“Accuracy in Dead-Reckoning Based Distributed Multi-Player Games”
Aggarwal et al., NetGames 2004
More on Dead-Reckoning

- Dead reckoning: estimate/predict an entity’s position based on its history
  - Previous positions
  - Velocity
  - Acceleration

- Dead reckoning works!
- But how well?
Accuracy of Dead-Reckoning

- Information exchange: via DR vector
  \[(x_0, y_0, z_0, vx_0, vy_0, vz_0)\]

- An entity’s movement is calculated using the current DR until the new one is received.
Accuracy (cont.)

- Delay cause inconsistencies in between sender’s view and receiver’s view
Accuracy (cont.)

• Does it matter?
Types of errors

- Before export error (we’ve seen this)
- After export error
How to cope with errors

• Synchronized global clock: DR vectors now include a timestamp

• The time difference is used to calculate the movement of an entity. This offsets the after export errors

• What about before export error: Unavoidable
Evaluation

• **BZFlag**: First person shooter game. Players drive tanks in battle field

• Three motion styles are tested: Linear, Circular, and Random

• Various delay values are tested: 100ms, 300ms, 800ms
Linear Motion

Linear Motion: 100ms Delay
Linear Motion

Linear Motion: 300ms Delay

![Graph showing error over time with two lines representing Local Dead Reckoning and Global Dead Reckoning. The graph peaks at around 5025 seconds, with the Local Dead Reckoning line being more erratic.]
Circular Motion

Circular Motion: 100ms Delay
Circular Motion

Circular Motion: 300ms Delay

![Graph showing error over time for Local Dead Reckoning and Global Dead Reckoning]
Random Motion

Random Motion: 300ms Delay

[Graph showing error over time with labels for Local Dead Reckoning and Global Dead Reckoning]
Evaluation results

- Delay is proportionate to error rate

- Linear vs Circular: No difference because BZFlag detects circular motions and use circular DR algorithm

- Random Motion: Very high error rate. Synchronized clock still does a good job.
Evaluation

• Some unexpected (but explainable) results with Random Motion test cases: error rate higher in some cases

• Due to the zero discrepancy between position, possible to overshoot trajectory.
Conclusion

• Simple yet effective technique to increase accuracy in Online Multiplayer Game

• Higher accuracy leads to better game experience

• Still does not address the bigger problem: before export error