Towards Adaptive Hour of Code

Tomáš Effenberger *Masaryk University Brno, Czech Republic*

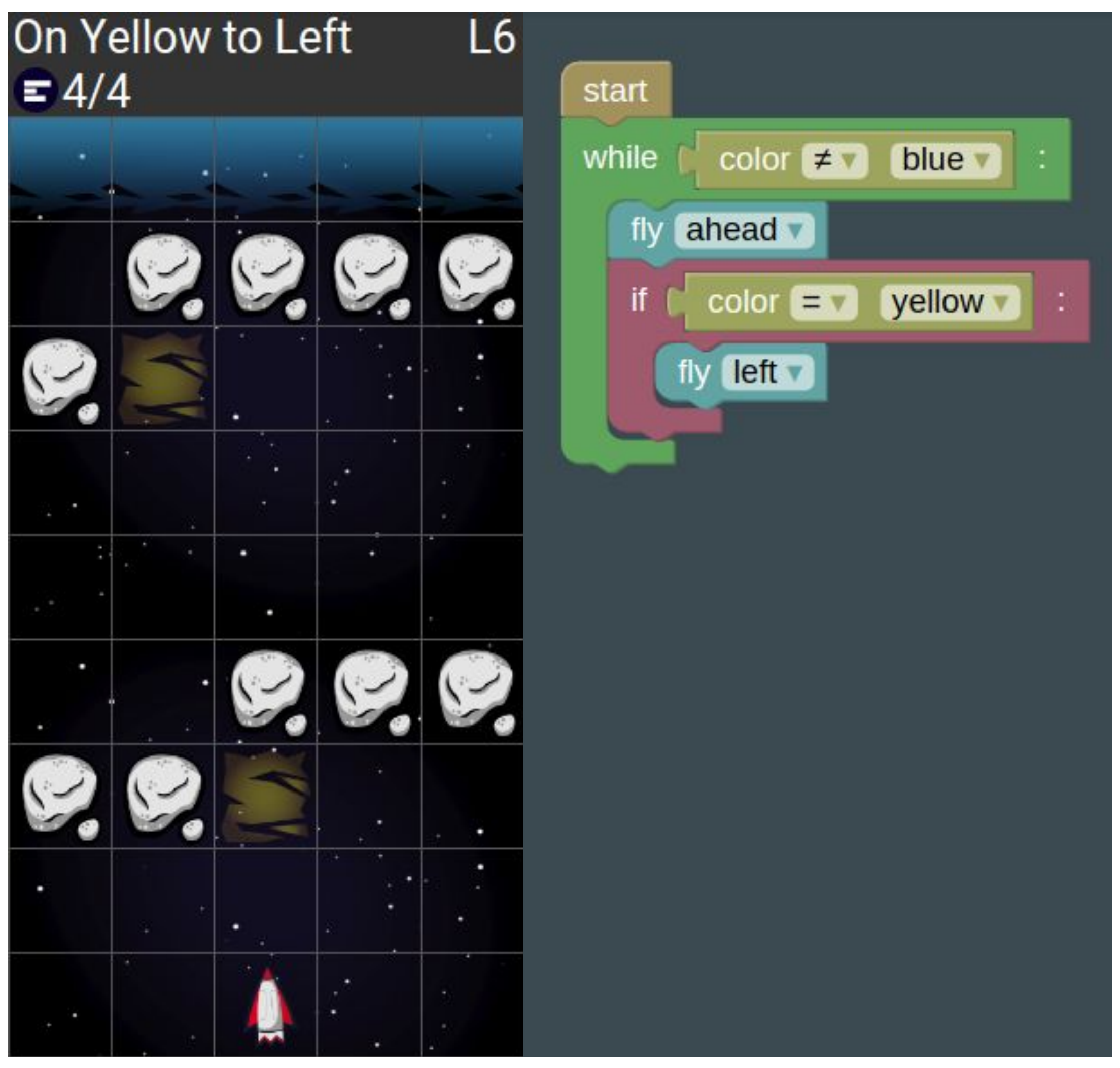
Motivation

– introductory programming

– millions of students

– typically fixed sequence of problems

Goal: adaptive behavior



```
start
while color ≠ blue :
    fly ahead
    if color = yellow :
        fly left
```

Research Questions

1. How to organize tasks for a personalized Hour of Code?
2. How to measure performance on programming problems?
3. How to predict the future performance?
4. How to recommend the next problem to solve?

