A0M33BEP
Unmanned vehicles

Course 13: Flight trajectory planning

David Šišlák
sislakd@fel.cvut.cz

Motivation

- UAVs operations in dynamic large-scale environment

- **domain properties**:
  - 4D problem - 3D world considering time
  - excluded zones (positions and time) are given by physical obstacles and no-flight zones
  - excluded zones can be dynamically updated after each planning

- desired trajectory planner properties:
  - optimization-based search in 4D space
  - high performance
  - can be used for nonholonomic vehicle

State space search

- **algorithm properties**
  - time complexity – e.g. number of searched states
  - space complexity – e.g. number of states in memory
  - quality of solution – optimal, complete (find a solution if it exists)
  - effective branching factor – number of states after expansion

$$N = 1 + b^* + (b^*)^2 + \cdots + (b^*)^d$$

- **Uniformed** methods
  - breadth-first search (BFS)
  - depth-first search (DFS)
  - depth-limited search (DLS)
  - iterative deepening search (IDS)
  - bi-directional search
- **Informed** methods
  - use heuristics to select appropriate state to expand
  - good heuristics minimize searched state space to find optimal solution
  - best-first search – A* search, greedy algorithm
  - localized search – hill-climbing algorithm

Breadth-first search

```
begin
    open :=  [Start], closed :=  []
    while (open <>  []) do begin
        X :=  FIRST(open)
        closed := closed + [X], open := open - [X]
        if X = GOAL then return(SUCCESS)
        else begin
                E := expand(X)
                E := E - closed
               open := open + E
            end
   end
return(failure)
end.
```

Best-first search

```
begin
    open :=  [Start], closed :=  []
    while (open <>  []) do begin
        X :=  BEST(open)
        closed := closed + [X], open := open - [X]
        if X = GOAL then return(SUCCESS)
        else begin
                E := expand(X)
                E := E - closed
               open := open + E
            end
   end
return(failure)
end.
```

Search with heuristics

- operations on lines 9 and 10:
  - if a node is already in open
    - do nothing iff f(e) of the existing one is better
    - replace iff f(e) of the existing one is worse
  - if a node is already in closed
    - remove from E iff f(e) of the existing one is better
    - remove from closed iff f(e) of the existing one is worse

Evaluation function

- evaluation function f(…) used as a criterion in the algorithm
  - c(m,n) – cost of step from m to n
  - g(m) – cost of all steps from the start to the state m
  - h*(n) – real cost of all steps from the state n to the goal

- examples of evaluation functions
  - f(m,n) - evaluation function used for the transition from the state m to the state n

  - f(m,n) = c(m,n) … hill-climbing search – can stuck in a local minima
  - f(m,n) = h*(n) … greedy algorithm – ~~optimal~~, ~~complete~~
  - f(m,n) = g(n)+h*(n) where g(n) = g(m)+c(m,n) … A* algorithm
    – optimal, complete

A* algorithm

- » $h^*(n)$ is unknown and cannot be used directly
- » $h(n)$ is heuristics – estimation of $h^*(n)$ value
- » evaluation function is then $f(m,n) = g(n)+h(n)$ usually denoted as $f(n)$ only

- » admissible heuristics in A*
  - » for every n: $0 \leq h(n) \leq h^*(n)$
  - » guarantee optimality of A*

- » monotonic heuristics
  - » for every n1 and n2 where n2 is child of n1

    $$h(n1)-h(n2) \leq c(n1,n2) \text{ and } h(goal) = 0$$

  - » every monotonic heuristics is adminissible

- » dominance of heuristics
  - » h2 dominates over h1 iff for every n $h1(n) \leq h2(n)$
  - » A* with h2 will use less state space than with h1

A* algorithm remarks

- has exponential memory complexity
  - usually out of memory earlier than defined time limit

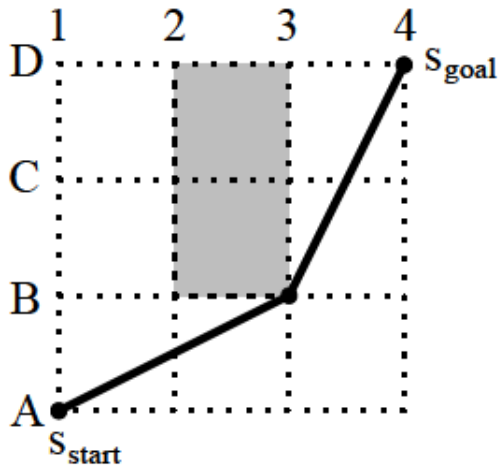
- iterative deepening A* (IDA*) – optimal, complete; reduces memory req.
  - limits the search branches with estimated maximum f for a solution

```
1.  f_limit=f(n_0)
2.  do A* with nodes of which f <= f_limit,
     in f_1 remember the smallest f >  f_limit
3.  if solution found
                          then return(solution)
                          else f_limit = f_1 and go to the step 2.
```

- memory bounded A* (MA*) – limits the size of open, removes worst state if no space
  - can stuck in local optimum

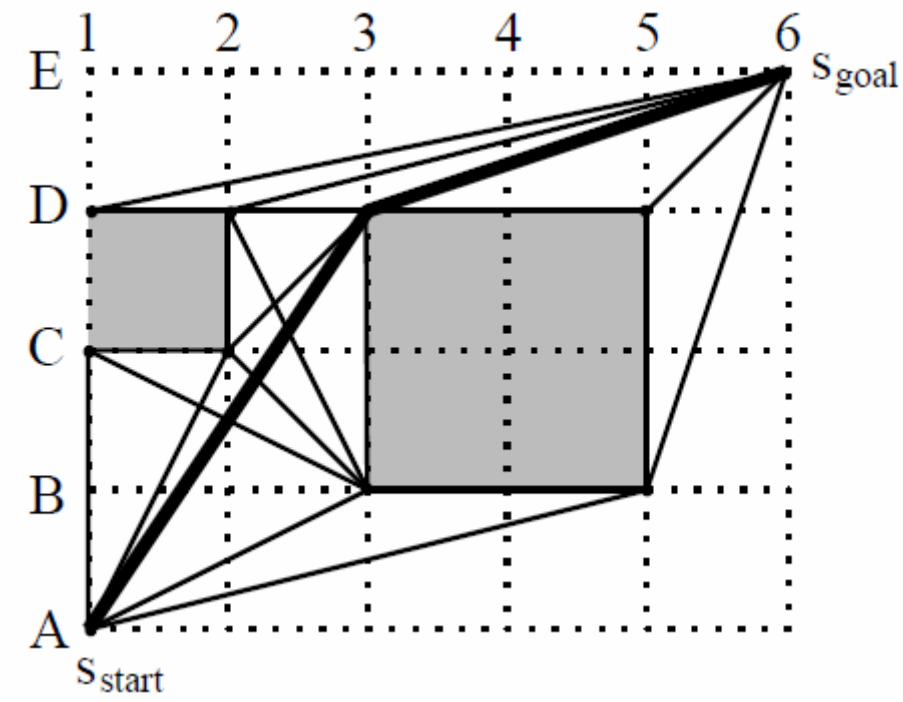
State space search for trajectory planning

- planning in a grid
  - 4-neighbors
  - 8-neighbors
  - any-angle
- no widely accepted common benchmarks for trajectory planners -> reduced comparison problem
- reduced problem – any-angle path planning in a grid
  - grid cells are either blocked or unblocked
  - start and goal locations are in grid vertices
  - path is a sequence of linked line elements which ends are in grid positions, any line element cannot intersect any blocked cell



A* algorithm in reduced domain

» search over visibility graph



» NODES is the set of all corner vertices + start and goal vertices

A* algorithm in reduced domain

```
Search(s_start, s_goal)
    g(s_start) ← 0;
    h(s_start) ← c(s_start, s_goal);
    parent(s_start) ← false;
    OPEN ← {s_start};
    CLOSED ← ∅;
    while OPEN ≠ ∅ do
        s_c ← RemoveTheBest(OPEN);
        if s_c = s_goal then return s_c;
        Insert(s_c, CLOSED);
        foreach s_d ∈ Candidates(s_c) do
            if Contains(s_d, CLOSED) then continue;
            if Intersect(s_c, s_d) then continue;
            g(s_d) ← g(s_c) + c(s_c, s_d);
            h(s_d) ← c(s_d, s_goal);
            parent(s_d) ← s_c;
            ProcessNode(s_d);
        end
    end
    return false;
end

Candidates(s_c)
    return NODES;
end

ProcessNode(s_d)
    InsertOrReplaceIfBetter(s_d, OPEN);
end
```

Theta* algorithm

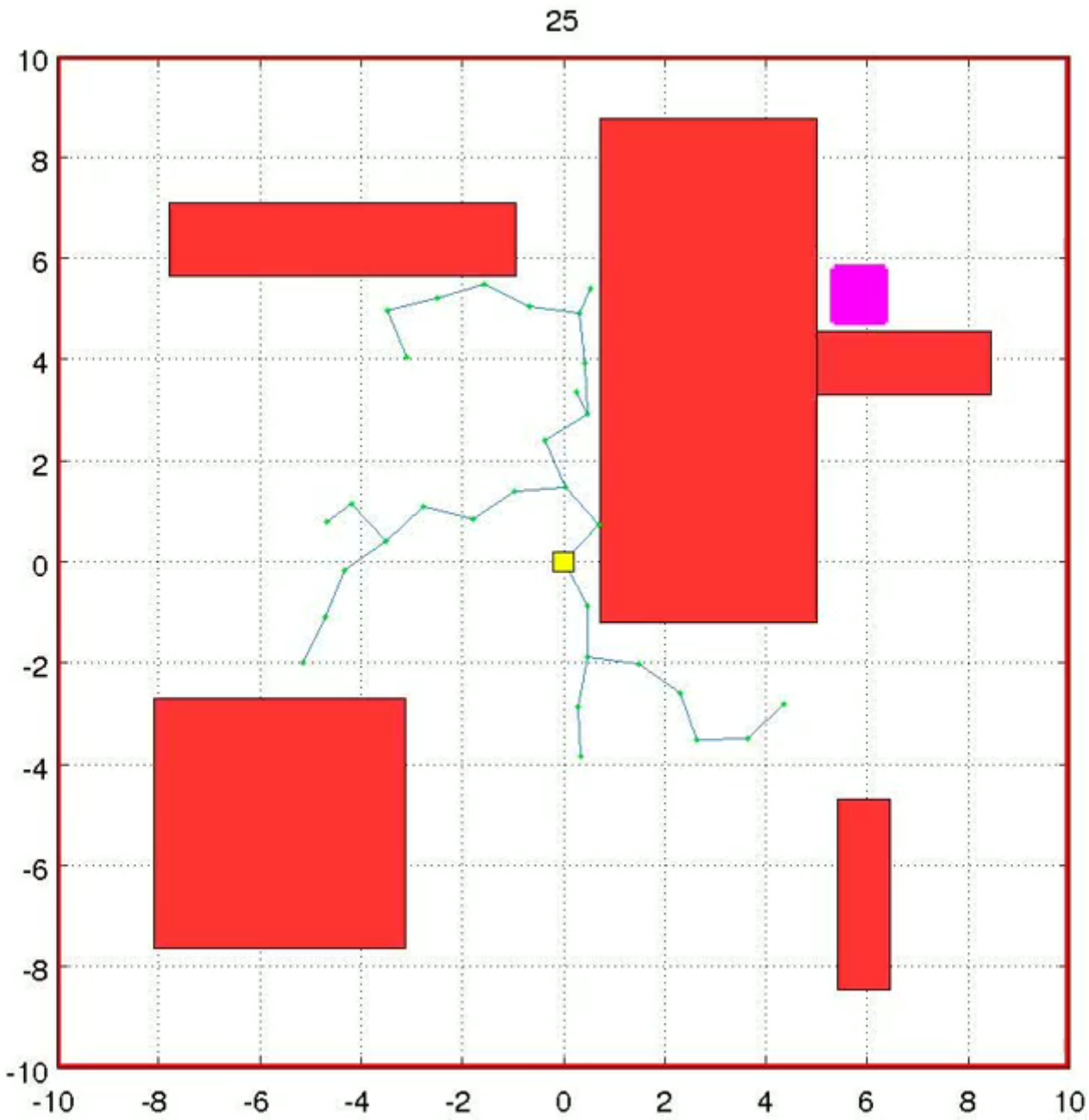
- works like A* algorithm except
  - generates only 4 children
  - reduces path after each expansion

- is not optimal

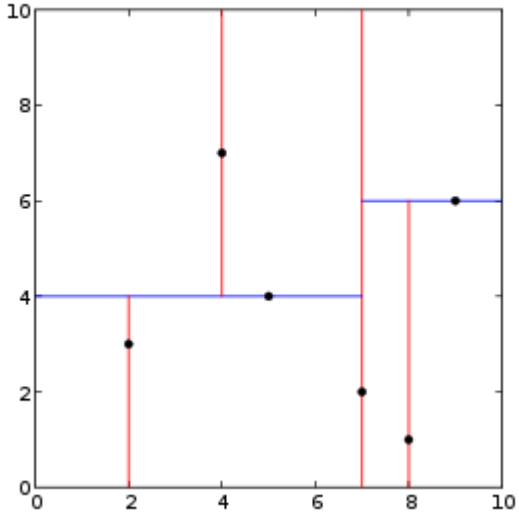
- but is faster than may other non-optimal versions
  - field D* (FD*) – which is using linear interpolation along grid-edges

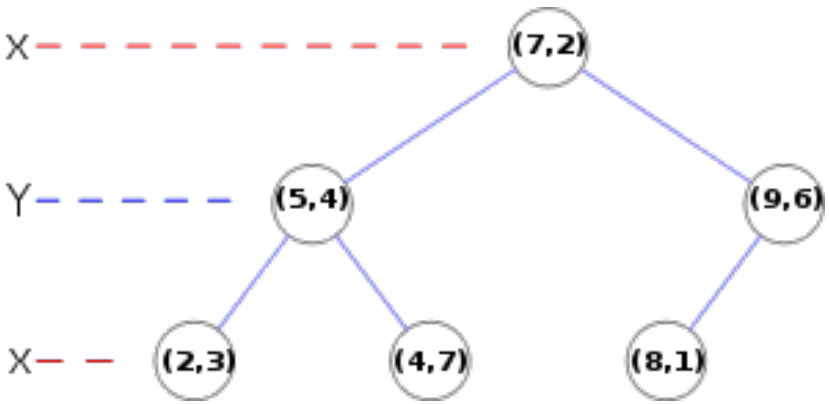
RRT – Rapid random trees search

```
RRT(s_start, s_goal)
    τ.init(s_start);
    for k=1 to K do
        s_rand ← RandomVertex();
        s_near ← τ.nearest(s_rand);
        s_new ← τ.stopping_configuration(s_near, s_rand);
        if s_new ≠ s_near then
            τ.add_vertex(s_new);
            τ.add_edge(s_near, s_new);
        end
        s_near ← τ.nearest(s_goal);
        if not Intersect(s_near, s_goal) then
            return τ, s_near;
    end
    return false;
end
```


