UNIVERSITY OF TWENTE.

APPLICATION DEVELOPMENT

LECTURE 5: ARDUINO PART 2, DESIGNING CLASSES REVISITED

```
class AppDev {
}
```

Part of SmartProducts

INTRODUCTION

APPLICATION DEVELOPMENT

- Design a class
- Project planning
- Arduino programming part 2
- Assignment

Fjodor van Slooten
W241 *(Horst-wing West)*
f.vanslooten@utwente.nl

Please store kits properly.
Stack max. 4 pieces!

slides @ vanslooten.com/appdev

DESIGN A CLASS 'PRODUCT'

IN C++ (ARDUINO)

- Read assignment: we are going to make a *simplified* version of a vending machine…
- … which sells 'Products', which only needs (displays) the name, and does calculations with the price

```
class Product {
private:
   String name;
   int price; // in cents
public:
   // constructor (fills in name and price)
   Product(String n, int p);
   // methods (getters)
   String getName();
   int getPrice();
}
```

Step 1b: Identify methods …

Step 1: *analyze object* in real world…:
It is: a product
Map to properties: color, contents, type, name, dimensions, price…

DESIGN A CLASS 'PRODUCT'

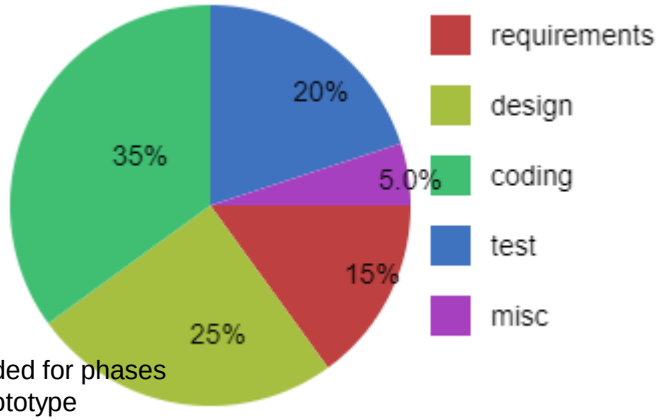
```
#include "Product.h"

// constructor:
Product::Product(String n, int p) {
  // assign values to class-variables name and price
  name = n;
  price = p;
}

// methods:
String Product::getName() {
  // return the name
  return name;
}

int Product::getPrice() {
  // return the price
  return price;
}
```

Step 2: add pseudo code which describes what methods should do
Step 3: add code to methods (change pseudo code into real code)

PROTOTYPE FOR PROJECT

HOW DO YOU DO?

- We are now in second half of project
- Usually, developing software (coding + testing) takes 55% of time
  (design + requirements = 40%)
- So to have enough time left, every group should now be developing prototype



Estimation of time needed for phases in development of a prototype

Prepare to vote

Internet

1

2

This presentation has been loaded without the Shakespeak add-in.
Want to download the add-in for free? Go to
http://shakespeak.com/en/free-download/.

TXT

1

2

Voting is anonymous

Regarding REQUIREMENTS specification we are

A. Just started
B. Busy, unknown how much time this will take
C. Almost ready
D. Done

The question will open when you start your session and slideshow.

Votes: 42

Closed

Internet

TXT

This text box will be used to describe the different message sending methods. The applicable explanations will be inserted after you have started a session.

Shakespeak

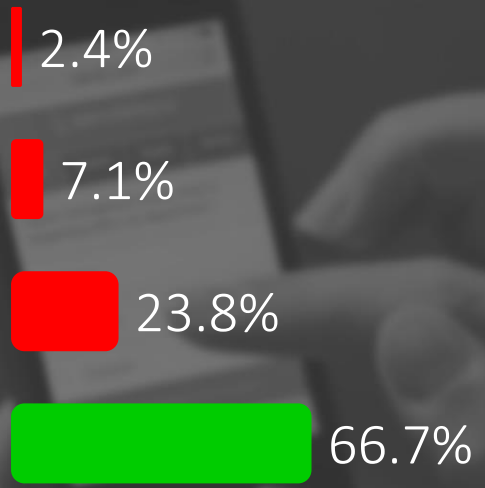
Regarding REQUIREMENTS specification we are

A. Just started

B. Busy, unknown how much time this will take

C. Almost ready

D. Done



Regarding DESIGN (for prototype) we are

A. Just started
B. Busy, unknown how much time this will take
C. Almost ready
D. Done

The question will open when you start your session and slideshow.