Theoretical Framework

Proxy goal: optimal challenge *(zone of proximal development)*

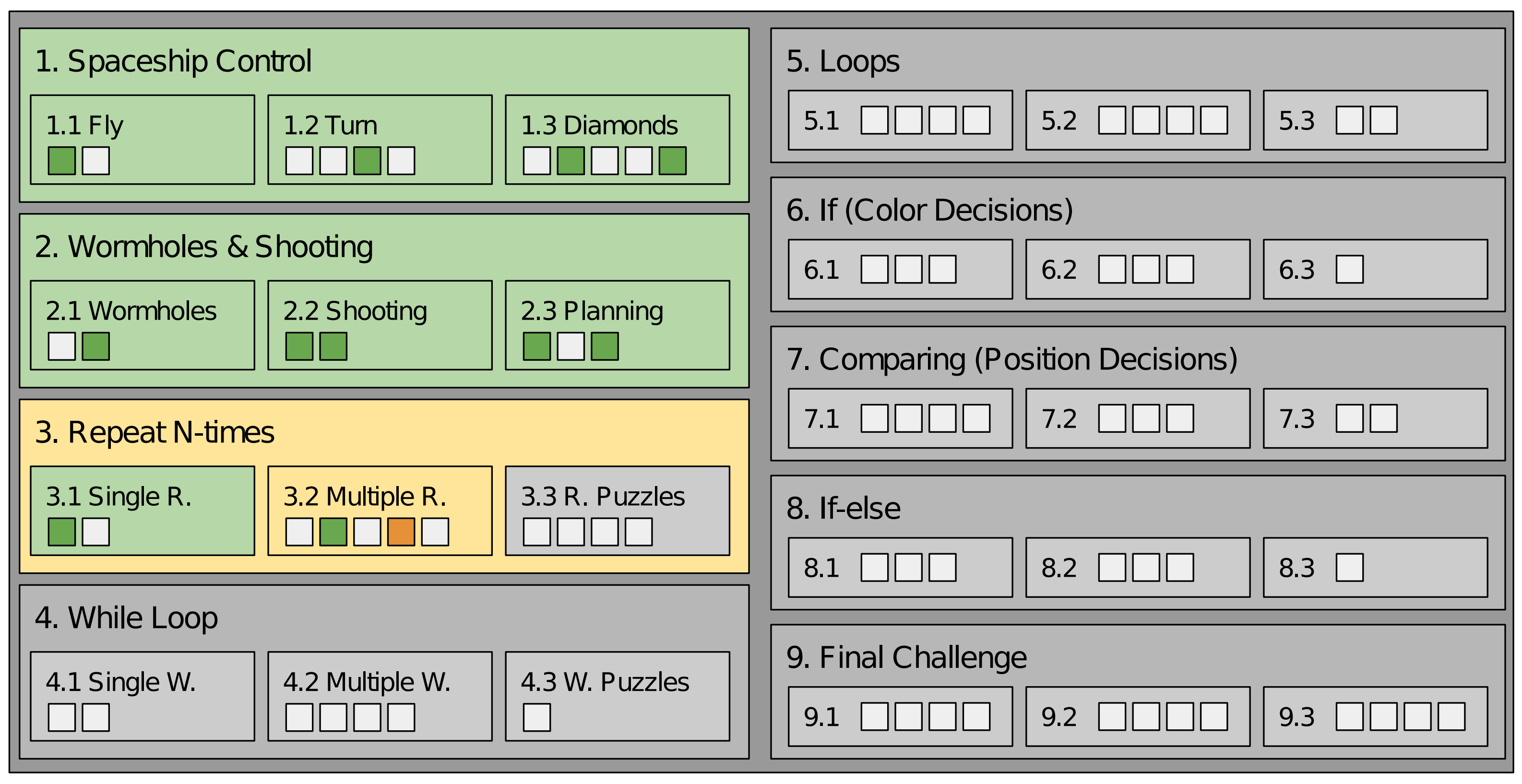
Adaptivity: design loop, outer loop

– domain model: hierarchical, linearly ordered levels

– performance measure: solving time, code submissions, ...

– student model: sum of performances per sublevel

– tutor model: mastery learning, random choice within sublevels



Methods

Exploratory analysis:

– problem difficulties, students' performance, concepts

– proxy evaluation of models

Online experiments:

– to compare tutor models

– proxy for learning: performance on *control tasks* (chosen randomly after each problem set)

Simulated experiments:

– exploring methodological issues

– ordering bias, attrition bias, learning, ...

Data

Multiple programming exercises and interfaces:

Exercise	Interface	Problems	Students	Attempts
RoboMission	blocks	85	3,800	62,500
Turtle Blockly	blocks	77	11,000	63,600
Turtle Python	text	51	2,400	11,900
Python	text	73	2,000	10,700

Granularity of data:

– RoboMission: every code edit

- Turtle: every code execution

– Python: every code submit

Expected Contribution

recommendations on the modeling approaches and evaluation methods in the context of introductory programming

Adaptive Learning

Research group, Masaryk university Brno

www.fi.muni.cz/adaptivelearning/