {GotoValidPosition, 1/6}, {Follow, 1/6}, {UseSmartObjectNode, 1/3}, {GotoNode, 1/3},
  {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {GotoValidPosition, 1/6}},
  {Follow, 1/6}, {UseSmartObjectNode, 1/3}, {GotoNode, 1/3},
  {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {GotoValidPosition, 1/6}},
  {Follow, 1/6}, {UseSmartObjectNode, 1/3}, {GotoNode, 1/3},
  {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {}}, {Delta01, {GotoValidPosition, 1/6}},
  {Follow, 1/6}, {UseSmartObjectNode, 1/3}, {GotoNode, 1/3}]

Delta01 appears in F.E.A.R. with the demolitions expert Douglas Holiday during "Mission 9: Bishop" [1], immediately after a dogfight in an elevator lobby; as an SFOD-D teammate, its rôle is to escort and protect Holiday while the player leads them to Aldus Bishop and his explosive belt. As Holiday deals with Bishop, Delta01 walks around the room, stopping from time to time, around the peaks shown in the above plots.

Delta01 waiting for Holiday to blast a wall.
The "actions" need for planning

We now turn to investigate the need for planning when considering three sets of actions: (i) defensive actions, (ii) offensive actions and (iii) intermediate actions, hoping for an answer to: is there more pressure to plan for defensive, offensive or intermediate actions? The plots below show that there is more pressure on planning to attack than on planning to escape danger; but probably the most important result of this section is that the need for planning in F.E.A.R. is largely due to the pressure on planning for intermediate actions (which concern moving, inspecting and animation purposes).
First, let's get all the actions of all the sessions:

```math
In[97]:= AllActions = Apply[Union, Map[GetActions, AllFEARSessions]]
```

```math
```

How many actions are there?

```math
In[98]:= Length[AllActions]
```

```math
Out[98]= 55
```

We now build the set of actions whose objective is to lower danger:

```math
```

What is the proportion of defensive actions across all sessions (sorted in increasing order)?

```math
In[100]:= da = Sort[Cases[GetActionFrequencies[Apply[Join, AllFEARSessions]], {Apply[Alternatives, DefensiveActions], _}], (Last[H1] < Last[H2]) &]
```

```math
Out[100]= {{{DodgeRollParanoid, \( \frac{3}{9785} \)}, {LongRecoilHelmetPiercing, \( \frac{9}{9785} \)}, {FlushOutWithGrenade, \( \frac{13}{9785} \)}, {SuppressionFire, \( \frac{14}{9785} \)}, {DodgeCovered, \( \frac{3}{1957} \)}, {LongRecoilExplosive, \( \frac{17}{9785} \)}, {DodgeShuffle, \( \frac{22}{9785} \)}, {SuppressionFireFromCover, \( \frac{71}{9785} \)}, {LongRecoilBullet, \( \frac{16}{1957} \)}, {EscapeDanger, \( \frac{213}{9785} \)}}
```

```math
In[101]:= Length[DefensiveNPCsPR = Sort[Apply[Join, Map:GetAllNPCsPlanningRanges, Map[(Cases[H, {___, Apply[Alternatives, DefensiveActions], ___}], Last[H1] < Last[H2]) &], AllFEARSessions]]]
```

```math
Out[101]= 256
```

```math
In[102]:= Length[DefensiveAPR = Sort[Apply[Join, Map:GetSessionPlanningRanges, Map[(Cases[H, {___, Apply[Alternatives, DefensiveActions], ___}], Last[H1] < Last[H2]) &], AllFEARSessions]]]
```

```math
Out[102]= 441
```

Let's now plot the need for planning for these defensive actions:

```math
In[103]:= PlotTheNeedForPlanning[DefensiveNPCsPR, DefensiveAPR, 0, 12600, 200, 50 / 3]
```
We'll briefly comment the previous plots with the following plot about offensive actions below. As reloading actions are always associated with attacking actions in the sessions we study, we choose to integrate them in the set of offensive actions:

```math
OffensiveActions = \{\text{Attack, AttackCrouch, AttackFromAmbush, AttackFromArmored, AttacKFromArmoredBounded, AttackFromCover, AttackFromView, AttackGrenadeFromCover, AttackLungeUncloaked, AttackMelee, AttackMeleeUncloaked, AttackReady, AttackTurretCeiling, BlindFireFromCover, Charge, ReloadCovered, ReloadCrouch}\};
```

