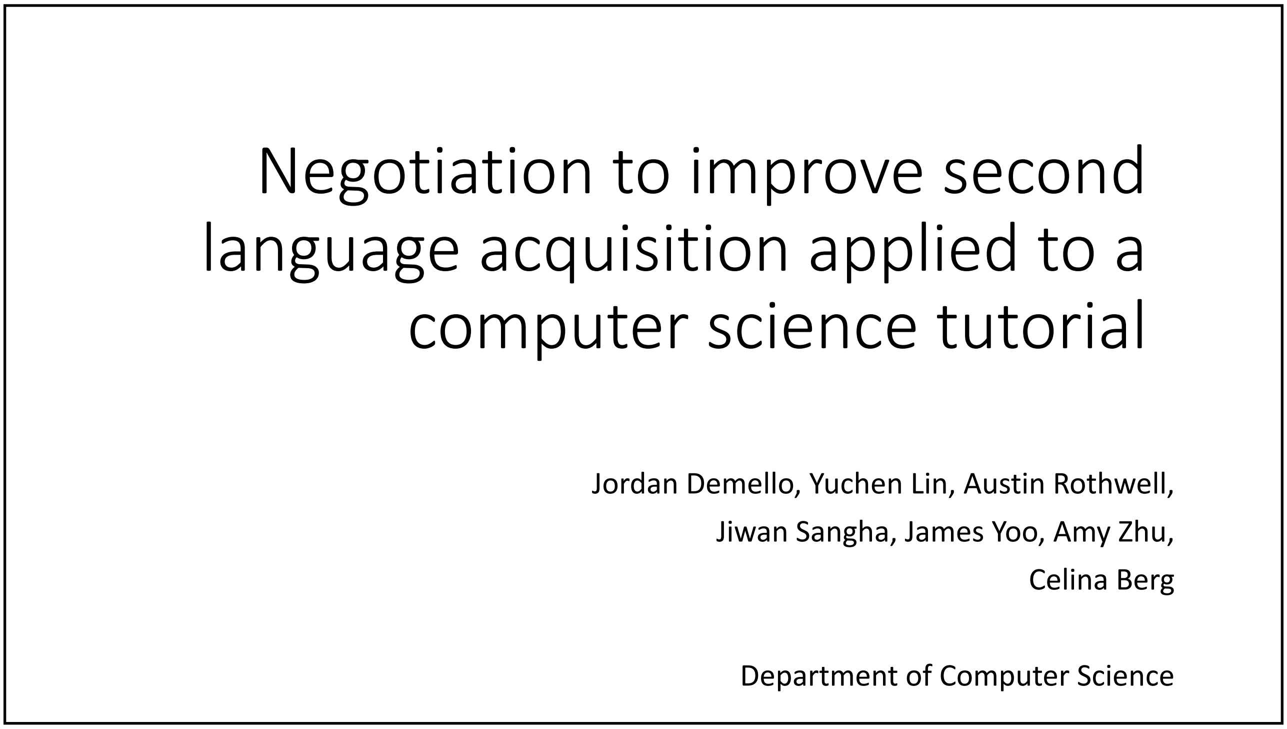
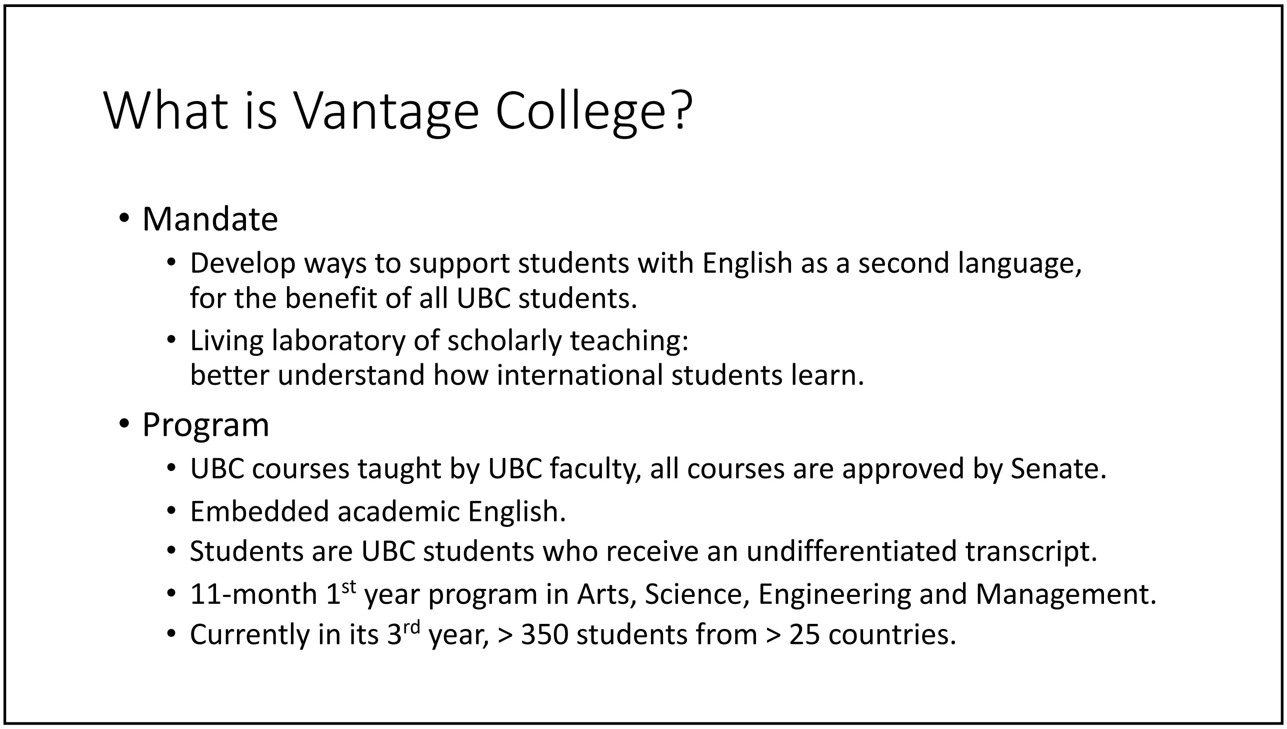


