MAD MAN Engagement Tactics

Movement, Action, Discussions, Motivation, Awareness, Noise

By using these techniques, you'll get data on how students are doing. You're a scientist. Your hypothesis is that everyone gets it - now you need data to determine if you're right.

How the traditional instructor engages students and tests whether everyone gets it or not:

Instructor: "Does everyone understand?"
One student answers: "Yup!"
Instructor: "Cool. Let's move on."

The traditional instructor **has no idea whether the rest of the class got it or not.** The traditional instructor's data is awful.

You will be an elite instructor. You will engage students differently! You will be a MADMAN of engagement! You will have optimal data as to whether ALL your students get it or not. And you will calibrate pace based on that data.

Do not speak 15 minutes straight without using an engagement technique.

Vary up how you engage students. Don't use the same technique every time you engage.

If you'd like, you could watch these tactics used in a classroom with play by play explanations:
https://www.youtube.com/watch?v=VA1GJ4aTrWs&list=PLQHSrcR-RsBSvHNrPI8zIibewvURhpzRf&index=1

Engagement Tactics

1. **Movement (get students to move)**

   Note: if you are in a small room without space for students to move then movement will be limited to engagement where the students stay in their seats (contributed by Student Success Manager: Ilyssa Rosenzweig):

   - "Put both hands up if you think this will output to true."
   - "Make an x across your chest if you think the div will be green."
   - "Make a circle with your hands if you think this will yield an error."

a. Instructor/TAs command students who have finished an exercise to help students that have not completed the exercise.

   Literally ask the strong student to get up and hover over a group of students that have not completed an exercise.
b. Instructor directs students to stand up and move to parts of the room to explain code in bigger groups away from their desks
c. Instructor brings all students to the front of class and gets them to move to the left of the room or right of the room based on the output of the coding demonstration
    i. If (1==1){ alert('hi') }else { alert('bye') }
        1. If you think this will alert hi, move to the left of the room. If you think this will alert bye then move to the right of the room
d. Beach ball technique (contributed by Josh Madewell - Instructor at UT Austin)
    i. Toss the ball to a student to explain first chunk of code. If the student doesn't know it then he/she can pass it to someone else. The student passes it to someone else and that student explain the next chunk of code.

2. **Action (action that Instructors and TAs take)**
    a. Walk around & Scan Screens During Exercises
        i. During an exercise at the quarter point, **scan screens** - if a student asks you for help - just let them know that you have to quickly scan screens and that you'll be right back to help them.

           <u>If you don't see code on their screen then ask them to pull it up.</u>

           Target Weak Students: Look for students who don't get it based on the code on their screen. Identify who they are.

           Target Strong Students: Students that have finished the exercise. An easy way to find these students is to say: "raise your hand if you finished" - direct these students to help those that haven't finished.

    b. Make Announcements to the entire class during Exercises
        i. During coding exercises, instructors and TAs walk around **look at screens** and actively **make announcements** to get the majority of students to understand and finish the coding exercise:

           It'll be difficult to do this, because it's so easy to end up helping a SINGLE student. The exercise will go by in 15 minutes, and there will be potentially a dozen or so students that have issues that you and your TAs have not gotten to.

To combat this, you make announcements based on what you see on student screens. If you spot things wrong on someone's screen, you make announcements to the entire class.

This is how you can get the majority of the class to finish the exercise.

Examples of announcements:

1. "Guys, if your code is not colored in sublime text it means that you did not save it as a html file"
2. "Guys do you use colons inside of a constructor function?... Exactly! You use equal signs inside of a constructor function!"

c. Rapport among TAs and Instructor: TAs and Instructors can ask each other questions during demos and background lectures to drive points home.

Here's an example:

Jim (TA): "Sean (Instructor) what happens when you make a constructor function lower cased?"

Sean replies: "Good point Jim. Nothing happens. You could do that. But you won't be following the best practice. And I want everyone here following the best practices."

3. **Discussions (get students to explain code to each other)**

Discussions are the #1 technique instructors miss.

Example: "Everyone explain lines 1-5 on the board to your partners."

While students discuss, Instructors/TAs walk around and digest what the students are saying, and they **participate in the discussion**.

If Instructors/TAs hear misconceptions, they make announcements to the class.

If you see students not discussing - then command them to do so: "hey can you move closer to them and discuss the code on the board?"

After the discussion, answer questions students had to the entire class and call on students to explain the code on the screen to the group.

When students are watching, they're not thinking. They're not digesting the material correctly. To combat this, you create discussions.