What is the proportion of offensive actions across all sessions (sorted in increasing order)?
In[105]:= \[0X0\] oa = Sort[Cases[GetActionFrequencies[Apply[Join, AllFEARSessions]], 
{Apply[Alternatives, OffensiveActions], _}], (Last[H1] < Last[H2]) &]

Out[105]= 

\{\{Charge, \frac{1}{9785}\}, \{\text{AttackCrouch}, \frac{2}{9785}\}, \{\text{AttackFromArmoredBounded}, \frac{4}{9785}\}, \\
\{\text{AttackLungeUncloaked}, \frac{1}{1957}\}, \{\text{ReloadCrouch}, \frac{6}{9785}\}, \{\text{AttackMeleeUncloaked}, \frac{7}{9785}\}, \\
\{\text{ReloadCovered}, \frac{11}{9785}\}, \{\text{AttackFromView}, \frac{13}{9785}\}, \{\text{AttackMelee}, \frac{1}{515}\}, \\
\{\text{AttackGrenadeFromCover}, \frac{24}{9785}\}, \{\text{AttackFromAmbush}, \frac{71}{9785}\}, \\
\{\text{BlindFireFromCover}, \frac{84}{9785}\}, \{\text{AttackFromArmored}, \frac{108}{9785}\}, \{\text{AttackTurretCeiling}, \frac{25}{1957}\}, \\
\{\text{Attack}, \frac{153}{9785}\}, \{\text{AttackReady}, \frac{63}{1957}\}, \{\text{AttackFromCover}, \frac{546}{9785}\}\}\]

In[106]:= \[0X0\] Length[OffensiveNPCsPR = Sort[Apply[Join, Map[GetAllNPCsPlanningRanges, Map[(Cases[#, {___, Apply[Alternatives, OffensiveActions], ___]}) &, AllFEARSessions]]]]]

Out[106]= 1245

In[107]:= \[0X0\] Length[OffensiveAPR = Sort[Apply[Join, Map[GetSessionPlanningRanges, 
Map[(Cases[#, {___, Apply[Alternatives, OffensiveActions], ___]}) &, 
AllFEARSessions]]]]]

Out[107]= 1460

The next plots show how the pressure on attacking is much more important than the pressure on defending (see above plots); not only are the curves more chaotic, showing a more important need for planning even after several seconds, but the curves also show a continuous need for planning when attacking, with no real disruption:

In[108]:= \[0X0\] PlotTheNeedForPlanning[OffensiveNPCsPR, OffensiveAPR, 0, 12600, 200, 50/3]
% of planning time over 6 frames

% of planning time over 7 frames

% of planning time over 8 frames

% of planning time over 9 frames
We end with actions which are neither offensive nor defensive:

\[
\text{IntermediateActions} = \text{Complement} [\text{AllActions}, \text{DefensiveActions}, \text{OffensiveActions}]
\]

\[
\text{Out} = [\text{Animate}, \text{DismountNodeUncloaked}, \text{DismountVehicle}, \text{FaceNode}, \text{Follow}, \text{FollowPlayer}, \text{GotoNode}, \text{GotoNodeOfType}, \text{GotoTarget}, \text{GotoTargetLost}, \text{GotoValidPosition}, \text{Idle}, \text{IdleOnVehicle}, \text{IdleTurret}, \text{InspectDisturbance}, \text{KnockDownBullet}, \text{KnockDownExplosive}, \text{LookAtDisturbance}, \text{LookAtDisturbanceFromView}, \text{MountNodeUncloaked}, \text{MountVehicle}, \text{ReactToDanger}, \text{SurveyArea}, \text{TraverseBlockedDoor}, \text{TraverseLink}, \text{TraverseLinkUncloaked}, \text{UseSmartObjectNode}, \text{UseSmartObjectNodeMounted}]
\]

What is the proportion of these intermediate actions across all sessions (sorted in increasing order)?

