AI ARTIFICIAL INTELLIGENCE SUMMIT

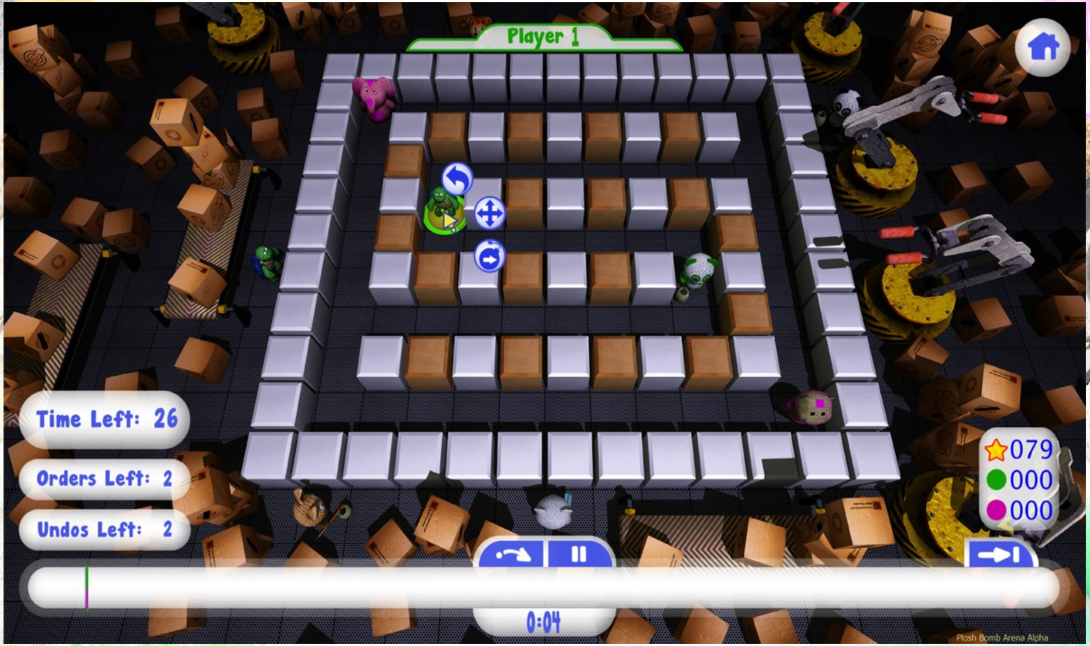
A Stitch in Time: Metaplanning for AI with Unusual Time Controls

Chris Hazard, PhD
CEO, Hazardous Software
Marq Singer
Principal Sw Engr, Hazardous Software

CYBER THREAT DEFENDER
Special thanks to IBM security & research thought leadership
Hazardous® Software

