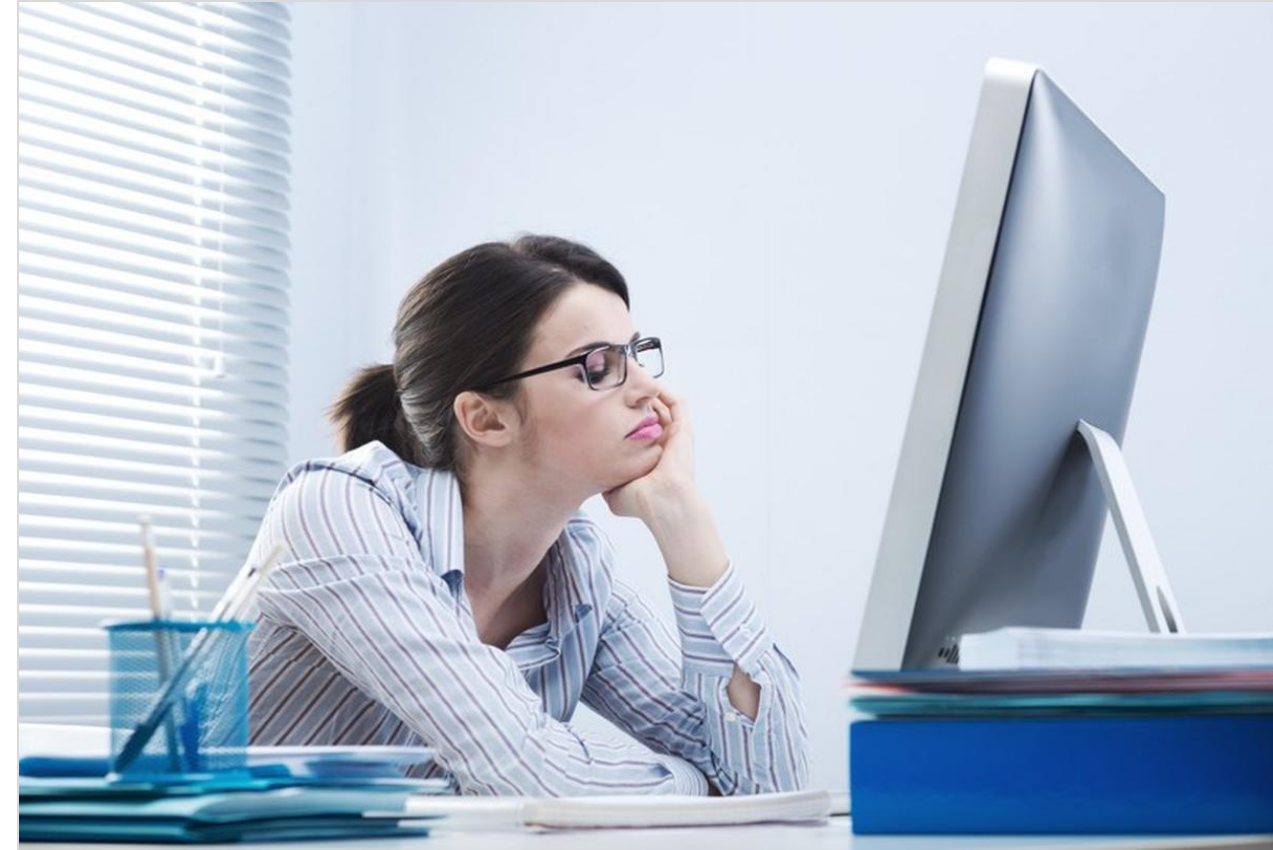


## Students are Bored When Learning About Software Testing

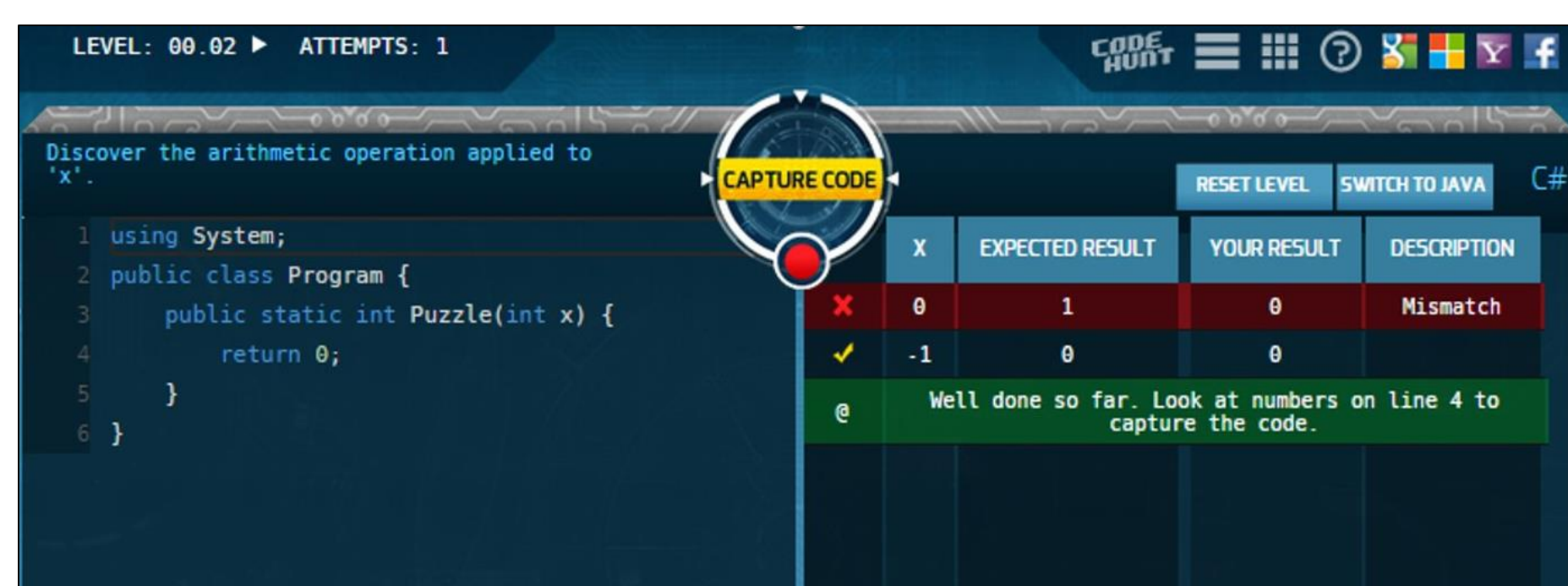
- In introductory CS courses, many students find learning about programming concepts, such as testing to be difficult and uninteresting



