Automatic Curriculum Generation of Training Scenarios in Virtual Reality for an Efficient Skill Acquisition

Isaac Gonzalez[1], Edward Kim[2], Alton Sturgis[2], Boxi Fu[2*], James Hu[2*], Yunzhong Xiao[2*], Kyle Cui[2*], Daniel He[2], Zachary A. Pardos[2], Bjorn Hartmann[2], Sanjit A. Seshia[2]

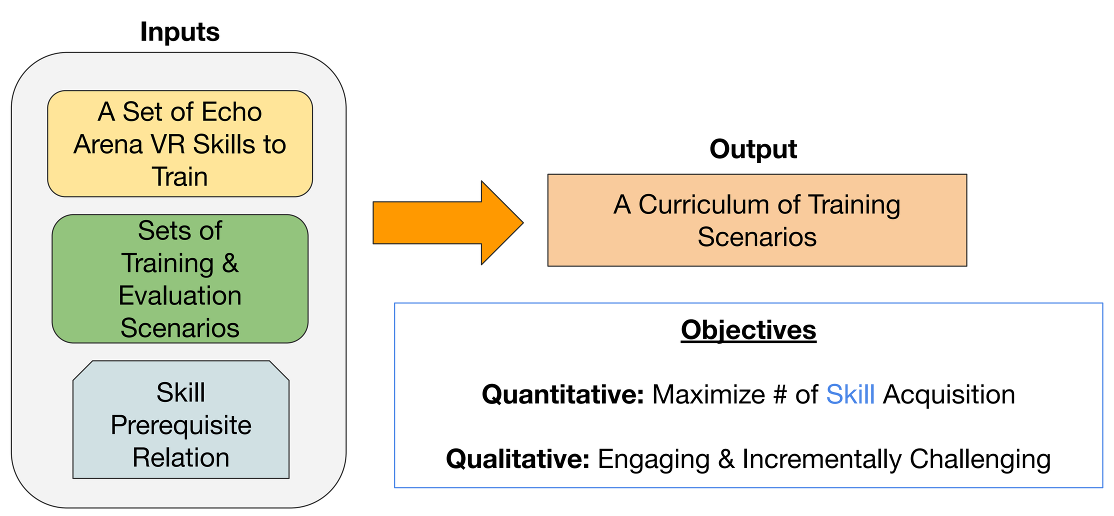
[1]Cerritos College, [2]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, *Contributed equally to the project

2022 Transfer-to-Excellence Research Experiences for Undergraduates Program (TTE REU Program)

Abstract

In classrooms, instructors often have difficulty teaching a course due to variations in students' prior knowledge and learning curves. Some students may get bored or some may fall behind. Consequently, instructors host office hours to personalize the education for students falling behind, but this is increasingly difficult as the student-to-instructor ratio increases. This motivates the need for an intelligent tutoring systems (ITS), an algorithm which personalizes a tutorial based on each student's performance. In this project, we aim to construct an ITS algorithm to help individuals quickly learn physical skills in virtual reality (VR).

