CSE 291 I: Usability of Programming Languages ("Programmers Are People Too")

Michael Coblenz

Homework

- Dylan on homework 1
- Homework 2
  - Rust error message challenges
  - Bank: use Coq instead of COBOL?
    - Reliability
    - Recruiting

USABILITY STUDY OVERVIEW

- Running usability studies requires:
  - Ethics approval
  - Recruiting
  - Training
  - Task design
  - Data collection/analysis

ETHICS REVIEW

- For research: need to submit proposal to Institutional Review Board (IRB)
- For class: no need to get IRB approval (IRB only supervises *research*)

ETHICS

- What if incentive is too high?
- What if incentive is too low?
  - IRB reviews incentives
- What if recruitment is misleading?
  - IRB reviews recruitment materials

PARTICIPANT PRE-SCREENING

- Can issue a pre-test to avoid wasting time on unqualified participants.
- Issues:
  - How will you incentivize people to take the test?
  - Can you use the test results in your research?

Which of the following might be a valid Java constructor invocation?

malloc(sizeof(Square))

Square.new(5)

square(5)

new Square(5)

In Java, *encapsulation* refers to:

Preventing clients from improperly depending on

Serializing data correctly so that it is transmitted

Using the `capsule` keyword to protect secret da

```
void test() {
  ArrayList list1 = new ArrayList()
  list1.add(1);

  ArrayList list2 = list1;
  list2.add(2);

  System.out.println(list1.size());
}
```

If `test()` is run, what is the output?

1

2

Do not use any external resources to answer this question.

Which statements are true of interfaces in standard Java?

	True	False
Interfaces have no field declarations unless they are `public static final`.	○	○
Methods in interfaces are public by default.	○	○
Methods in interfaces (except for `default` methods) lack bodies.	○	○
A class can implement no more than one interface.	○	○

DEMOGRAPHICS

- Collect information if you want it!
- Programming experience? Languages?
- If they tell you, you can use it…
- e.g. Gender_____

TRAINING

- How will you prepare your participants?
- People don't read.
- People think they understand but in fact do not.
- Teach…and then assess.
- Or: decide that no training is necessary.

Obsidian Tutorial

Write a contract called **Person** that has an **Owned** reference to a **House** and a **Shared** reference to a **Park**. The **House** and **Park** contracts are given below.

```
contract House {

}

contract Park {

}
```

Please write your answer in the VSCode window (code1.obs). You may compile your code to check your answer.

```
contract Money {
    ...
}

contract Wallet {
    Money@Owned m;

    Wallet@Owned() {
        m = new Money();
    }

    transaction spendMoney() returns Money@Owned {
        ...
    }

    transaction receiveMoney(Money@Owned >> Unowned mon) {
        ...
    }
}
```

What is **m** in the above code fragment above?

- ◯ A Money object
- ◯ An Owned reference to a Money object
- ◯ An Owned object
- ◯ All of the above
- ◯ None of the above

TASKS

- This is the hardest part of study design.
- You will not get this right the first time.
- Solution: pilot repeatedly.
- But: you can use data from your "pilots" if you follow protocol.
- (a true "pilot" involves throwing the data out)
- What is the distribution over task times?

USABILITY STUDY TASKS

- Choose an *interesting* task
  - One that you think might be hard
  - One that is central to the usability of your design
- Can't test everything