- How can we make the process more motivating and relevant for learners?

## Hypothesis: Gamification Can Increase Motivation when Learning about Software Testing Principles

- Past studies have used gamification to improve motivation, engagement, and enjoyment levels
  - CodeHunt is a game where beginners can learn to find bugs in their code



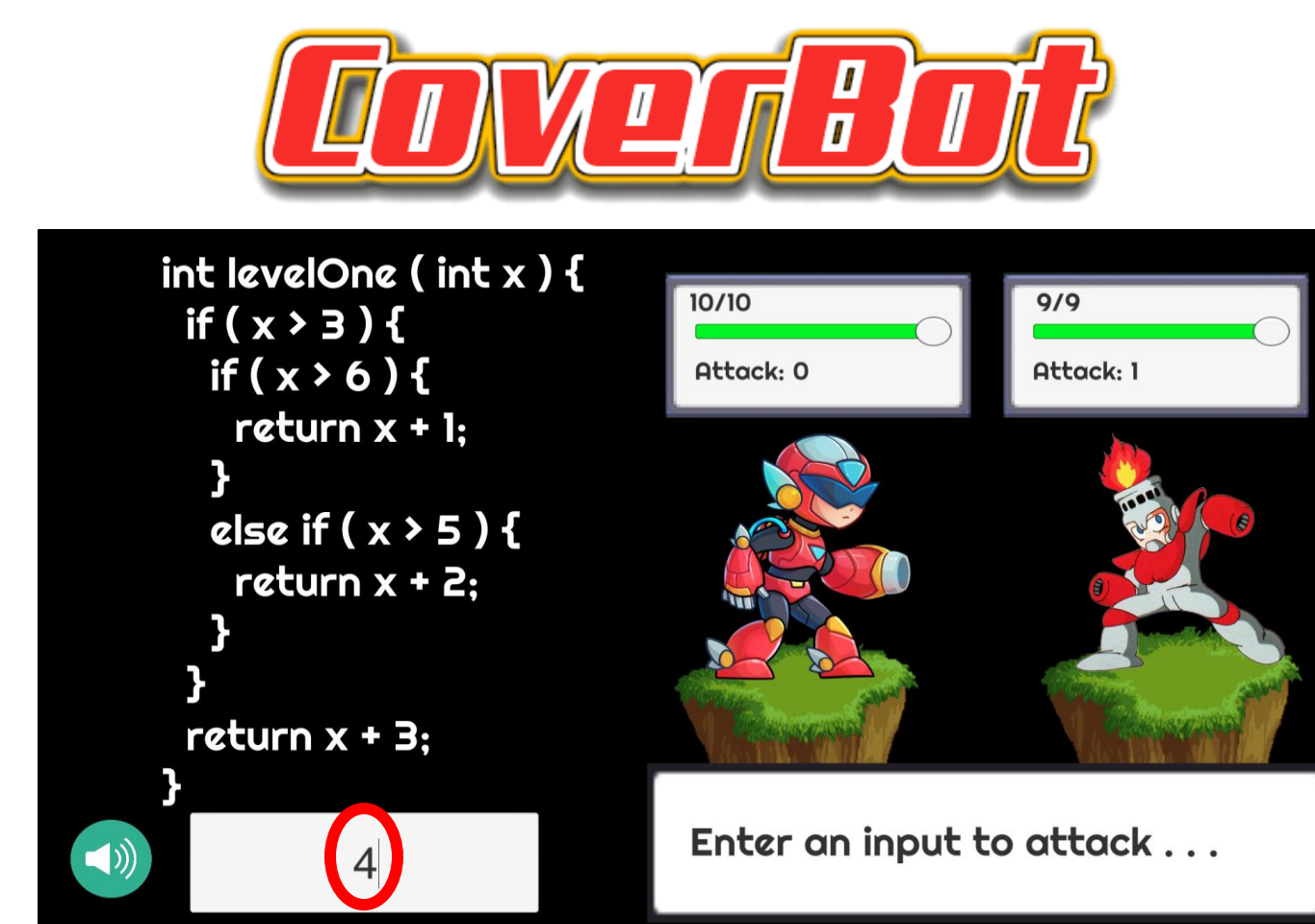
- We want to take this process and apply it to learning about specific software testing principles

- We focus on statement coverage:
  - Execution of every line in a source of code

```
1 Prints (int a, int b) {
2   int result = a+ b;
3   If (result> 0)
4     Print ("Positive", result)
5 Else
6   Print ("Negative", result)
7 }
```