Negotiation to improve second language acquisition applied to a computer science tutorial

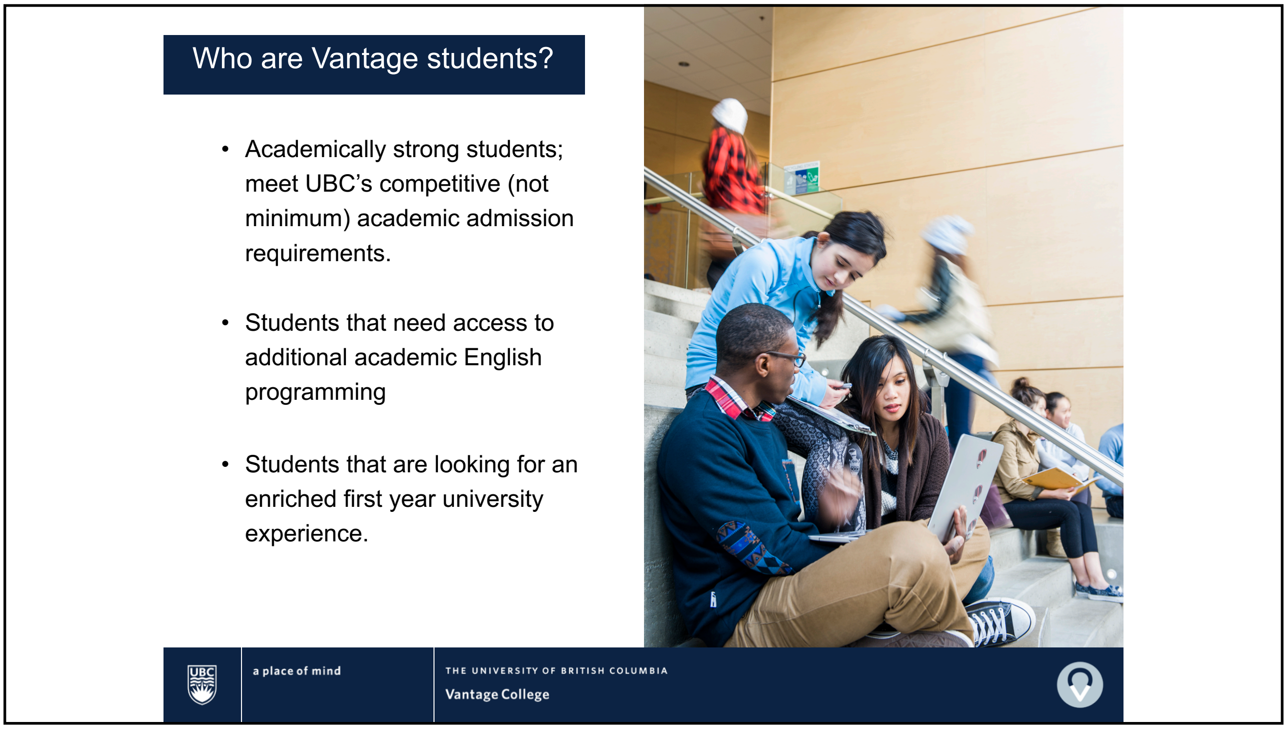
Jordan Demello, Yuchen Lin, Austin Rothwell,
Jiwan Sangha, James Yoo, Amy Zhu,
Celina Berg

Department of Computer Science



What is Vantage College?

- Mandate
  - Develop ways to support students with English as a second language, for the benefit of all UBC students.
  - Living laboratory of scholarly teaching: better understand how international students learn.
- Program
  - UBC courses taught by UBC faculty, all courses are approved by Senate.
  - Embedded academic English.
  - Students are UBC students who receive an undifferentiated transcript.
  - 11-month 1st year program in Arts, Science, Engineering and Management.
  - Currently in its 3rd year, > 350 students from > 25 countries.

