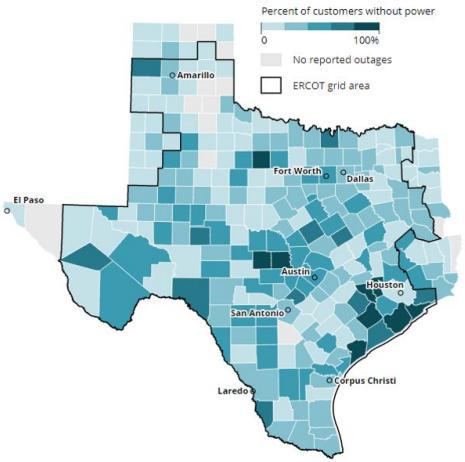
Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

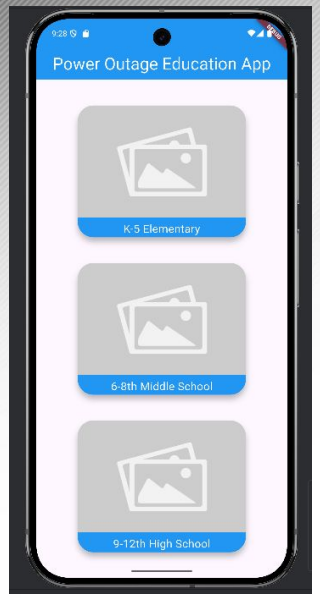
Team 28: Power Outage Education App
Bi-Weekly Update 3

Jackie Villanueva, Aidan Petropoulos, Joey Raphael
Sponsor: Dr. Mladen Kezunovic
TA: Swarnabha Roy

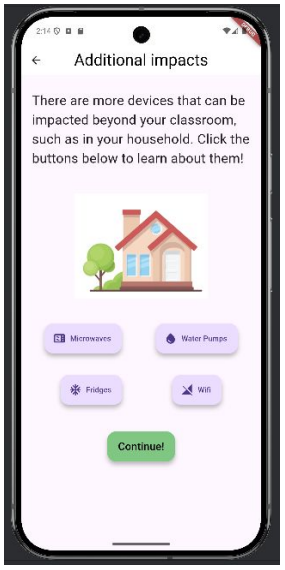
Project Summary

- Problem statement: “People are often uninformed about the best course of actions to take before and during a power outage.This lack of knowledge can result in lack of preparation and uninformed decision-making that can cause further harm.”
  - The 2021 Texas Freeze
  - Outages caused by natural disasters
- Solution: Develop an app that provides different age-specific knowledge to educate students about the impact of power outages utilizing engaging, interactive apps. By informing people about what measures to take before, during, and after a power outage, the harm caused by power outages can be mitigated.





Project/Subsystem Overview



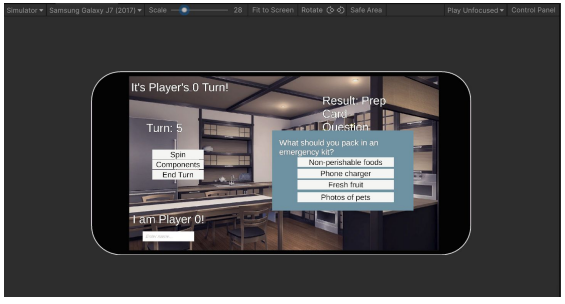
Power Outage Education App

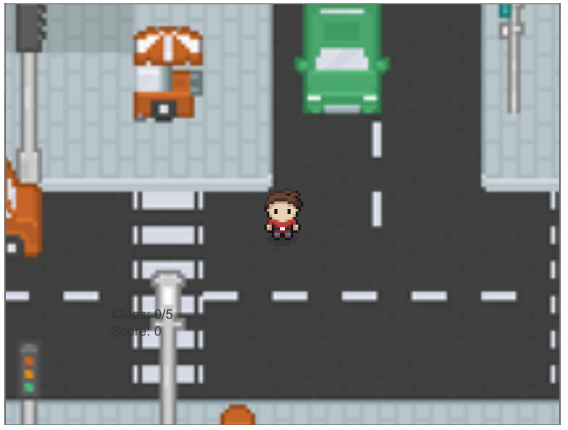
Homepage UI (Flutter | Team)

