Computational Thinking in Middle School: Design Thinking Outline

- Conversation about the typical delivery of CT across grades 6-8
- Open discussion of pain points in teaching CS in middle school
- Learning progressions: palette and block restrictions
    - Specific computational thinking concepts and sequences
    - Instructors can check and uncheck in design mode
    - Instructors can still use these
        - Instructors can put blocks in the game interface, but the students can't add any new ones.
        - Students can't do anything to the blocks already there
    - Left click exclude is allowed for simple ones: individual blocks
    - Quest: several games, increasing amount of freedom allowed as you're learning concepts
        - Different difficulty levels for different games
- Focusing on CT concepts via points
    - Emphasizing looping: the loop has a higher point value than doing it individually
        - Explain the concept of the points; preset configurations
    - Explain scoring system options for Parson's Puzzles and Constructionist Video Games
    - Parson's operates on a specific gaming rubric
        - Wrong spot decrements score
    - Constructionist: set your own goals for learning objectives, based on instructor preferences.
    - Record the number of points that students get in their games, translate into leaderboards.
    - Teacher gets to weight various ways that students can get incentivized.
    - Several ways of assessing mastery: actual mastery plugin with the Hairball analysis, and PECT analysis.
        - Pictures as a login as a student: spider graph is the mastery of results
        - Automated results from CT concepts
    - Deep dive on principals of Constructionist Video Games
- Gameful Direct Instruction: create a Parson's Puzzle
    - Create sprites and put them in the stage
    - Game templates under construction
        - Might be a calculation, maze, etc.
    - Enter question, enter hint
        - Find a number: have you tried a loop? Canned vs. dynamic hints.
    - Saving/loading Parson's Puzzles in Design and Play modes

    - Load from a student view:
        - Automatically take care of which palettes are available
        - Experiment from a selective Parson’s palette.
    - When you load as a student, the student doesn’t see any scripts in the scripts pane. The goal is to match what the teacher has created.
    - Discuss which elements of CT concepts teachers might want to deliver via Parson’s Puzzles
- Gameful Constructionism: game-objective editor and feedback system
    - Beginning quest: once upon a programmer
    - Capability to create game objectives using block-based software testing grammar: teacher task
    - Expect to have games pre-built and have templates and games customizable with content relevant for your school.
    - Make copies of your own games. Borrow games and objectives from the affinity space.
    - Blocks will have a drop own: don’t have to know underlying names of a block
    - Explore options for student view of feedback in real-time
    - Explore teacher capability, time, and willingness to engage in this activity
- Gameful Intelligent Tutoring: recommender and auto-hinting
    - Outer loop next-task guidance via Mission/Quest/Game recommendations
    - Ordered vs. unordered access to games in the quest
    - Guidance toward additional practice games when automated evaluation of mastery low.
    - Describe inner loop and opportunities for problem-solving support through individualized just-in-time and on-demand hinting
    - Explore teacher interest in throttling hinting at the class and individual levels
- Infusing CT across the curriculum
    - Demonstrate Mission/Quest/Game creation, and elicit requirements that would enable teachers from CS and other subjects to fuse learning goals
    - Discuss game-based learning that facilitates game-based assessment
    - Review CS/CT standards and means of incorporation, tracking, and reporting
    - Identify teacher wish-list for Scratch and CT in middle school