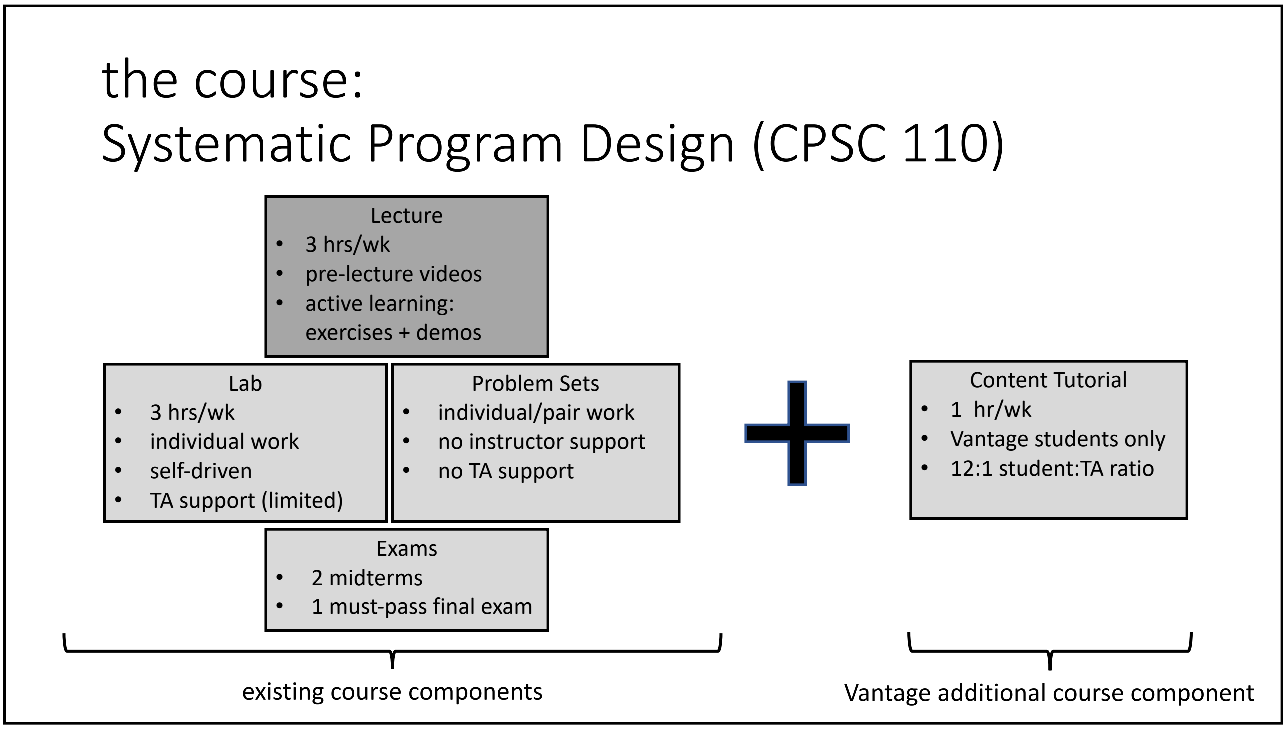


Who are Vantage students?

- Academically strong students; meet UBC's competitive (not minimum) academic admission requirements.
- Students that need access to additional academic English programming
- Students that are looking for an enriched first year university experience.

a place of mind

THE UNIVERSITY OF BRITISH COLUMBIA
Vantage College



the course:
Systematic Program Design (CPSC 110)

Lecture
- 3 hrs/wk
- pre-lecture videos
- active learning: exercises + demos

Lab
- 3 hrs/wk
- individual work
- self-driven
- TA support (limited)