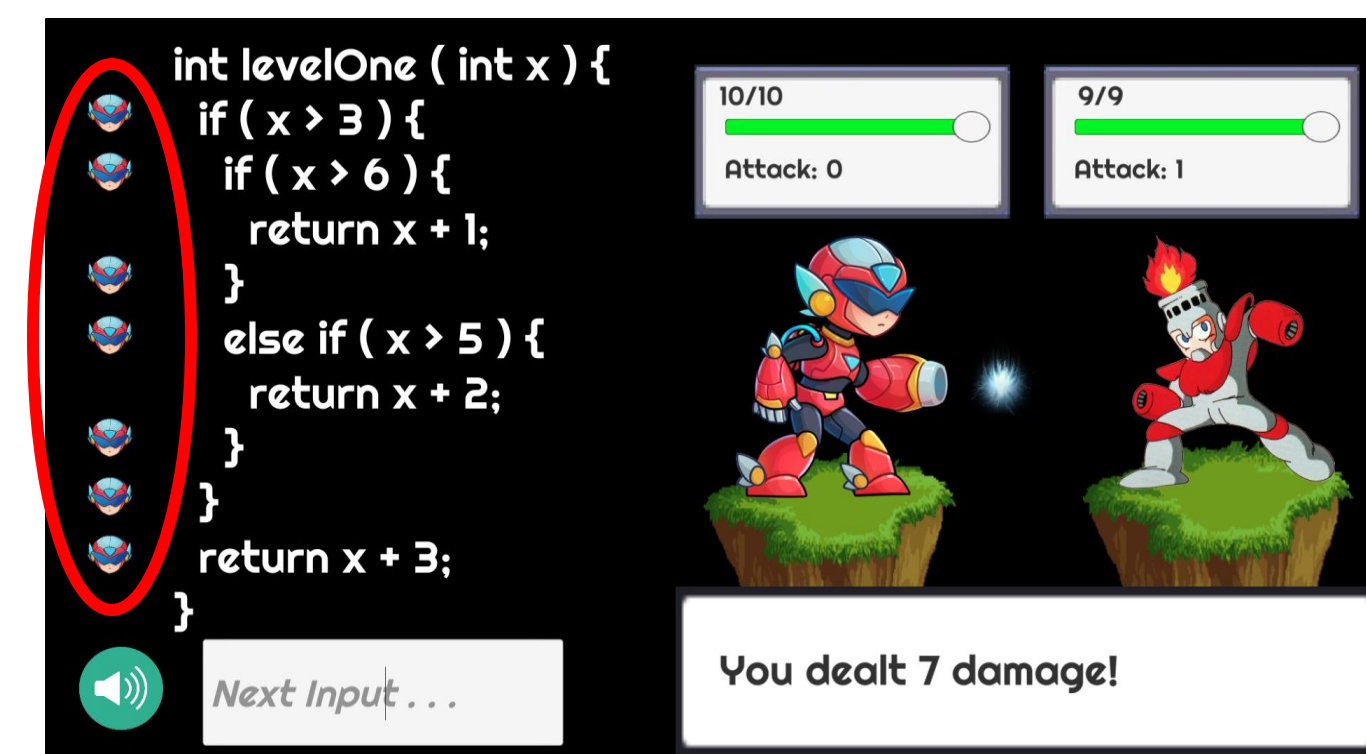
## Solution: A Game To Make Learning Testing Fun

- Gamification is the application of game-design elements and principles in a non-game context. Some examples of gamification can include:
  - Leveling progression
  - Graphics/Sound Effects