25
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

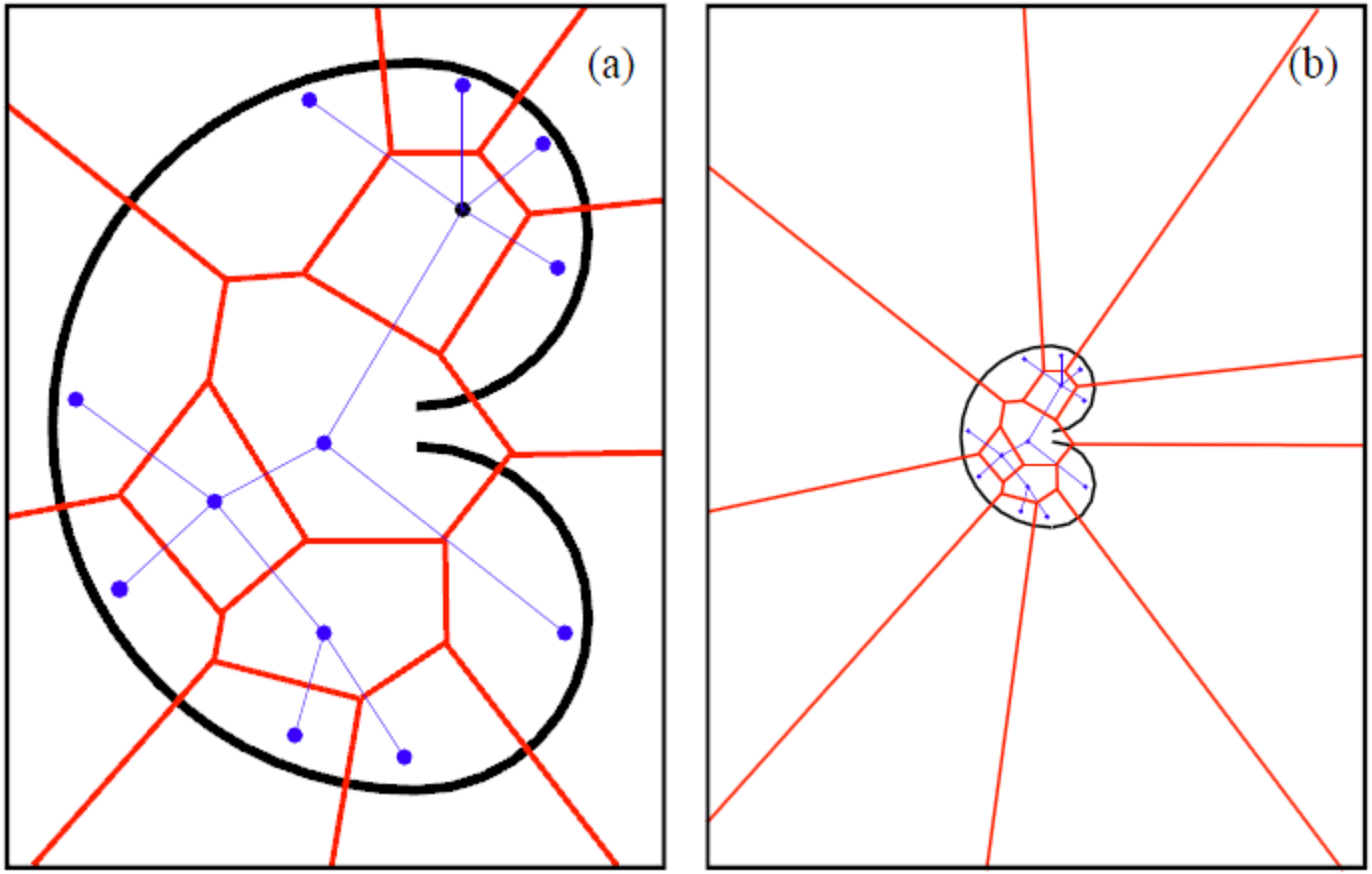
KD-tree structure



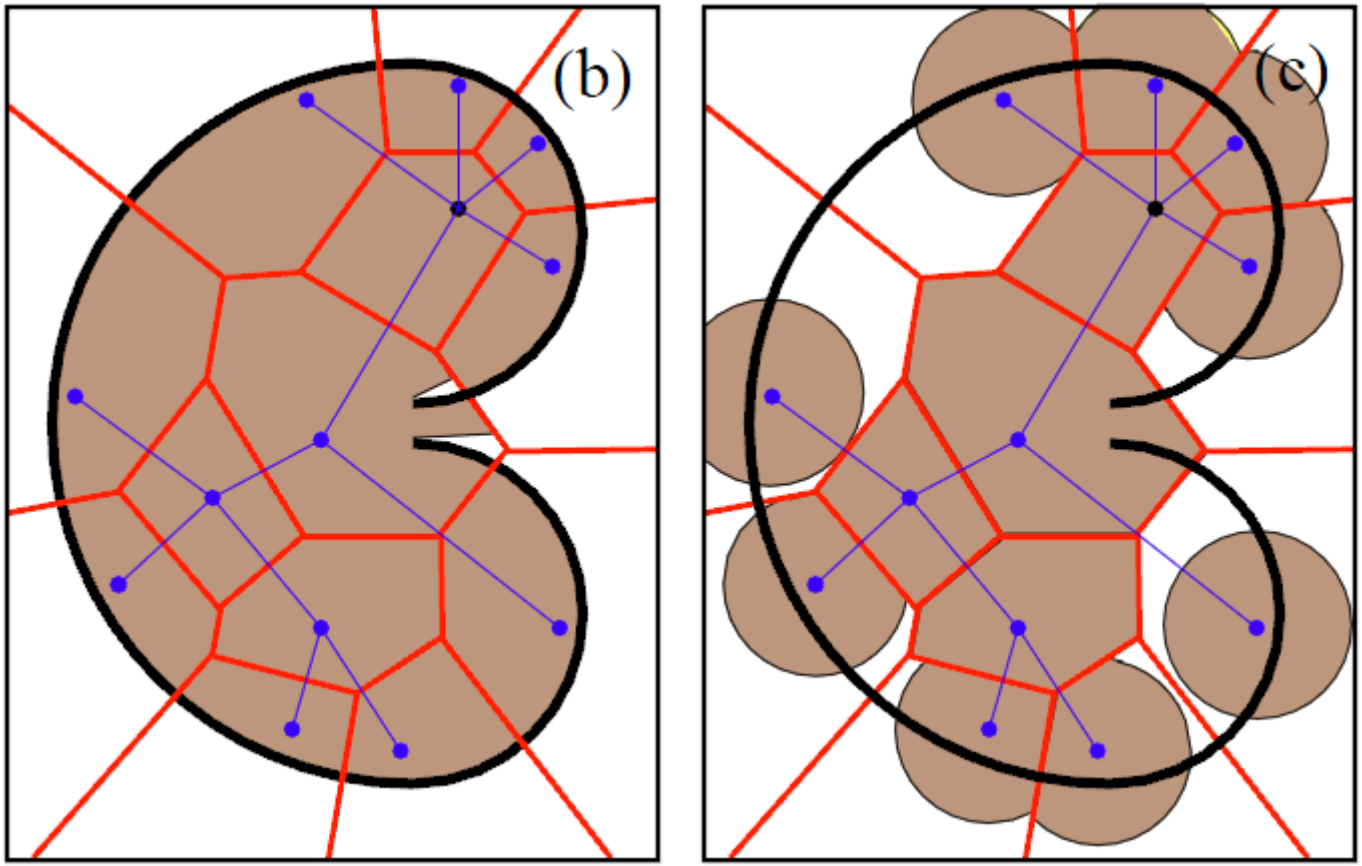


- complexity
  - insert new point O(log n)
  - remove point O(log n)
  - query nearest O(log n)

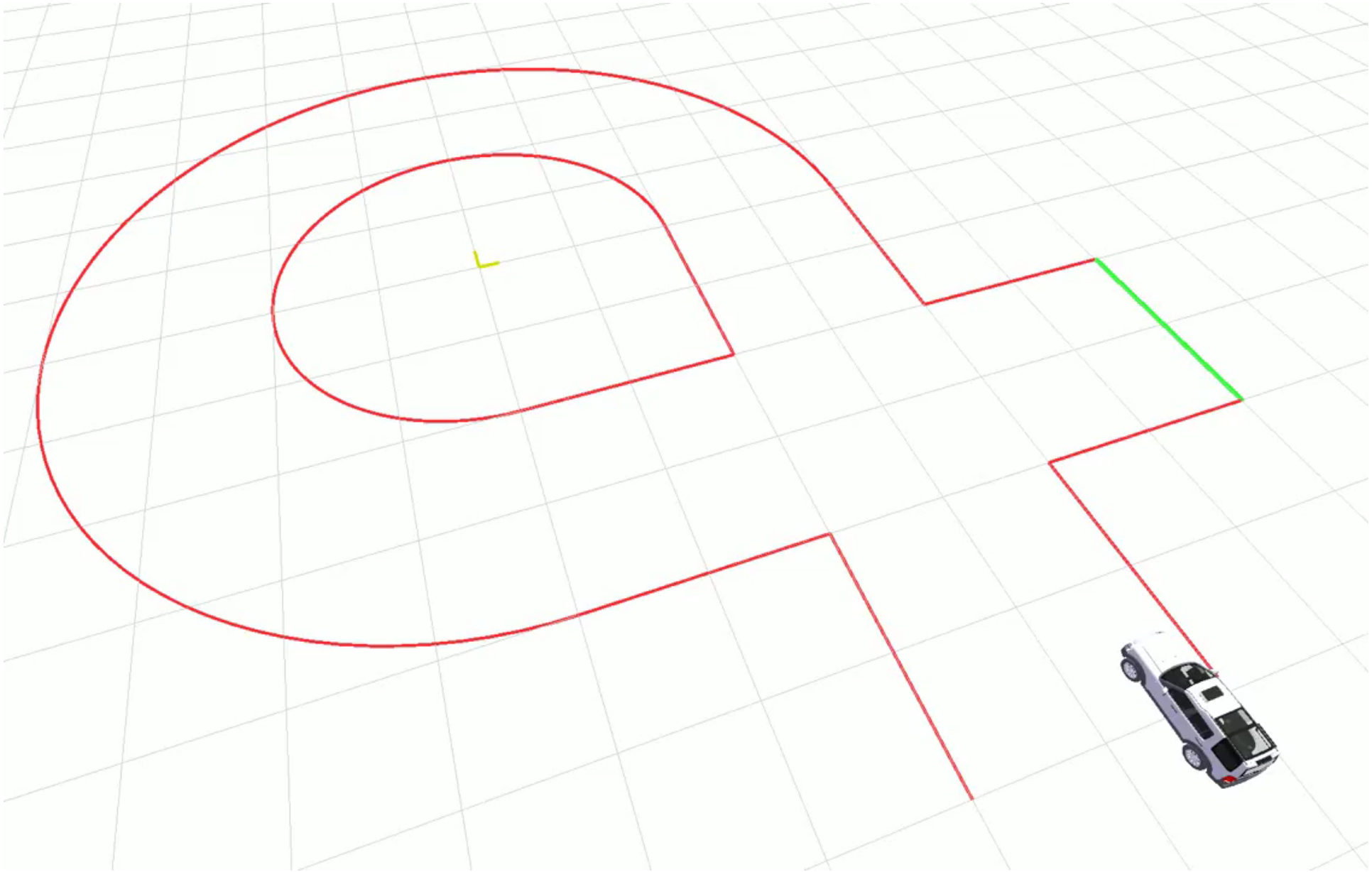
RRT – Rapid random trees search



RRT – Rapid random trees search



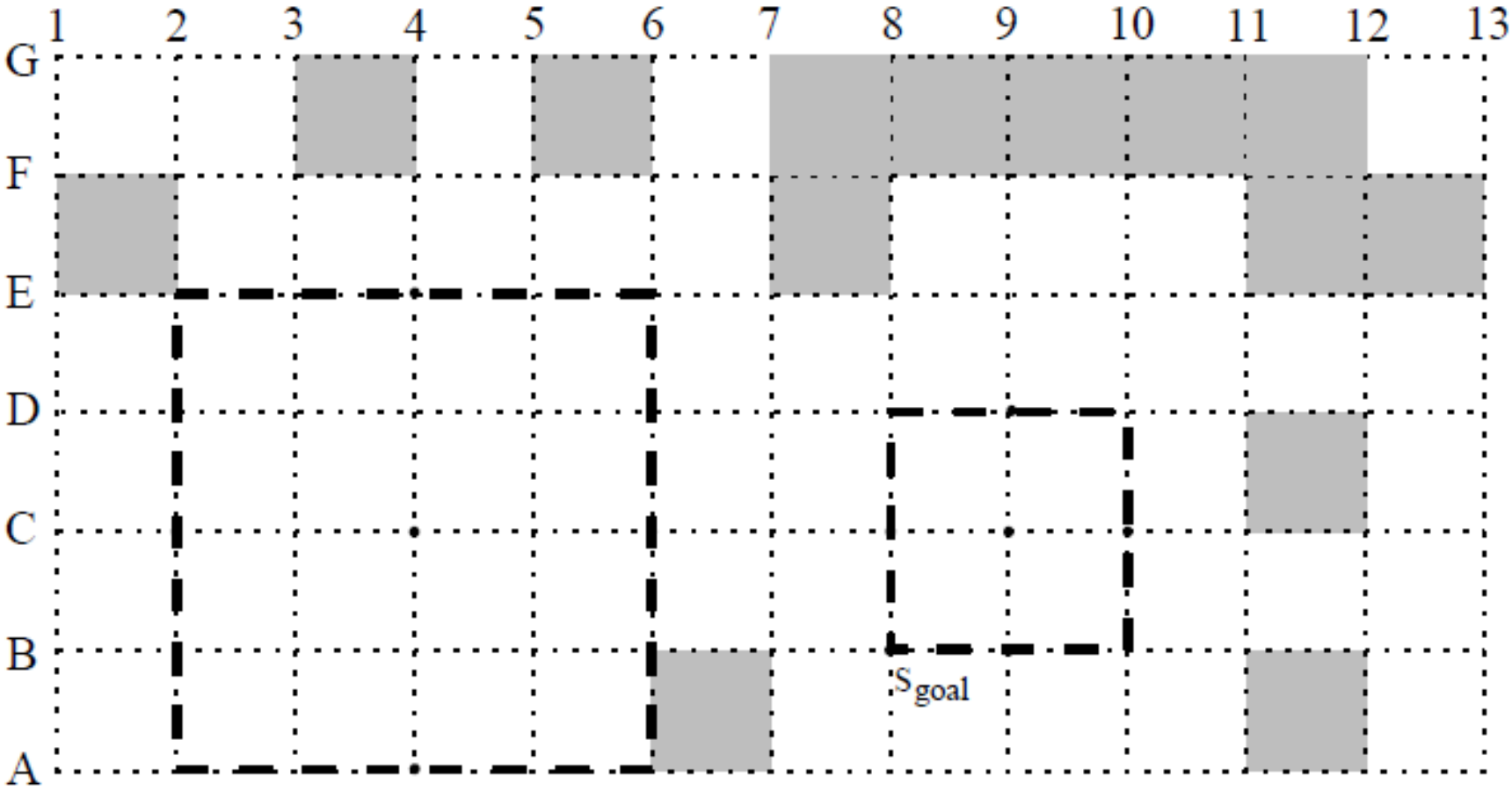
RRT – Rapid random trees search



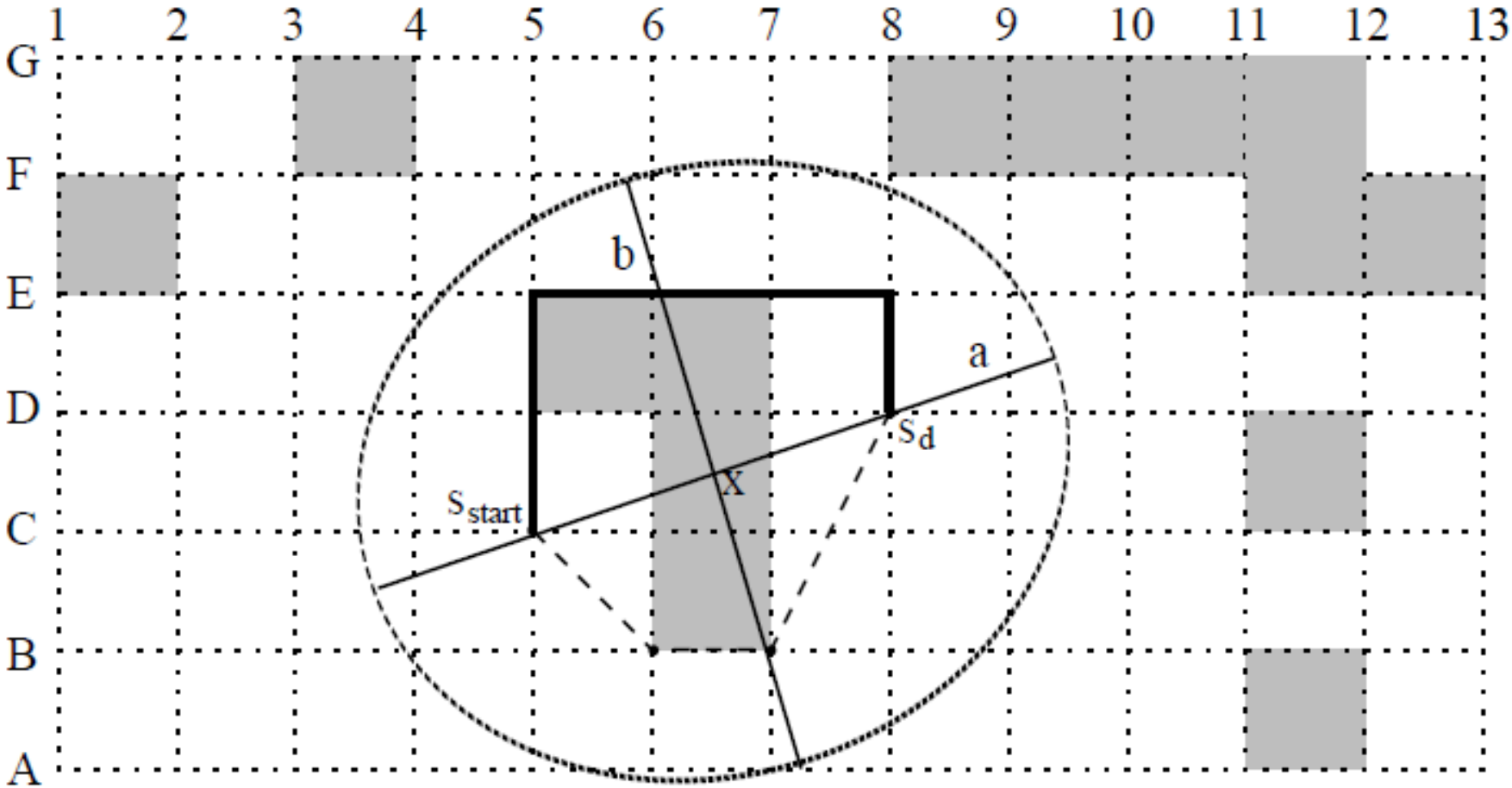
Accelerated A* (AA*)