Problem Sets
- individual/pair work
- no instructor support
- no TA support

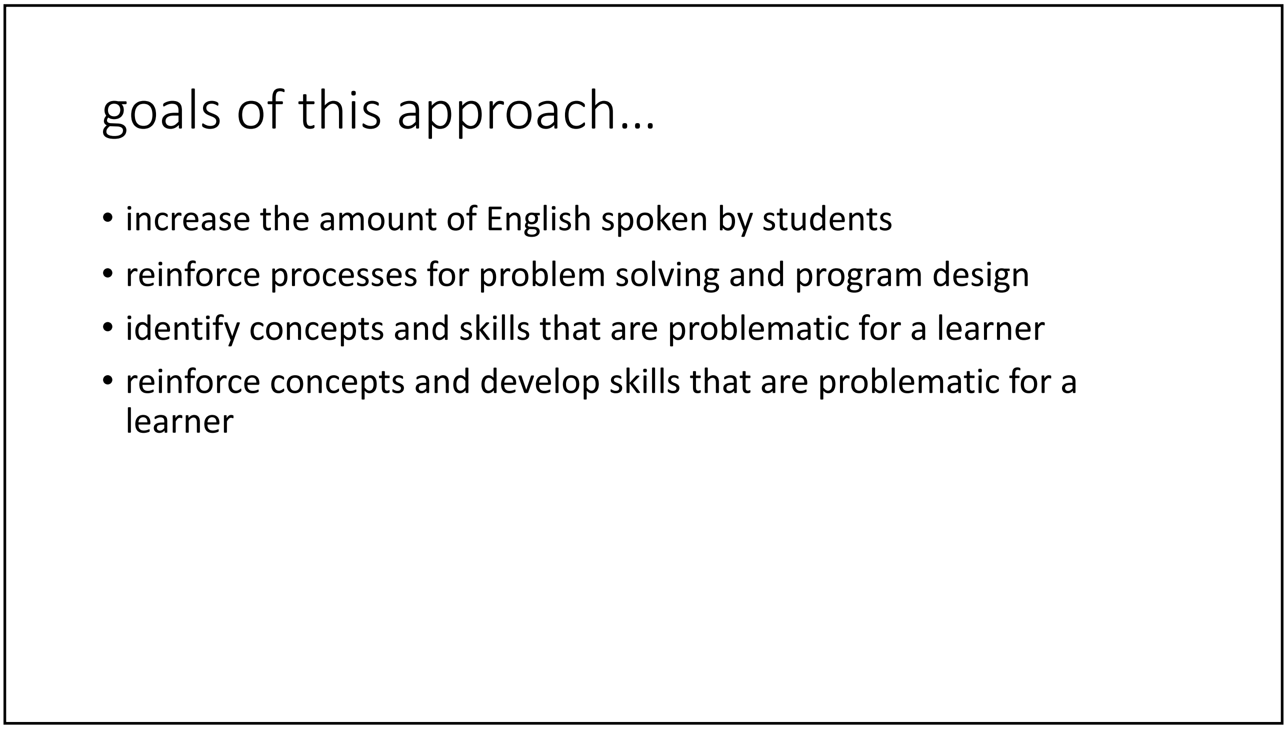
Exams
- 2 midterms
- 1 must-pass final exam