Votes: 44

Closed

This presentation has been loaded without the Shakespeak add-in. Want to download the add-in for free? Go to http://shakespeak.com/en/free-download/.

Internet

This text box will be used to describe the different message sending methods. The applicable explanations will be inserted after you have started a session.

TXT

Shakespeak

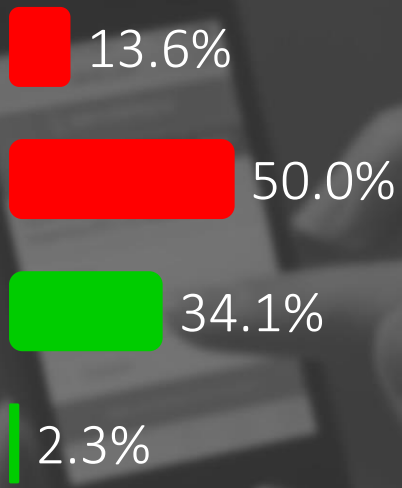
Regarding DESIGN (for prototype) we are

A. Just started

B. Busy, unknown how much time this will take

C. Almost ready

D. Done



= programming

Regarding CODING (for prototype) we are

A. Not started yet
B. Just started
C. Busy, unknown how much time this will take
D. Almost ready
E. Done

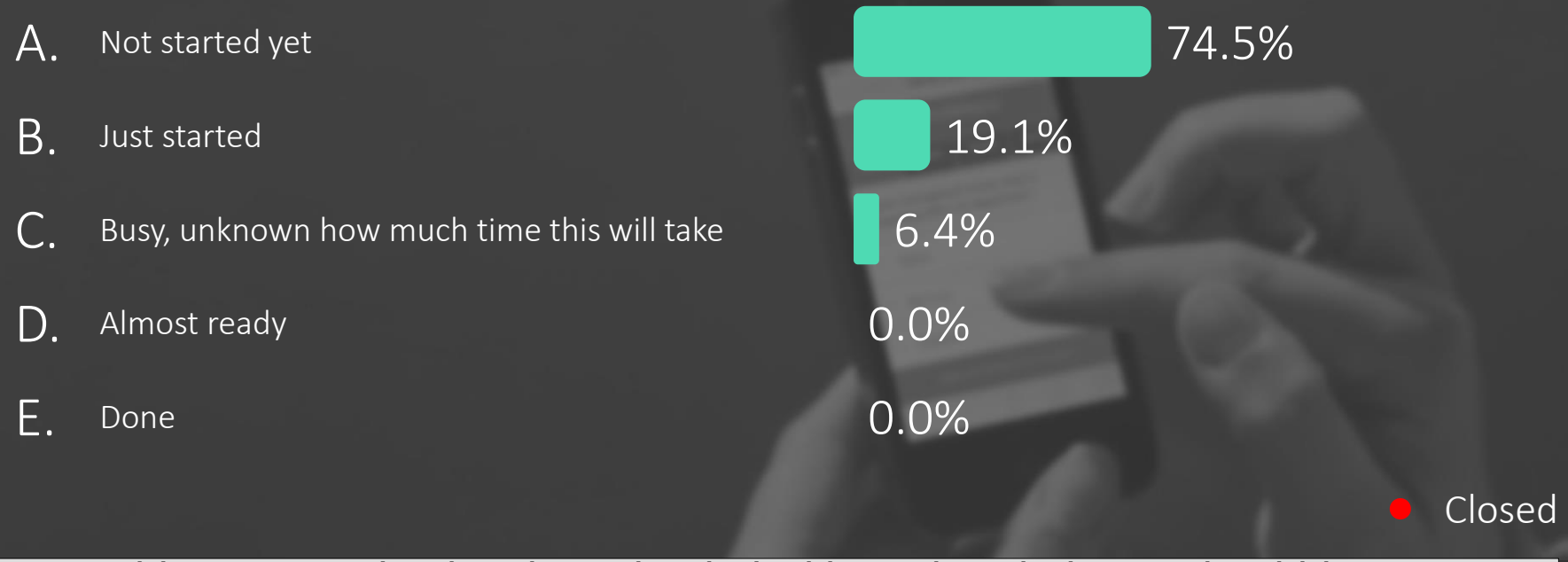
The question will open when you start your session and slideshow.

