An ACT-R model of collaborative skill acquisition for Coop Space Fortress

Cvetomir Dimov
Carnegie Mellon University
Coop Space Fortress

Game description

* **Goal**: kill fortress located at center of screen

* **Roles**: one player baits the fortress, the other shoots it down; bait should fly slowly

* **Incentivization**: point gains from killing fortress and losses from dying or missing shots
Coop Space Fortress

Game description

- **Goal**: kill fortress located at center of screen
- **Roles**: one player baits the fortress, the other shoots it down; bait should fly slowly
- **Incentivization**: point gains from killing fortress and losses from dying or missing shots
Coop Space Fortress

Game description

- **Goal**: kill fortress located at center of screen
- **Roles**: one player baits the fortress, the other shoots it down; bait should fly slowly
- **Incentivization**: point gains from killing fortress and losses from dying or missing shots
Coop Space Fortress

Game description

• **Goal**: kill fortress located at center of screen

• **Roles**: one player baits the fortress, the other shoots it down; bait should fly slowly

• **Incentivization**: point gains from killing fortress and losses from dying or missing shots
Studies with Coop Space Fortress

Study 1
- **Goal:** pilot
- **Duration:** ~ 1 h (20 games x 3 minutes)
- **Sample:** 14 teams

Study 2
- **Goal:** transfer from Space Track
- **Duration:** ~ 2 h (2 sessions)
- **Sample:** 39 teams
Study Results: Performance

Study 1: Score

Study 2 (control condition): Score
Study Results: Performance

Study 1: Kills

Study 2 (control condition): Kills
Study Results: Performance

**Study 1: Deaths**

**Study 2 (control condition): Deaths**
Study Results: Role Consistency

Study 1: Bait proportion

Study 2 (control condition): Bait proportion
Study Results: Anticipatory Actions
Study Results: Anticipatory Actions

![Chart showing orientation relative to aim at fortress (deg) for different conditions]

- Aim at fortress during:
  - End of turn
  - Shot

Orientation relative to aim at fortress (deg)

Difference in two orientations:
- < 3.35 (median)
- >= 3.35 (median)
Study Results: Anticipatory Actions
Study Results: Anticipatory Actions
Study Results: Anticipatory Actions

Other anticipatory actions:

• Preparing to stop when outside
• Avoiding crashing into outer border or fortress
• Bait prepares to slow down before having entered
ACT-R Model: Skill Acquisition Process

1. Cognitive operators
2. Transition from declarative phase to procedural phase
3. Control tuning

Anderson, Betts, Bothell, Hope, & Lebiere (2019)
ACT-R Model: Operators

Game State

Action
ACT-R Model: Operators

Fortress alive → Fortress vulnerable → Target other → Self behind

Action
ACT-R Model: Operators

Fortress alive → Fortress vulnerable → Target other → Self behind

Aim at fortress → Shoot
ACT-R Model: Operators

- **Fortress alive?**
  - **YES**
  - **Fortress vulnerable?**
    - **YES**
    - **Target other?**
      - **YES**
      - **Self behind?**
        - **NO**
        - **Aim at fortress**
          - **Shoot**
        - **YES**
          - **Aim at fortress**
  - **NO**
  - **NO**
- **NO**
- **YES**
  - **Target other?**
    - **YES**
    - **Self behind?**
      - **NO**
      - **Aim at fortress**
        - **Shoot**
      - **YES**
        - **Aim at fortress**
  - **YES**
    - **Self behind?**
      - **NO**
      - **Aim at fortress**
        - **Shoot**
      - **YES**
        - **Aim at fortress**

ACT-R Model: Production Compilation

P1: Trigger operator retrieval
R1: Retrieve operator
P2: Execute operation
ACT-R Model: Production Compilation

P1: Trigger operator retrieval
   ↓
R1: Retrieve operator
   ↓
P2: Execute operation
   ↓
P2b: Directly execute operation
ACT-R Model: Control Tuning

• Control tuning tunes parameters to the specific task from feedback about reward rate:
  1. Assume a quadratic relationship between parameter values and reward rate
  2. Assume an exponentially decaying noise around reward rate
  3. Pick a parameter value and sample its reward for a certain period (positive and negative feedback associated with events)

• Example: Learning **bait speed** (positive feedback from fortress missiles and negative feedback from dying)
ACT-R Model: Control Tuning

- Stochastic gradient descent: Adjust control parameter as a function of error magnitude and error direction after an action.

- Example: learning when to release key press when turning as a function of feedback about how far off the resultant orientation was from the target orientation and whether it under- or overshot the target.
ACT-R Model: Anticipatory Actions

• All anticipatory actions in model rely on motion extrapolation based on the ship’s current position and speed:
  1. Avoid crashing into obstacles
  2. Determine aim at fortress at time of shot
  3. If bait’s future location is inside the hexagon, prepare to slow down
  4. If future location is outside of hexagon, prepare to stop
ACT-R Model: Role Adoption

Role adoption is modeled through reinforcement learning. Two productions that select bait/shooter compete with each other. The production utility is updated:

- Each fortress kill brings positive reward
- A role switch brings negative reward
ACT-R Model: Performance

Study 1: Score

Study 2 (control condition): Score
ACT-R Model: Performance

Study 1: Kills

Study 2 (control condition): Kills
ACT-R Model: Performance

Study 1: Deaths

Study 2 (control condition): Deaths
ACT-R Model: Role Consistency

Study 1: Bait proportion

Study 2 (control condition): Bait proportion
Thank you