Improving Programming-by-Demonstration With Better Semantic Expression

Thesis Proposal

Richard McDaniel

November 14, 1995

Thesis Committee:

Brad Myers (chair)

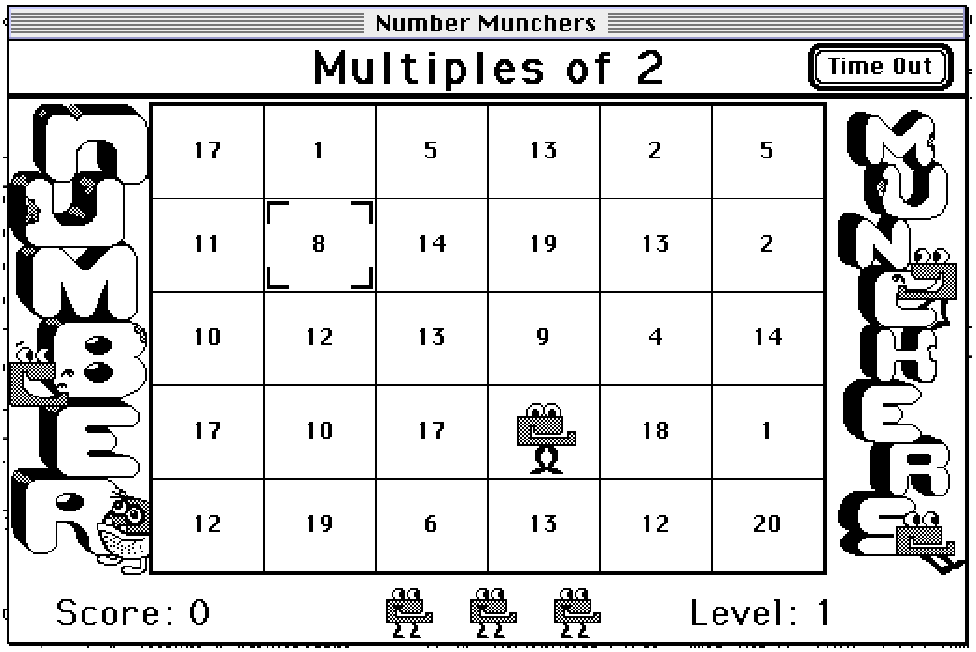
David Garlan

Roger Dannenberg

David Canfield Smith (Apple Computer)

For Example

Number Munchers



Outline

- Introduction
- Domain
- Plan of attack
- Related work
- Contributions
- Schedule

Programming-By-Demonstration (PBD)

Definition

To make a program using PBD, the user demonstrates the task by manipulating the system interface.

Uses for PBD

- Generating macros for an interface.
- Making applications for a small domain.

Intention

- Broaden domain of PBD.

Approach

- Apply and test non-language techniques that allow desired semantics to be expressed.

Thesis Statement

I claim that new techniques that I will develop and test will increase the expressive power of programming-by-demonstration to enable nonprogrammers to build a class of applications that currently requires programming.

Domain

The Big Domain

- Interactive Software
- Video Games
- Educational Games

The Actual Domain

- ***Complete*** applications
- "Board Games"

"Board Games" Is a Figurative Term

- The medium is a computer, not cardboard and paper.

Applications In Domain

- Computerized board games - Chess, Monopoly
- Video games - PacMan, Lemmings
- Educational games - Number Munchers, Reader Rabbit, Playroom