Hazardous Software

Plosh: Time Bomb



Player 1

Time Left: 26

Orders Left: 2

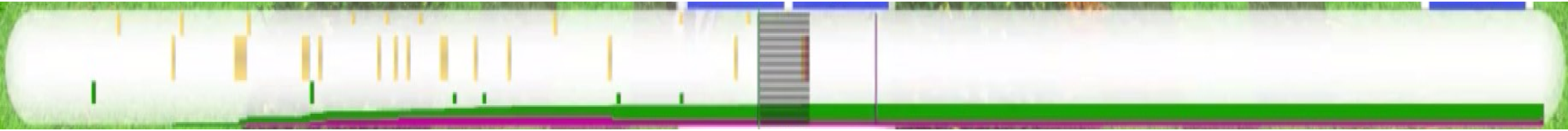
Undos Left: 2

☆079
●000
●000

0:04

Plosh: Time Bomb

The Timeline



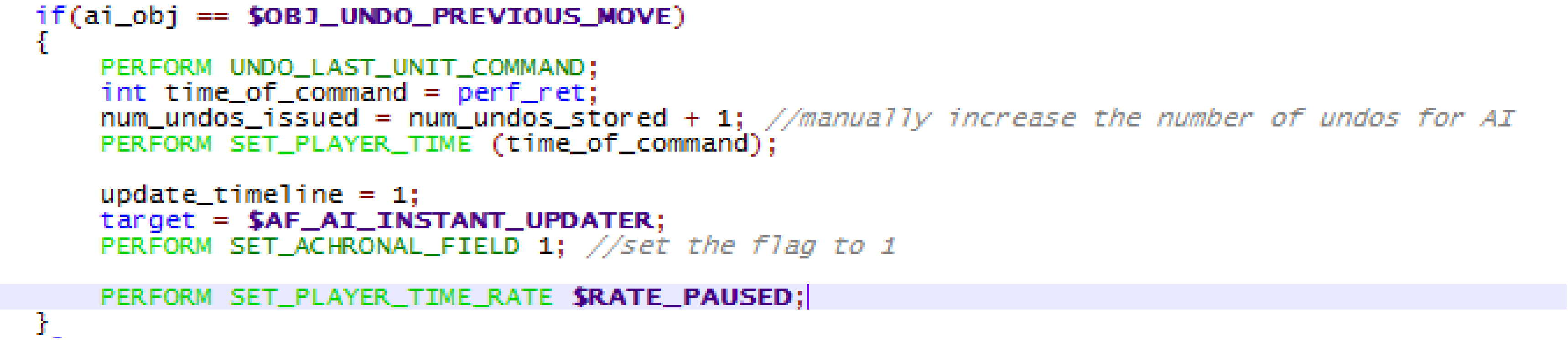
- Access to position, event, and state data
- Lots can happen in 90 seconds

The Problem

- Standard approach
- Worry about what to avoid or approach & how to get there
- Future events will change the outcome

- Plosh: past matters
- Mistakes must be undone
- A move in the past may appear benign but game-winning
- Past can be rewritten out from under you

Time Management

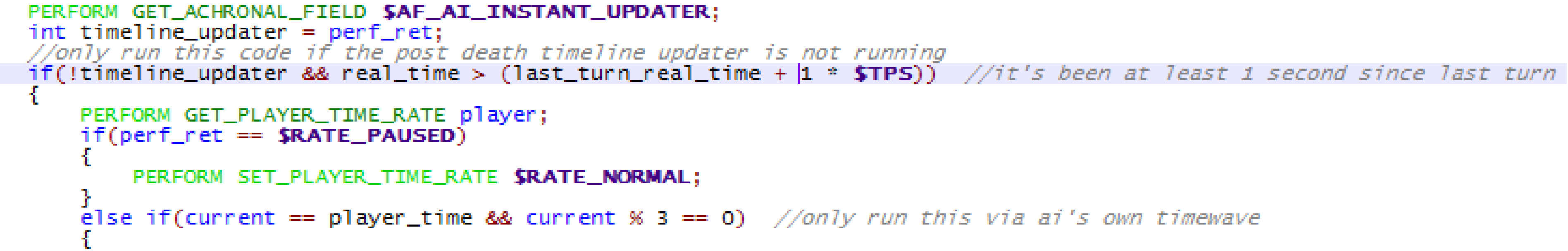


```
if(ai_obj == $OBJ_UNDO_PREVIOUS_MOVE)
{
    PERFORM UNDO_LAST_UNIT_COMMAND;
    int time_of_command = perf_ret;
    num_undos_issued = num_undos_stored + 1; //manually increase the number of undos for AI
    PERFORM SET_PLAYER_TIME (time_of_command);

    update_timeline = 1;
    target = $AF_AI_INSTANT_UPDATER;
    PERFORM SET_ACHRONAL_FIELD 1; //set the flag to 1

    PERFORM SET_PLAYER_TIME_RATE $RATE_PAUSED;
}
```

Time Management: Instant Updating



```
PERFORM GET_ACHRONAL_FIELD $AF_AI_INSTANT_UPDATER;
int timeline_updater = perf_ret;
//only run this code if the post death timeline updater is not running
if(!timeline_updater && real_time > (last_turn_real_time + 1 * $TPS))  //it's been at least 1 second since last turn
{
    PERFORM GET_PLAYER_TIME_RATE player;
    if(perf_ret == $RATE_PAUSED)
    {
        PERFORM SET_PLAYER_TIME_RATE $RATE_NORMAL;
    }
    else if(current == player_time && current % 3 == 0)  //only run this via ai's own timewave
    {
```