+

Content Tutorial
- 1 hr/wk
- Vantage students only
- 12:1 student:TA ratio

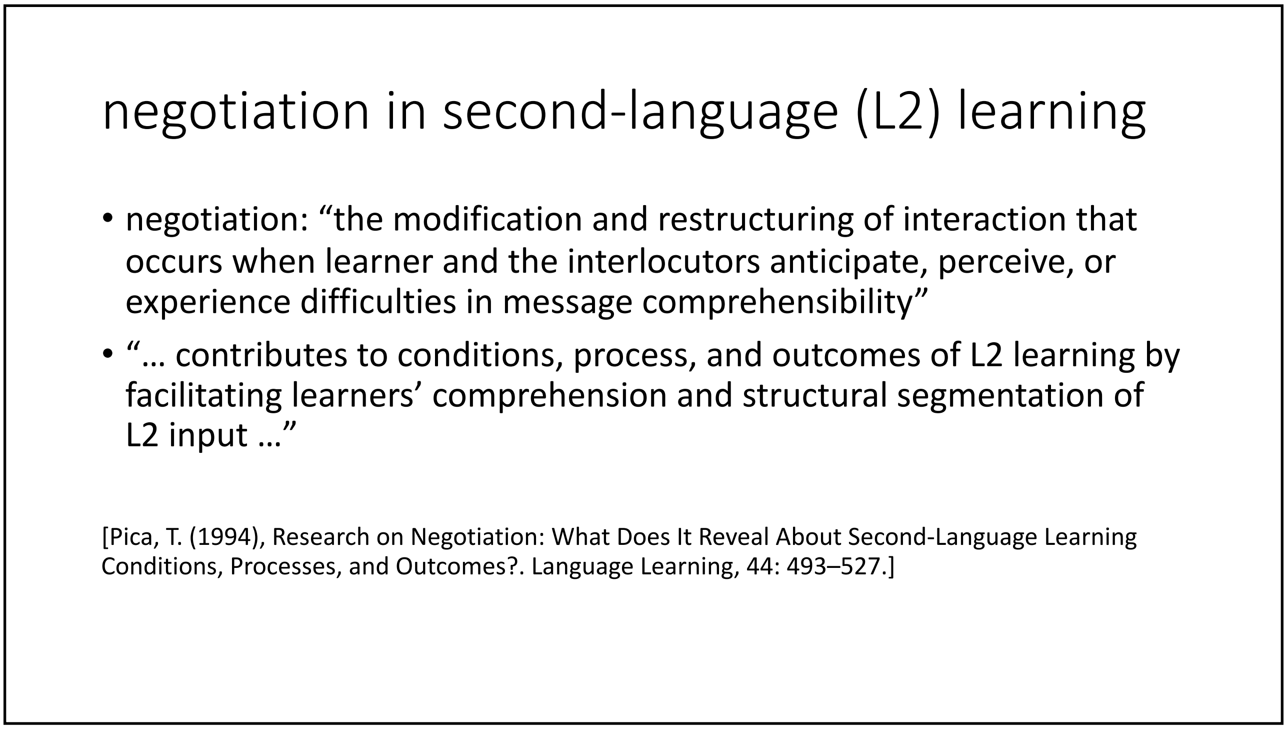
existing course components

Vantage additional course component



goals of this approach…

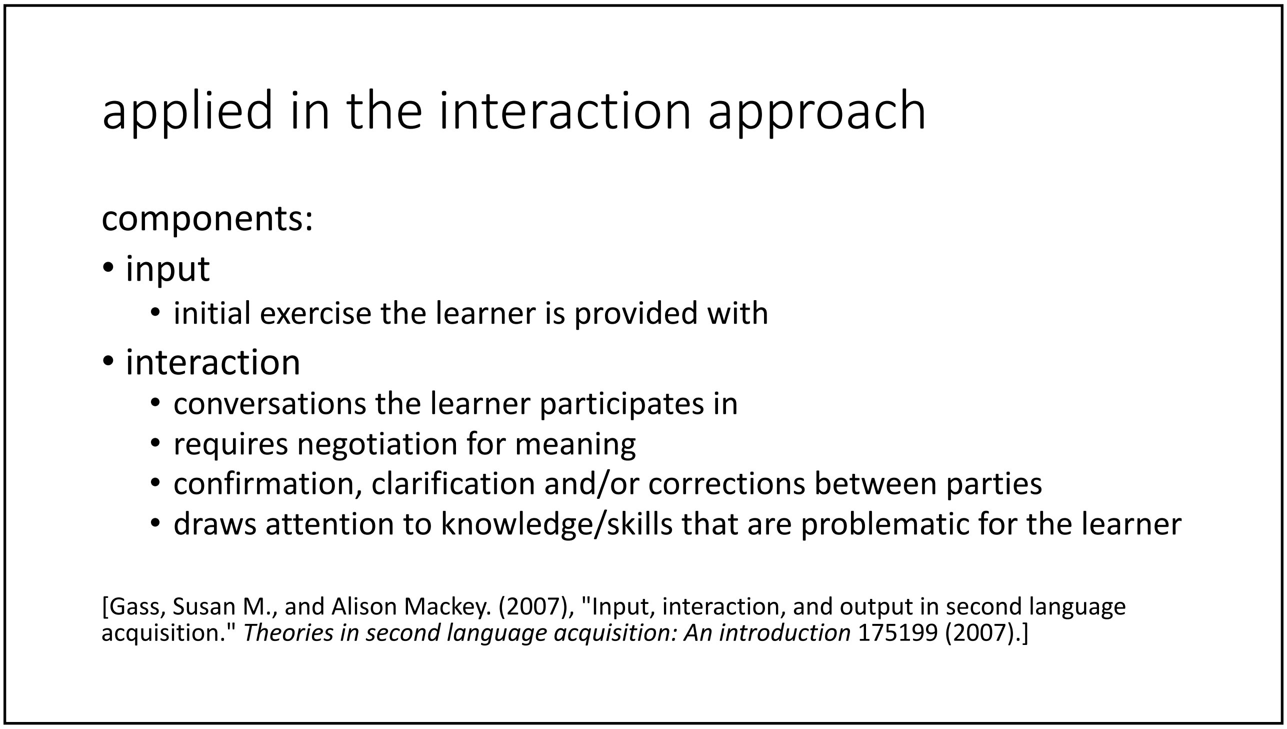
- increase the amount of English spoken by students
- reinforce processes for problem solving and program design
- identify concepts and skills that are problematic for a learner
- reinforce concepts and develop skills that are problematic for a learner



negotiation in second-language (L2) learning

- negotiation: “the modification and restructuring of interaction that occurs when learner and the interlocutors anticipate, perceive, or experience difficulties in message comprehensibility”
- “… contributes to conditions, process, and outcomes of L2 learning by facilitating learners’ comprehension and structural segmentation of L2 input …”

[Pica, T. (1994), Research on Negotiation: What Does It Reveal About Second-Language Learning Conditions, Processes, and Outcomes?. Language Learning, 44: 493–527.]