Domain

Properties of Board Games

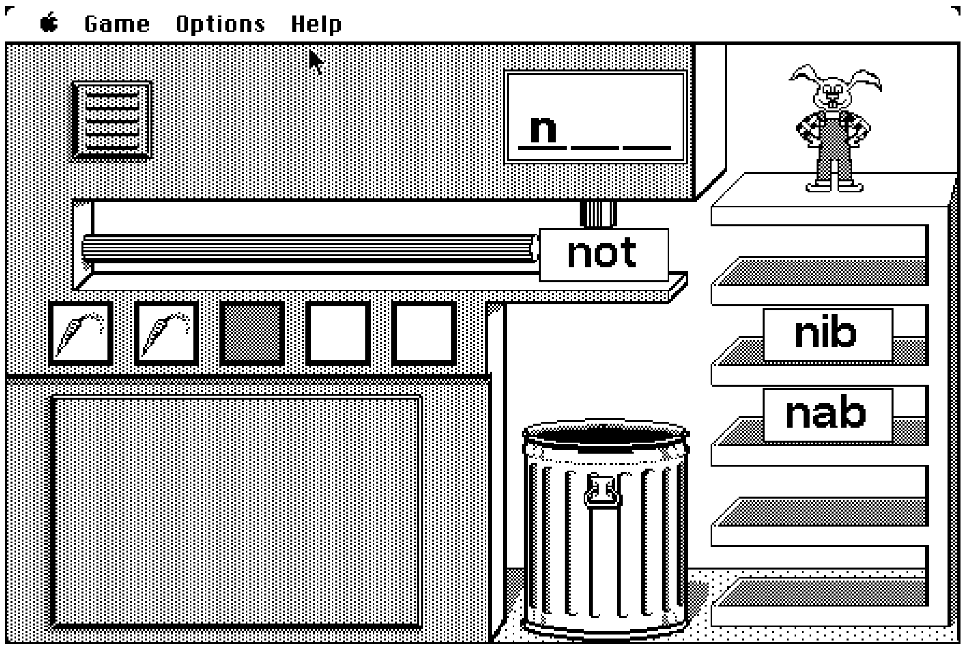
- Two-dimensional, object oriented graphics.
- Standard object terminology - pieces, board, cards, tiles.
- Does not rely on “physics.” No need for complex mathematics.

Rules Vs. Strategy

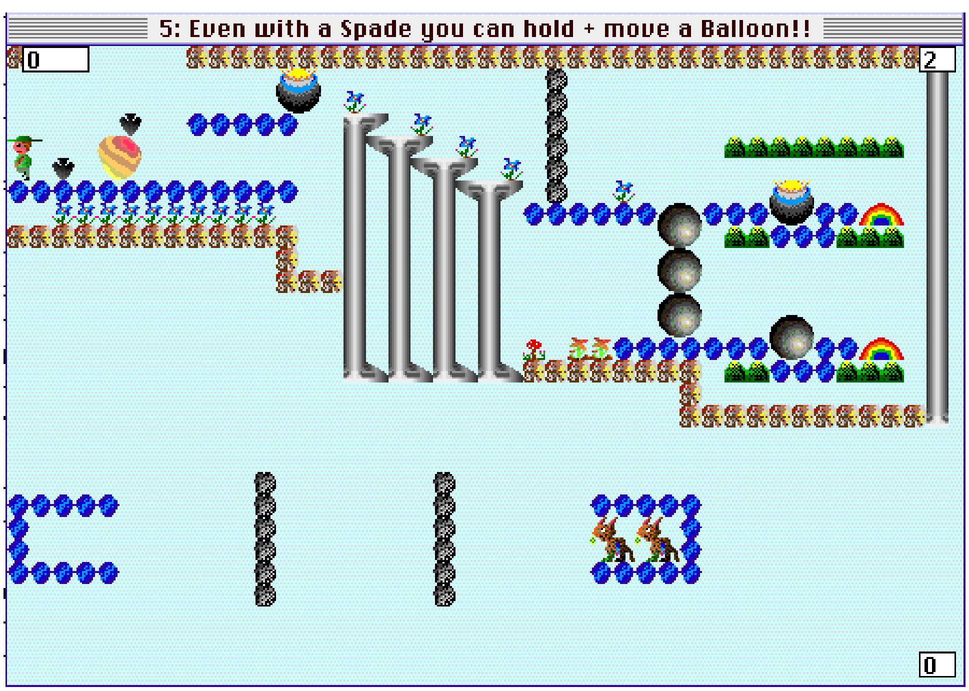
- Computer cannot serve as a player.
- Computer acts as referee for human players.
- Computer can perform some autonomous actions.
  - timer events
  - animations
  - moving “monsters”

Example Board Games

Reader Rabbit



Gold Pusher



Why Board Games?