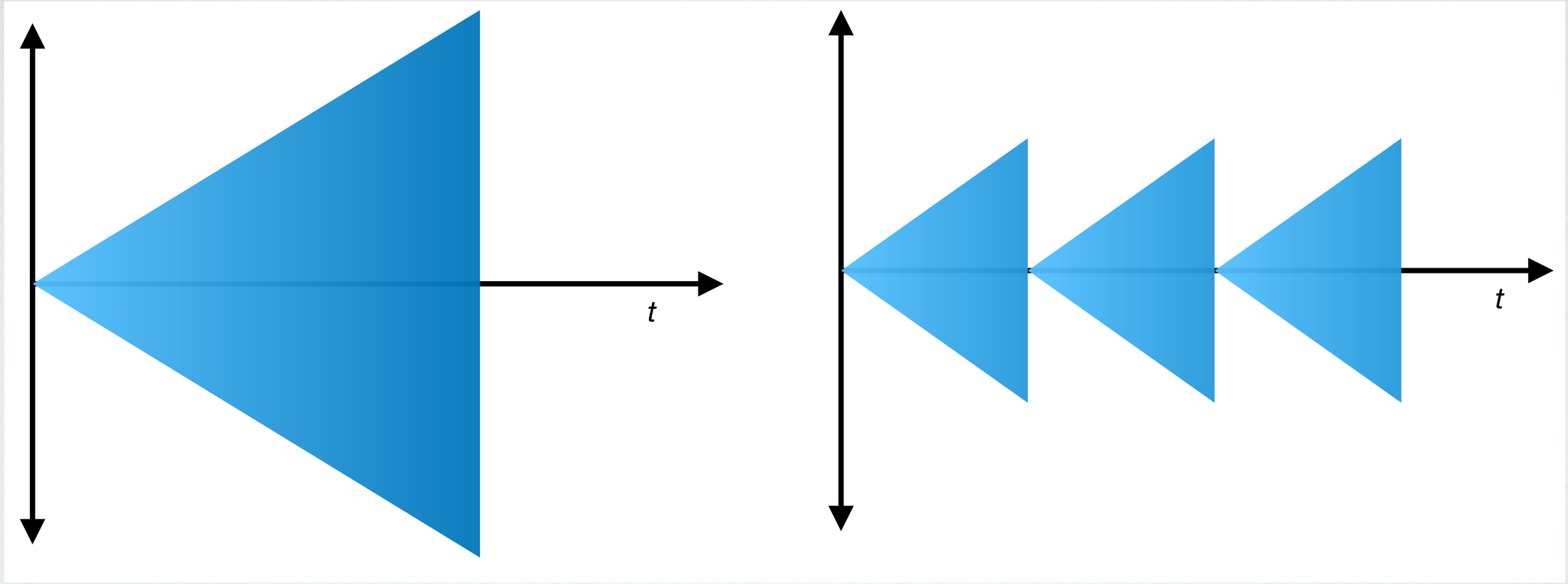
TASK IDEAS

- Write a program according to this specification.
- Are there bugs in this code? If so, what are they?
- Fill in the missing code…
- What does this code do?
- Answer these questions about this code.

TASK DESIGN

- Must carefully restrict tasks!
- People will get stuck on irrelevant things
- Decide how much help to provide
- Ideally: scope task to focus on the variable of interest
- *Constrain the task as much as possible.*

DECOMPOSING TASKS



Monolithic task

Subtasks

DATA COLLECTION

- Think-aloud
- Audio recordings
- Videos
- Screen capture
- Eye tracking
- Post-study survey

- Take lots of notes!, including timestamps! You do not want to watch the videos.
- Include a clock on the screen.

THINK-ALOUD

- Two varieties: concurrent and retrospective
- "Please keep talking."
- Can't use timing as a dependent variable due to effect of explanations.

TASK CONTEXTS

- Pencil/paper
- Text editor
- IDE
- Compiler?
- Debugger?
- Test cases?

TASK EXAMPLES

- Q: What challenges do web programmers encounter when using PHP to write web apps?
  - (who?)
- Plan: Recruit people who say they've made at least one web site in PHP
  - (what does one web site mean?)
- Task attempt 1: "write a gradebook app in PHP. You have 1 hour."
  - What is a gradebook app?
  - Where do you think people will get stuck?

TRY 2

- Refine task:
  - Here is a gradebook app, but the component that shows a student their grades is incomplete. Write `displayGrades()`, which will display a student's grades.
  - What format?
  - Are you measuring PHP, or the MySQL API, or the particular database schema, or something else?

TRY 3

- Refine research question
- What problems do PHP programmers encounter when handling errors?
- Put them in a situation where they will encounter errors!
- What kind of errors?
- Hypotheses:
  - They will forget to check for errors
  - They will misinterpret error codes
  - They will have trouble figuring out the causes of errors they encounter, even when the errors are common (e.g. "server unreachable")

```
main asset contract Auction {
  Participant@Unowned seller;

  state Open;
  state BidsMade {
    // the bidder who made the highest bid so far
    Participant@Unowned maxBidder;
    Money@Owned maxBid;
  }
  state Closed;

  ...


  transaction bid(Auction@Shared this,
                  Money@Owned >> Unowned money,
                  Participant@Unowned bidder) {
      if (this in Open) {
        // Initialize destination state,
        // and then transition to it.
        BidsMade::maxBidder = bidder;
        BidsMade::maxBid = money;
        ->BidsMade;
      }
      else {
        if (this in BidsMade) {
          //if the newBid is higher than the current Bid
          if (money.getAmount() > maxBid.getAmount()) {
            //1. TODO: fill this in.
            //   You may call any other transactions as needed.
            maxBidder.receivePayment(maxBid);
            maxBidder = bidder;
            maxBid = money;
          }
          else {
            //2. TODO: return the money to the bidder,
            //   since the new bid wasn't high enough.
            //You may call any other transactions as needed.
            bidder.receivePayment(money);
          }
        }
        else {
          revert ("Can only make a bid on an open auction.");
        }
      }
    }
}
```

```
contract Auction {
  // the bidder who made the highest bid so far
  address maxBidder;
  uint maxBidAmount;

  // 'payable' indicates we can transfer money to this address
  address payable seller;

  // Allow withdrawing previous money for bids that were outbid
  mapping(address => uint) pendingReturns;

  enum State { Open, BidsMade, Closed }
  State state;
  ...
  function bid() public payable {


    if (state == State.Open) {
      maxBidder = msg.sender;
      maxBidAmount = msg.value;
      state = State.BidsMade;
    }


    else {
      if (state == State.BidsMade) {
        //if the newBid is higher than the current Bid
        if (msg.value > maxBidAmount) {
          //1. TODO: fill this in.
          //   You may call any other functions as needed.
          pendingReturns[maxBidder] += maxBidAmount;
          maxBidder = msg.sender;
          maxBidAmount = msg.value;
        }
        else {
          //2. TODO: return the newBid money to the bidder,
          //   since the newBid wasn't high enough.
          //You may call any other functions as needed.
          pendingReturns[msg.sender] += msg.value;

        }
      }
      else {
        revert ("Can only make a bid on an open auction.");
      }
    }
  }
}
```

CONCLUSION

- Running usability studies requires:
  - Recruiting
  - Training
  - Task design
  - Data collection/analysis
- Task design is probably the trickiest. Start early and pilot!