Power Outage Lesson (Flutter | Jackie)

Board Game (Unity | Joey)

LittleDoers Game (Unity | Aidan)





Project Timeline

Base Subsystem Functionality (completed 9/14)	Flutter/Unity Integration (to complete 9/28)	Refine Integration and Subsystems (to complete by 10/25)	Testing with Students (to complete by 11/15)	Finalize App Design (to complete by 11/29)	Demo (to complete by 12/4)

- Green: Completed
- Yellow: In Progress
- Red: Behind Schedule
- White: Not started

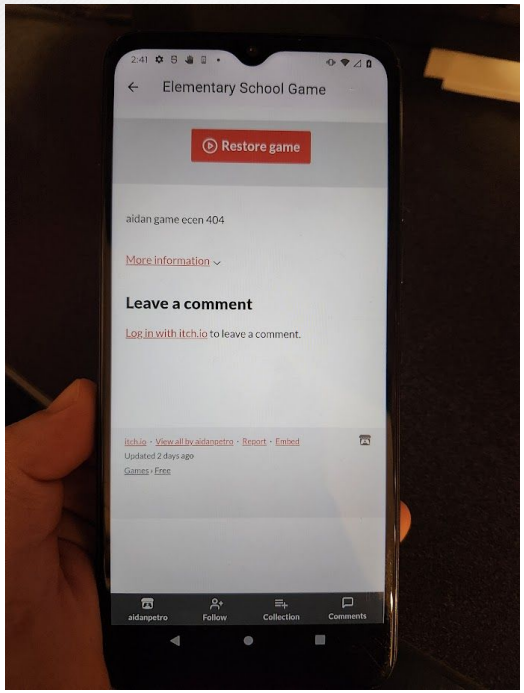
Elementary School

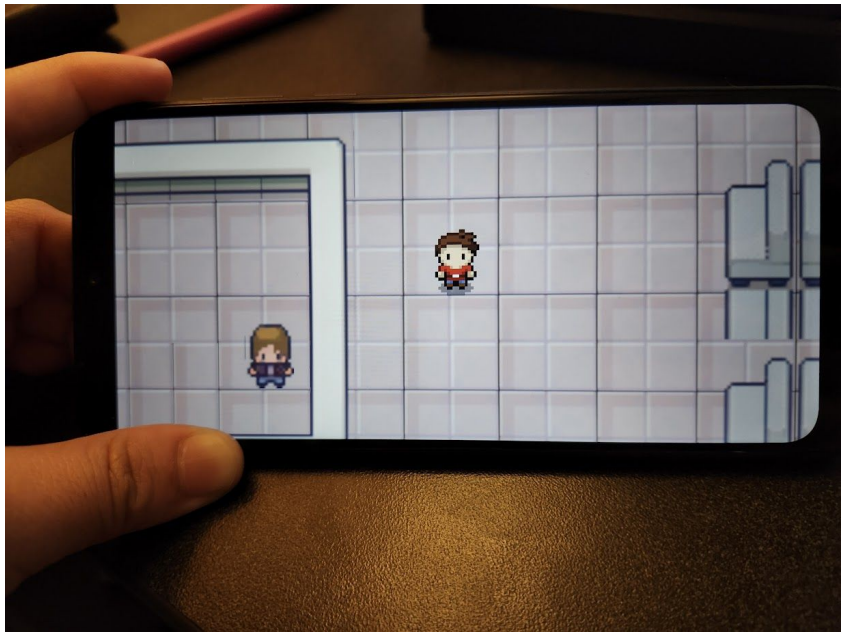
Aidan Petropoulos

Accomplishments since last update 18 hrs of effort	Ongoing progress/problems and plans until the next presentation
- Completed integration of Unity game into Flutter via webGL build function in Unity	- Touch screen not functioning when used via integration method - Functional testing with devices given from Doseum - On-going functional testing of app integration including UI and game performance