Votes: 47

Closed

Internet
TXT
This text box will be used to describe the different message sending methods. The applicable explanations will be inserted after you have started a session.
Shakespeak

Regarding CODING (for prototype) we are



Regarding TESTING the prototype we are

A. Not started yet
B. Just started
C. Busy, unknown how much time this will take
D. Almost ready
E. Done

The question will open when you start your session and slideshow.

Votes: 45

Closed

This presentation has been loaded without the Shakespeak add-in. Want to download the add-in for free? Go to http://shakespeak.com/en/free-download/.

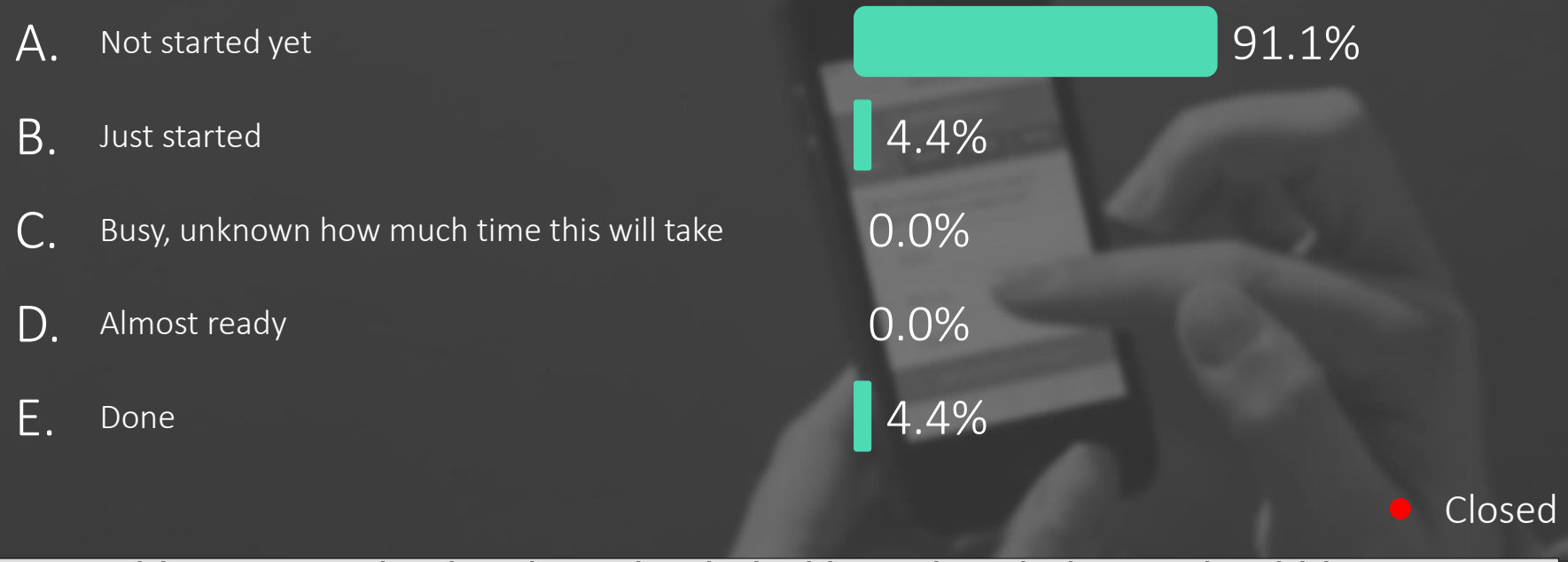
Internet

This text box will be used to describe the different message sending methods. The applicable explanations will be inserted after you have started a session.

TXT

Shakespeak

Regarding TESTING the prototype we are



Did you discuss a planning with your tutor?

A. Yes, and the phases just mentioned were also discussed
B. Yes, but different phases were discussed
C. Yes, but we did not have a clear planning
D. We still have to do this
E. No

The question will open when you start your session and slideshow.

Votes: 45

Closed

This presentation has been loaded without the Shakespeak add-in. Want to download the add-in for free? Go to http://shakespeak.com/en/free-download/.