4. **Motivation (pump your students up)**
    a. Increase your voice during key points + use hand gestures
    b. Increase Morale by getting students to clap for one another
        i. Get students to clap for one another after someone answers a question correct. This will increase morale.
    c. Compliment students individually.
        i. "The way you structured your object was superb. Your functions are short and meaningful. Love your code!"
    d. Compliment groups during project week.
        i. "Your API got shut down. You were down a developer. But that didn't stop you. You guys delivered a product just in time for your presentation. You guys killed it. This is what being a developer on a team is all about. Getting through adversities to deliver core features no matter what."
    e. Compliment your class
        i. "We just finished our first week of JavaScript, and it was difficult. You had a really tough hangman and psychic game assignment. Some of you prevailed, some of you dealt with bugs that you couldn't debug in time. However, all of you wrote JavaScript. All of you used logic, came up with pseudocode, struggled and had little victories throughout the process. I promise you that all of you have grown from this assignment. I promise that by the end of this course, if you keep working hard, and you keep your head up high, you will be able to complete this assignment in 6 hours or less in one sitting. It's happened before and it will happen again!"
    f. Fight Speech when they want to give up
        i. "There is no way in hell that I'm going to let any one of you give up. If you think that you should give up - then you are wrong. We reverse engineer our homeworks into lesson plans and exercises... Redo the exercises. Redo them and you'll be able to do the homework. Use your exercises as references during homeworks. Quitting is not an option. You must code. You will prevail."

5. **Awareness (make sure you don't hear from the same students all the time + get increase your students' interest)**
    a. Asking Questions to your class

        Most instructors ask questions to the entire class: "How do we add a class to this element?" Then the same few strong students answer. Most instructors only hear from the students that get it.

        Don't talk to the class as if they are one person!

Because, you'll never ever hear from the students that kind of get it or don't get it.

You want to hear from the students that kind of get it and don't get it. Who cares about hearing an answer from someone that knows the answer?

Instead of talking to the class as if they are one person, cold call a row of students or a particular student. This will show you if an entire group of students don't understand a concept.

i. Cold call a single student
    1. "Eric, what does the code on line 32 do?"
ii. Cold call two students
    1. "Mary or Elise, why is the code on line 34 wrong?"
iii. Cold call a group of students
    1. "Vincent's row - what does line 32 do?"

b. Use your students in your examples. By adding them into the examples, students learn about their classmates, pay attention and have fun!
    i. Instead of something boring like foo or hello world, ask: "Jennifer what did you do over the weekend"
        1. Then you would update the example code to be var events = ["Jen watched all of breaking bad over the weekend"]
        2. You would then ask another student the same question and add to that array.
    ii. "Brian what do you do for fun?"

        Then you modify the code to be this:

        <h1>What we do for fun</h1>
        Brian catches pokemon for fun

c. Say and do interesting things to help students remember/understand
    i. Use mnemonics to help students remember things (contributed by Nicole Thurnua (UCLA TA) and)
        1. Mkdir : to help students remember this - say "Mortal Kombat Drrrrrrr"
        2. MySQL : "whose SQL is this? It's mine! It's not <insert student's name here>'s SQL it's my sql."

6. Noise (get your students to increase their noise)

a. See what percentage of the class understands something and who particularly doesn't
    i. "If you think this will output to true, stomp your feet."

ii. “If you think this will break, clap your hands.”

If you notice that very few of your students understand something, go into it for a bit until you think you’ve cleared the misconception

b. Increase Morale by getting students to clap for one another
   i. Get students to clap for one another after someone answers a question correct. This will increase morale.

Dealing with students who are feeling down/want to quit:

MOPE
Motivate, **Analyze**, **Plan**, **Empathize**.

Motivate: Motivate the hell out of them. Give them praise. Anything that they have done semi-ok/great, compliment them on it.

Observe: Are they coming to office hours? Do they code in a quiet place? How many hours are they putting aside? Are they redoing past homeworks & activities? Are they slow typers? etc.

Plan: Create a step by step plan for them and follow up.

Empathize: I’ve been in your shoes… Give specific examples.

Here’s a plan that was created by an instructor for a student that was really struggling by the end of week 5 (timers)

take notes on things that you don't understand or have trouble with while going through this plan - and be ready to discuss this with your tutor.

And come in for office hours

sun dec 5 -

3pm - 5pm -

slack pavan about wanting tutoring for friday dec 10th (7:30 to 10:00pm around)

4.1

jQuery drinklist

OnClick Basic

Sandwich click

trigger random

...

6pm - 10pm

watch tic tac toe video that pavan made
take notes

look for these files in the class content repo:
tic-tac-toe.html
tic-tac-toe.md

and then

do it on your own

11pm ...

get ready for the ajax lesson on monday

https://www.freecodecamp.com/challenges/trigger-click-events-with-jquery

tue dec 7

7:30pm - 9:30

1. I would set up the skeleton for week 5 hw and push it up to github and heroku AND read the prompt before you do it

2.

4.2 activities

10 - 12

continue with 4.2 or go to 4.3

thur 9

7:30pm - 9:30

1. 4.3 activities

10 - 12

read through the code for the stop watch activity - DON'T look through time conversion function and relate it to what i happening - don't do it

now do the same thing for the slide deck activity

...

add onto your week 5 hw

git push origin master
git push heroku master

fri 10th

7:30pm - 9:30

hw 5

10 - 12

hw 5

sat 11th - hw 5 due - quiz game

Outside resources on these subjects:

Movement
Action
Discussions