What Is Good About Board Games?

- There are useful applications in the domain.
- Provides the needed level of sophistication.

Challenges Provided By Domain

- Many conditions
- Multi-step paths and connections
- Temporal relationships
- Counting things

Conditions

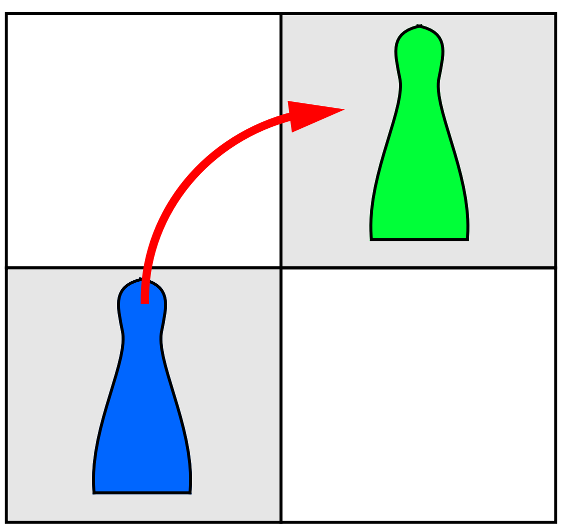
A condition makes a behavior perform different actions under different circumstances.

Examples

- Moving into an occupied space.
- Modes such as turn order or game phases.

Why Is This a Challenge?

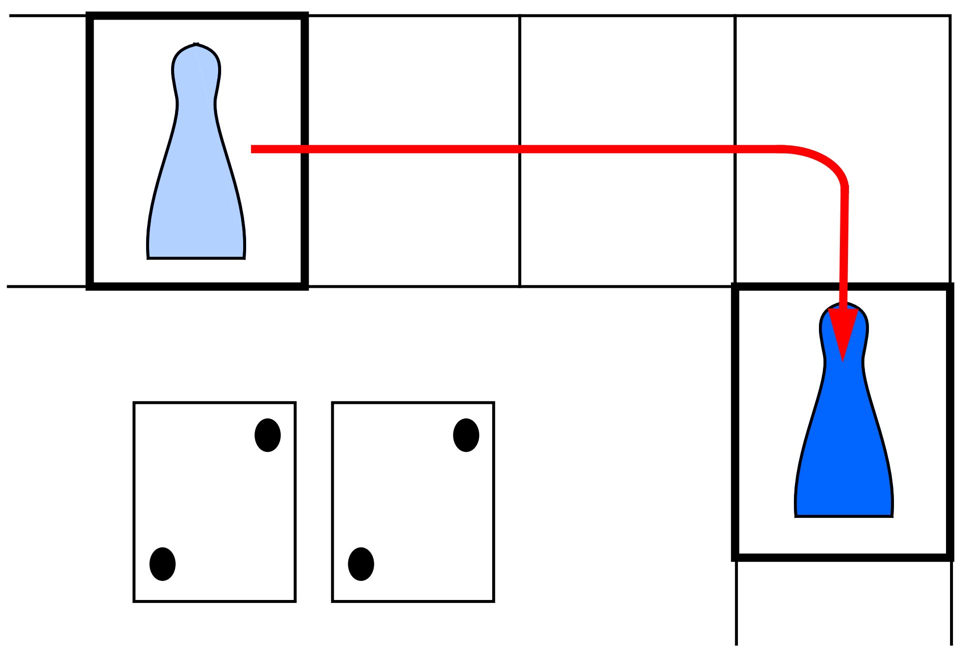
- We desire the system to infer most conditions.
  - Conditions are difficult to program.
  - Other systems assign conditions manually through dialog boxes.