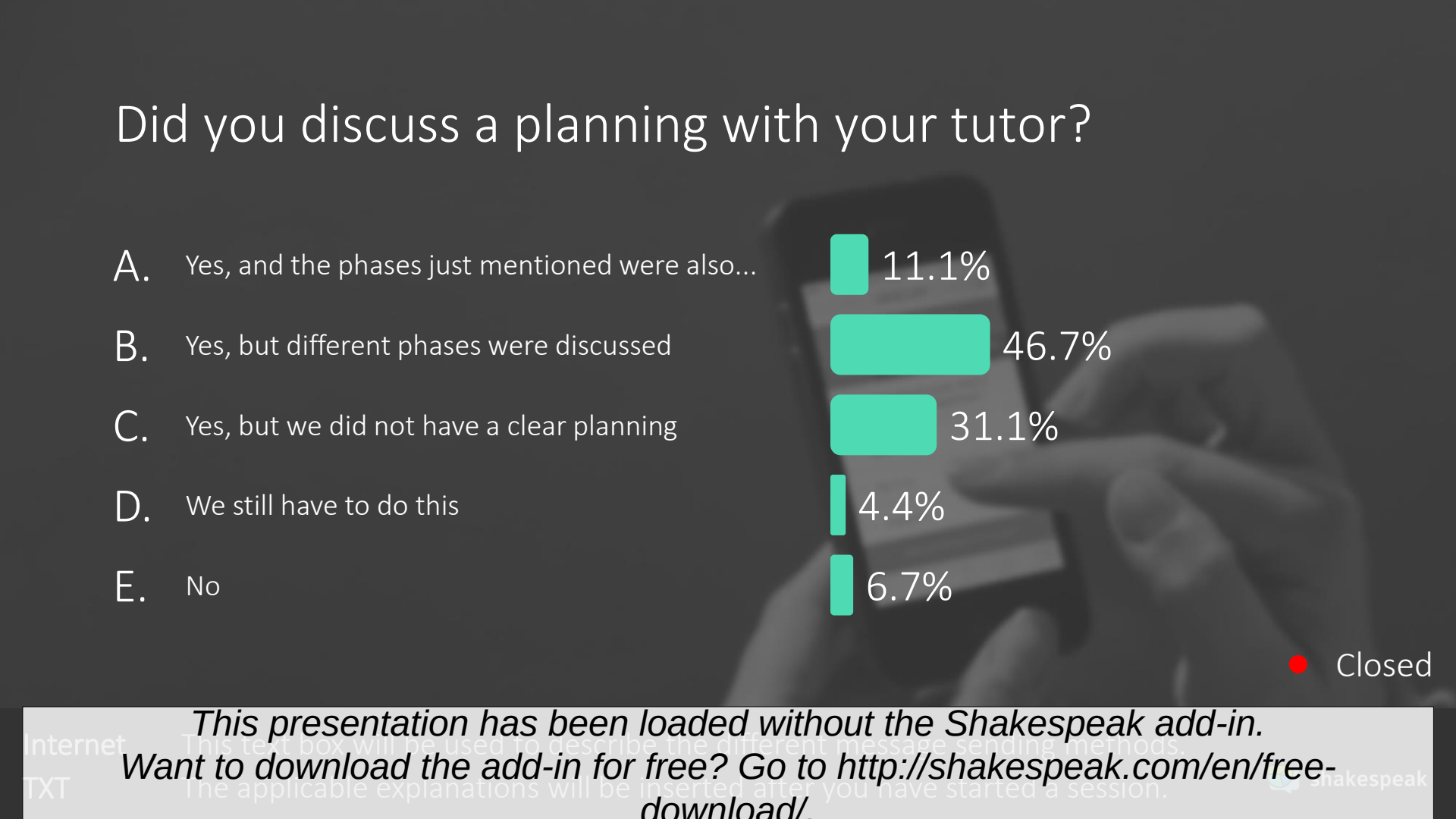
Internet

TXT

This text box will be used to describe the different message sending methods. The applicable explanations will be inserted after you have started a session.

Shakespeak

Did you discuss a planning with your tutor?

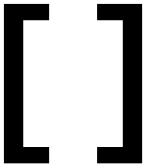


This presentation has been loaded without the Shakespeak add-in. Want to download the add-in for free? Go to http://shakespeak.com/en/free-download/.