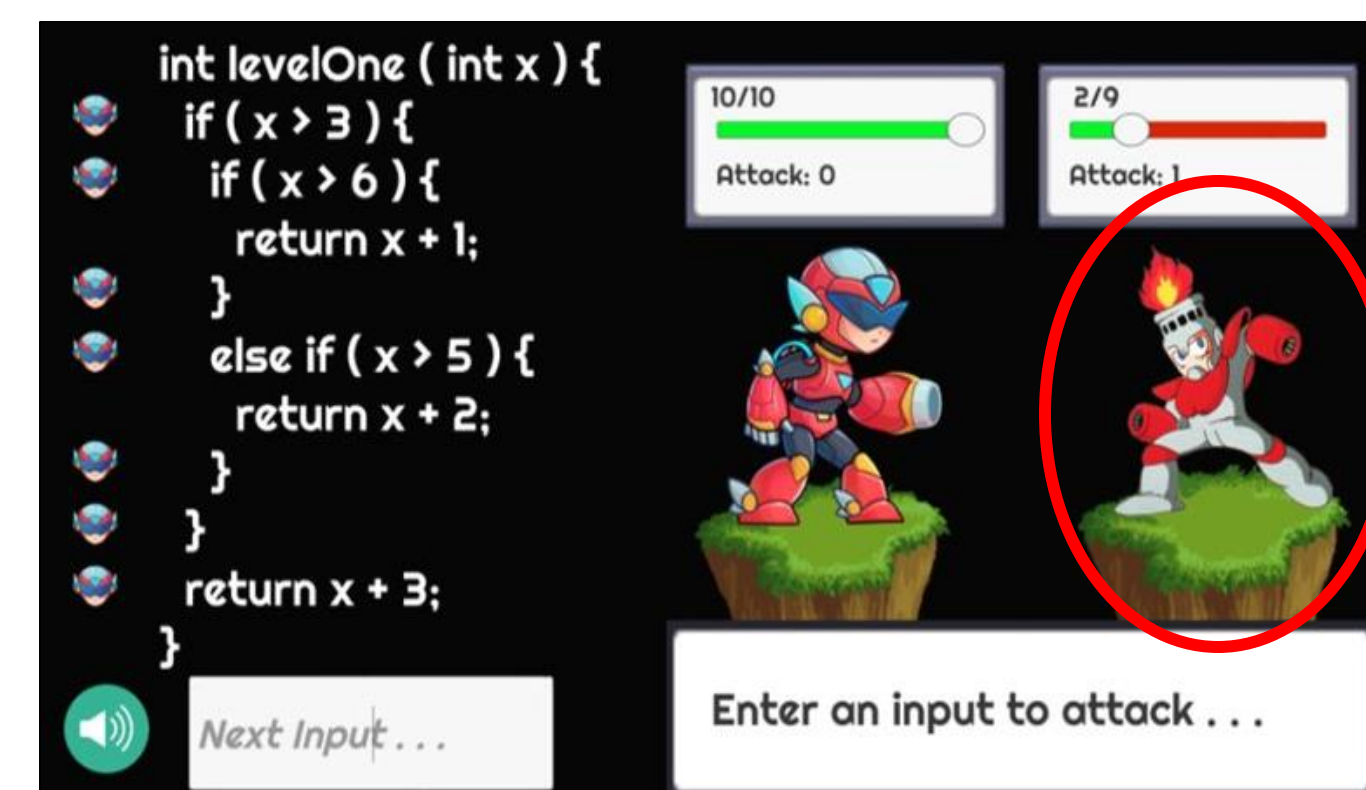


### Gamified Version

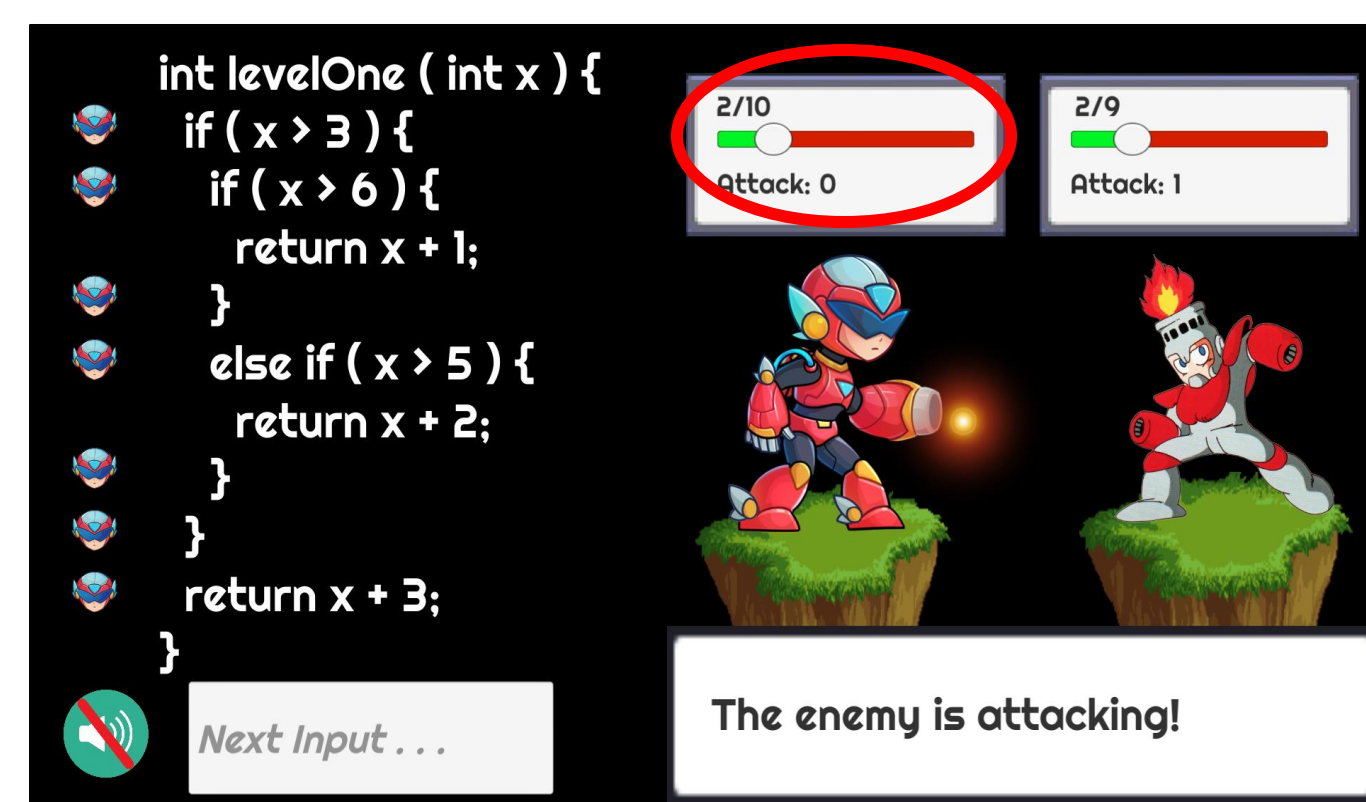