Elementary School

Aidan Petropoulos





Middle School Game

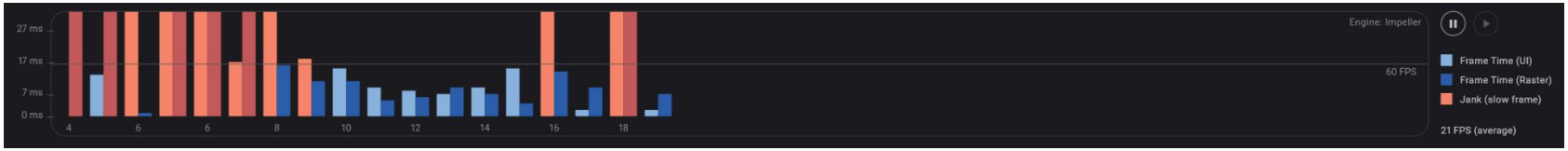
Joey Raphael

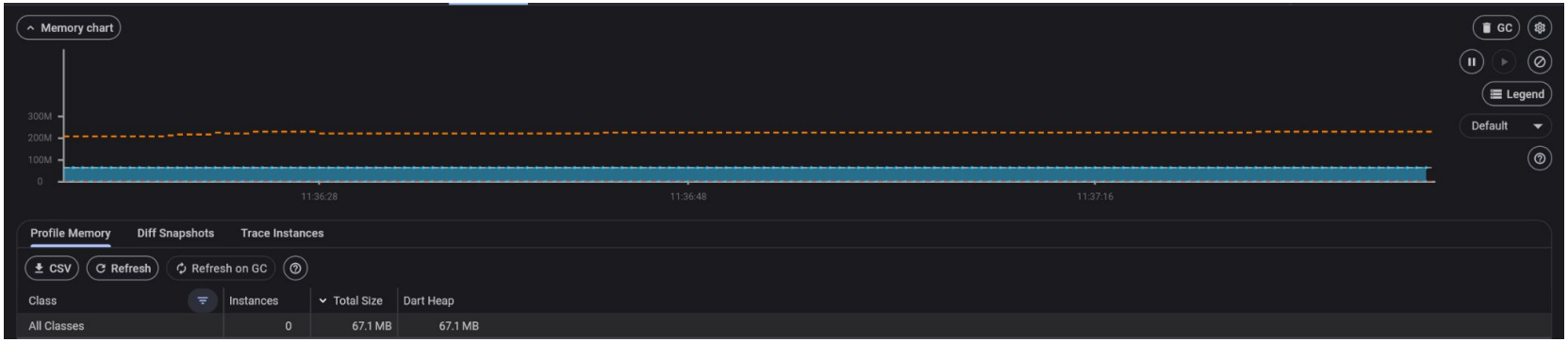
Accomplishments since last update 16 hrs of effort	Ongoing progress/problems and plans until the next presentation
- Integrated Unity game into Flutter app, refined UI of embeds - Verified UI scales properly for supported devices	- Test system resource usage - Resolve crashing from memory / system resource usage

Middle School Game

Joey Raphael

- Uses only on average 67 MBs of memory (target memory usage is < 300 MBs)
- Average FPS is 21 FPS (target is 30 FPS), average frame times exceed 3 ms
  - Exploring causes: unoptimized load times, unnecessary calls