Research Problem



Background

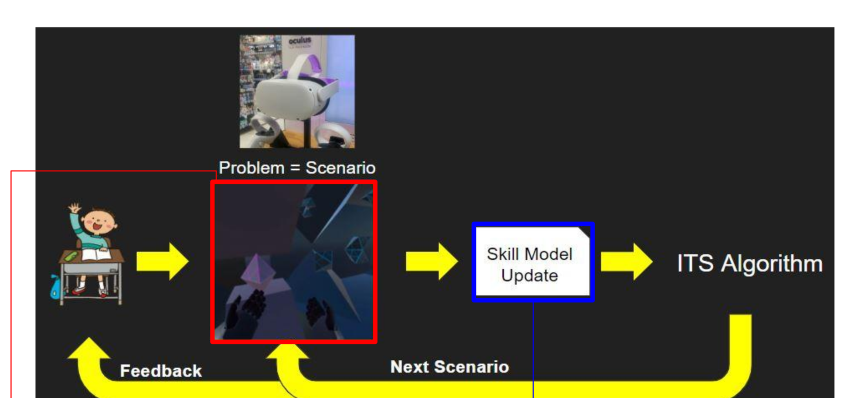


Fig. 1. Intelligent Tutoring Systems

Intelligent Tutoring Systems is a personalized tutorial algorithm that is able to track skill acquisition, giving feedback to an individual and calculating the probabilities of whether a skill is known or how close the trainee is to acquiring it.

```
h1andh3Region = MeshVolumeRegion(trimesh.creation.box((1, 1, 1)),
        dimensions = (2, 10, 6), position = (7, 0, 0))
h2Region = MeshVolumeRegion(trimesh.creation.box((1, 1, 1)),
        dimensions = (2, 2, 2), position = (14, 0, 0))

hoop1Point = OrientedPoint in h1andh3Region
hoop2Point = OrientedPoint in h2Region
hoop3Point = OrientedPoint in h1andh3Region

require (distance from hoop1Point to hoop3Point) > 5

h1 = Hoop at hoop1Point,
        facing 90 deg

h2 = Hoop at hoop2Point

h3 = Hoop at hoop3Point,
        facing 90 deg
```

Fig. 2. Code from scenario "BT_ST" that randomly distributes the three hoops within their instantiated regions

Scenic, a scenario modeling language, was created with the goal of modeling and generating distributions over static and dynamic condition over scenarios.

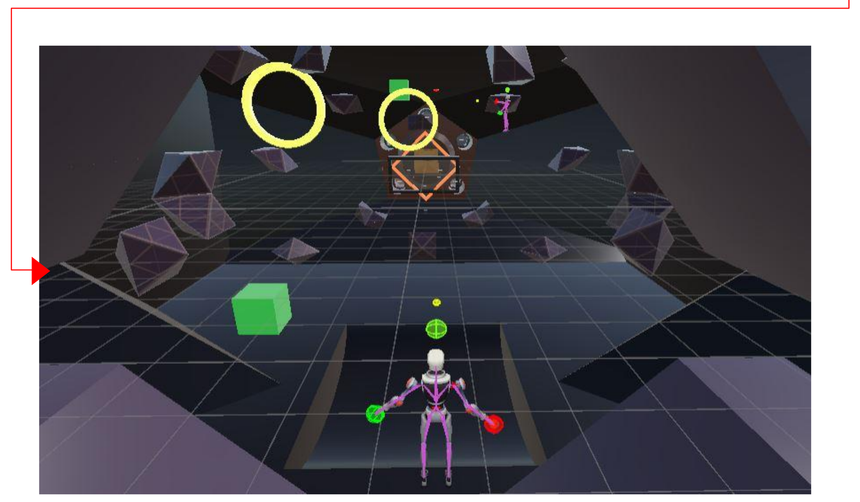


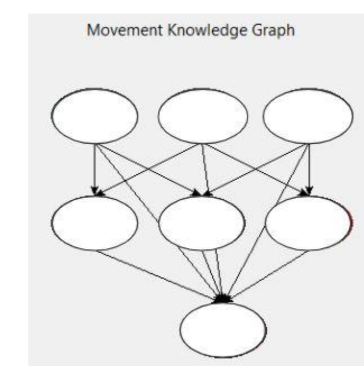
Fig. 3. Scenario "Static_Pass_BT_GR_ST" generated onto Unity

Knowledge Tracing is a bayesian model which determines when a trainee has mastered or learned a skill. Given a correct / incorrect response of a trainee with respect to a task in a training scenario, the model outputs the probability of a mastery of a skill.

Methodology

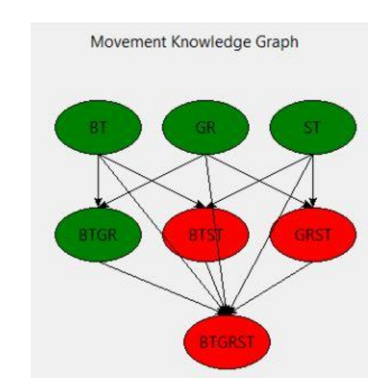
Step 1: Knowledge Representation

- A generated Knowledge Graph is built upon a set of skills and their prerequisite relation to train.
- Each node is a skill.
- The directed edges represent the prerequisite relations.