LISTS

ALSO CALLED: ARRAYS

Head First: p59-69, 134-137 ***Aan de slag met*:** 7.4-7.6



- List of primitive types
- E.g. `byte, int, double`
- Java:
  - Always initialize with new
  - Lists of objects: use class **`ArrayList`**

Java:

```
// declare list of 4 integers:
int[] list = new int[4];

// assign values:
for (int i = 0; i < list.length; i++) list[i] = i;

int sum = 0;
for (int e : list) sum += e;
System.out.println(sum);
```

Arduino/C++:

```
// declare list of 5 led-pins:
byte leds[] = {2, 3, 4, 5, 13 };

// turn on the second led in the list:
digitalWrite(leds[1], HIGH);

// turn leds on one by one:
for (int i = 0; i < sizeof(leds); i++) {
   digitalWrite(leds[i], HIGH);
   delay(1000); // wait 1 sec.
}
```

Used in test-sketch for assignment 5a.

tutorialspoint: java arrays

ARRAYLIST

ONLY IN JAVA!

Practice: assignment 4b: bouncing balls

- Keeps list of objects
- Assignment 5b: used in Controller
- Methods:

  add(Object)

  get(int)

  size()

```
// declare list which contains balls:
ArrayList<Ball> balls;

// create new ball b:
Ball b = new Ball();

// add ball b to list:
balls.add(b);

// get ball number 5 from the list:
Ball b = balls.get(5);

// how many balls in the list?:
System.out.println("Number of balls: "+balls.size() );
```

ARRAYLIST

JAVA VERSION

Practice: assignment 5b:
Controller has list of products

```
ArrayList<Product> products; // list of products

products = new ArrayList<Product>(); // new empty list

// add a product:
products.add(new Product("Cola", 100));

// method:
private boolean checkPayment() {
   // Get product from list:
   Product p = products.get(chosenItem);

   // If balance is higher or equal to price of chosen product, return true
   if (balance >= p.getPrice()) {
      return true;
   }
   gui.displayMessage("Balance insufficient.");
   return false;
}
```

In assignment: check methods of class Product to get name and price of a Product

ARRAY OF OBJECTS

C++ VERSION

Practice: assignment 5a:
Controller has list of products

```
Product * products[NUM_PRODUCTS]; // list of products, new empty list

// add a product:
products[0] = new Product("Cola", 100);

// method:
bool Controller::checkPayment() {
    // Get product of list:
    Product * p = products[chosenItem];

    // If balance is higher or equal to price of chosen item, return true
    if (balance >= p->getPrice()) {
      return true;
    }
    gui->displayMessage("Balance insufficient.");
    return false;
}
```

In assignment: check methods of class Product to get name and price of a Product

LOOP TROUGH A LIST

JAVA VERSION

- for-each loop:

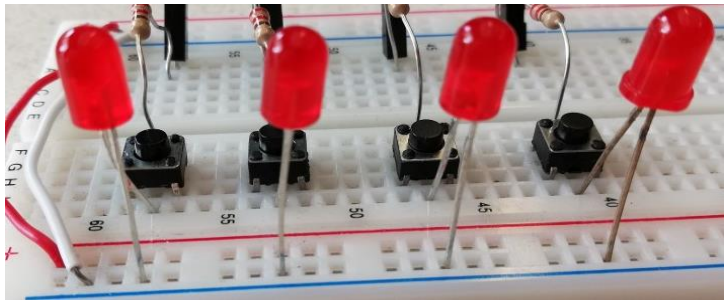
```
public void paintComponent(Graphics g) {
  super.paintComponent(g);
  // Draw all balls by calling the paintComponent() method for each ball:
  for (Ball b : balls) { // for each ball in the list..
      b.paintComponent(g); // draw the ball
  }
}
```

- ‘normal’ for loop:

```
for (int i = 0; i<balls.size(); i++) {
   balls.get(i).paintComponent(g);
}
```

ARDUINO PROGRAMMING PART 2

- Buttons and LEDs
- LED:
  - Set pin as output: `pinMode(13, OUTPUT);`
  - `digitalWrite(13, HIGH);`
- Button: `buttonState = digitalRead(6);`