\[
\text{ia} = \text{Sort} [\text{Cases}[\text{GetActionFrequencies}[\text{Apply}[\text{Join}, \text{AllFEARSessions}]]],
\quad \text{Apply}[\text{Alternatives}, \text{IntermediateActions}],_2], (\text{Last}[1] < \text{Last}[2]) \&]
\]

\[
\text{Out} = \begin{align*}
\text{KnockDownExplosive}, & 19785 \\
\text{FollowPlayer}, & 39785 \\
\text{KnockDownBullet}, & 79785 \\
\text{DismountNodeUncloaked}, & 79785 \\
\text{MountVehicle}, & 89785 \\
\text{LookAtDisturbanceFromView}, & 99785 \\
\text{IdleOnVehicle}, & 21957 \\
\text{GotoTargetLost}, & 21957 \\
\text{DismountVehicle}, & 21957 \\
\text{TraverseLinkUncloaked}, & 169785 \\
\text{MountNodeUncloaked}, & 219785 \\
\text{GotoNodeOfType}, & 229785 \\
\text{UseSmartObjectNodeMounted}, & 239785 \\
\text{GotoValidPosition}, & 269785 \\
\text{TraverseBlockedDoor}, & 429785 \\
\text{Follow}, & 101957 \\
\text{SurveyArea}, & 519785 \\
\text{IdleTurret}, & 589785 \\
\text{ReactToDanger}, & 699785 \\
\text{TraverseLink}, & 1199785 \\
\text{FaceNode}, & 1239785 \\
\text{LookAtDisturbance}, & 1399785 \\
\text{Animet}, & 2499785 \\
\text{Idle}, & 5269785 \\
\text{InspectDisturbance}, & 5779785 \\
\text{GotoTarget}, & 6389785 \\
\text{GotoNode}, & 24549785 \\
\text{UseSmartObjectNode}, & 25669785
\end{align*}
\]

\[
\text{Length}[
\text{IntermediateNPCsPR} = \text{Sort} [\text{Apply}[\text{Join}, \text{Map}[\text{GetAllNPCsPlanningRanges}, \text{Map}[\{\text{Cases}[#, \{\text{___}, \text{Apply}[\text{Alternatives}, \text{IntermediateActions}], \text{___}\}]\} & \text{AllFEARSessions}]]]]]
\]

\[
\text{Out} = 4419
\]

\[
\text{Length}[
\text{IntermediateAPR} = \text{Sort} [\text{Apply}[\text{Join}, \text{Map}[\text{GetSessionPlanningRanges}, \text{Map}[\{\text{Cases}[#, \{\text{___}, \text{Apply}[\text{Alternatives}, \text{IntermediateActions}], \text{___}\}]\} & \text{AllFEARSessions}]]]]]
\]

\[
\text{Out} = 4741
\]

From the following plots, it is rather immediate to observe that the need for planning we studied in the previous subsection is due to actions which concern non-fighting moments in F.E.A.R.:
Actions mostly responsible for the planning effort

We end by gathering, on the same plot, the proportions of intermediate (blue), offensive (red) and defensive (green) actions (the x-axis is normalized and corresponds to the percentage of actions in one of the three sets of actions, with respect to their increasing proportions; the y-axis is the proportion of an action in the generated plans, across all sessions):

```
In[114]:= Show[ListPlot[Module[{t = Apply[Plus, da[[All, 2]]]}, 100.
   Table[{i/Length[da], Apply[Plus, Take[da[[All, 2]], -i]]/t, {i, 1, Length[da]}]],
   PlotStyle -> {RGBColor[0, 1, 0], PointSize[0.015]}],
   ListPlot[Module[{t = Apply[Plus, ia[[All, 2]]]}, 100.
   Table[{i/Length[ia], Apply[Plus, Take[ia[[All, 2]], -i]]/t, {i, 1, Length[ia]}]],
   PlotStyle -> {RGBColor[0, 0, 1], PointSize[0.015]}],
   ListPlot[Module[{t = Apply[Plus, oa[[All, 2]]]}, 100.
   Table[{i/Length[oa], Apply[Plus, Take[oa[[All, 2]], -i]]/t, {i, 1, Length[oa]}]],
   PlotStyle -> {RGBColor[1, 0, 0], PointSize[0.015]}],
   PlotRange -> All, Frame -> True, GridLines -> Automatic]
```

We observe that only a small percentage of the actions are responsible for planning ("responsible" as these actions made it in a plan: the planning component effort ended in inserting these actions in the plans given to the NPCs): 15% of the intermediate actions and about 30% of both defensive and offensive actions are reponsi-
apply for about 80% of the planning effort for these actions.

```
In[115]:= Module[{l = Take[da, -3]}, {l, N[100 Apply[Plus, 1[[All, 2]]] / Apply[Plus, da[[All, 2]]]]}]
```

```
Out[115]= {SuppressionFireFromCover, 71/9785, 16/957, EscapeDanger, 213/9785, 79.6499}
```