- extension of A* algorithm
- no-preprocessing of an environment
- key concepts:
  - **adaptive state generation** using 4-successors at maximum -> reduction of branching factor to constant
  - search for any-angle path using **progressive path truncation** applied to every partial path represented by a state

AA* - adaptive state generation



AA* - progressive path truncation



AA* algorithm

```
Candidates(s_c)
    sq ← DetectMaxSquare(s_c);
    return UsableSideCenters(sq);
end

ProcessNode(s_d)
    foreach s_n ∈ EllipseMbs(CLOSED, s_start, s_d)
    do
        if g(s_n) + c(s_n, s_d) < g(s_d) then
            if not Intersect(s_n, s_d) then
                g(s_d) ← g(s_n) + c(s_n, s_d);
                parent(s_d) ← s_n;
            end
        end
    end
    InsertOrReplaceIfBetter(s_d, OPEN);
end
```

Evluation – randomized grids

- » grids size 100x100 with randomly blocked cells, start and goal positions
- » four different densities of obstacles: 5%, 10%, 20% and 30%
- » results averaged from 500 tasks for the same configurations provided to all algorithms
- » each generated task was validated by A* and then by others

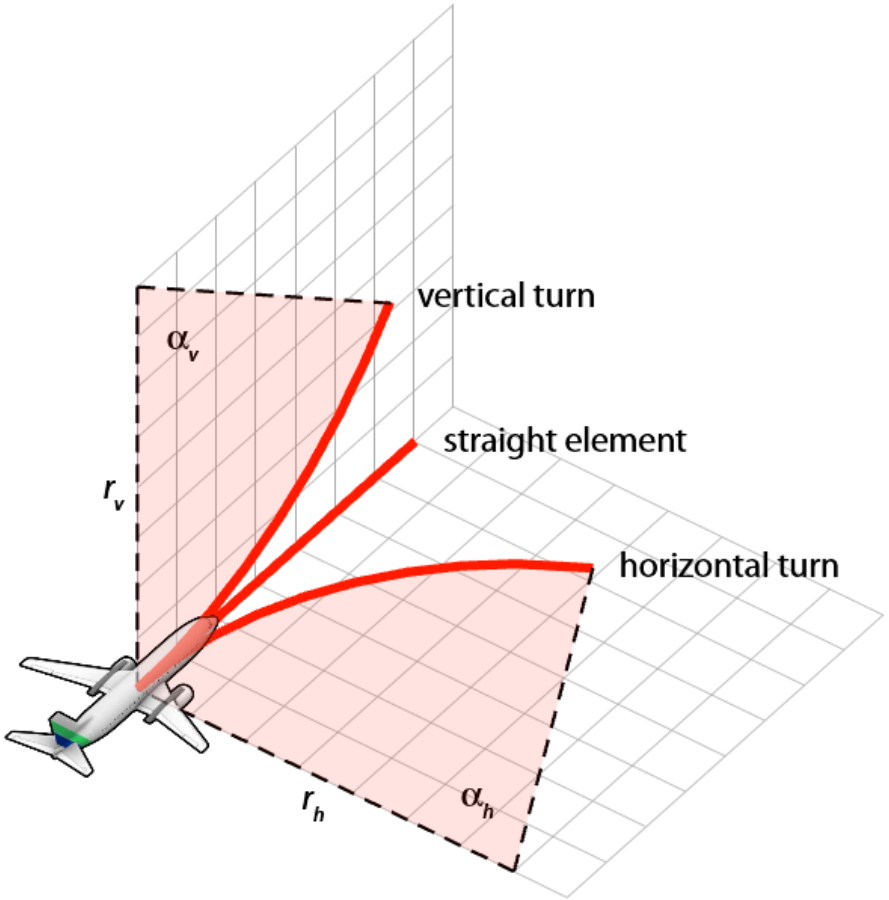
Configuration	Shortest Paths A*	Θ*	AA*	RRT PS	dynamic bi-RRT PS
5% blocked cells	54.210 (10.084)	54.345 (0.023)	54.210 (0.079)	73.834 (0.003)	73.411 (0.0004)
10% blocked cells	53.190 (11.896)	53.428 (0.026)	53.190 (0.082)	79.558 (0.013)	75.896 (0.0015)
20% blocked cells	53.301 (18.476)	53.623 (0.036)	53.301 (0.101)	85.207 (0.032)	78.030 (0.0037)
30% blocked cells	53.206 (31.493)	53.611 (0.049)	53.206 (0.129)	85.344 (0.077)	81.566 (0.0089)

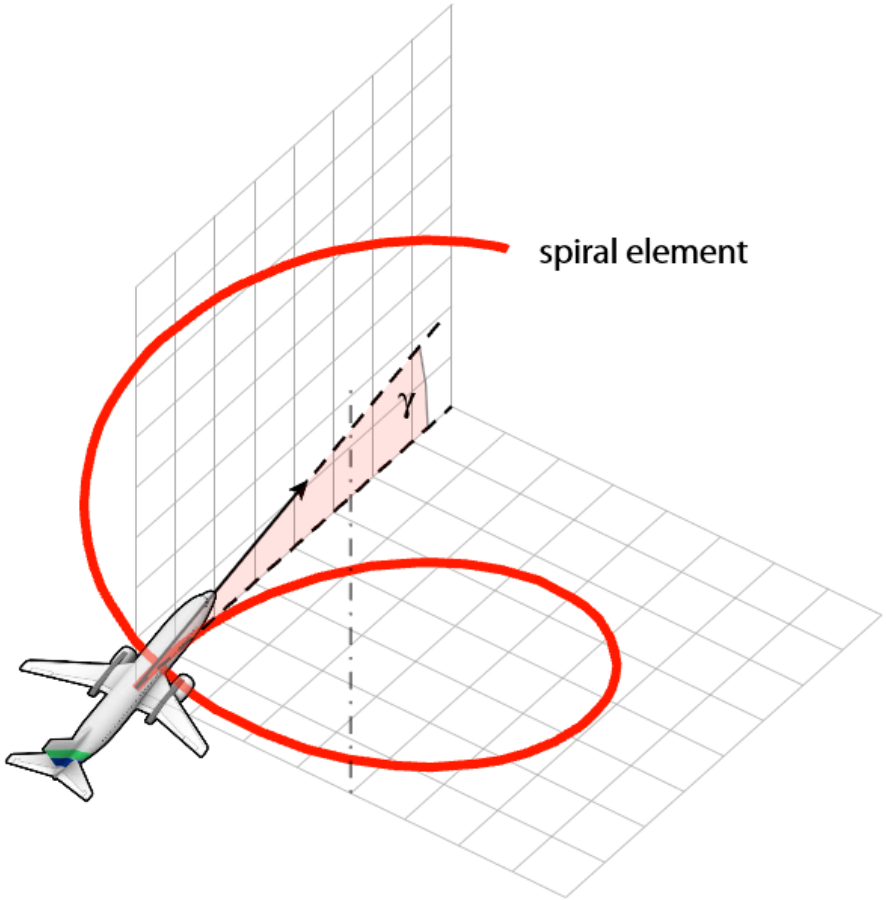
Table: Path lengths and run-times (in parenthesis), each averaged from 500 runs for random grids of size 100 x 100.

Planning for nonholonomic vehicle

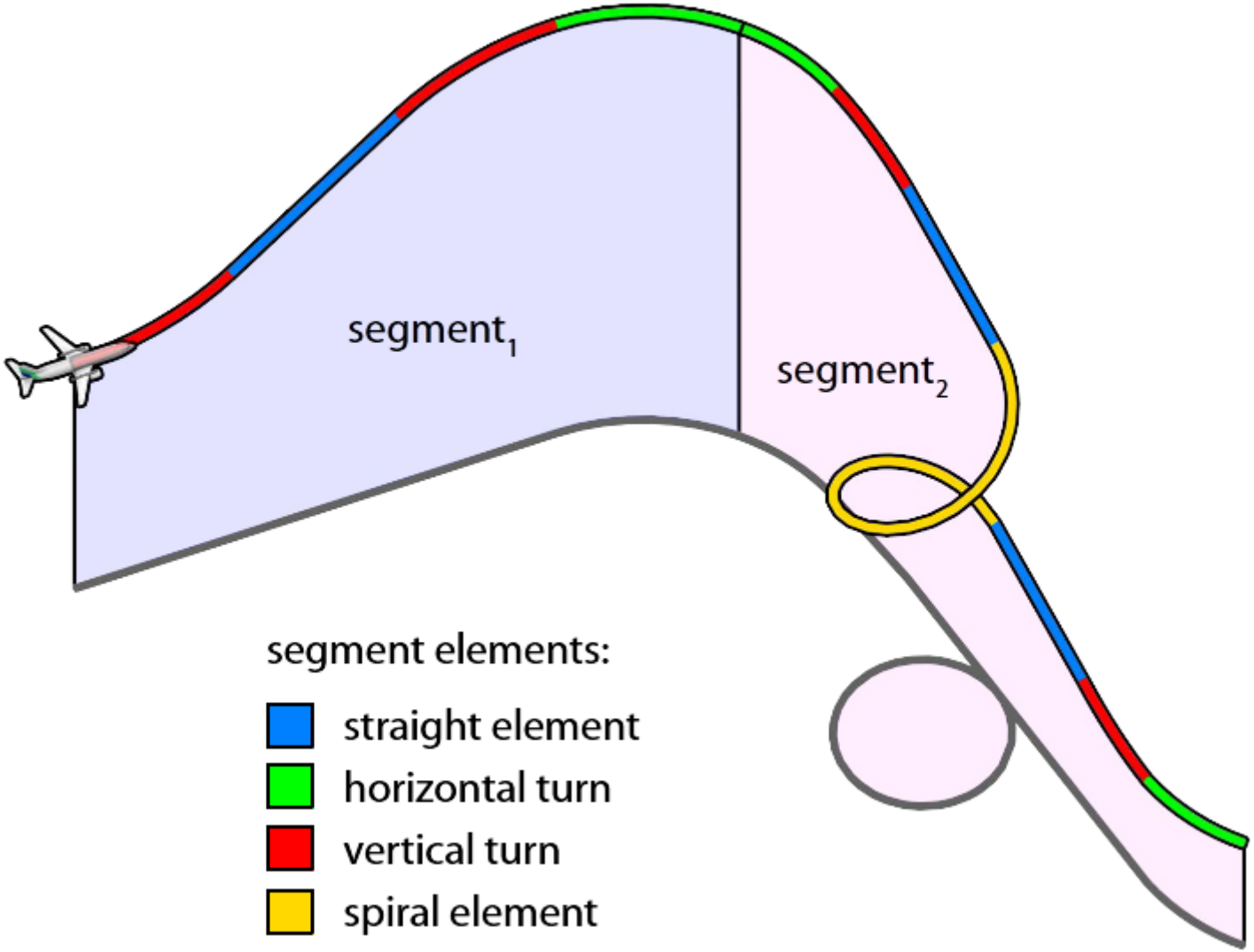
- path specifies motion trajectory for airplane reference point
- airplane dynamics can be converted to flight envelope elements:
  - straight
  - horizontal turn
  - vertical turn
  - spiral
- elements parameters are constrained by
  - minimum horizontal turn radius
  - minimum vertical turn radius
  - maximum airplane climb/descend rate