1. Analyze code and enter input



2. Wait and see which lines were executed



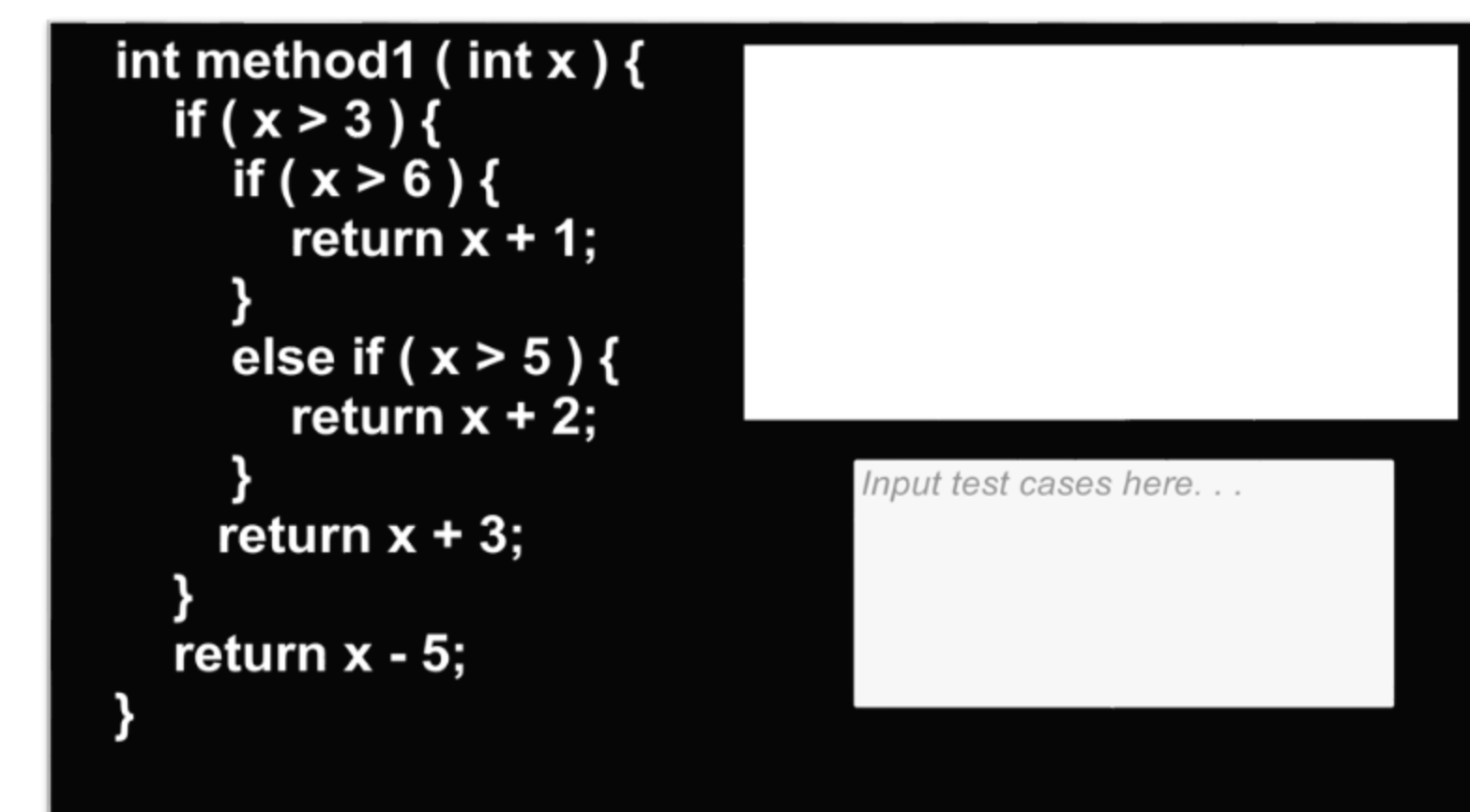