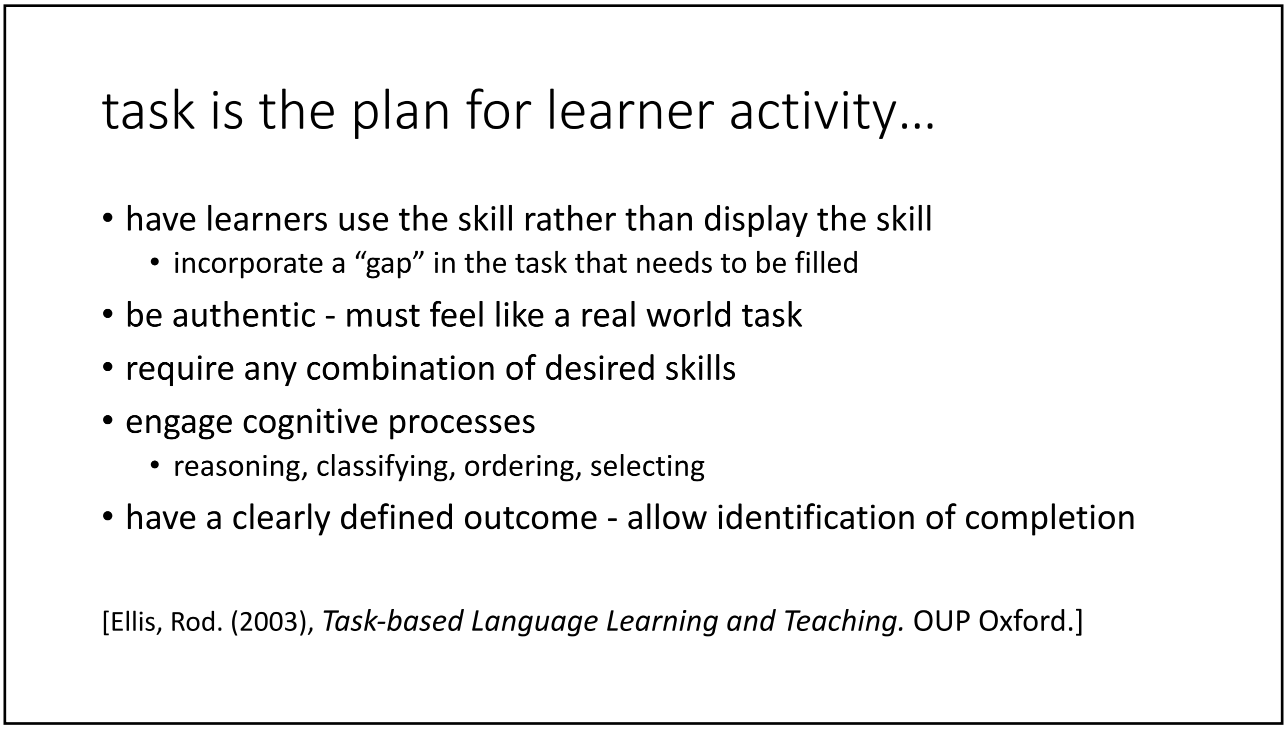


applied in the interaction approach

components:

- input
  - initial exercise the learner is provided with
- interaction
  - conversations the learner participates in
  - requires negotiation for meaning
  - confirmation, clarification and/or corrections between parties
  - draws attention to knowledge/skills that are problematic for the learner

[Gass, Susan M., and Alison Mackey. (2007), "Input, interaction, and output in second language acquisition." *Theories in second language acquisition: An introduction* 175199 (2007).]

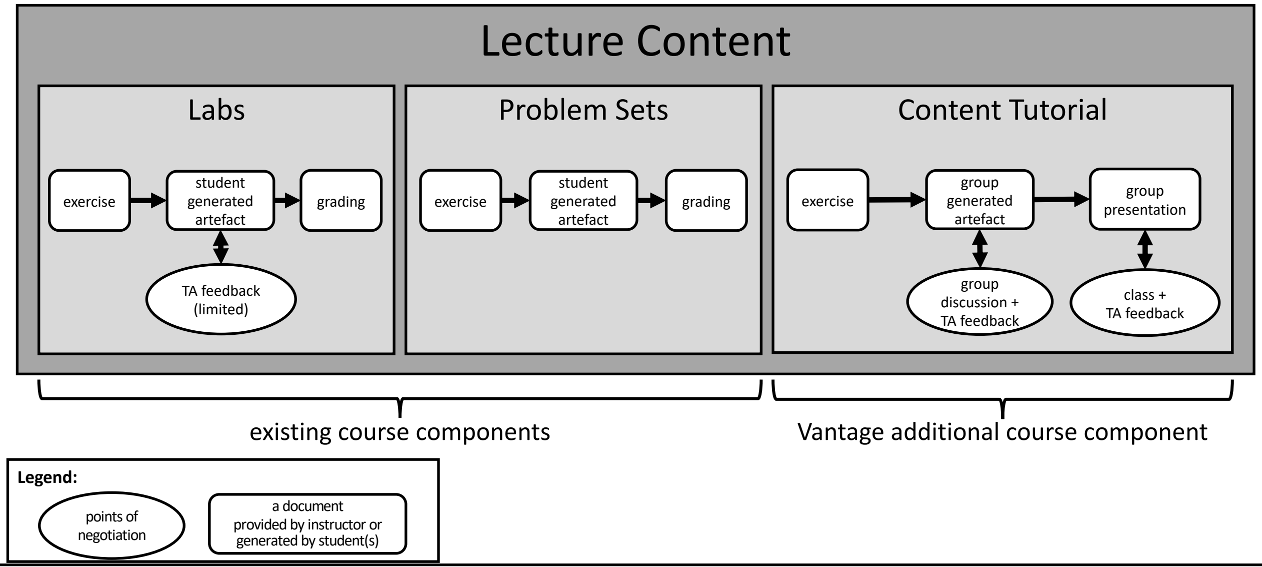


task is the plan for learner activity…

- have learners use the skill rather than display the skill
  - incorporate a "gap" in the task that needs to be filled
- be authentic - must feel like a real world task
- require any combination of desired skills
- engage cognitive processes
  - reasoning, classifying, ordering, selecting
- have a clearly defined outcome - allow identification of completion

[Ellis, Rod. (2003), *Task-based Language Learning and Teaching.* OUP Oxford.]