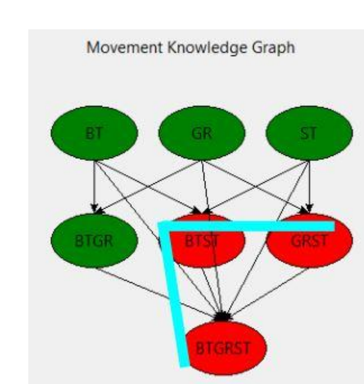
Step 2: Knowledge Identification

- Identify prior knowledge (i.e. fully color the knowledge graph).
- Green = Skill Learned
- Red = Skill Not Learned



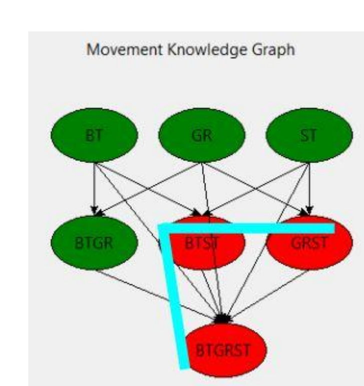
Step 3: Identifying Zone of Proximal Development (ZPD)

- ZPD is a concept from cognitive science, which defines a "boundary" of knowledge [1].
- The ZPD within the knowledge graph is the area of yet learned skills that are close to the skills which the trainee mastered.



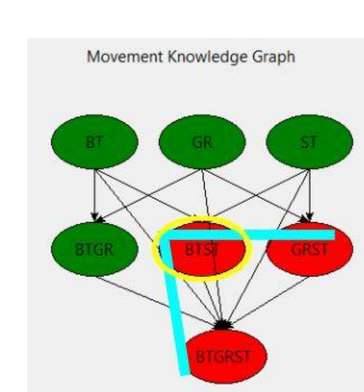
Step 4: Incremental Curriculum Generation

- Within the ZPD, the node with the least number of prerequisites will have its scenario run next.
- If there are two or more nodes with an equal amount of prerequisites, they will be chosen randomly.



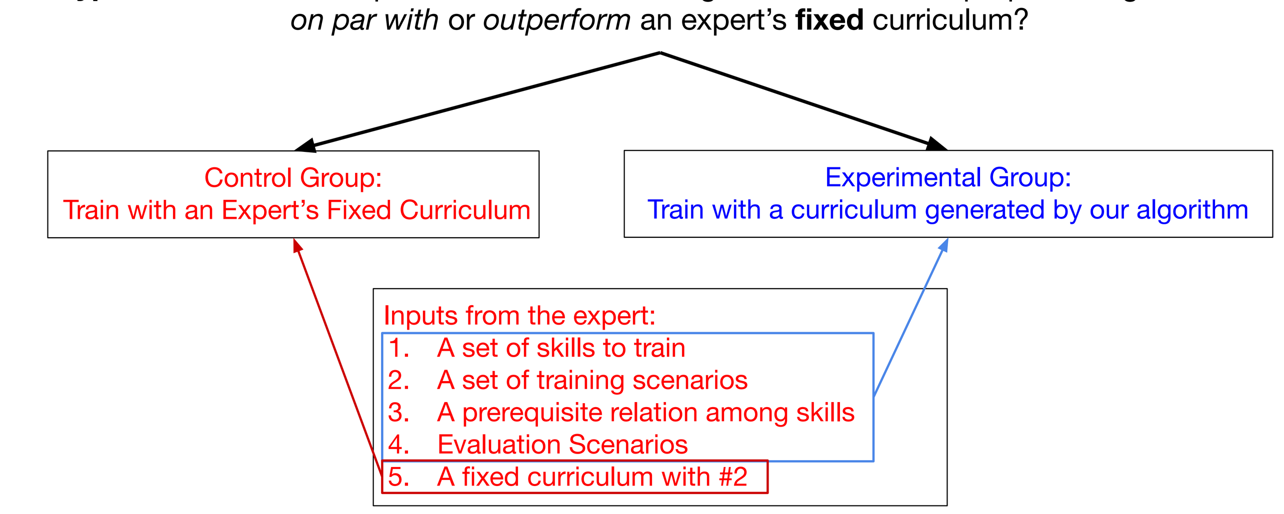
Step 5: Knowledge Tracing (KT) - Skill Acquisition Criterion

- KT models a mastery of a single skill.
- Each node has its own KT model.
- KT model determines when to change the color of a node (i.e. skill) from red to green.



Preliminary Evaluation

Hypothesis: Would the personalized curriculum generated from our proposed algorithm be *on par with* or *outperform* an expert's **fixed** curriculum?