Multi-Step Paths and Connections

Two Meanings:

- Graphical paths
  - Objects following a trajectory.
  - Determining legal next positions.
- Nested object descriptions
  - Objects described relative to other objects.
  - Actions with multiple arguments.



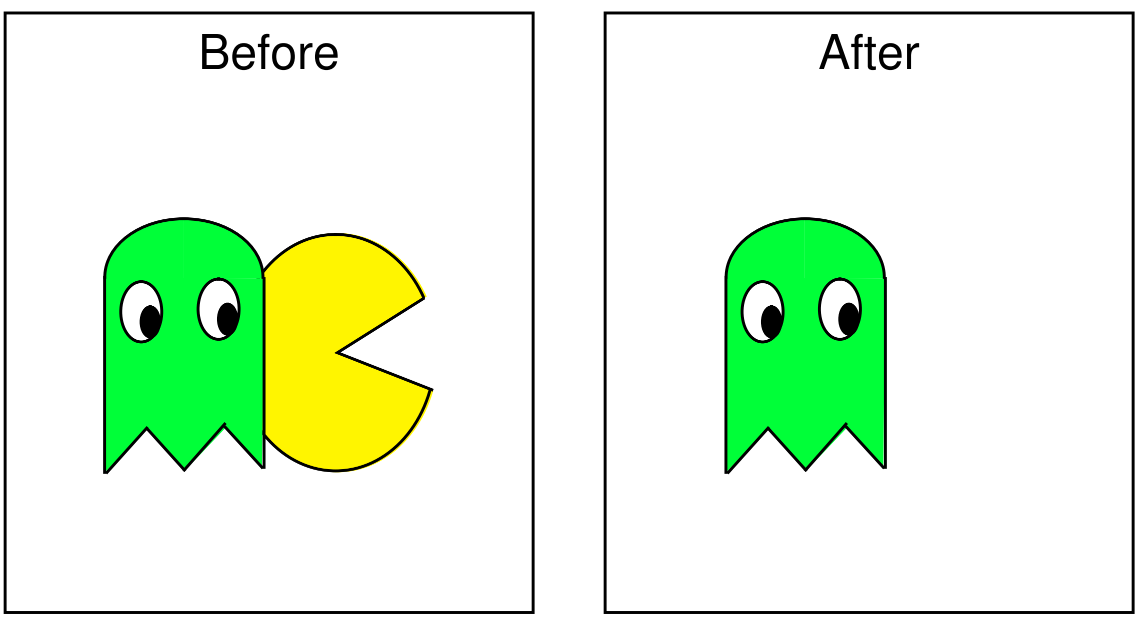
Temporal Relationships

Uses of Temporal Data

- Object trajectory depends on past location.
- Phases of turn occur in particular order.
- Animated actions depend on timing.

Why Is This Challenging?

- The past is overwritten with the present.
- One cannot manipulate what is not available.
- How much past should be remembered?
- How much detail should be remembered?



