Modeling *Space Fortress*

Marc Destefano
August 5, 2010
• don’t fly too fast
• keep the ship within the hexagons
• capture appropriate bonuses when available
• destroy the fortress as often as possible
• destroy the mines as quickly as possible
Ship Flight

- Thrust is acceleration-based
- Space is frictionless
- Orientation and motion are decoupled
turn
thrust
ship velocity
thrust velocity
new ship velocity
stop turning
turn
Experiment
Results
Strategies Learned

• Waiting for the mine
• Using VLCTY as a counter
• Increased capture of PNTS bonuses
High-performing player, Average VLNER at mine onset over time

strategy change!
Final Bonus Capture

- No bonus captured
- SHOTS bonus captured
- PNTS bonus captured

Low-performing player

Medium-performing player

High-performing player
Tactics

• Stay close to inner hexagon, moving slowly, but fast enough to prevent fortress from firing

• Keep movement perpendicular to fortress as often as possible

• Aim ship to face fortress as often as possible

• Small movements, rapid keypresses
Average time Thrust key depressed over time
High-performing vs. low-performing player

Game Number
Most common orientation angle differences for starting ship turning

Difference of angle between ship orientation and angle of ship to fortress

\[ \rho = 13.79 \]
Most common velocity angle differences for starting to thrust

\[ \bar{x} = 96.8, \rho = 17.14 \]

Difference of angle between ship's velocity vector and angle of ship to fortress
Model
Procedural Module
- Pattern Matching
- Production Selection
- Production Execution

Goal Module
- Goal Buffer

Imaginal Module
- Imaginal Buffer

Temporal Module (pacemaker)

Temporal Buffer (accumulator)

Visual-Location Buffer

Visual Buffer

Vision Module

Manual Buffer

Motor Module

Device Module

External World (Space Fortress)

Declarative Module
- Retrieval Buffer

Manual Buffer

Imaginal Buffer

Temporal Module (pacemaker)

Visual-Location Buffer

Visual Buffer

Vision Module

Motor Module

Device Module

External World (Space Fortress)

Audio Module

Aural Buffer

Vocal Buffer

Speech Module

(Taatgen, van Rijn, & Anderson, 2007)
Global Parameters

(sgp :esc t :er t :v t :mas 1
:bll 0.5 :ans 0.25 :lf 0.5
:visual-movement-tolerance 40.0
:trace-detail high :test-feats nil
:motor-feature-prep-time 0
:default-punch-delay 0.06
:visual-attention-latency 0.05
:do-not-harvest imaginal)
Global Parameters

visual movement tolerance set to 40 degrees (!)

(sgp :esc t :er t :v t :mas 1
:blt 0.5 :ans 0.25 :lf 0.5
:visual-movement-tolerance 40.0
:trace-detail high :test-feats nil
:motor-feature-prep-time 0
:default-punch-delay 0.06
:visual-attention-latency 0.05
:do-not-harvest imaginal)
Global Parameters

Set instant motor feature preparation (Kieras, 2009)

(sgp :esc t :er t :v t :mas 1
:bll 0.5 :ans 0.25 :lf 0.5
:visual-movement-tolerance 40.0
:trace-detail high :test-feats nil
:motor-feature-prep-time 0
:default-punch-delay 0.06
:visual-attention-latency 0.05
:do-not-harvest imaginal)
Global Parameters

do not automatically harvest the imaginal buffer

(sgp :esc t :er t :v t :mas 1
 :bll 0.5 :ans 0.25 :lf 0.5
 :visual-movement-tolerance 40.0
 :trace-detail high :test-feats nil
 :motor-feature-prep-time 0
 :default-punch-delay 0.06
 :visual-attention-latency 0.05
 :do-not-harvest imaginal)
Implementation Details

- Imaginal buffer used as a non-harvested “scratchpad” to manage issue of interruptible tracking
- Device module modifies a fixed list of location and object chunks that represent the visicon
- Visual object chunks contain slots that assume ability to determine orientation, etc.
- ACT-R drives the simulation by stepping a frame every 33ms in the event scheduler
Flight Pattern Productions

(p turn-right-in-circle
  =goal>
  ISA fly-ship
  state standard-flight-pattern
  =visual>
  ISA ship
  > orient-diff 2
  - vel 0
  > fortress-distance 95
  < fortress-distance 190
  ?manual>
  processor free
  ==> 
  +manual>
  ISA press-key
  key d)

(p thrust-in-circle
  =goal>
  ISA fly-ship
  state standard-flight-pattern
  =visual>
  ISA ship
  - vel 0
  > vel-diff 95
  > fortress-distance 95
  < fortress-distance 175
  < orient-diff 5
  ?manual>
  processor free
  ==> 
  +manual>
  ISA press-key
  key w)
```
(p turn-right-in-circle
  =goal>
  ISA  fly-ship
  state standard-flight-pattern
  =visual>
  ISA  ship
  > orient-diff 2
  - vel 0
  > fortress-distance 95
  < fortress-distance 190
  ?manual>
  processor free
  ==>  +manual>
  ISA  press-key
  key  d
)

(p thrust-in-circle
  =goal>
  ISA  fly-ship
  state standard-flight-pattern
  =visual>
  ISA  ship
  - vel 0
  > vel-diff 95
  > fortress-distance 95
  < fortress-distance 175
  < orient-diff 5
  ?manual>
  processor free
  ==>  +manual>
  ISA  press-key
  key  w
)
```

“Butter Zone” Productions
Outside the “Butter Zone”

- Very difficult to pull patterns out of the data
- Most common expert subject claim: “I tried to get to a tangent”
Model Evaluation
• don’t fly too fast ✓
• keep the ship within the hexagons ✓
• capture appropriate bonuses when available ✓
• destroy the fortress as often as possible ✘
• destroy the mines as quickly as possible ✘

<table>
<thead>
<tr>
<th>PNTS</th>
<th>CNTRL</th>
<th>VLCTY</th>
<th>VLNER</th>
<th>IFF</th>
<th>INTRVL</th>
<th>SPEED</th>
<th>SHOTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>30</td>
<td>35</td>
<td>0</td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>
PNTS
(fortress kills, mine destructions, taking damage, bonuses, etc.)

CNTRL
(keeping the ship within the hexagons)

VLCTY
(keeping the ship moving but not too quickly)

SPEED
(destroying mines quickly)
Discussion
Challenges for ACT-R

- More sophisticated motor control
- Disappearing objects
- Closed-loop control with continuous feedback
- Interruptible tracking
Focused Attention

Distractibility

Skilled performance