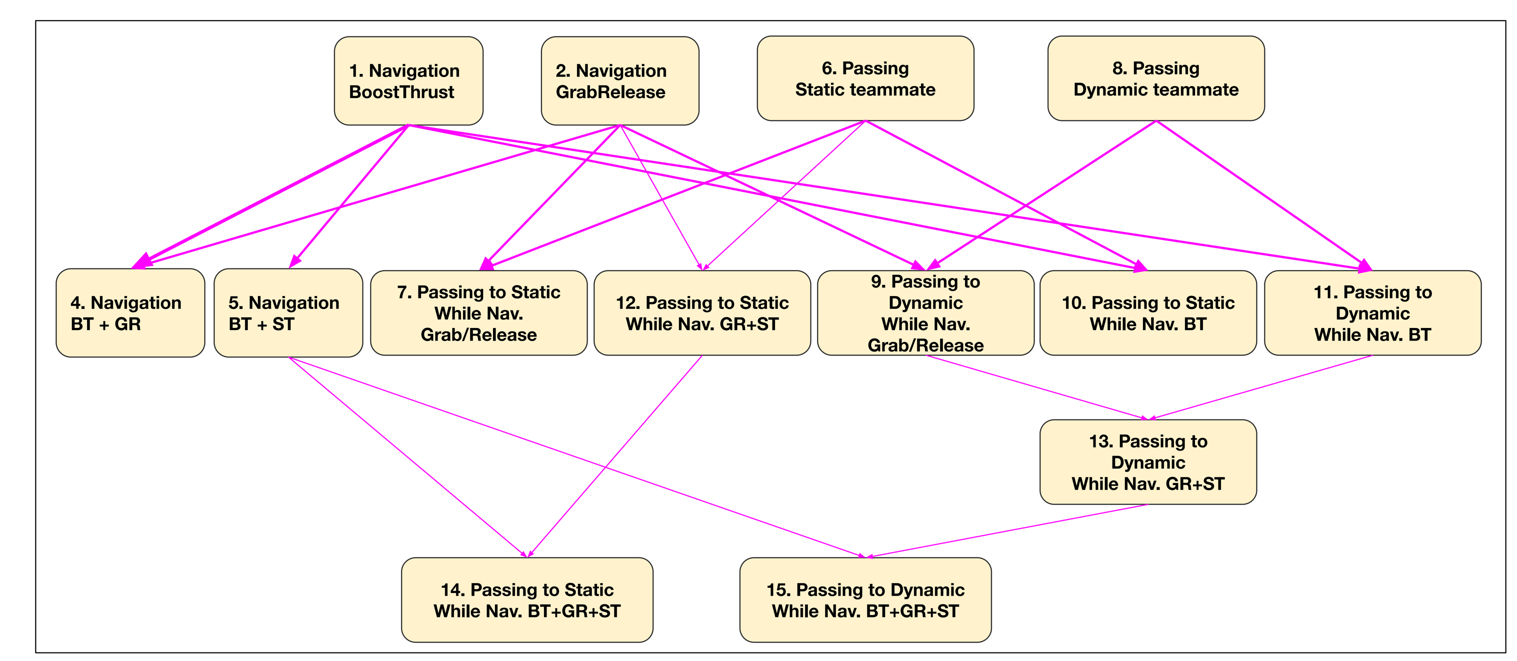


Fig. 4. Consulted with four experts of the Echo Arena VR game and created a Knowledge Graph on a set of 15 skills needed to train a novice.

Conclusion

We constructed an ITS algorithm that uses Bayesian Knowledge Tracing and a total of 15 scenarios created with Scenic to help individuals improve their skills for the Echo Arena VR game.

References

[1] R. Luckin, "Designing children's software to ensure productive interactivity through collaboration in the zone of proximal development (zpd)," *Information Technology in Childhood Education*, vol. 2001, pp. 57–85, August 2001.

Acknowledgements

I would like to thank my mentor Edward Kim and the team for the opportunity to work alongside them and helping me to catch on what they have been working on. I thank Professor Sanjit Seshia for allowing me to take part in the project under his supervision. I would also like to thank the Hopper-Dean Foundation for funding my expenses that allowed me to work on my research for the nine weeks I was here.

Contact Information
Isaac D. Gonzalez
Email: gonzaisaac5151@gmail.com
Phone: (562) 355-7149

Support Information
This work was funded by the Hopper-Dean Foundation.