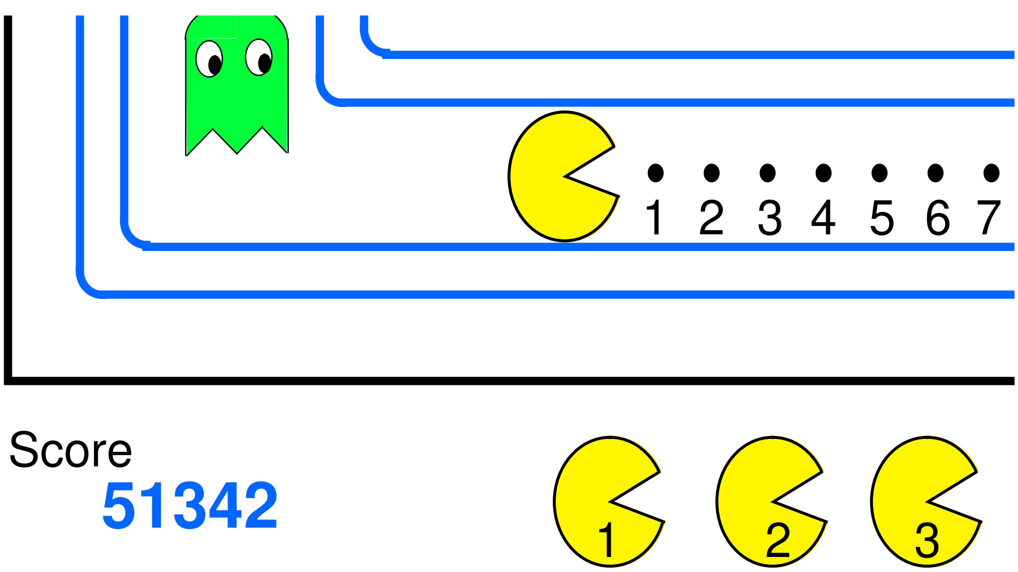
Counting Things

Uses For Numbers

- Counting a class of objects.
- Counting a relative distance.
- Keeping score.
- Counting a series of events.
- Repeat counts in behaviors.

Issues

- Integrating counting with inferencing.
- Some numbers should be determined implicitly.
- Some numbers do not appear in game objects.



Plan of Attack

New Demonstration Techniques

- Interaction
- Deck of cards
- Annotation objects
- Multiple views

Inferencing

- Knowledge representation
- Induction
- Plan recognition

Evaluation

- What to measure
- Kinds of tests

Interaction

Current PBD Systems Use Augmented Macro Recorder

- Separate record phases for stimulus and response.
- Must plan each demonstration carefully in advance.
- Difficult to specify negative examples.

Nudges

- Author runs the system and notices errors.
- In response mode, author corrects errors by manipulating misbehaving objects.
- Two special nudges:
  - Do Something! - behavior already specified but object is not responding.
  - Stop That! - object does something it is not supposed to do.

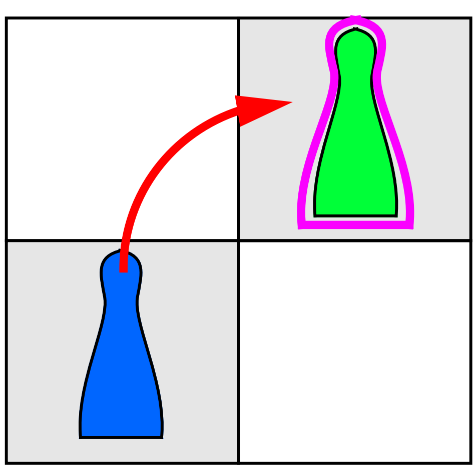
Interaction

Hint Highlighting

- Mark objects that contribute to the behavior.
  - predicates of a condition
  - intermediate stages of multi-step descriptions

Properties of Nudges and Hints

- Stimulus inferred by noting what events are referenced in the response.
- Permits Run/Build mode to be independent of Response mode.
- Behavior demonstrated ***as needed***.
- Hints provided voluntarily. (Hints may be requested in case of impasse.)