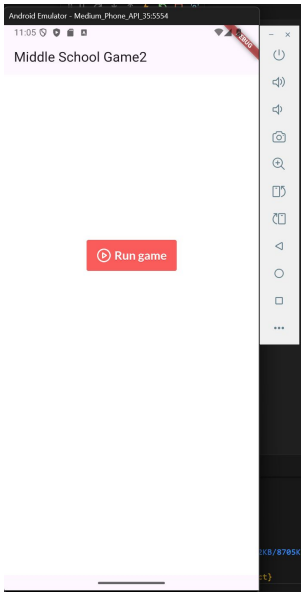


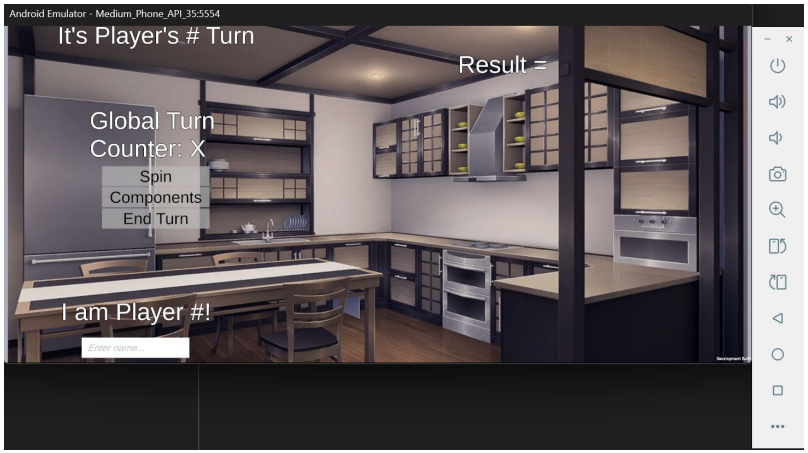


Middle School Game

Joey Raphael

- All buttons visible on 16:9 aspect ratio, 1080p resolution, 6.3-6.5 inch screen size
  - No trimming of text, buttons are not clipped by screen borders





High School App

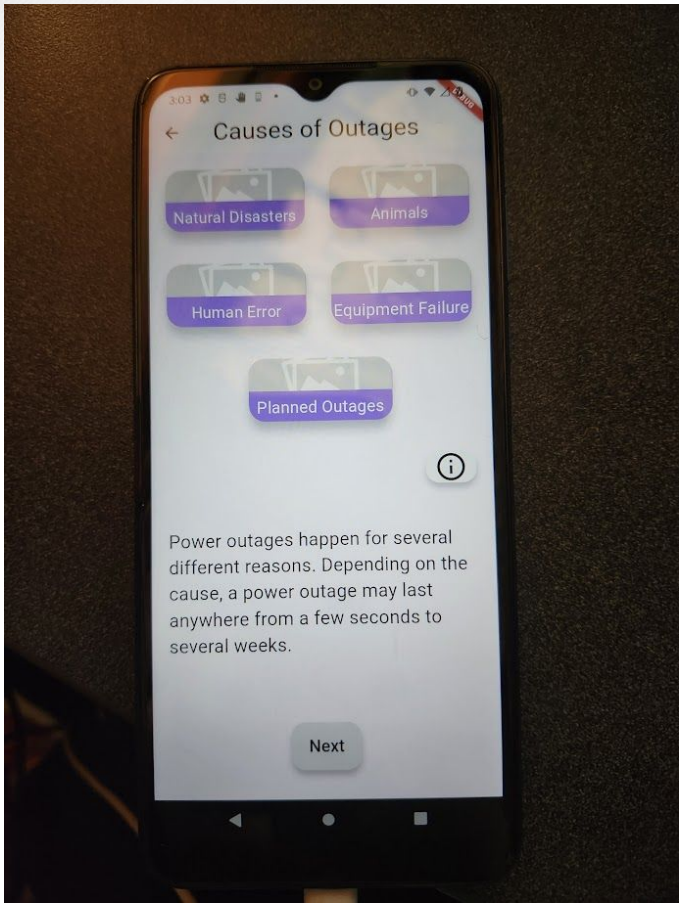
Jackie Villanueva

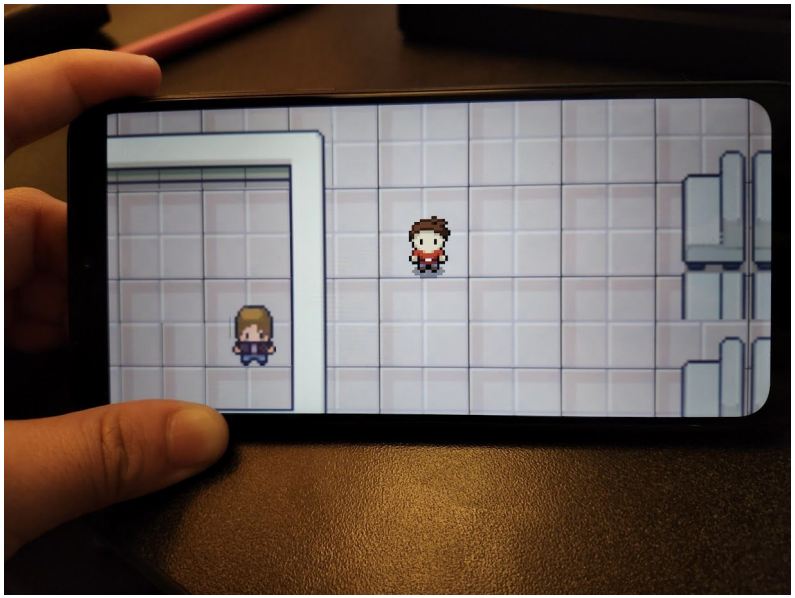
Accomplishments since last update 16 hrs of effort	Ongoing progress/problems and plans until the next presentation
- Integration of Unity subsystems onto Flutter subsystem - Updated Java version of Flutter app to 21 and Gradle to 8.5.0 - Simulated app following integration on phone and validated flutter subsystem functionality	- Continue planning for testing with students - Refine UI for the Flutter App - Remove placeholders - Add unique pre-quiz for students that have used the app before (Middle School)