3. Keep going and defeat the enemy!



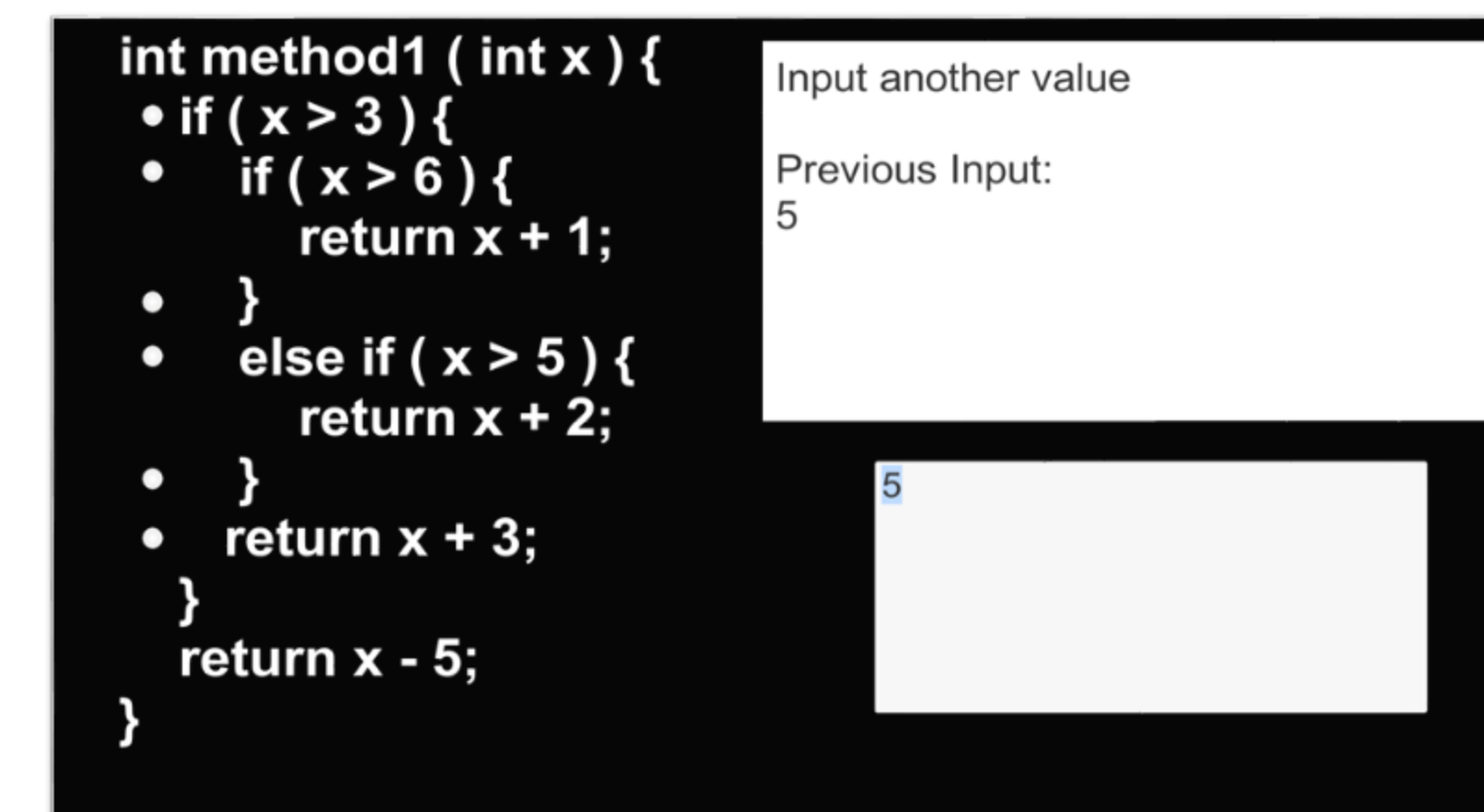
4. If no new lines are executed, you take damage!