What is the length of the plans in which SuppressionFireFromCover, LongRecoilBullet and EscapeDanger are involved?

```
In[116]:= Map[Function[{action}, {action, Union[Map[Last, Select[Apply[Join, AllFEARSessions], (MemberQ[Flatten[#], action]] &]]]], Take[da, -3][[All, 1]]]
```

```
Out[116]= {{SuppressionFireFromCover, {1}}, {LongRecoilBullet, {1}}, {EscapeDanger, {1}}}
```

Here are offensive actions for which planning happens the most:

```
In[117]:= Module[{l = Take[oa, -4]}, {l, N[100 Apply[Plus, 1[[All, 2]]] / Apply[Plus, oa[[All, 2]]]]}]
```

```
```

What is the length of the plans in which AttackTurretCeiling, Attack, AttackReady and AttackFromCover are involved?

```
In[118]:= Map[Function[{action}, {action, Union[Map[Last, Select[Apply[Join, AllFEARSessions], (MemberQ[Flatten[#], action]] &]]]], Take[oa, -4][[All, 1]]]
```

```
Out[118]= {{AttackTurretCeiling, {1}}, {Attack, {1, 2}}, {AttackReady, {1}}, {AttackFromCover, {1, 2}}}
```

And here is the set intermediate actions for which planning happens the most:

```
In[119]:= Module[{l = Take[ia, -6]}, {l, N[100 Apply[Plus, 1[[All, 2]]] / Apply[Plus, ia[[All, 2]]]]}]
```

```
Out[119]= {Animate, 249/9785, Idle, 526/9785, InspectDisturbance, 577/9785, GotoTarget, 638/9785, GotoNode, 2545/9785, UseSmartObjectNode, 2566/9785, 89.4817}
```

What is the length of the plans in which Animate, Idle, InspectDisturbance, GotoTarget, GotoNode and UseSmartObjectNode are involved?

```
In[120]:= Map[Function[{action}, {action, Union[Map[Last, Select[Apply[Join, AllFEARSessions], (MemberQ[Flatten[#], action]] &]]]], Take[ia, -6][[All, 1]]]
```

```
Out[120]= {Animate, {1}}, {Idle, {1, 2}}, {InspectDisturbance, {2, 3}}, {GotoTarget, {1, 2, 3}}, {GotoNode, {1, 2, 3, 4}}, {UseSmartObjectNode, {1, 2, 4}}
```

We can understand these sets of actions as a result of a design decision: NPCs shall mostly plan for staying alive (escaping danger or recoiling), even while attacking (cf. the greatest importance for attacking from cover); and it is worth noting the importance of Turrets for attacking, while the proportions for GotoNode and UseSmartObjectNode corroborate the previous study of the peaks in the need for planning.

If these sets of actions, for which planning happens the most, were to be different for each level of the game, the above curves would show a flat look, with a higher average proportion value on the y-axis. We could also think of a steady increase of proportions instead of an important increase for the first 30% as observed in the above curves.

A last word about the above plan lengths: all of the defensive actions appear in plans of length 1 (and not only the first 30%) whereas all offensive actions can appear in plans of length 1 or 2 (very unfrequent actions such as reloading actions always appear in plans of length 2). As for the need for planning, we could again conclude that planning happens more for offensive actions than for defensive actions, but this result is also related to the...
granularity of these actions, which, again, is a design decision: smaller grain defensive actions would entail longer defensive plans.

Conclusions

In this study, we observed plans in F.E.A.R. are never longer than 4 actions while almost 99% of the plans generated during playing the game are 1 or 2-action long; we also discovered there are plans in F.E.A.R. which are always generated with the same sequence of actions, thus suggesting that these could be implemented as "fixed" plans such as HTNs [2]. We also found that the need for planning is greater when attacking than when defending, although the overall need for planning in F.E.A.R. was due to intermediate actions, from surveillance to animation purposes. We eventually discovered that 20% of the 55 available actions are responsible for 80% of the planning activity.