Metaplanning

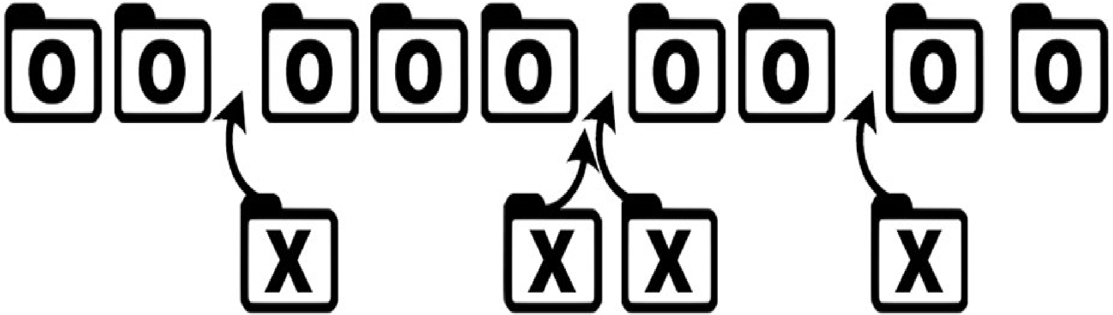
Planning where each game state is a plan, and each action is a change to the plan

Can minmax on plans!

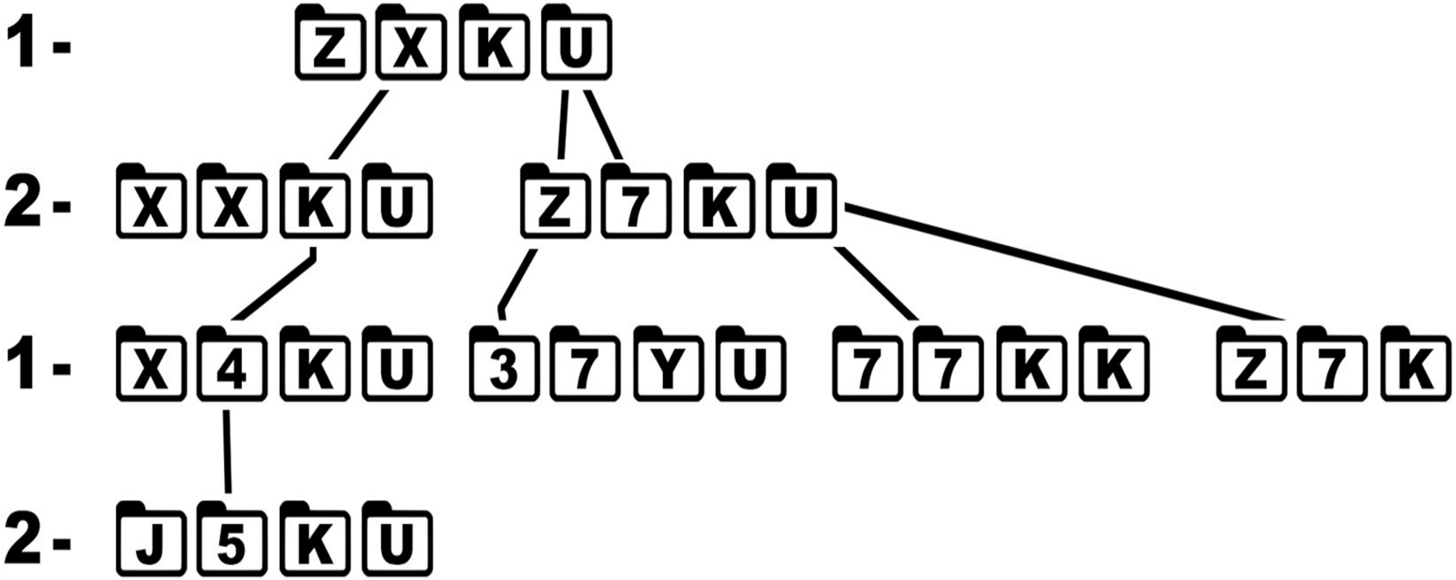
Metaplanning Strength

- Tight loop: leverage cached timeline
- Full metaplanning: plans of plans
- Replanning (e.g., P. Breimyer, P. Wurman. "PBA*: Using Proactive Search to Make A* Robust to Unplanned Deviations." AAAI. 2008.)