### Non-Gamified Version (Experimental Control)

*Represents learning about statement coverage in a traditional classroom setting*



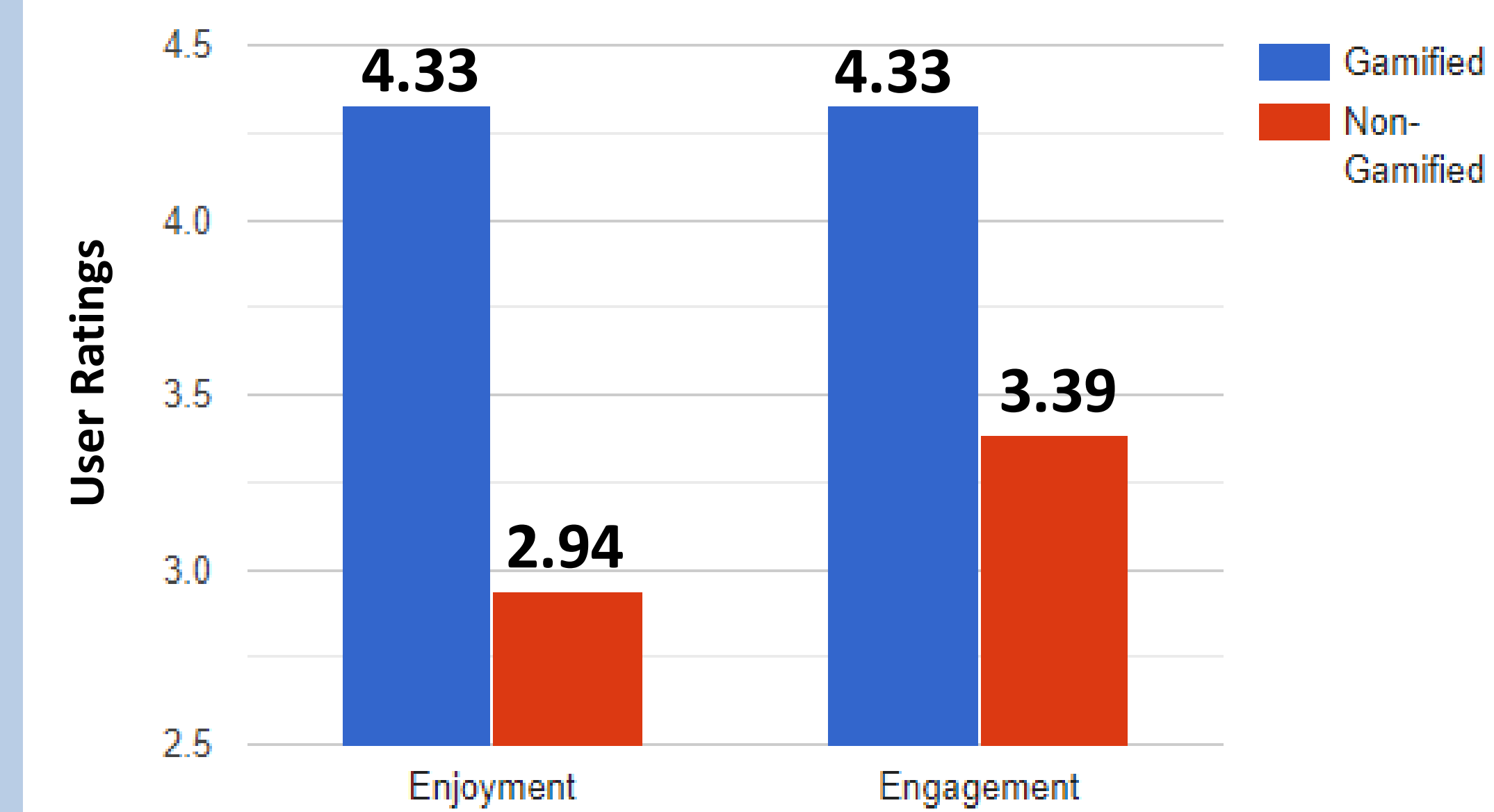
1. Analyze code and enter input



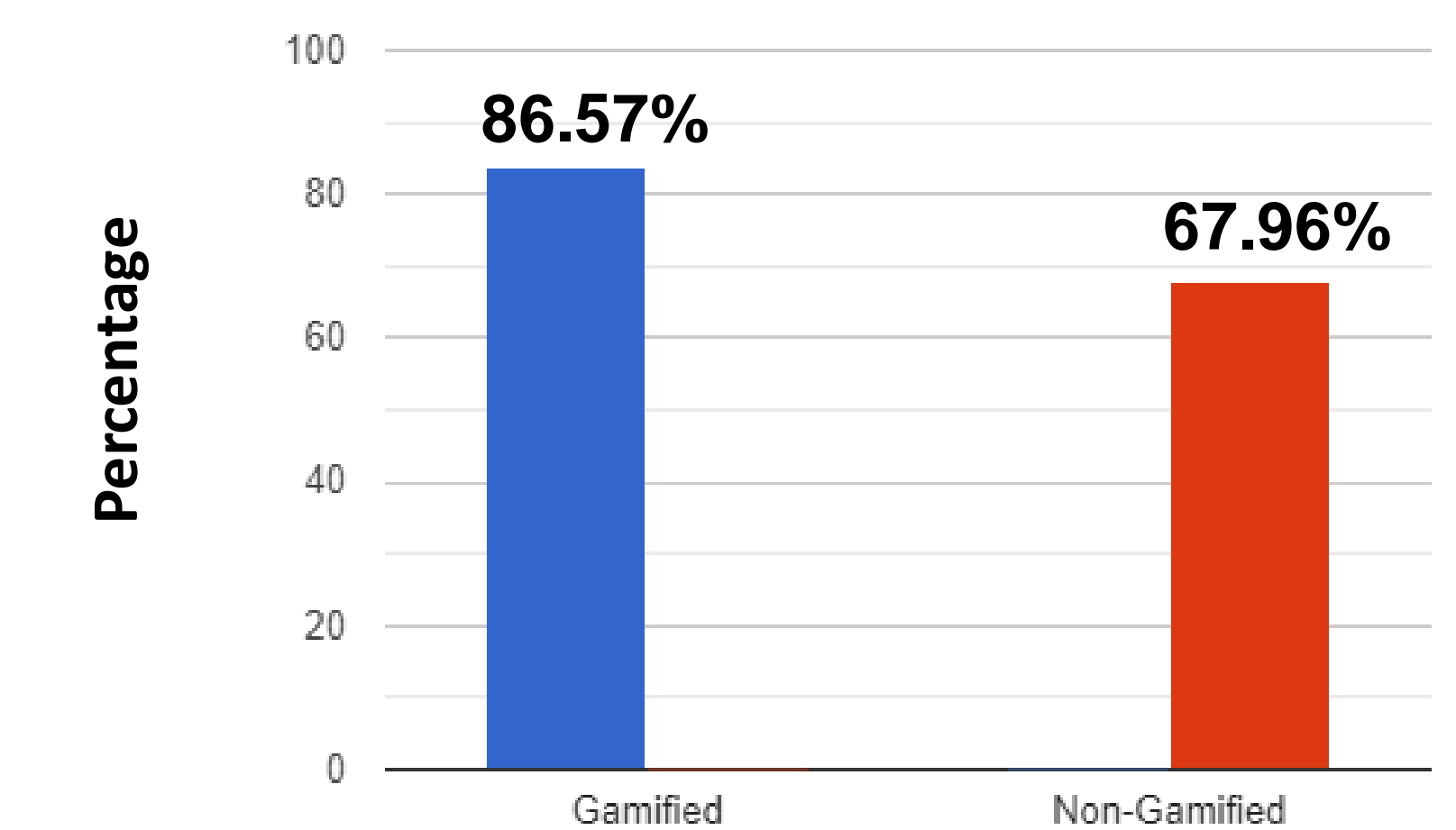
2. Input numbers till all lines are executed

