ITSs
Intelligent Tutoring Systems

Goals

- Goal of ITSs
- What ITSs need to do
- How ITSs work
- When they are successful (and not)

What problem are we solving?

- Not enough human tutors to help students
- Individualize learning beyond the 30-to-1 ratios in classrooms

Customizable aspects of instruction

Where you are in the curriculum (content)

Difficulty of problems

What parts of content we review / spiral and build upon

Feedback you get

Analogies / examples used to explain new concepts

Account for learning differences

What they need to do

- Adapting to the individual student
  - Change the example they are giving
  - Feedback on errors
  - Speed at which the material is taught
  - Learning trajectory – order and depth that things are taught
  - How fine or coarse grain the learning goals are

What are ITSs
Distinguishing Features

ITS has two particular jobs:

1) Choose the next problem
2) Provide feedback

What are ITSs
Distinguishing Features

Model of what the student knows and feels

Decide what information to give to a student (pedagogical model)

Domain knowledge

Learner module – addresses mental state of student

User Interface

Detect when the user makes errors, what those errors are

What are ITSs
Distinguishing Features

- Boiled Down: Two main jobs:
  - Given incorrect answer, provide useful feedback
  - Given behavior, choose next problem

What would perfect, personalized computer-based instruction look like?

Whenever a student makes an error, the ITS provides feedback that addresses the underlying issue phrased in a way the student understands

Feedback is not too frequent, frustrating student

Engaging so student stays interested (not over Zoom)

What would perfect, personalized instruction look like?

- Perfectly paced schedule (topic when you're ready)
- They know how to motivate you
- Positive but constructive feedback
- Respond to emotional state, adjust pedagogy accordingly
- Bring elements of student's external life to teaching
- Uses an interface adapted for student needs
- Knows sources of errors so can provide instruction / feedback appropriately
- Trust between tutor and student

What are the components of feedback?

-

What are the components of feedback?

- Source of the error
- Number of times the student has made that error
- Difficulty of the content
- Emotional state of the student
- Immediacy
- Positive or negative reinforcement

What are things to consider when designing feedback?

What are things to consider when designing feedback?

- Level of feedback
- Frequency of feedback
- The exact sentences to use

Which are the hardest / have the most variance?

- Showing empathy and acknowledging learner's emotional state
- Understanding how the student responds to different types of feedback
- Figuring out the true source of the error
  - Depth of the error

What are the components of choosing the next problem?

-

What are the components of choosing the next problem?

- The average pacing the student has done so far.
- Incremental beyond the last question
- Mastery model – continue asking questions on the same subject / level of difficulty until mastery is achieved (mastery != perfect)
- Types of problems (worked example, faded worked examples) -> remove scaffolding as students progress within level
- What their answer was for the last one (whether it's correct, in what way it's incorrect)

What are things to consider when choosing the next problems?

- How well they've been doing on the problems so far.
- What subgoals still need to be achieved
- How long they've been working on the same skill
- Someone could get something "right" but have an approach that's not efficient or is missing some element
- How easily students get frustrated

What aspects have the most variance for choosing the next problem?

- Mental state of the student
- The sources of errors
- Student's outside interests
- Choosing / offering from a wide variety of instructional approach / problem type

Midterm

- In-Class on Thursday
- Three goals of this class:
  - C# programming (O-O Concepts)
  - Game design (appeal to wide variety of users)
  - Learning theory (needs to teach using good pedagogy)

MVP

- Next week
- Read specs carefully
  - Menu screen that shows how to navigate to different levels or splash screen or something
  - One introductory level to show me how you introduce new, simple concepts
  - Two intermediate levels that are adjacent in difficulty to show how you exercise the same skill at different difficulties
  - 2+-d movement with collision detection
  - Don't need all your bells and whistles
  - Legal to break all copyright within an educational project

Where would ITSs be useful? Where would ITS NOT be useful?

- Make list of your own for learning & gaming reasons why / when / where ITSs should work (5 min)
- Go to breakout groups of 2-3 to combine and discuss your lists and make a cohesive argument, enter into Gradescope (10 min)
- Share arguments (15 min)

Where would ITSs be useful?

- Language learning (vocabulary)
  - Easy to detect errors
  - Decide what level of error to give feedback on
- Middle School +
  - Younger students have short attention spans
- Problems w/ multiple concrete stages
  - Easier to diagnose the error
- Situations with clear correct answers, no subjectivity
  - Memorization / multiplication / vocabulary / spelling
- Situations where learners are already motivated
- Reviewing / practicing rather than learning

NOT useful

literature / essays / things with no specific right answer
Little extra benefit over flash cards
Not useful if learner is not used to / doesn't trust computers
Complex questions for which they cannot show work
Group or collaborative learning - device might change the dynamic
If devices need to be shared, can't individualize
Not a complete replacement because some thrive in group / classroom environment
Feedback hard if it's based on building / exploring in a constructionist

Blended teacher & ITS

- First interactions always with the teacher
  - Allows student questions
  - ITS used for practice / homework
- Intro video could replace initial instruction, then questions are for the teacher
- Can be used well outside the classroom
- Teacher could go around and provide explanations / narrow down sources of errors

What's the real story?

- Disconnect between programmers & education
  - Only programmers can add content
  - Education experts needed to craft good feedback.
  - Interdisciplinary work is hard
- Systems are expensive and on top of teachers
- Role of teacher during instruction not well defined
- Limitations on the types of problems given and types of instruction given
- Teachers don't appreciate their roles being disrespected by implying this software is better

Design your own EdTech / ITS system

- Join in groups of 2-3 to combine and discuss your lists and make a cohesive list of suggestions (10 min)
- Share suggestions (20 min)
- (not putting into Gradescope, just sharing out)

How do you improve…

- Technical learning
- Motivation / Emotion
- Do not limit yourself to desktop
  - Use all features of modern devices

Replacing homework

- Works with teacher for learning expectations
- Specific amount of practice each week
- Problems match the learning level better
- Teacher is sent info on which and on what students are struggling
- Does not remove interaction b/c it's only replacing worksheets
- Need motivation – time or skill level? Each session ends at time and/or skill level.
- Nice differentiation
- Levels the playing field for students who can't afford tutor
- Could integrate a badge system
- Harder to "show your work"