- Flutter subsystem functions completely following integration
- Issues discovered with running the Middle School system but opens the Elementary subsystem

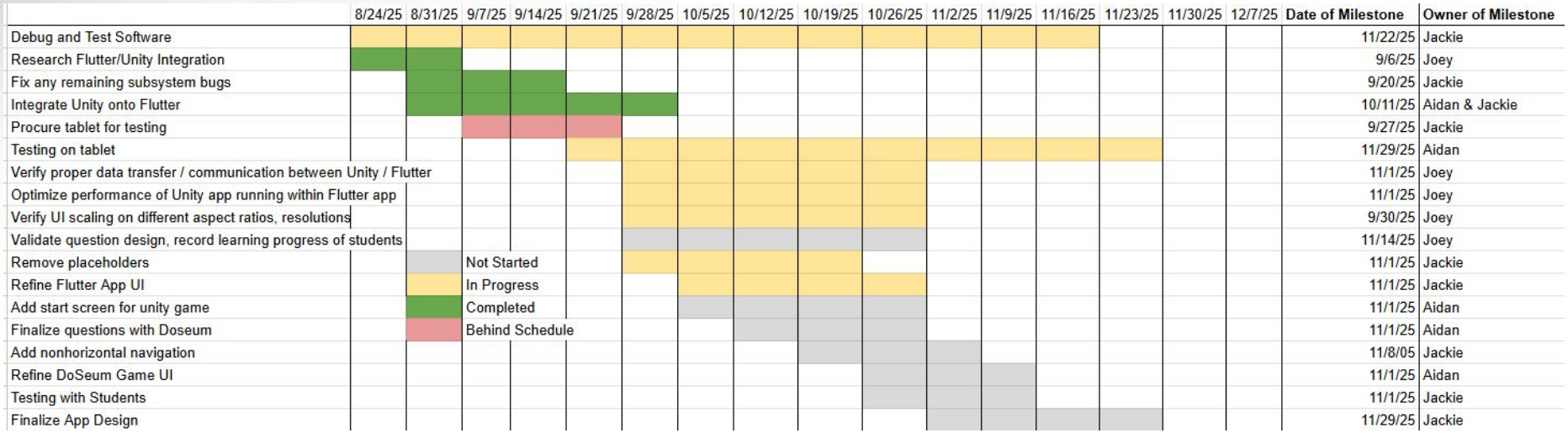
High School App

Jackie Villanueva





Execution Plan



	Date of Milestone	Owner of Milestone
Debug and Test Software	11/22/25	Jackie
Research Flutter/Unity Integration	9/6/25	Joey
Fix any remaining subsystem bugs	9/20/25	Jackie
Integrate Unity onto Flutter	10/11/25	Aidan & Jackie
Procure tablet for testing	9/27/25	Jackie
Testing on tablet	11/29/25	Aidan
Verify proper data transfer / communication between Unity / Flutter	11/1/25	Joey
Optimize performance of Unity app running within Flutter app	11/1/25	Joey
Verify UI scaling on different aspect ratios, resolutions	9/30/25	Joey
Validate question design, record learning progress of students	11/14/25	Joey
Remove placeholders	11/1/25	Jackie
Refine Flutter App UI	11/1/25	Jackie
Add start screen for unity game	11/1/25	Aidan
Finalize questions with Doseum	11/1/25	Aidan
Add nonhorizontal navigation	11/8/05	Jackie
Refine DoSeum Game UI	11/1/25	Aidan
Testing with Students	11/1/25	Jackie
Finalize App Design	11/29/25	Jackie