Planning



Metaplanning / Metaminimax



Metaplanning in Plosh: 1st Pass

- After every order, traverse timeline
  - evaluate time of interest (TOI) list
  - prioritize and focus on
    - earliest death
    - last focus/turn time
    - if more than 1 plosh, last idle time per plosh

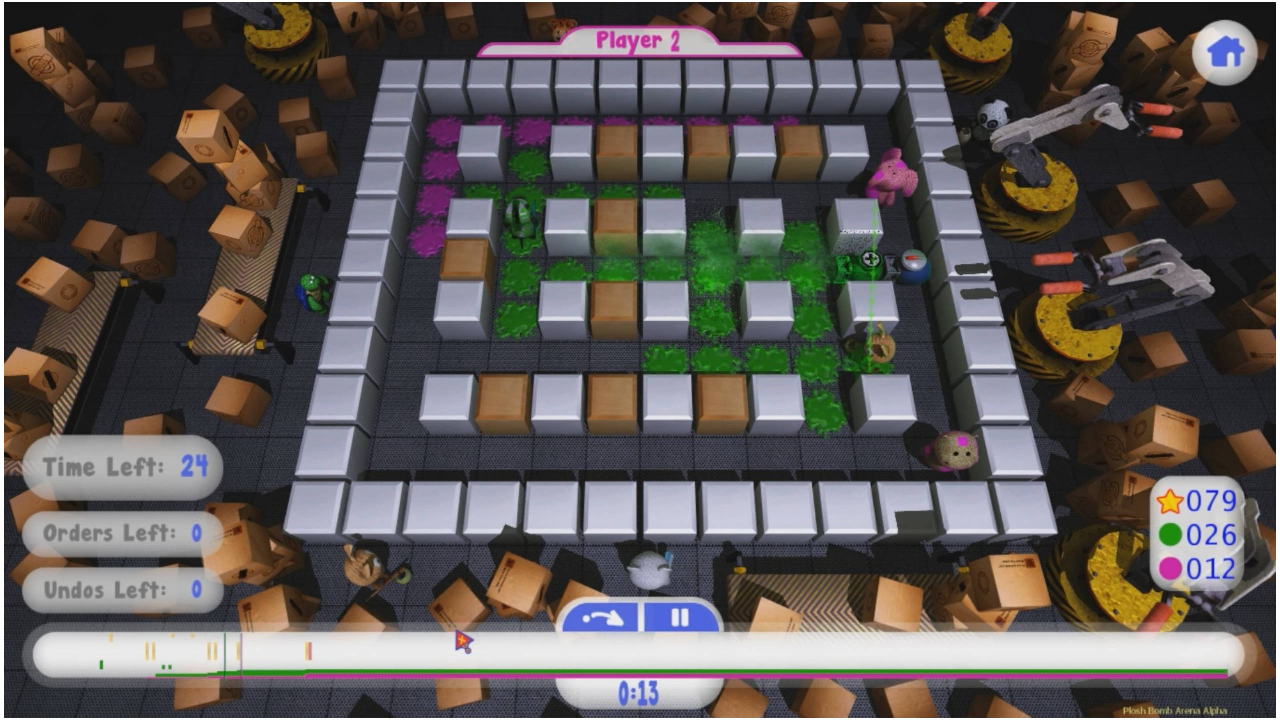
Metaplanning in Plosh: 1st Pass

Evaluate and prioritize every instant on timeline, scored from 0-15 based on situation:

- first possible danger
- any danger point
- the most dangerous point
- highest number of upgrades
- sudden changes in map paint coverage

1st Pass Heuristics

- If can't drop a bomb (recharging, limited ability to change things), then deprioritize
- At least one second different from other TOIs
- If cannot reach any enemy (they can't reach you), then deprioritize
- Check for enemy (or own) bombs
- Also check similarity to other TOIs - if too similar, take best
- Earlier in timeline prioritized
- Evaluate severity of bomb inverse of distance away