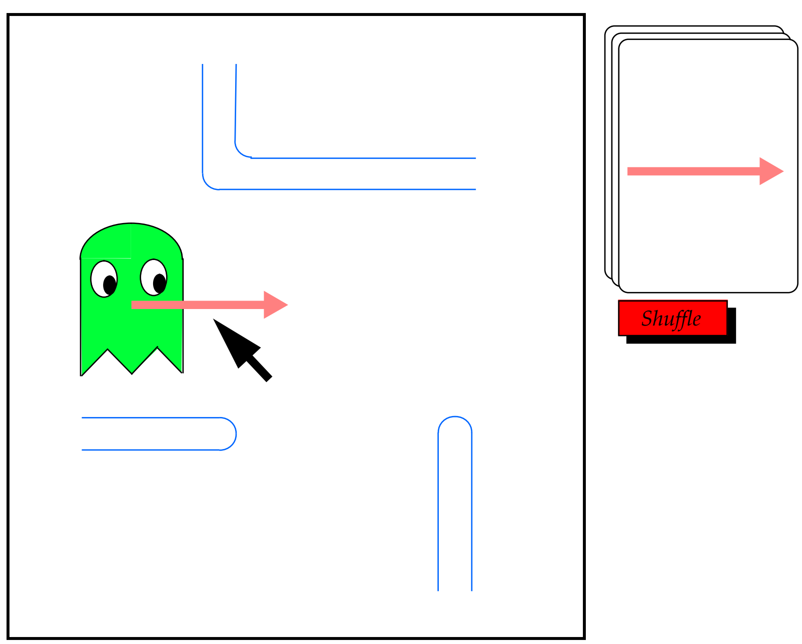
Deck of Cards

Uses

- Hold data
- Sequencing
- Random events

Advantages

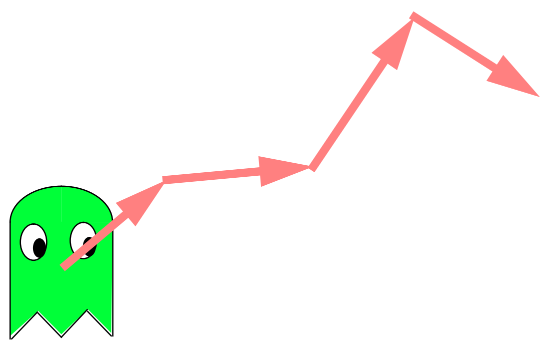
- Most users are familiar with concept.
- Operations are simple and well understood.
- Autonomous card playing can be powerful.



Annotation Objects

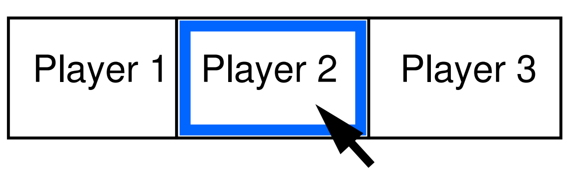
Guide Objects

- Indicate graphical constraints between objects.
- Placeholders, dividers, markers.
- Animation paths, hidden connections.



Mode Switches

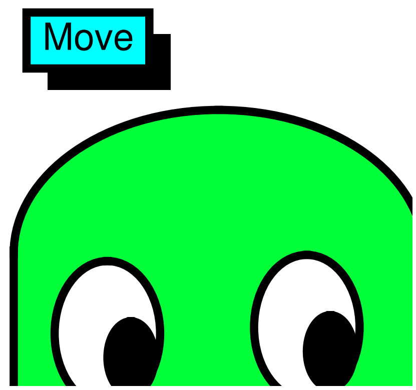
- Keep track of state variables.
- Generate custom events.



Annotation Objects

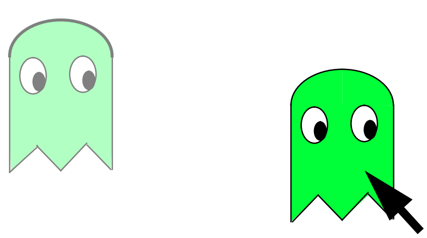
Behavior Icons

- Manipulable icons above objects associated with behavior.
- Cut, copy, paste transfers behavior between objects.