Planning for nonholonomic vehicle

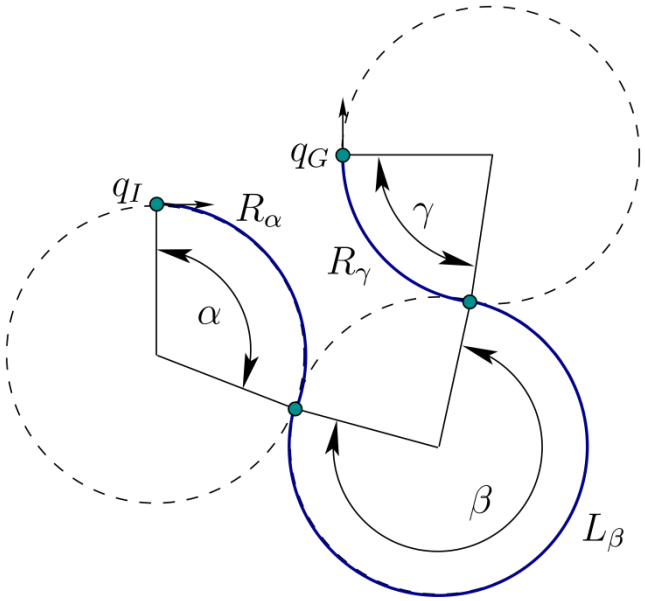




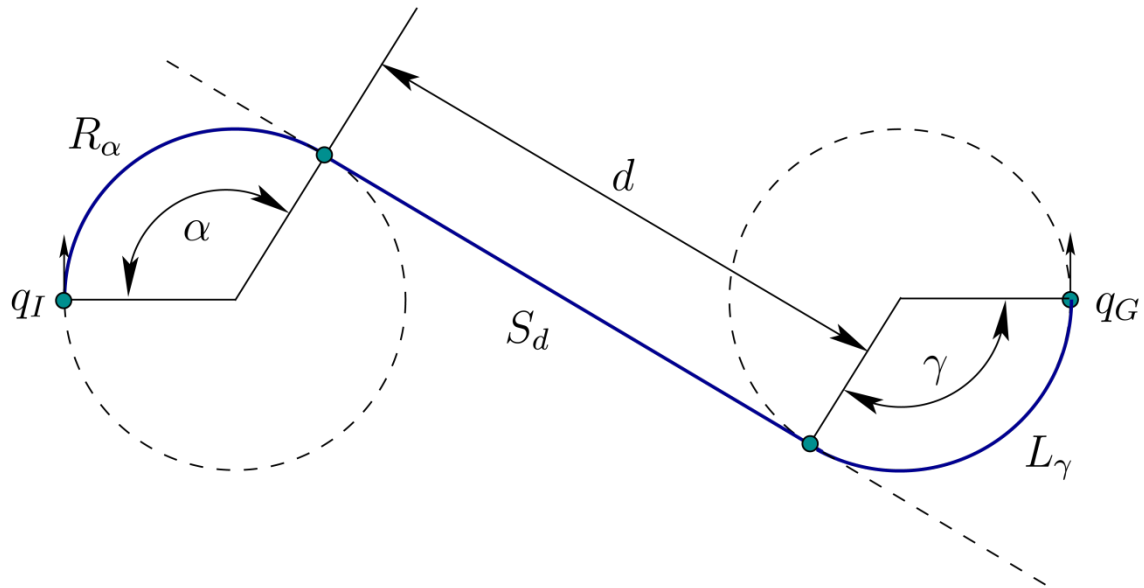
Planning for nonholonomic vehicle



Dubin car problem

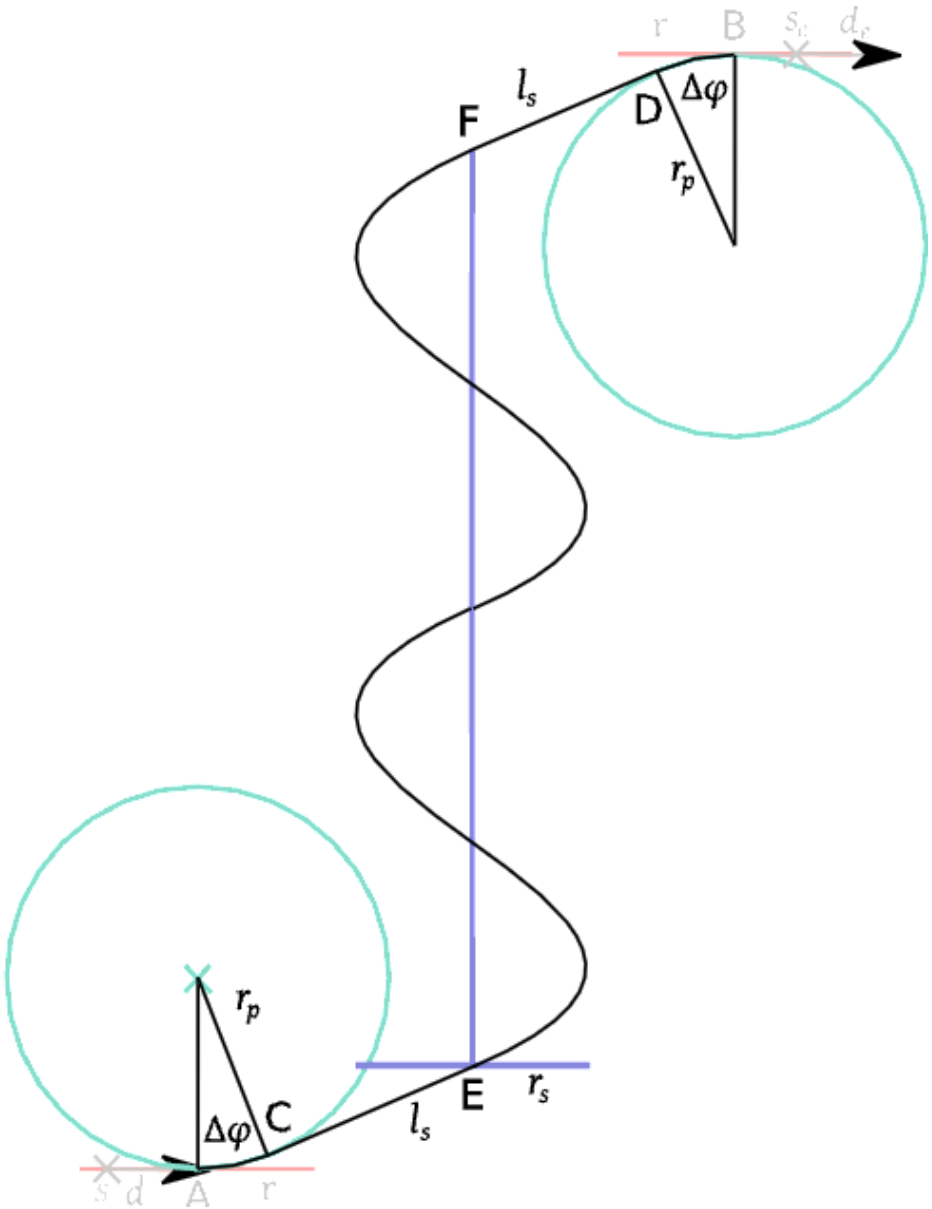


$R_\alpha L_\beta R_\gamma$



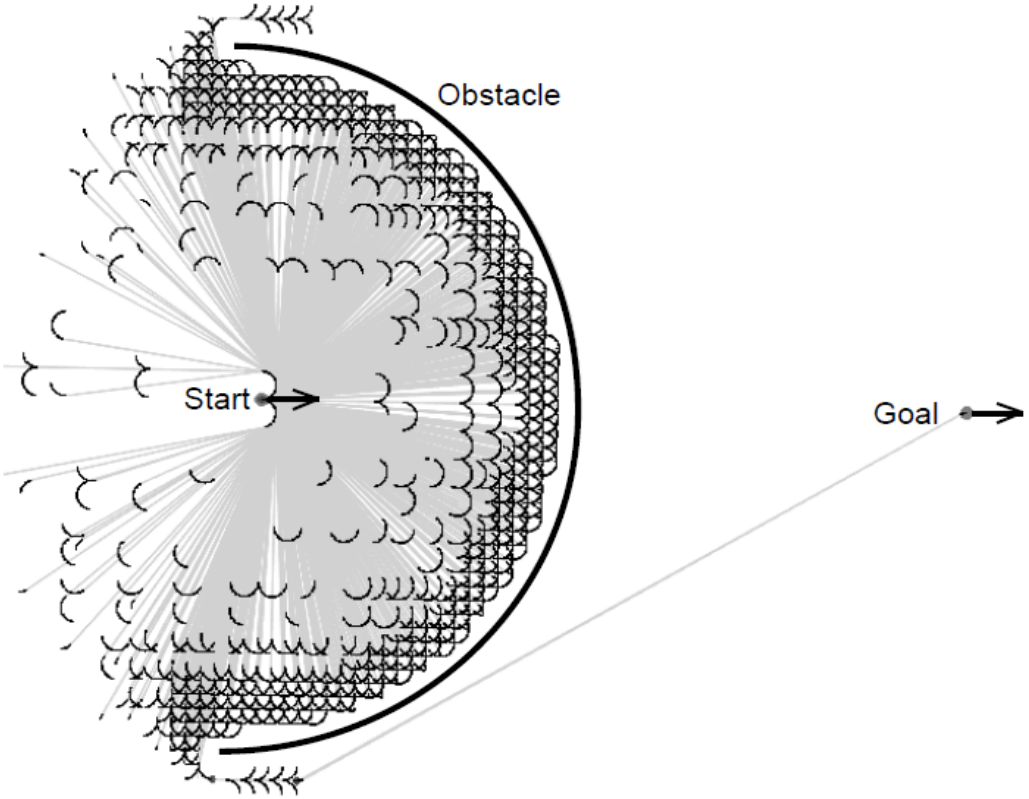
$R_\alpha S_d L_\gamma$

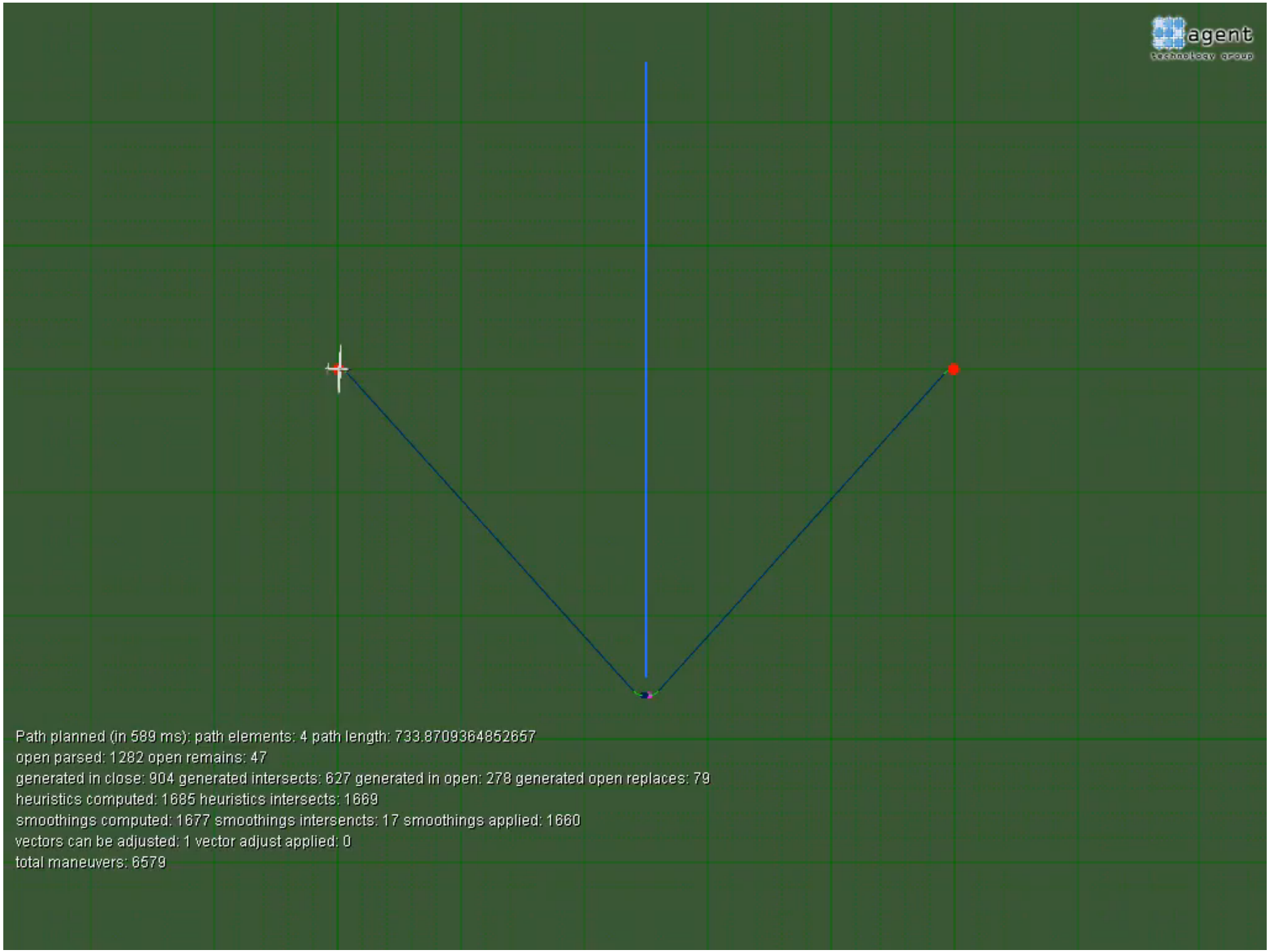
Dubin car problem

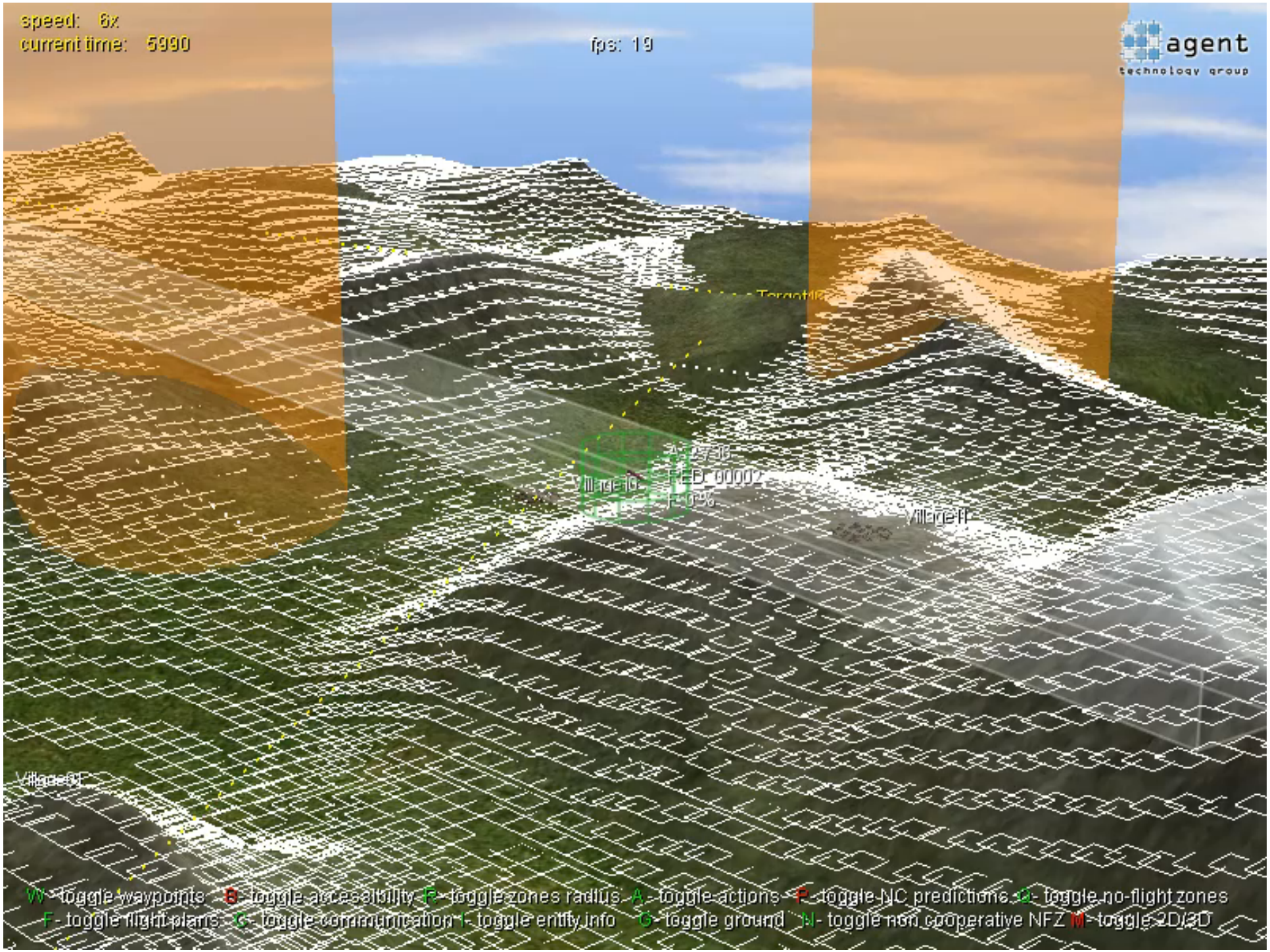


AA* for nonholonomic planning

- » **adaptive sampling** – removes the trade-off between speed and search precision
- » minimal sampling step – **search precision**
- » progressive path smoothing
- » similarity check
- » heuristics based on shortest connection