Time Left: 24

Orders Left: 0

Undos Left: 0

2nd Pass: Deep Dive

- Focus on top (3-5) TOIs per Plosh
- Jump to each TOI, execute over time areas
- Evaluate top 2 best actions

2nd Pass: Deep Dive

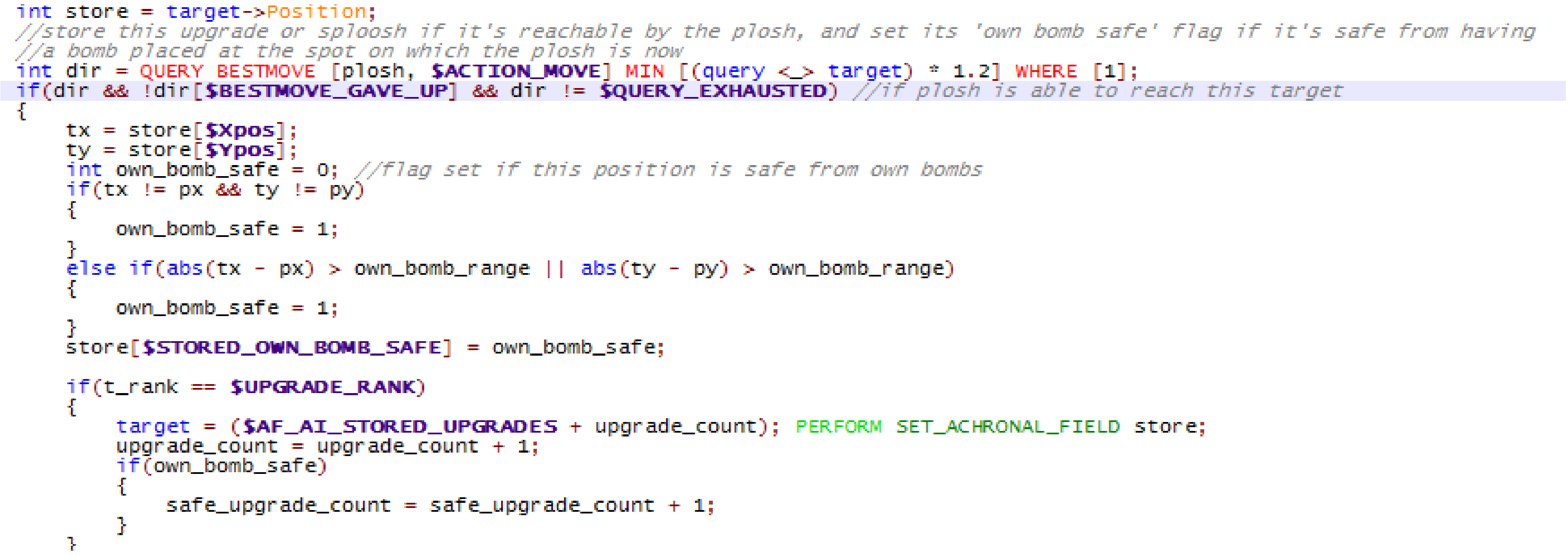
- If death
  - Jump to time of death, undo previous move
  - Jump back to previous move, reevaluate
  - Repeat *entire* process
- If plosh is in danger, evaluate at the current instant
- if plosh not in danger, run to end of action
  - Don’t second guess metaplan

Undoing Death

Deep Dive: No Immediate Danger

- Prefer less utilized plosh
- Mark everything visible as safe/unsafe from bomb drop in four cardinal directions
- Don’t bomb unless everything plosh cares about is safe
- Account for bomb cascades