Temporal Ghosts

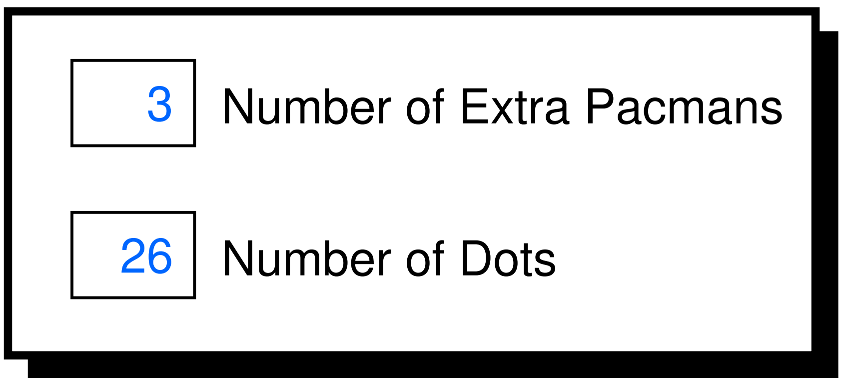
- Dimmed, translucent image of objects in previous state.
- Past states can be accessed directly.



Multiple Views

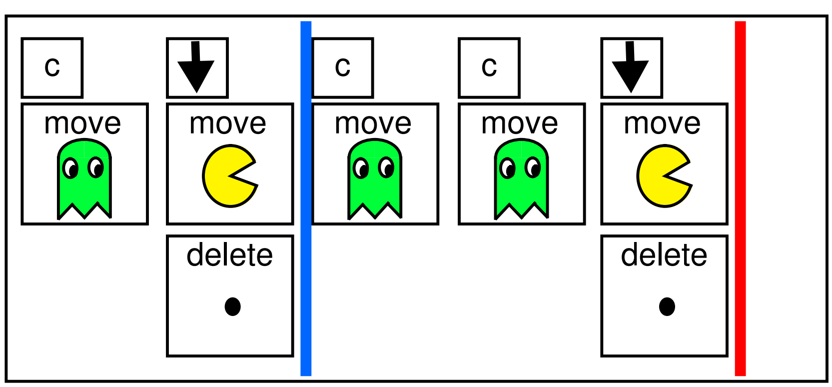
Counting Sheet

- Counts objects.
- Generates descriptions for numbers.
- Generates events for key numbers.



Timeline

- Can move forward and back in time.
- Graphical view of events and actions.
- Allows events and actions to be manipulated.
- Can regulate number of ghost objects.



Inferencing

Knowledge Representation

- Language for representing learned behavior.
  - must be searchable
- Encoding of general board game idioms and patterns.

Induction Algorithms

- Plan recognition
  - builds program structure
  - finds iteration and branches
- Classification
  - data description
  - predicate of a condition

Evaluation

The Actual Project

- Implement techniques as a unified tool.
- Implement inferencing algorithm.
- Evaluate tool on nonprogrammers.

What To Measure

- Completeness - use tool to build demonstration applications.
  - Number Munchers
  - Chess
  - something larger, like an adventure game
- Usefulness - find associate to use system - announce system to CMU community.
- Successfulness - formally test techniques with nonprogrammers.

Related Work

Programming-By-Demonstration

- Tools to build applications.
  - Peridot, Marquise, Demo/Demo II
- Tools to build games.
  - KidSim, Pavlov
- Tools that use hints.
  - Demo II (guidewires), Maulsby's work (Turvy, Moctec, Cima), KATIE (mentions use of highlighting)

Inductive Learning

- Classifiers (pure inductive learning)
  - ID3, decision tree, concept space
- Plan recognition
  - SIERRA, PROUST

Expected Contributions

Improving PBD and Expanding Domain of PBD

- Interaction techniques
  - Nudges
  - Hint Highlighting
- Deck of cards metaphor
- Annotation objects
  - Behavior icons
  - Temporal ghosts
  - Mode switches

Combining Techniques In a Unified, Well-Designed Framework

- Provides techniques as a tested whole.
- Can use system to build applications.

Schedule

Paper Prototyping (1 month)

- Find subjects to show techniques.
- Feedback to determine emphasis.

Implementation (8 months)

- Build user interface.
- Incorporate learning algorithms.
- Build demonstration applications.

User Testing (3 months)

- Announce system to CMU.
- Find subjects (must be nonprogrammers).
- Test how well they can make Number Munchers.

Write the Thesis (6 months)