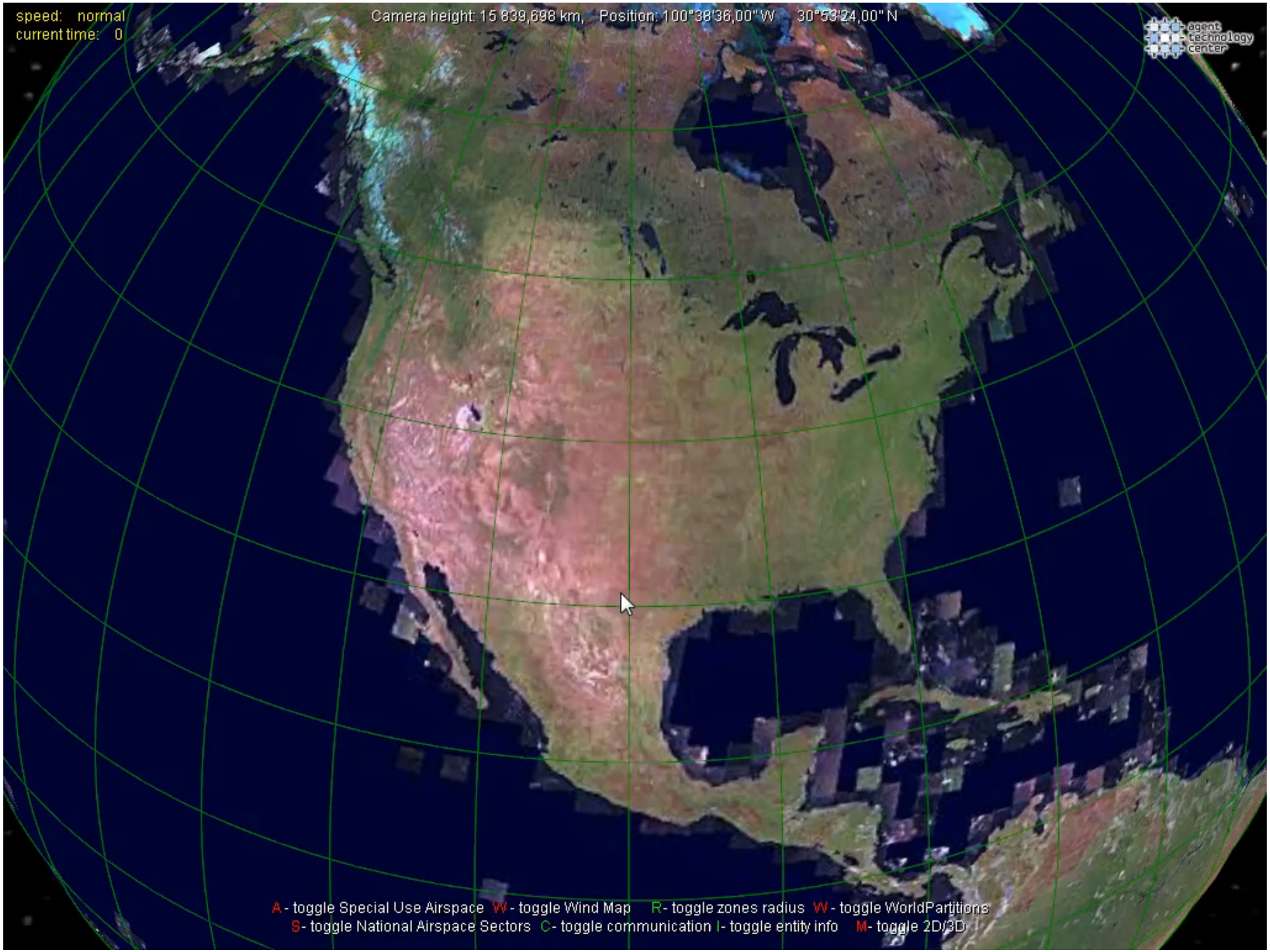


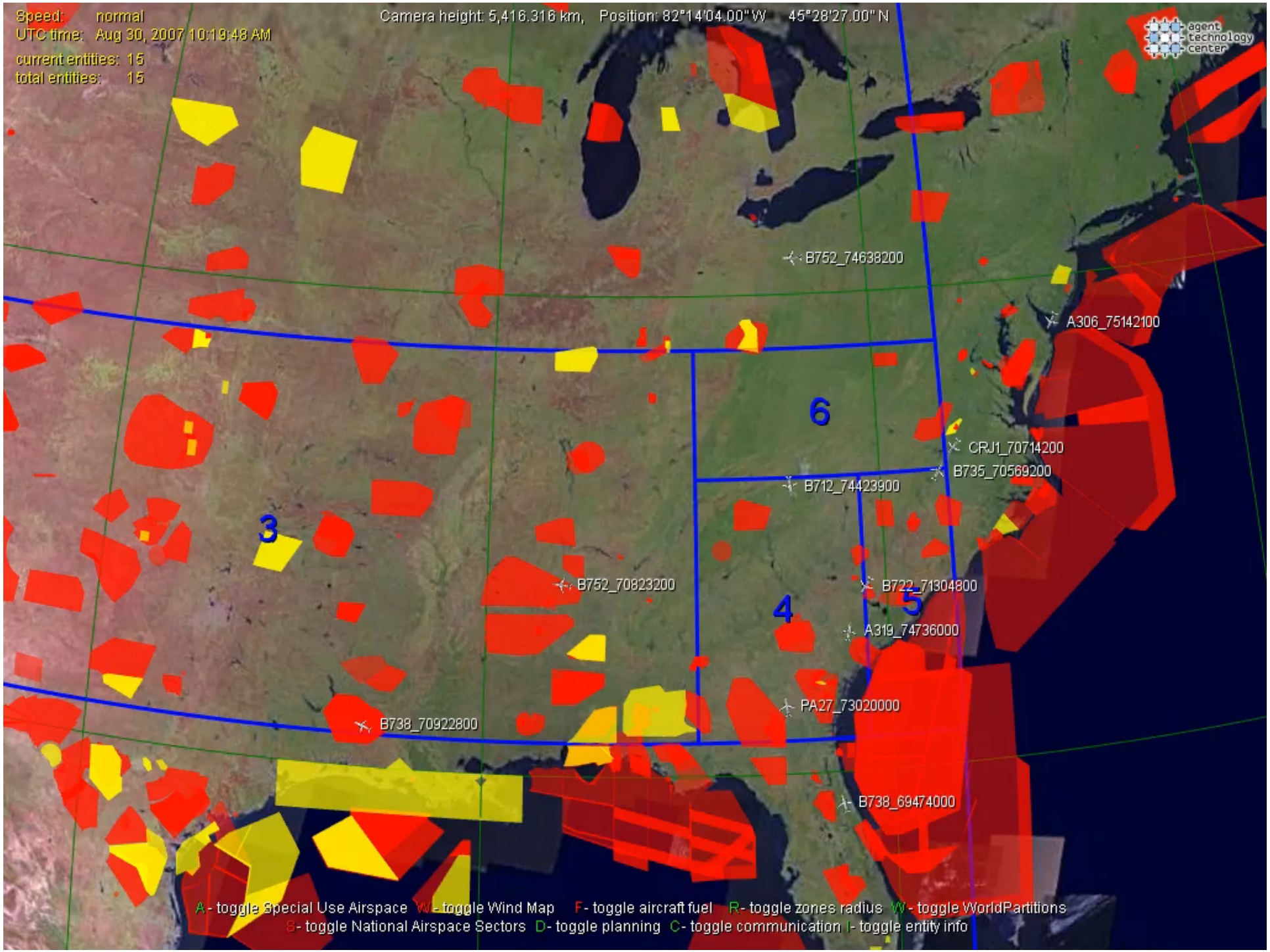

agent
technology group
Path planned (in 589 ms): path elements: 4 path length: 733.8709364852657
open parsed: 1282 open remains: 47
generated in close: 904 generated intersects: 627 generated in open: 278 generated open replaces: 79
heuristics computed: 1685 heuristics intersects: 1669
smoothings computed: 1677 smoothings intersencts: 17 smoothings applied: 1660
vectors can be adjusted: 1 vector adjust applied: 0
total maneuvers: 6579

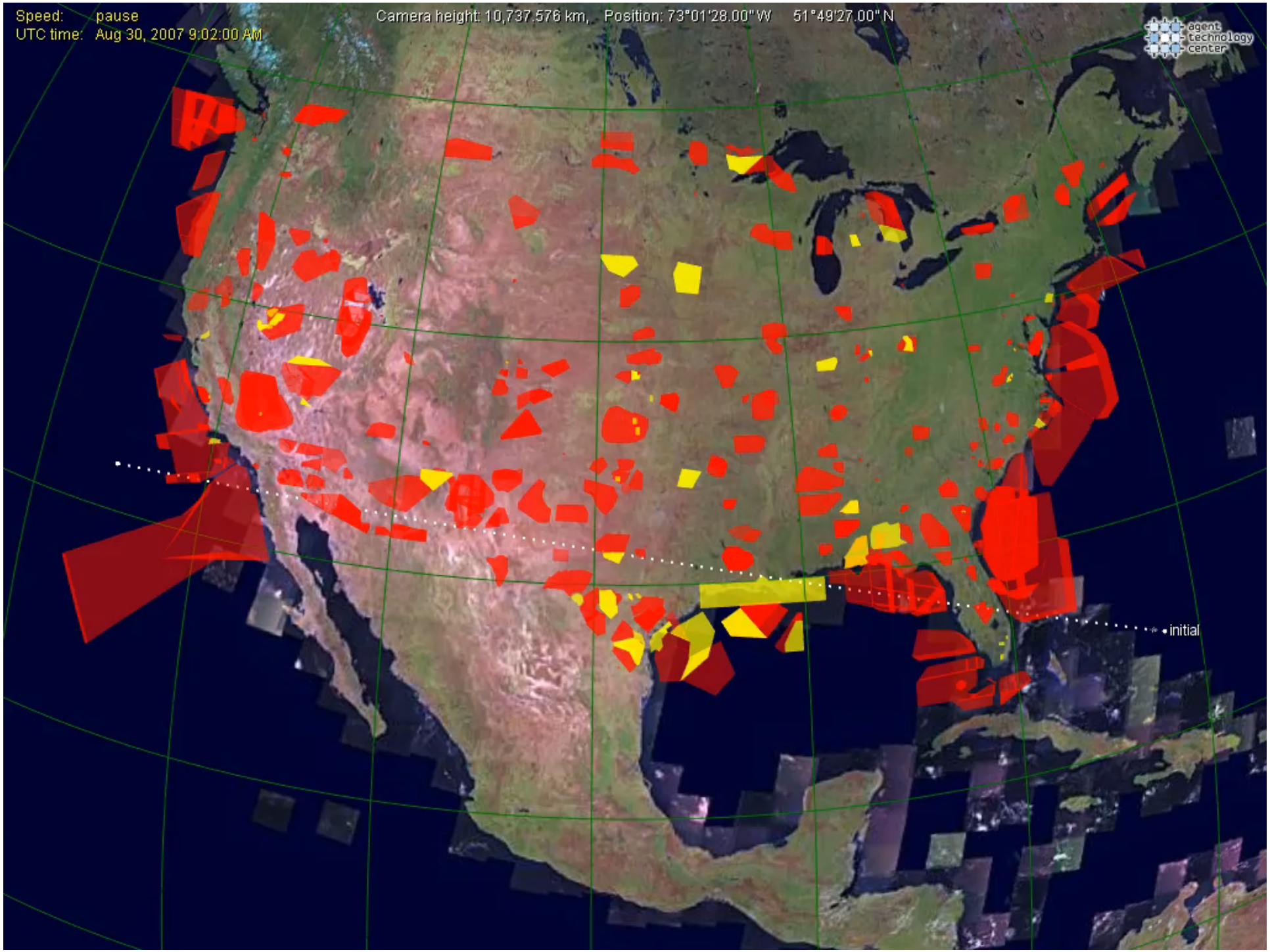

speed: 6x
current time: 5990
fps: 19
agent
technology group
Village II
W - toggle waypoints B - toggle accessibility R - toggle zones radius A - toggle actions P - toggle NC predictions O - toggle no-flight zones
F - toggle flight plans C - toggle communication I - toggle entity info G - toggle ground N - toggle non cooperative NFZ M - toggle 2D/3D

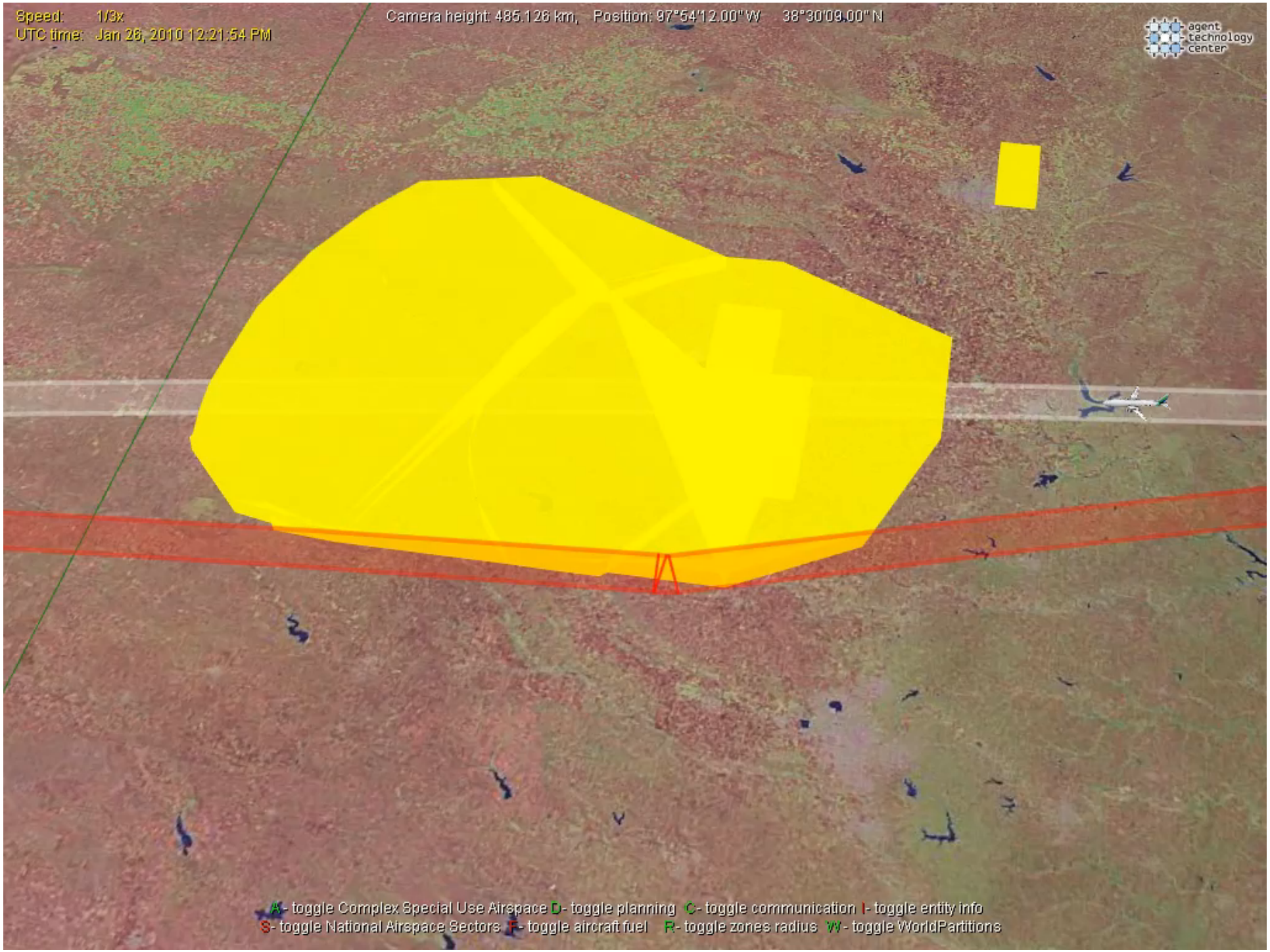
Procerus planning

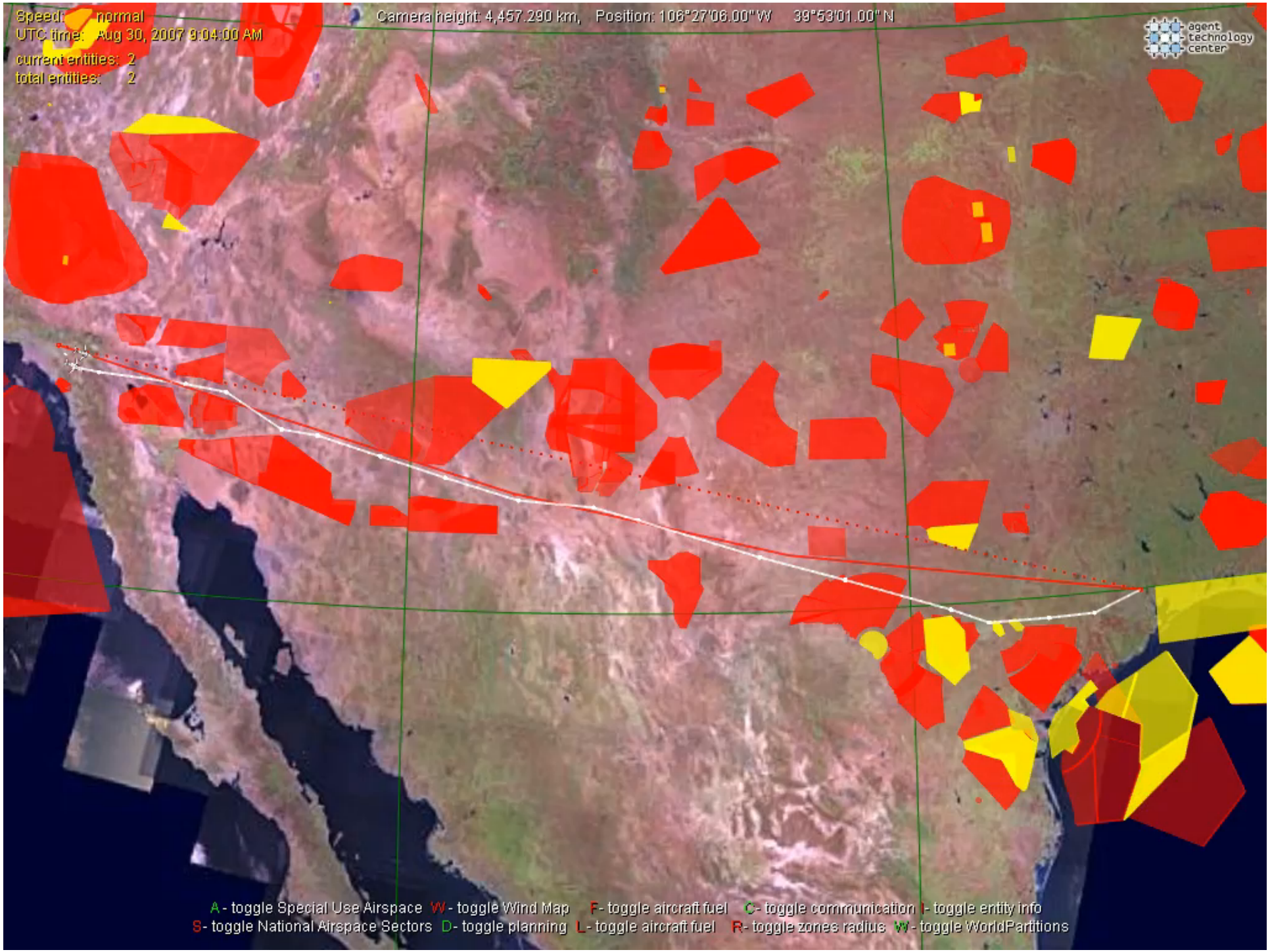

WP1
WP2
WP5
LAND
WP3
Procerus_1
TAKE_OFF
WP4


speed: normal
current time: 0
Camera height: 15 839,698 km, Position: 100°38'36,00" W 30°53'24,00" N
agent technology center
A - toggle Special Use Airspace W - toggle Wind Map R - toggle zones radius W - toggle WorldPartitions
S - toggle National Airspace Sectors C - toggle communication I - toggle entity info M - toggle 2D/3D