adding points of negotiation



Lecture Content

Labs

exercise

student generated artefact

grading

TA feedback (limited)

Problem Sets

exercise

student generated artefact

grading

Content Tutorial

exercise

group generated artefact

group presentation

group discussion + TA feedback

class + TA feedback

existing course components

Vantage additional course component

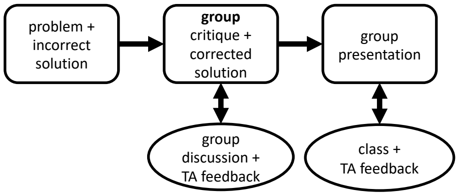
Legend:

points of negotiation

a document provided by instructor or generated by student(s)

types of tasks…

- Critique



problem + incorrect solution

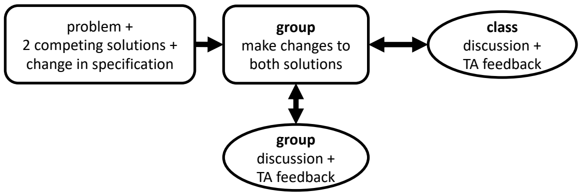
group critique + corrected solution

group presentation

group discussion + TA feedback

class + TA feedback

- Comparison



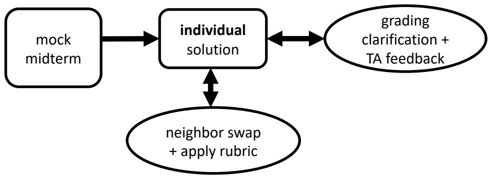
problem + 2 competing solutions + change in specification

group make changes to both solutions

class discussion + TA feedback

group discussion + TA feedback

- Practice midterm



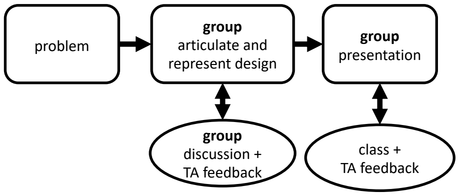
mock midterm

individual solution

grading clarification + TA feedback

neighbor swap + apply rubric

- Design



problem

group articulate and represent design

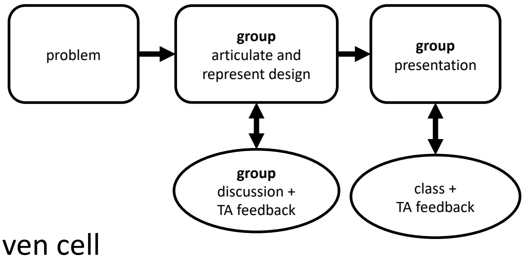
group presentation

group discussion + TA feedback

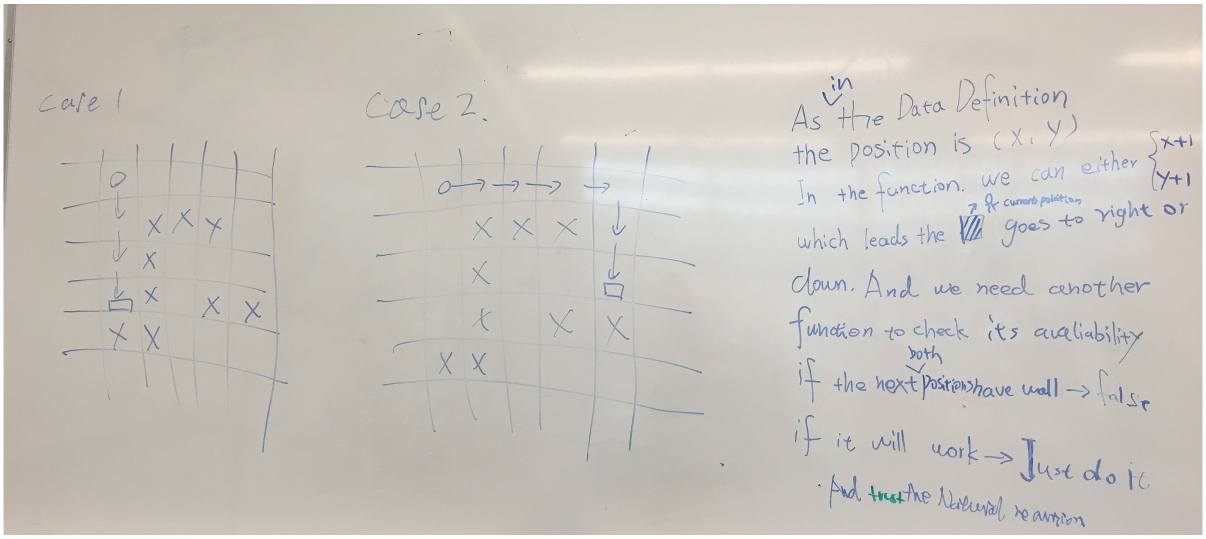
class + TA feedback

sample lesson plan - design

- Intended Learning Outcomes:
  - Identifying Generative Recursion problems from description
  - Difference between Generative and Structural Recursion
  - Generative Recursion – establishing what “problem”/ “changing information”
  - Template blending
- Task 1 – Discussion (5 minutes)
  - What makes this a gen rec problem?
  - Some helpful indicators…
    - No data defined to operate on
    - multiple possible next moves from any given cell


problem
group articulate and represent design
group presentation
group discussion + TA feedback
class + TA feedback