Acknowledgments

This work could have never been achieved without Jeff Orkin's implementation of Artificial Intelligence Planning in F.E.A.R. [3,4] and Monolith's release of the "F.E.A.R. Public Tools"; moreover, many thanks to Jeff Orkin for answering all my questions. Many thanks also to Arjen Beij (Guerilla Games) for his interest in this project and his questions about the peaks, which greatly improved this report; and to Alex Champandard (AIGameDev.com) and Carle Coté (Eidos Montréal) for their comments and feedback.

References


Appendix 1: the costs of actions in F.E.A.R.

In this study we viewed F.E.A.R. as a black box. We only considered the information dynamically generated and then recorded in the session files while playing the game. We never looked at static information about the game such as action costs, although used by the planning component as it looks for the cheapest plan.

The costs of the 55 actions from the session files, given to sort in alphabetical order and sorted in increasing cost, are the following:
OffCosts = Sort[
    Map[(First[Cases[ActionCosts, _]]) &, DefensiveActions], (Last[#1] < Last[#2]) &]

Out[123]=

{{ReloadCovered, 1}, {Charge, 1}, {AttackLungeUncloaked, 1}, {BlindFireFromCover, 2}, 
  {AttackGrenadeFromCover, 2}, {AttackMeleeUncloaked, 3}, {AttackMelee, 3}, 
  {AttackFromArmored, 4}, {AttackFromArmoredBounded, 4}, 
  {GotoTarget, 4}, {ReloadCrouch, 4}, {GotoTargetLost, 8}, 
  {TraverseLink, 2}, {TraverseLinkUncloaked, 1}, 
  {UseSmartObjectNode, 3}, {UseSmartObjectNodeMounted, 1}, 
  {Last[#1] < Last[#2]) &}

Now here are the actions cost of the three action classes we studied in the last section: defensive actions, 
offensive actions and intermediate actions (animation, patrolling, etc):

DefCosts = Sort[
    Map[(First[Cases[ActionCosts, _]]) &, DefensiveActions], (Last[#1] < Last[#2]) &]

Out[122]=

{{EscapeDanger, 0.5}, {SuppressionFireFromCover, 1}, 
  {DodgeCovered, 1}, {SuppressionFire, 2}, {DodgeRollParanoid, 2}, 
  {LongRecoilHelmetPiercing, 3}, {LongRecoilExplosive, 3}, 
  {LongRecoilBullet, 3}, {Follow, 3}, {FollowPlayer, 2}, 
  {GotoNode, 1}, {GotoValidPosition, 1}, {Idle, 2}, {IdleOnVehicle, 1}, {IdleTurret, 2}, 
  {InspectDisturbance, 2}, {KnockDownExplosive, 2}, 
  {KnockDownBullet, 2}, {ReactToDanger, 1}, {ReloadCovered, 1}, 
  {ReloadCrouch, 5}, {AttackLungeUncloaked, 1}, 
  {AttackLungeUncloaked, 1}, {Animat, 1}, {LookAtDisturbance, 1.5}, 
  {LongRecoilHelmetPiercing, 3}, {LongRecoilExplosive, 3}, 
  {LongRecoilBullet, 3}, {Follow, 3}, {FlushOutWithGrenade, 3}, 
  {DodgeRollParanoid, 2}, {BlindFireFromCover, 2}, {AttackGrenadeFromCover, 2}, 
  {UseSmartObjectNode, 3}, {UseSmartObjectNodeMounted, 1}, 
  {UseSmartObjectNode, 3}, {UseSmartObjectNodeMounted, 1}, 
  {Last[#1] < Last[#2]) &}