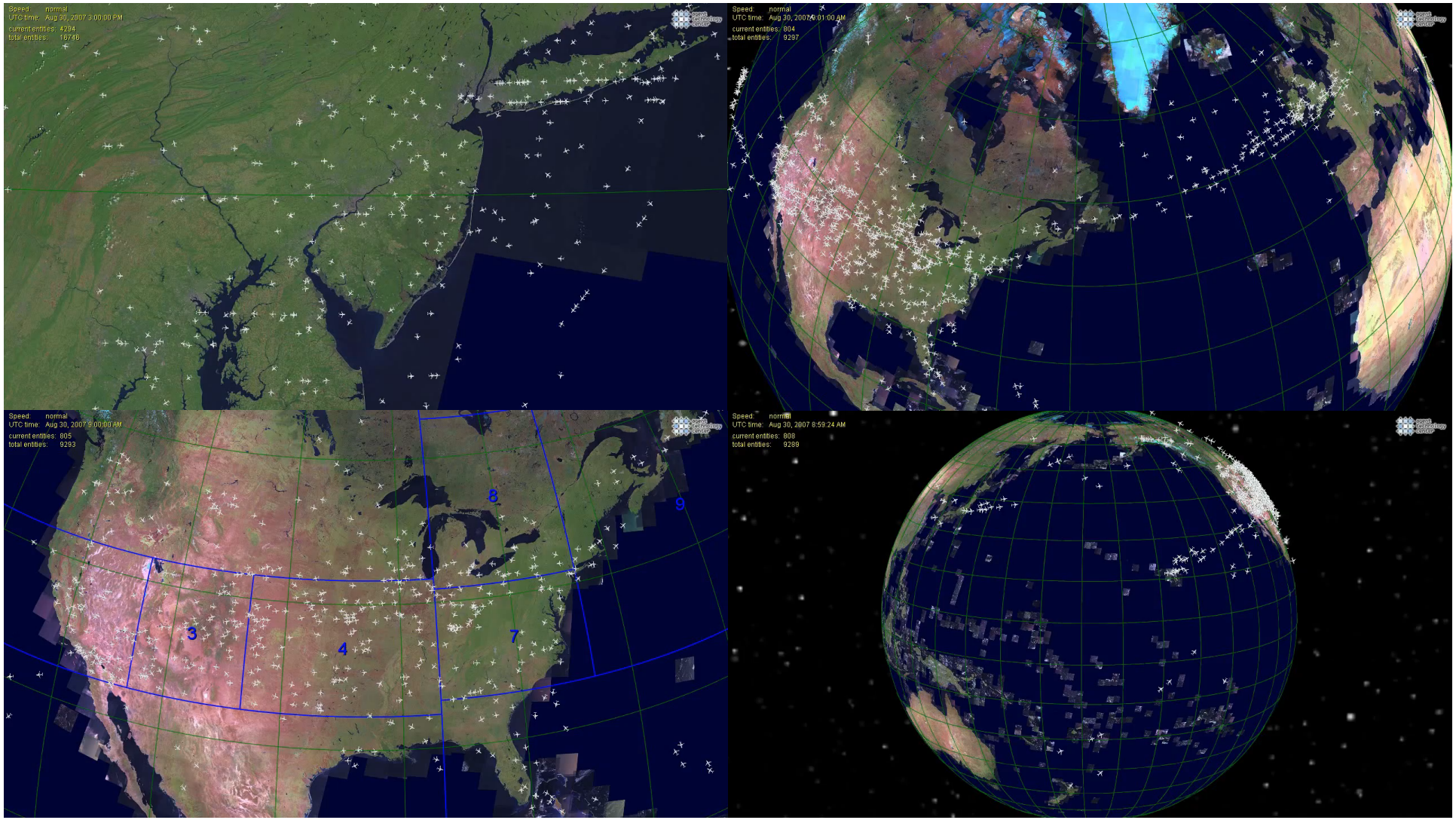

Speed: normal
UTC time: Aug 30, 2007 10:19:48 AM
current entities: 15
total entities: 15
Camera height: 5,416.316 km, Position: 82°14'04.00" W 45°28'27.00" N
agent technology center
B752_74638200
A306_75142100
6
CRJ1_70714200
B735_70569200
B712_74423900
3
B752_70823200
B722_71304800
4
5
A319_74736000
PA27_73020000
B738_70922800
B738_69474000
A - toggle Special Use Airspace W - toggle Wind Map F - toggle aircraft fuel R - toggle zones radius W - toggle WorldPartitions
S - toggle National Airspace Sectors D - toggle planning C - toggle communication I - toggle entity info

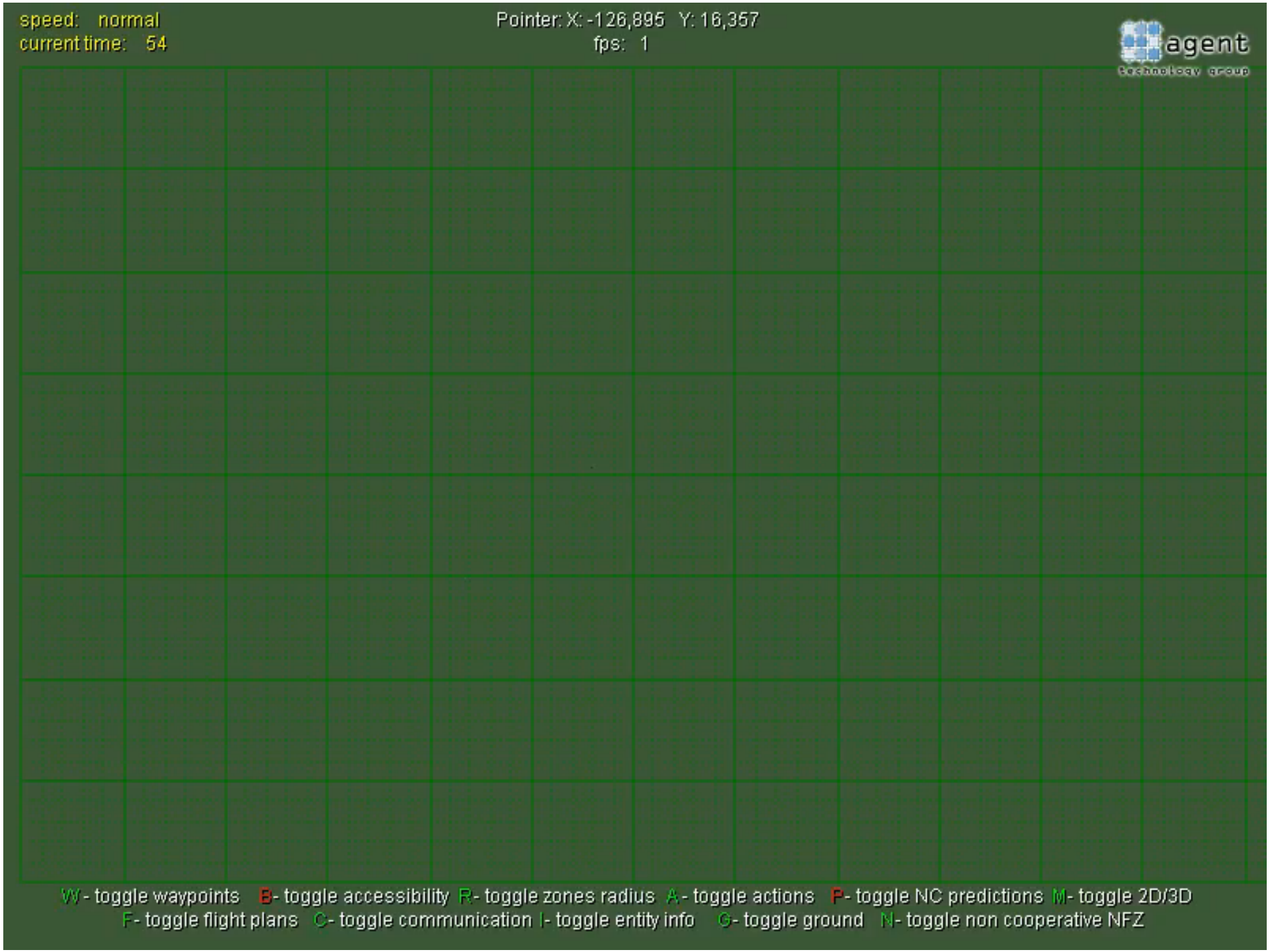

Speed: pause
UTC time: Aug 30, 2007 9:02:00 AM
Camera height: 10,737,576 km, Position: 73°01'28.00" W 51°49'27.00" N
agent technology center
initial

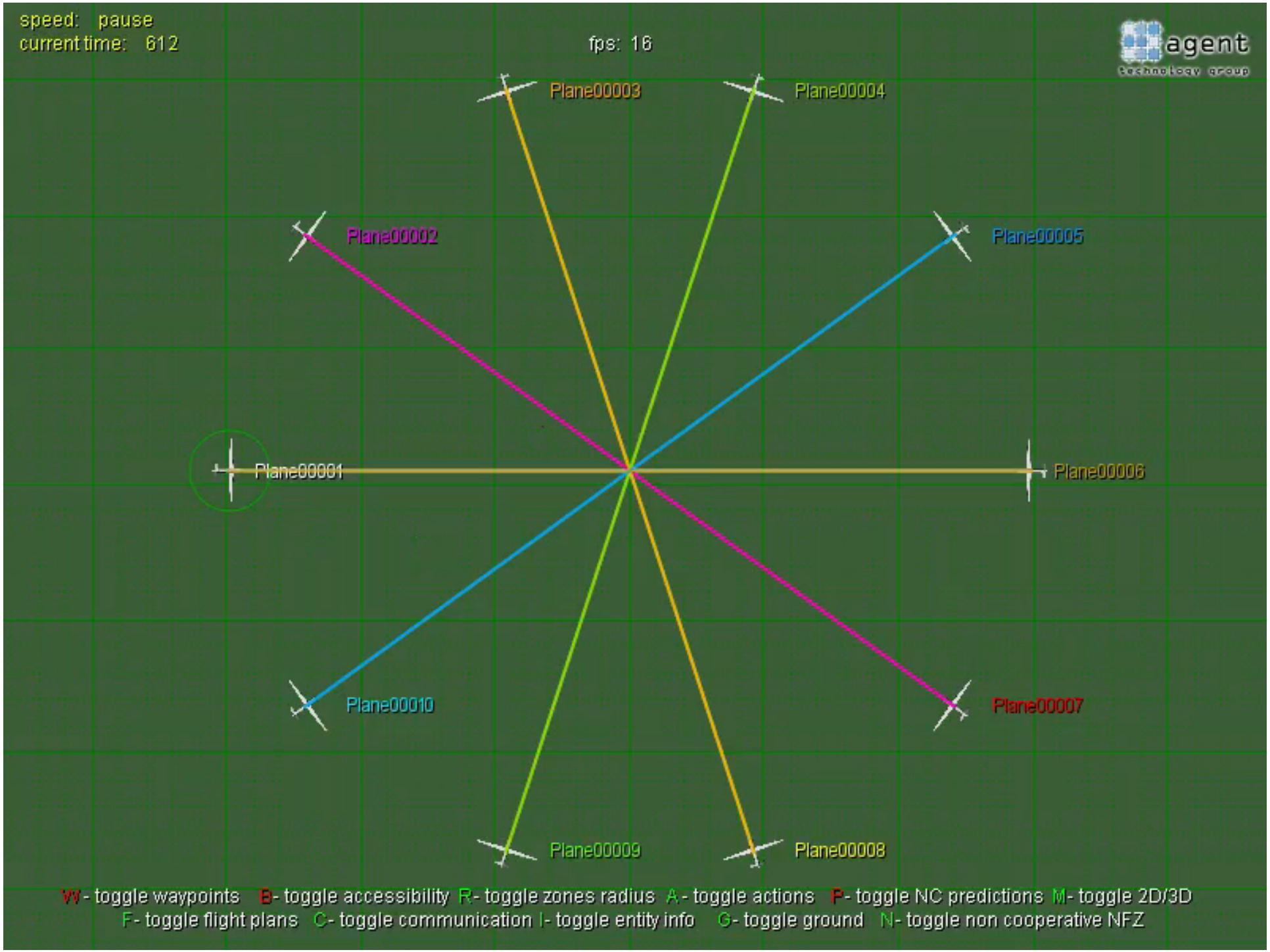

Speed: 1/3x
UTC time: Jan 26, 2010 12:21:54 PM
Camera height: 485.126 km, Position: 97°54'12.00" W 38°30'09.00" N
agent technology center
A - toggle Complex Special Use Airspace D - toggle planning C - toggle communication I - toggle entity info
S - toggle National Airspace Sectors F - toggle aircraft fuel R - toggle zones radius W - toggle WorldPartitions

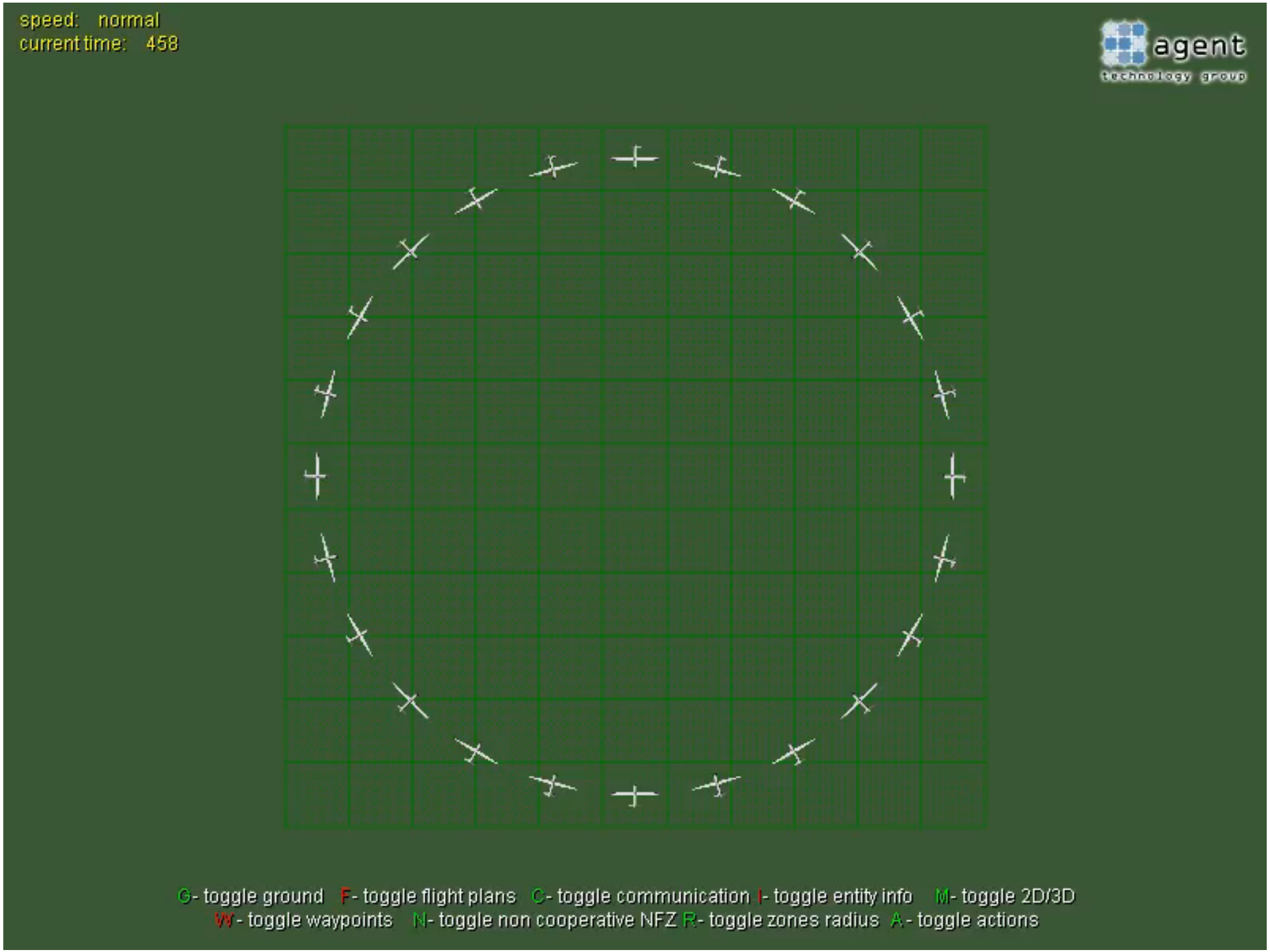

Speed: normal
UTC time: Aug 30, 2007 9:04:00 AM
current entities: 2
total entities: 2
Camera height: 4,457.290 km, Position: 106°27'06.00" W 39°53'01.00" N
agent technology center
A - toggle Special Use Airspace W - toggle Wind Map F - toggle aircraft fuel C - toggle communication I - toggle entity info
S - toggle National Airspace Sectors D - toggle planning L - toggle aircraft fuel R - toggle zones radius W - toggle WorldPartitions