Bomb Safety



```
int store = target->Position;
//store this upgrade or sploosh if it's reachable by the plosh, and set its 'own bomb safe' flag if it's safe from having
//a bomb placed at the spot on which the plosh is now
int dir = QUERY BESTMOVE [plosh, $ACTION_MOVE] MIN [(query <_> target) * 1.2] WHERE [1];
if(dir && !dir[$BESTMOVE_GAVE_UP] && dir != $QUERY_EXHAUSTED) //if plosh is able to reach this target
{
    tx = store[$Xpos];
    ty = store[$Ypos];
    int own_bomb_safe = 0; //flag set if this position is safe from own bombs
    if(tx != px && ty != py)
    {
        own_bomb_safe = 1;
    }
    else if(abs(tx - px) > own_bomb_range || abs(ty - py) > own_bomb_range)
    {
        own_bomb_safe = 1;
    }
    store[$STORED_OWN_BOMB_SAFE] = own_bomb_safe;

    if(t_rank == $UPGRADE_RANK)
    {
        target = ($AF_AI_STORED_UPGRADES + upgrade_count); PERFORM SET_ACHRONAL_FIELD store;
        upgrade_count = upgrade_count + 1;
        if(own_bomb_safe)
        {
            safe_upgrade_count = safe_upgrade_count + 1;
        }
    }
```

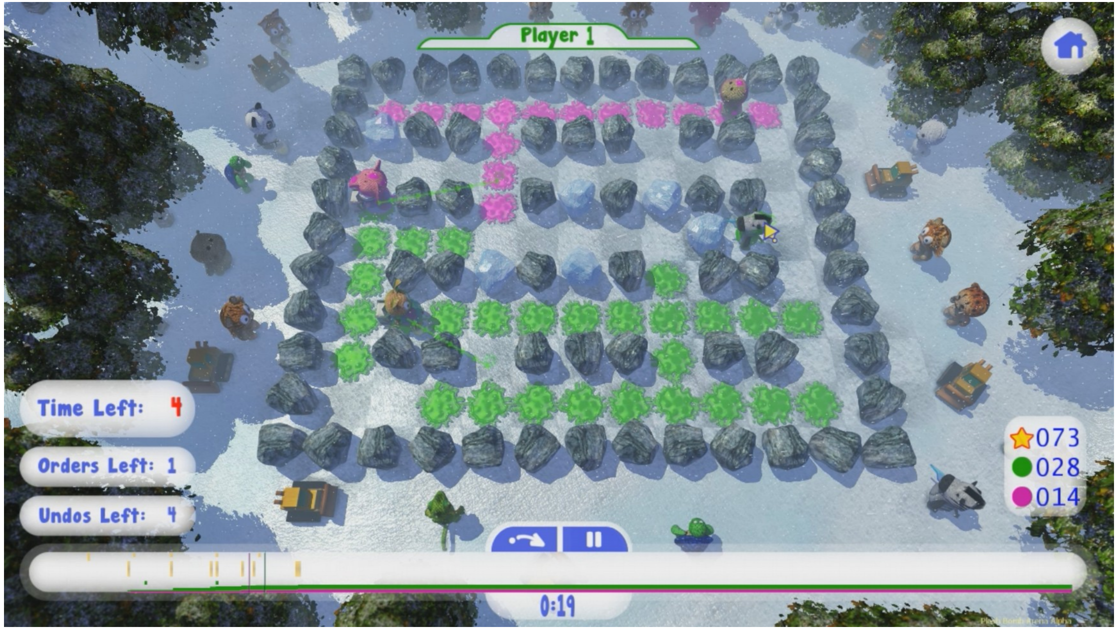
Deep Dive: Bomb Placing

- If place bomb, look at escapes
  - If upgrade easily reachable
  - Safe reachable location
  - Otherwise don’t bomb, just move (especially beginning of game on some maps)
  - Prefer hitting adversary with bomb blast

Deep Dive: Movement

- Find location with the least amount of paint in bomb coverage area
  - Randomize to remove contention
- Evaluate likely bomb blasts on the way

Deep Dive: Movement



3rd Pass: Selection & Execution

- Pick highest score TOI (prefer bombing efficacy, upgrades, overall safety)
- Validate impact of action on future plan
  - If adversary capabilities at position (bombs / territory) that will reduce plan strength, reevaluate plan with second best action
  - If adversary has strong position after both actions tried, try next best TOI
- Execute command
- If remaining moves, re-run entire process

Conclusions

Memory is cheap if state storage efficient

- Cache plans even if not making time manipulation game!

Likely to work well if can predict some player choices

Implicit iterative model of when AI should perform rare, costly actions

Questions?

cjhazard@hazardoussoftware.com

masinger@hazardoussoftware.com

http://hazardoussoftware.com

http://ploshgame.com