Out[121]=

Sort[ActionCosts = {{Animate, 1}, {Attack, 6}, {AttackCrouch, 5}, 
  {AttackFromAmbush, 4}, {AttackFromArmored, 4}, {AttackFromArmoredBounded, 4}, 
  {AttackFromCover, 4}, {AttackFromView, 4.5}, {AttackGrenadeFromCover, 2}, 
  {AttackLungeUncloaked, 1}, {AttackMelee, 3}, {AttackMeleeUncloaked, 3}, 
  {AttackReady, 7}, {AttackTurretCeiling, 6}, {BlindFireFromCover, 2}, 
  {Charge, 1}, {DismountNodeUncloaked, 1}, {DismountVehicle, 1}, {DodgeCovered, 1}, 
  {DodgeRollParanoid, 2}, {DodgeShuffle, 3}, {EscapeDanger, 0.5}, 
  {FaceNode, 1}, {FlushOutWithGrenade, 3}, {Follow, 3}, {FollowPlayer, 2}, 
  {GotoNode, 1}, {GotoValidPosition, 1}, {Idle, 2}, {IdleOnVehicle, 1}, {IdleTurret, 2}, 
  {InspectDisturbance, 2}, {KnockDownExplosive, 2}, 
  {KnockDownBullet, 2}, {ReactToDanger, 1}, {ReloadCovered, 1}, 
  {ReloadCrouch, 5}, {ReloadCrouch, 4}, {UseSmartObjectNodeMounted, 1}, 
  {UseSmartObjectNode, 3}, {UseSmartObjectNodeMounted, 1}, 
  {Last[#1] < Last[#2]) &}
\begin{verbatim}
In[124]:= IntCosts = Sort@
  Map[(First[Cases[ActionCosts, {#, _}]] &), IntermediateActions], (Last[#1] < Last[#2]) &]
Out[124]= {{UseSmartObjectNodeMounted, 1}, {TraverseLinkUncloaked, 1},
  {TraverseBlockedDoor, 1}, {SurveyArea, 1}, {ReactToDanger, 1}, {MountVehicle, 1},
  {MountNodeUncloaked, 1}, {IdleOnVehicle, 1}, {GotoValidPosition, 1},
  {GotoNodeOfType, 1}, {GotoNode, 1}, {FaceNode, 1}, {DismountVehicle, 1},
  {DismountNodeUncloaked, 1}, {Animate, 1}, {LookAtDisturbance, 1.5}, {TraverseLink, 2},
  {KnockDownExplosive, 2}, {KnockDownBullet, 2}, {InspectDisturbance, 2},
  {IdleTurret, 2}, {Idle, 2}, {FollowPlayer, 2}, {UseSmartObjectNode, 3},
  {LookAtDisturbanceFromView, 3}, {Follow, 3}, {GotoTarget, 4}, {GotoTargetLost, 8}}

The following plot gathers the actions costs (y-axis) according to the three classes and the colors we used
(blue for intermediate, red for offensive and green for defensive); we observe that it's cheaper to defend than to
attack, thus reinforcing that there is less pressure on planning to defend than on planning to attack:
\end{verbatim}

\begin{verbatim}
In[125]:= Show[ ListPlot[DefCosts[[All, 2]], Joined -> True, PlotLines -> {RGBColor[0, 1, 0], Thickness[0.02]}, PlotRange -> All, GridLines -> Automatic, Frame -> True], ListPlot[OffCosts[[All, 2]], Joined -> True, PlotStyle -> {RGBColor[1, 0, 0], Thickness[0.01]}, PlotRange -> All, GridLines -> Automatic, Frame -> True], ListPlot[IntCosts[[All, 2]], Joined -> True, PlotStyle -> {RGBColor[0, 0, 1], Thickness[0.01]}, PlotRange -> All, GridLines -> Automatic, Frame -> True]]
\end{verbatim}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{plot.png}
\caption{Actions costs according to classes and colors: blue for intermediate, red for offensive, green for defensive.}
\end{figure}

\section*{Appendix 2: NPC casting of the session files}

In this section, we visualize the appearance of NPCs along all session files as we did at the beginning of the
section titled “Actions and Plans” for session number 1123. We first recall the session files numbers:
\begin{verbatim}
In[126]:= isn
Out[126]= {1082, 1092, 1112, 1123, 1132, 1142, 1151, 1201, 1202, 1222, 1232, 12332, 1242, 1252, 1262, 1272, 1282}
\end{verbatim}

All these session files correspond to about 3 hours and 45 minutes sampled from playing F.E.A.R. between
level 2 (“Infiltration”) and level 11 (“Sayonara, Sucker”) (see [1] for details about these levels). We can observe
in the plots (x-axis is in seconds, y-axis represents NPCs) below that some NPCs seem to appear all the time
and with some kind of regularity; this is the case for rats, for instance. But the most noticeable is that NPCs
generally appear in small squads (3 or 4), thus leaving the time to the player to deal with these small waves of
enemies:
\begin{verbatim}
In[127]:= GraphicsColumn[Map[SessionCasting, AllFEARSessions]]
\end{verbatim}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{session_files.png}
\caption{Session files casting of NPCs.}
\end{figure}