Large-scale planning

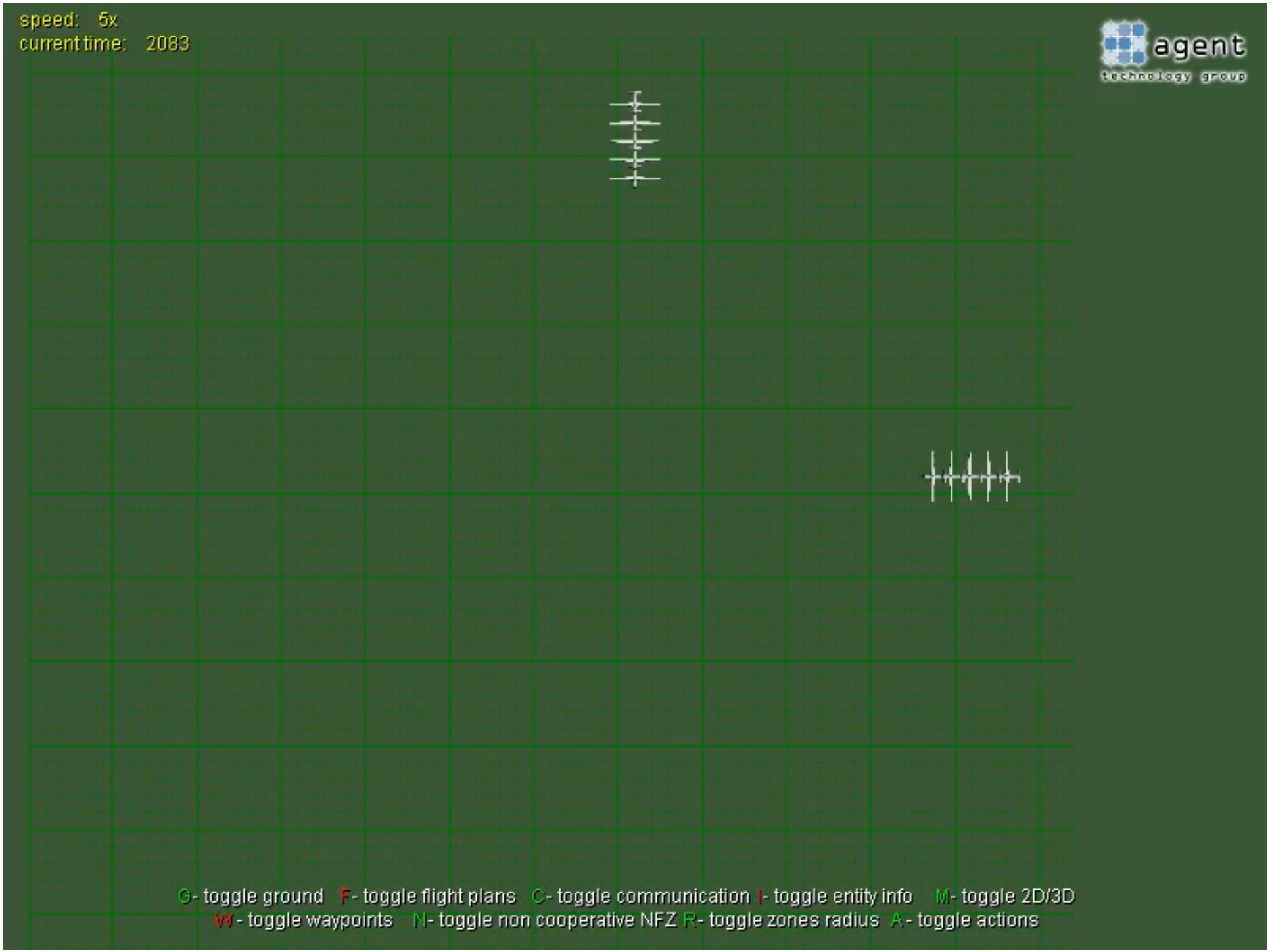


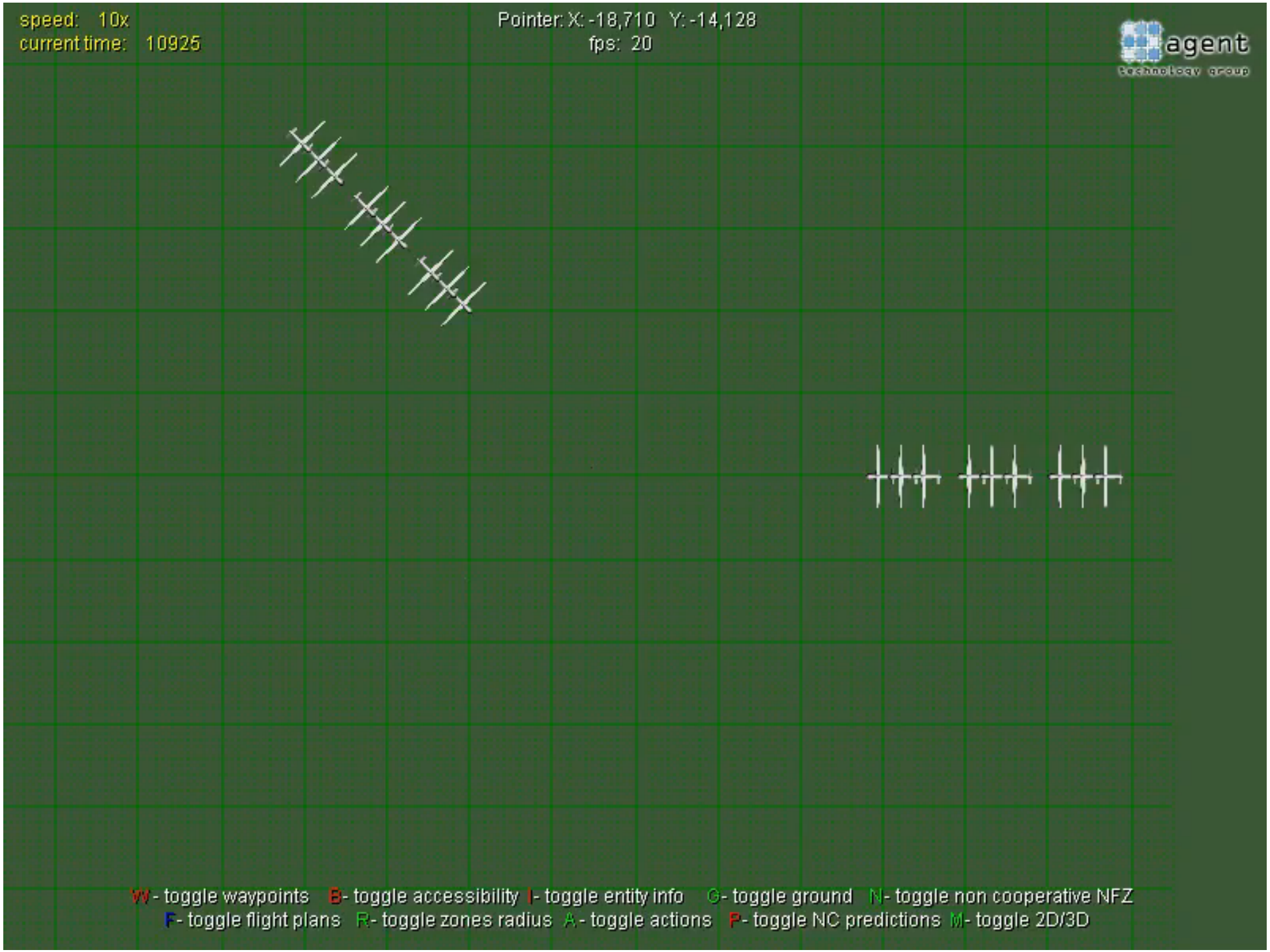

speed: normal
current time: 54
Pointer: X: -126,895 Y: 16,357
fps: 1
agent
technology group
W - toggle waypoints B - toggle accessibility R - toggle zones radius A - toggle actions P - toggle NC predictions M - toggle 2D/3D
F - toggle flight plans C - toggle communication I - toggle entity info G - toggle ground N - toggle non cooperative NFZ

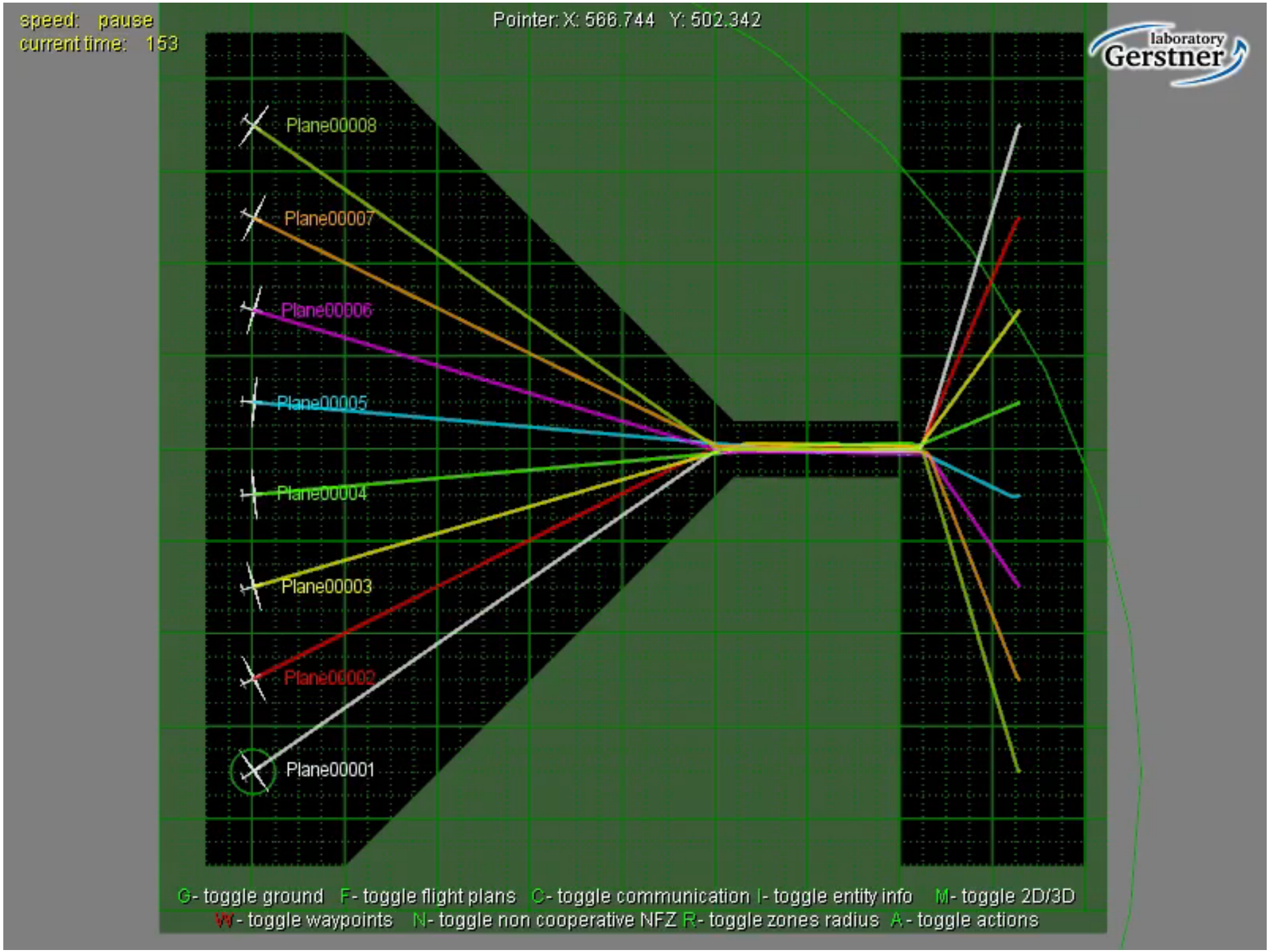

speed: pause
current time: 612
fps: 16
agent
technology group
Plane00003
Plane00004
Plane00002
Plane00005
Plane00001
Plane00006
Plane00010
Plane00007
Plane00009
Plane00008
W - toggle waypoints B - toggle accessibility R - toggle zones radius A - toggle actions P - toggle NC predictions M - toggle 2D/3D
F - toggle flight plans C - toggle communication I - toggle entity info G - toggle ground N - toggle non cooperative NFZ

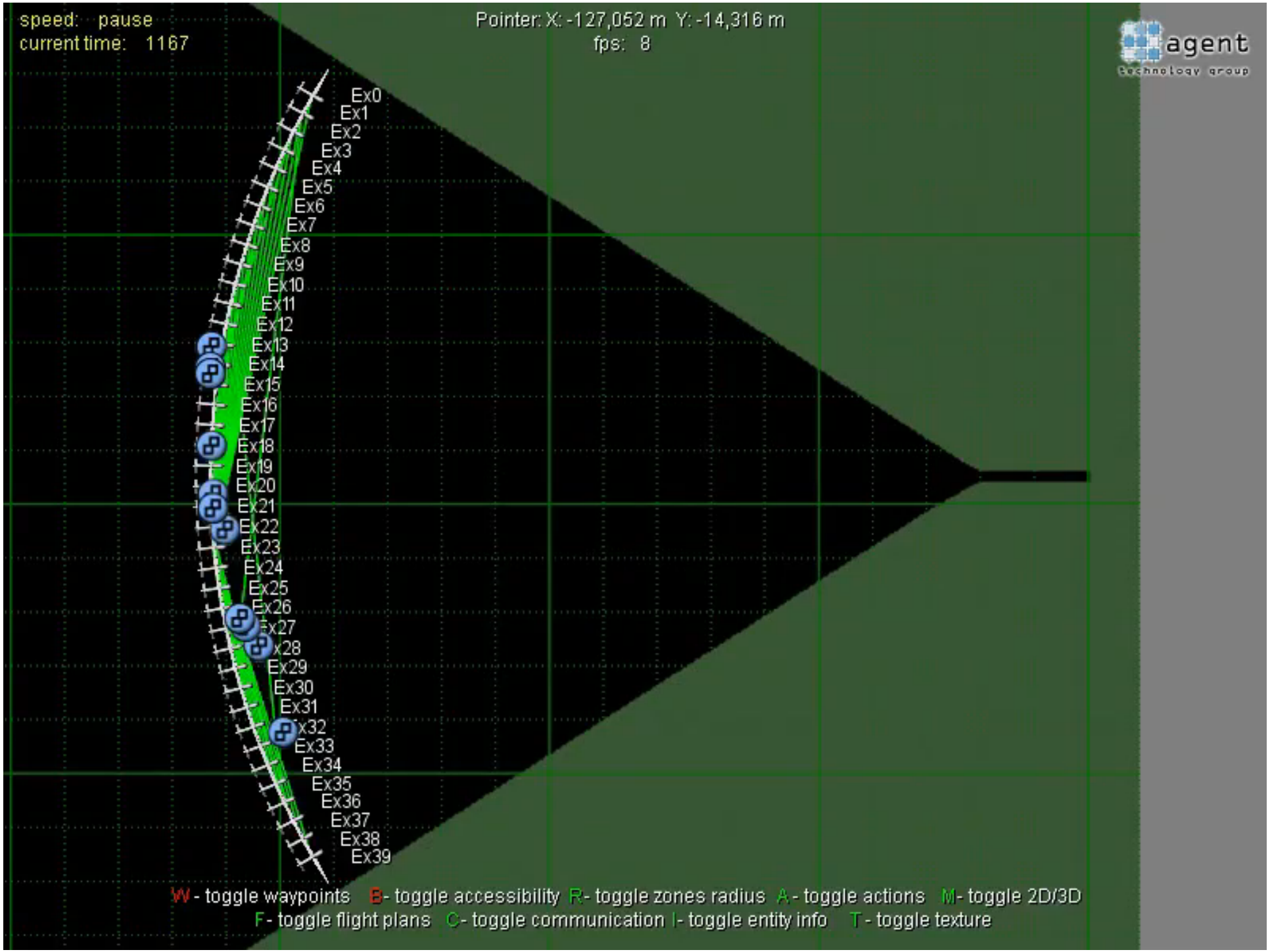

speed: normal
current time: 458
agent
technology group
G- toggle ground F- toggle flight plans C- toggle communication I- toggle entity info M- toggle 2D/3D
W- toggle waypoints N- toggle non cooperative NFZ R- toggle zones radius A- toggle actions


speed: normal
current time: 5355
agent
technology group
G- toggle ground F- toggle flight plans C- toggle communication I- toggle entity info M- toggle 2D/3D
W- toggle waypoints N- toggle non cooperative NFZ R- toggle zones radius A- toggle actions


speed: 5x
current time: 2083
agent
technology group
G- toggle ground F- toggle flight plans C- toggle communication I- toggle entity info M- toggle 2D/3D
W- toggle waypoints N- toggle non cooperative NFZ R- toggle zones radius A- toggle actions