## Users Preferred the Gamified Version

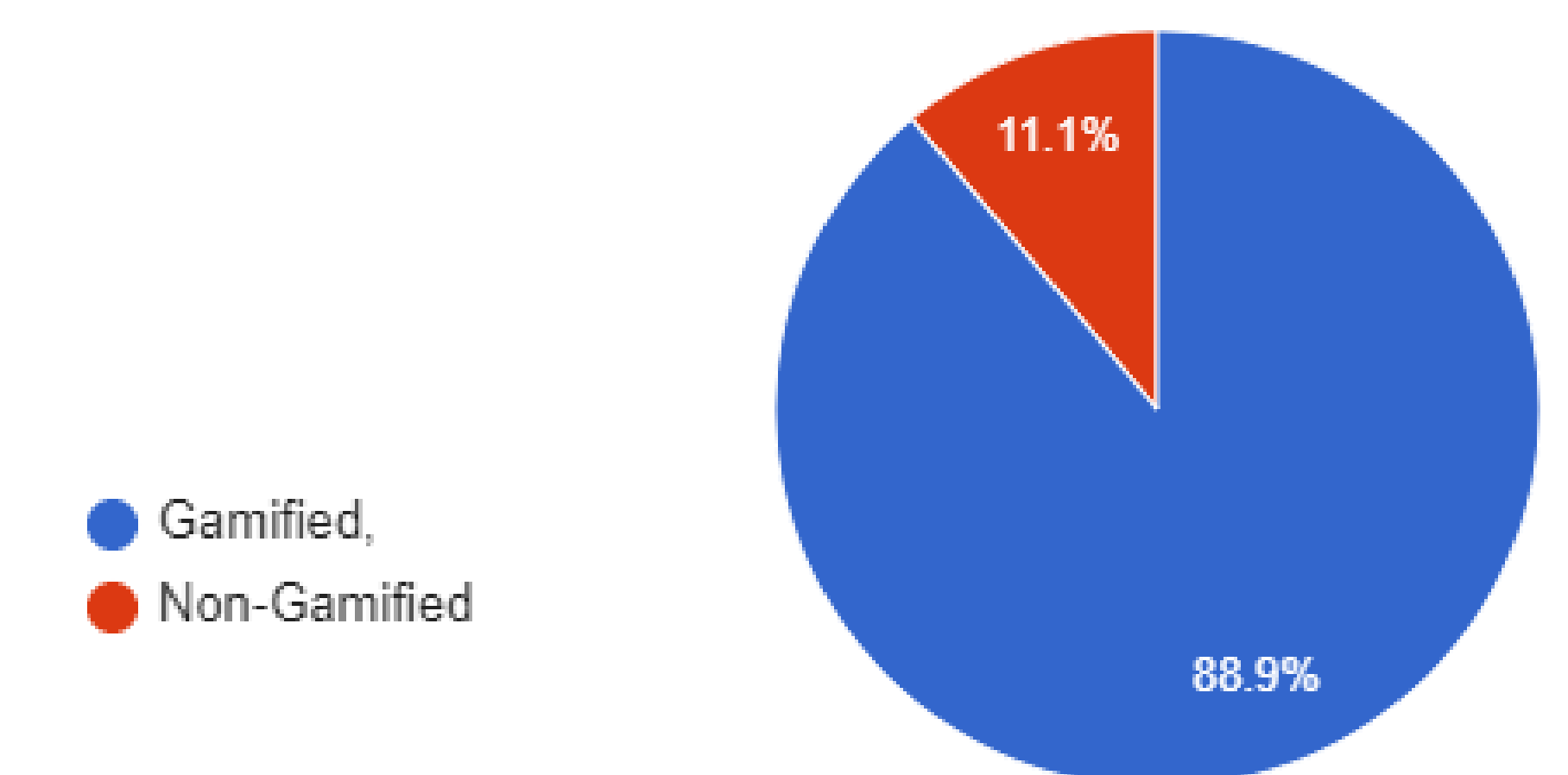
### Ratings of Enjoyment and Engagement



### Percentage of Successful Attempts



### Preferred Version



## Key Takeaways

- Methods of gamification are effective for helping students learn about software testing, specifically code coverage.
- We have contributed another methodology for learning about code coverage which heavily relied on 3 key gamification features:
  - Real-time feedback
  - Level progression
  - Animations

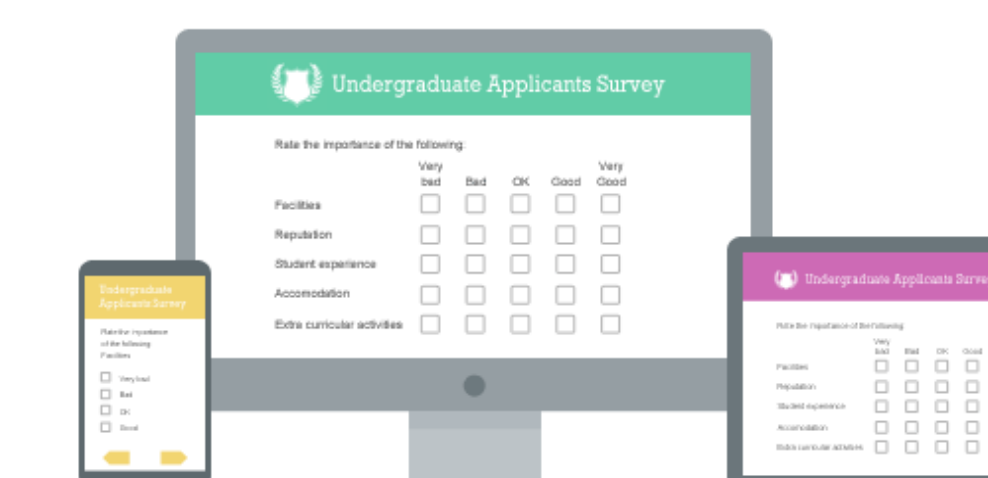
## User Studies: Within-Subjects Design

### Set-up:

- Two versions: gamified and non-gamified.
  - Each with four statement coverage tasks.
- Experiment conducted by a within-subjects design
  - Balancing the order between twenty subjects

### Surveying the Participants:

- Fill out a survey after each version regarding:
  - Enjoyment
  - Engagement
  - Quality of Learning



## Acknowledgements

We'd like to thank our research advisors William G. Griswold, and Sorin Lerner for their constant guidance and support. We'd also like to thank the Early Research Scholars' Program, Christine Alvarado, and Vignesh Gokul for providing us with this opportunity.

## References

"January Code Hunt Challenge." *TECHCOMMUNITY.MICROSOFT.COM*, 9 Jan. 2019, techcommunity.microsoft.com/t5/student-developer-blog/january-code-hunt-challenge/ba-p/311369.  
Nikolai Tillmann, Jonathan de Halleux, Tao Xie, and Judith Bishop. 2014. Code hunt: gamifying teaching and learning of computer science at scale. In *Proceedings of the first ACM conference on Learning @ scale conference (L@S '14)*. Association for Computing Machinery, New York, NY, USA, 221–222. DOI:https://doi.org/10.1145/2556325.2567870