- Task 2 – Establish the problem/next problems (30 minutes)
  - Do not let the students write code yet!!
  - Ask them to sketch out a picture of their solution (problems + next problems).
  - If they don’t know where to start ask them to:
    - label the axis of the maze (x,y values)
    - walk through the cell of the maze and identify the changing information:
      - Do the values of the cells change?
      - Does the position you are at in the maze change?
  - What they should have to share:
    - a clear description of what the problem is: a position
    - what it means to be in a solved position
    - an example problem (position 0,0) with next problems (list … … …)
    - in words, the steps to generate the next problems

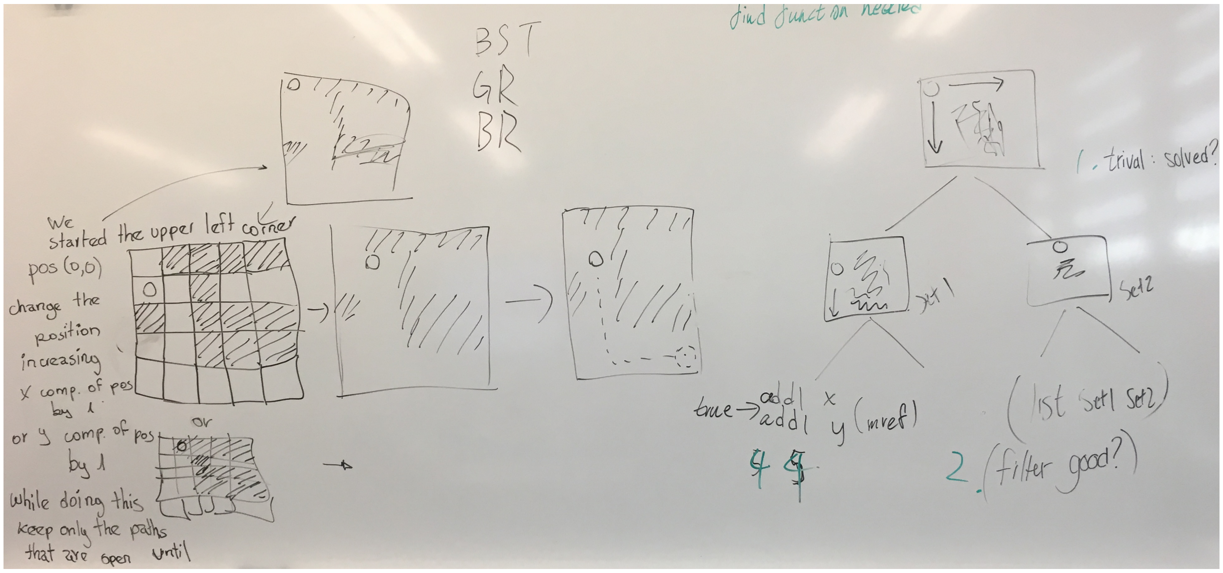


As in the Data Definition the position is (X, Y) In the function. we can either {X+1 / Y+1} which leads the ▨ & current position goes to right or down. And we need another function to check its avaliability

if the next both positions have wall → false

if it will work → Just do it

· And trust the Natural recursion



We started the upper left corner pos (0,0) change the position increasing x comp. of pos by 1 or y comp. of pos by 1 while doing this keep only the paths that are open until

true → add1 x
add1 y (mref)

(list set1 set2)

TA perspective: student engagement

high engagement

- split time: group/individual work + class presentation/discussion
- debate over correct/best solution to a problem presented
- working through problems
- getting hints + solutions to problems given

low engagement

- attendance
- presentations
- asking students to explain how to solve a problem in English
- focus on design vs. code solution

TA perspective: when learning moments occur

the moments

- groups solve a problem +
  group presentations + class critique
- debate over correct/best solution to a problem presented
- writing down problem-solving strategies and applying those strategies to a harder problem
- applying problem-solving strategies

the challenges

- getting students to collaborate outside of friend-groups
- weak presentation skills +
  non-attentive classmates
- writing down problem-solving strategies
- communicating ideas outside of code
- desire to jump right into
  writing a code solution

TA quotes

- “The intimacy afforded by the small class size allowed us to get to know everyone's names, and fostered a sense of community between us and the students.”
- “Overall teaching this group of students is exciting because most are very smart and determined to grasp the content.”
- “The best learning moments were when multiple students were having heated debates about the best answer to a question.”
- “Trying to ask abstract questions (the answers to which would be complex or ambiguous even in your native language, such as coding process) felt like drawing blood from a stone oftentimes.”
- “Hated having them leave class without providing them with a thorough explanation of the solution.”

next steps…

- How to de-emphasize solution and emphasize process?
- How to encourage articulation of process?
- How to encourage attendance?
- How to measure impact?
- Application to lecture or lab environment?