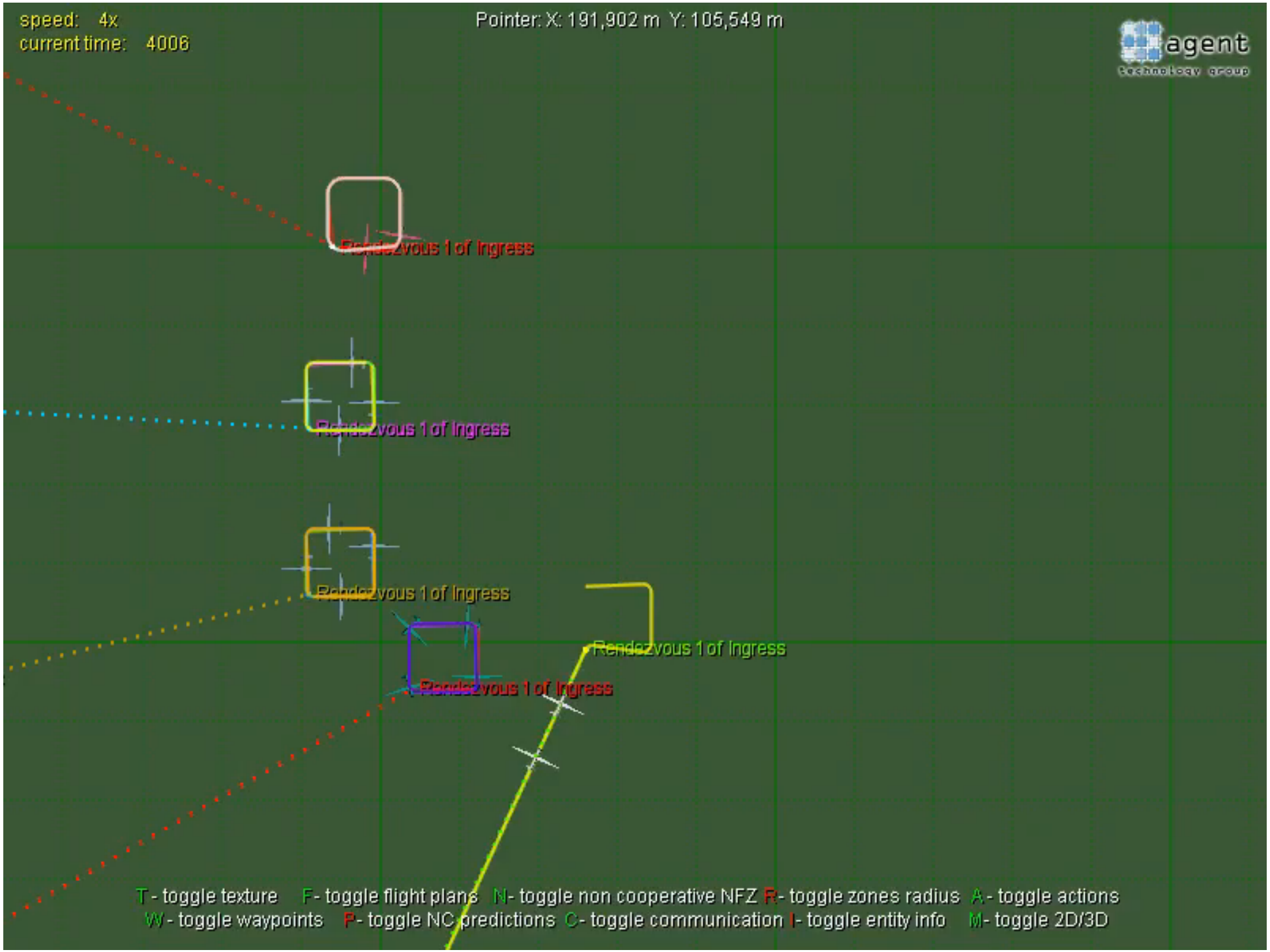

speed: 10x
current time: 10925
Pointer: X: -18,710 Y: -14,128
fps: 20
agent
technology group
W - toggle waypoints B- toggle accessibility I- toggle entity info G- toggle ground N- toggle non cooperative NFZ
F- toggle flight plans R- toggle zones radius A - toggle actions P- toggle NC predictions M- toggle 2D/3D

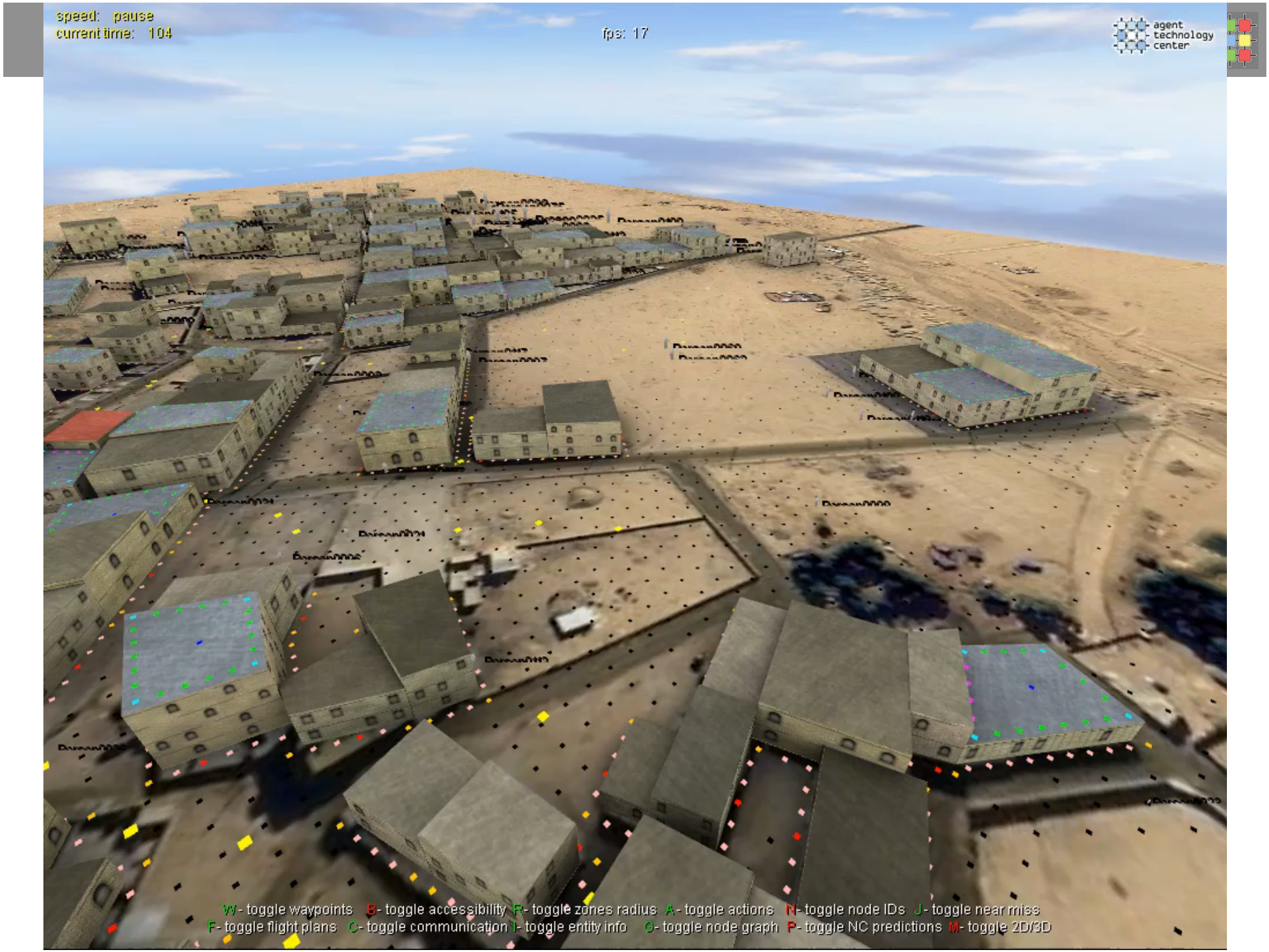

speed: pause
current time: 153
Pointer: X: 566.744 Y: 502.342
laboratory Gerstner
Plane00008
Plane00007
Plane00006
Plane00005
Plane00004
Plane00003
Plane00002
Plane00001
G- toggle ground F- toggle flight plans C- toggle communication I- toggle entity info M- toggle 2D/3D
W- toggle waypoints N- toggle non cooperative NFZ R- toggle zones radius A- toggle actions

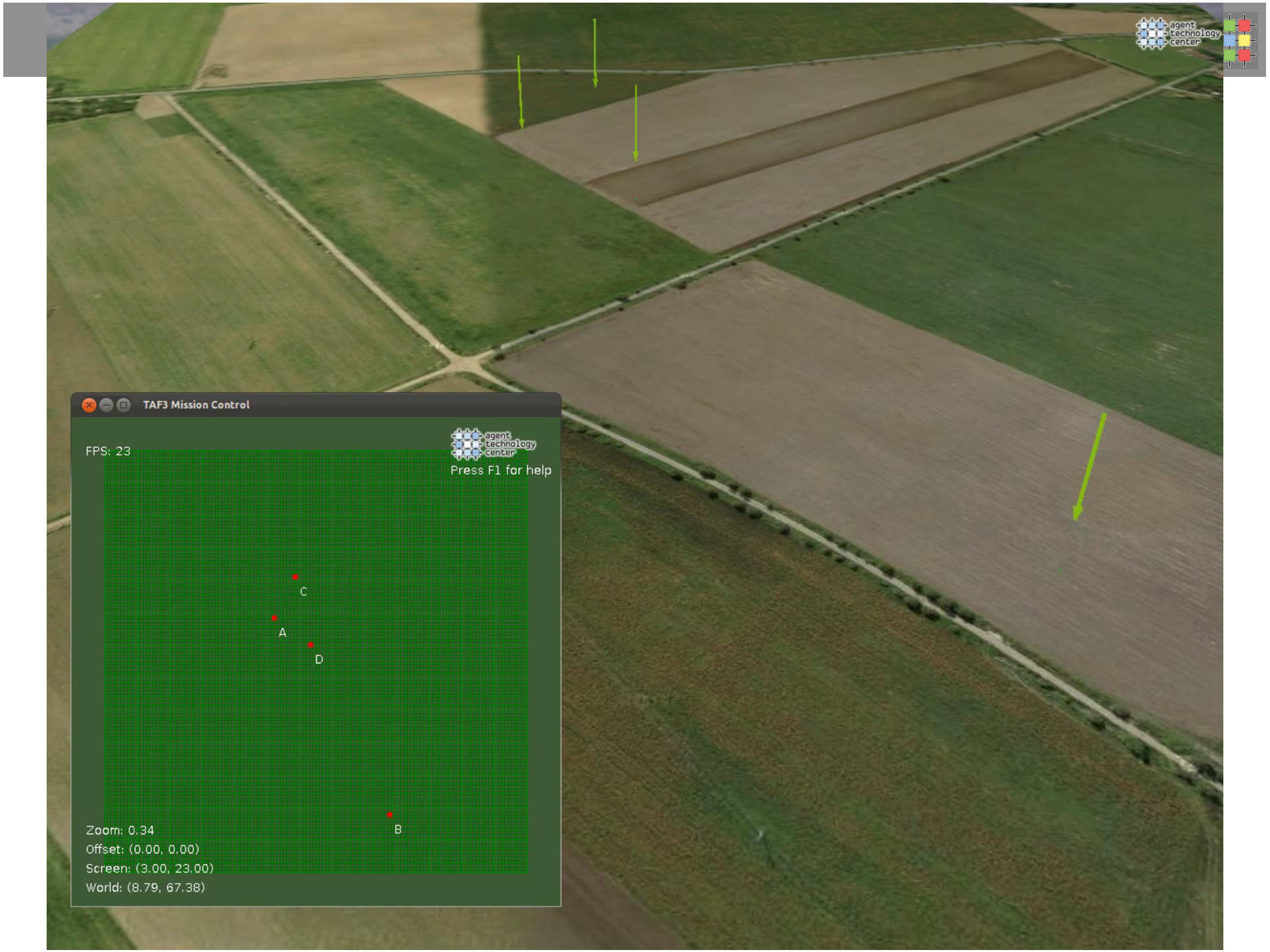

speed: pause
current time: 1167
Pointer: X: -127,052 m Y: -14,316 m
fps: 8
agent
technology group
Ex0
Ex1
Ex2
Ex3
Ex4
Ex5
Ex6
Ex7
Ex8
Ex9
Ex10
Ex11
Ex12
Ex13
Ex14
Ex15
Ex16
Ex17
Ex18
Ex19
Ex20
Ex21
Ex22
Ex23
Ex24
Ex25
Ex26
Ex27
Ex28
Ex29
Ex30
Ex31
Ex32
Ex33
Ex34
Ex35
Ex36
Ex37
Ex38
Ex39
W - toggle waypoints B- toggle accessibility R- toggle zones radius A- toggle actions M- toggle 2D/3D
F- toggle flight plans C- toggle communication I- toggle entity info T - toggle texture


speed: normal
current time: 383
laboratory Gerstner
F - toggle flight plans D - toggle planning debug N - toggle non cooperative NFZ R - toggle zones radius A - toggle a
P - toggle NC predictions E - toggle zone test layer C - toggle communication I - toggle entity info M - toggle 2D


Speed: 10x
Time: 39
Pointer: X: 24,826 m Y: 16,265 m
agent technology center
UAV_2
UAV_1
W - toggle waypoints B - toggle accessibility R - toggle zones radius A - toggle actions P - toggle NC predictions J - toggle near miss
F - toggle flight plans C - toggle communication I - toggle entity info G - toggle ground N - toggle non cooperative NFZ M - toggle 2D/3D


speed: 4x
current time: 4006
Pointer: X: 191,902 m Y: 105,549 m
agent
technology group
Rendezvous 1 of Ingress
Rendezvous 1 of Ingress
Rendezvous 1 of Ingress
Rendezvous 1 of Ingress
Rendezvous 1 of Ingress
T - toggle texture F - toggle flight plans N - toggle non cooperative NFZ R - toggle zones radius A - toggle actions
W - toggle waypoints P - toggle NC predictions C - toggle communication I - toggle entity info M - toggle 2D/3D


speed: pause
current time: 104
fps: 17
agent
technology
center
W - toggle waypoints B - toggle accessibility R - toggle zones radius A - toggle actions N - toggle node IDs J - toggle near miss
F - toggle flight plans C - toggle communication I - toggle entity info G - toggle node graph P - toggle NC predictions M - toggle 2D/3D


agent
technology
center
TAF3 Mission Control
FPS: 23
agent
technology
center
Press F1 for help
C
A
D
B
Zoom: 0.34
Offset: (0.00, 0.00)
Screen: (3.00, 23.00)
World: (8.79, 67.38)