Validation Plan

Milestone #	Test	Detail	Validation Method	Data	Status	Owner
1	App Functionality Testing	App successfully opens and does not crash, lag, or glitch.	Boot up the app and navigate through all systems of the app.	- Touch screen controls do not work on Elementary subsystem - Middle school subsystem crashes on boot up.	Fail	Jackie Villanueva
2	Subsystem Objective Completion	Each subsystem can be played from start to end.	Run through each individual subsystem and reach the 'end.'	- Touch screen controls do not work on Elementary subsystem - Middle school subsystem crashes on boot up.	Fail	Aidan Petropoulos
3	User Interface Consistency	Check that all buttons, menus, and text show up correctly and work the same way on both phones and tablets	Run the app on an Android phone and tablet. Confirm the correct functionality of each button. - Account for edge cases in which buttons are not directly pressed, visually inspect the collision detection of UI elements with Flutter Layout Explorer provided in DevTools - Modify code to log into console how long it takes for function to run after associated button is pressed on app. Button press to app action should not take longer than 10 ms for a response.	N/A	Untested	Joey Raphael
4	Offline and Online Functionality	The app has different interfaces for if it is online or offline.	Run the software with and without wifi. Check if the interface correctly differs between the two scenarios.	N/A	Untested	Aidan Petropoulos
5	Performance (Middle School System)	The game runs at an acceptable performance - Less than 100 ms latency between touch input and button response - Average of 30 FPS, frametimes less than 200 ms	- Load integrated app into test phone / tablet - Monitor system resource usage using Flutter resource profiler - Plot RAM usage, CPU usage, frametimes, over time and label events such as app loading, game loading, game start, game end, etc. - Verify that usage of system resources does not exceed target device specs i.e game does not use more than 4 GBs of RAM, utilize more than 4 cores in standard 8-core CPU, average FPS is >30, average frametimes are <3ms	- Output graphs detailing frametime over frame, memory usage over time, CPU usage over time with key events labeled (game loading, game start, game end, app start, etc.) - Frametimes were <3ms, averaged 27 ms - Game embedded in Flutter app utilized 67 MBs of RAM - Average FPS is 21 FPS	Partial Success	Joey Raphael
6	Nonlinear progression (High School Subsystem)	The user can navigate back to preivous tabs that have already been accessed	Progress through the high school subsystem. Using the appbar on the top, confirm that previous 'tabs' may be accessed, then return to the most current tab.	N/A	Untested	Jackie Villanueva
7	Flutter and Unity	Test that the Unity WebGL game runs properly inside the Flutter app, responds to touch input, and displays the correct scenes, buttons, and animations without distortion or delay	Click Elementary and middle school tabs from Flutter base. Play through the individual games for both.	- Touch screen controls do not work on Elementary subsystem - Middle school subsystem crashes on boot up.	Fail	Joey Raphael
8	Learning Outcomes (Elementary)	Students demonstrate improved understanding of foundational concepts after playing.	Pre- and post-game quizzes, teacher feedback, observation.	N/A	Untested	Aidan Petropoulos
9	Learning Outcomes (Middle School)	Students demonstrate improved understanding of targeted concepts after playing.	Pre- and post-game quizzes, teacher feedback, observation. - Adminster quiz before students play game, quiz will consist of questions from the game, record what questions were answered correctly or incorrectly on a per-student basis. - Adminster quiz after students play game, quiz will consist of same questions, record what questions were answered correctly or incorrectly - Determine overall accuracy of questions, identify which questions were difficult, determine "learning" by assessing if students were able answer question correctly after getting question initially wrong in pre-game quiz - Output bar graph of questions answered correctly, determine percentage of students who had increased test scores, determine average test score increase	- Output accuracy data and test scores as .csv files - Process into bar graphs to compare testing accuracy among different questions, students, etc.	Untested	Joey Raphael
10	Learning Outcomes (High School)	Students demonstrate improved understanding of advanced concepts after advancing through the curriculum.	Post content quiz, teacher feedback, observation	N/A	Untested	Jackie Villanueva

Thank you for your time!

Any questions?