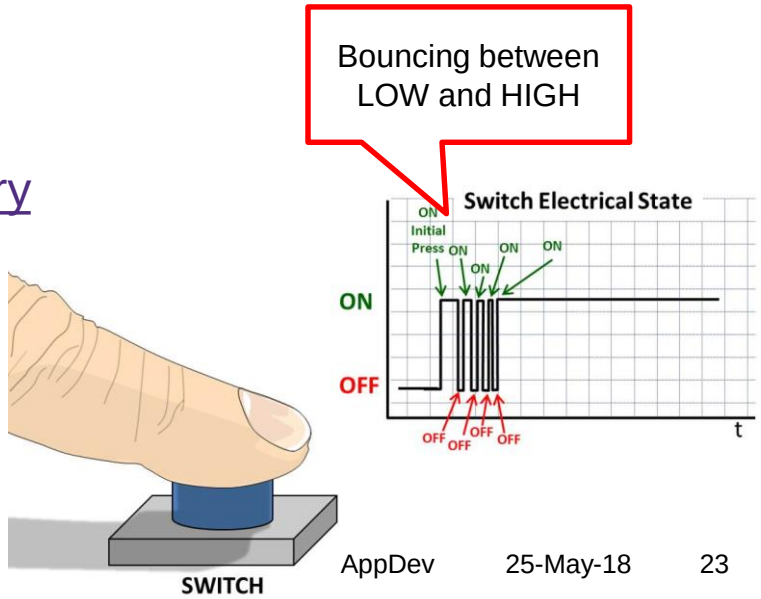


arduino.cc/en/Tutorial/Button

arduino.cc/reference/en/language/functions/digital-io/digitalwrite/

PRESSING A BUTTON…

- Interaction with button takes time:
- Switching from LOW to HIGH has a short period in which state is undefined
- We need to ‘de-bounce’
- In assignment, we use Bounce2-library for this



Example: multi_button_check_analog_input_lcd.ino

A LOT OF BUTTONS…

Use Bounce2 library

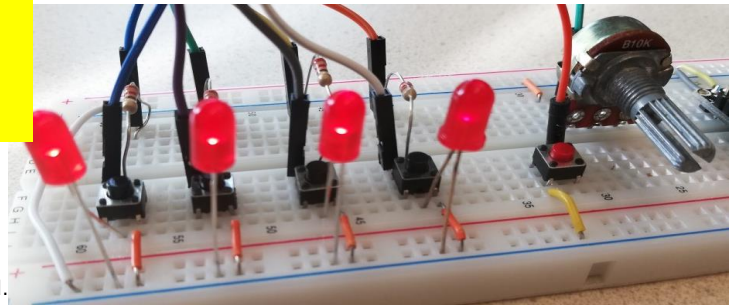
List of pin numbers

List of buttons

Setup the buttons in a for-loop

```
#include <Bounce2.h>

byte button_pins[] = {6, 7, 8, 9, 10 }; // button pins

#define NUMBUTTONS sizeof(button_pins)

Bounce * buttons = new Bounce[NUMBUTTONS];

void setup() {
  // Make input & enable pull-up resistors on switch pins
  for (int i=0; i<NUMBUTTONS; i++) {
    // setup the bounce instance:
    buttons[i].attach( button_pins[i], INPUT_PULLUP);
    buttons[i].interval(25); // interval in ms
  }
}
```

We use internal pull-up* resistors of Arduino, this saves us having to mount 5 additional resistors



* 'pull-up' means a resistor pulls-up the signal (so it is HIGH).
This also means we have to check the value to fall (becomes LOW) when the button is pressed.

CHECKING A LOT OF BUTTONS…

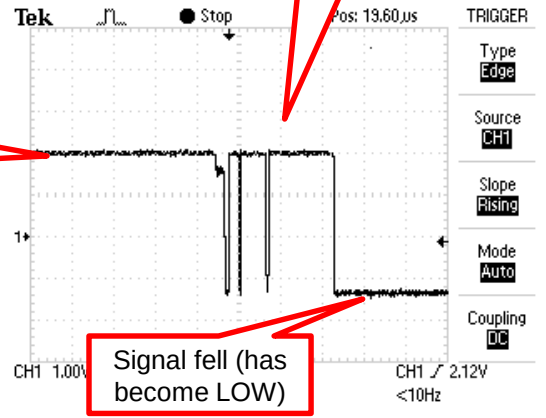
```
void loop() {
  // check all buttons if they are pressed:
  for (int i = 0; i<NUMBUTTONS; i++) {
    // Update the Bounce instance:
    buttons[i].update();
    // If it fell*, do something:
    if ( buttons[i].fell() ) {
      Serial.print(i);
      Serial.println(" press");
    }
  }
}
```

For-each button…

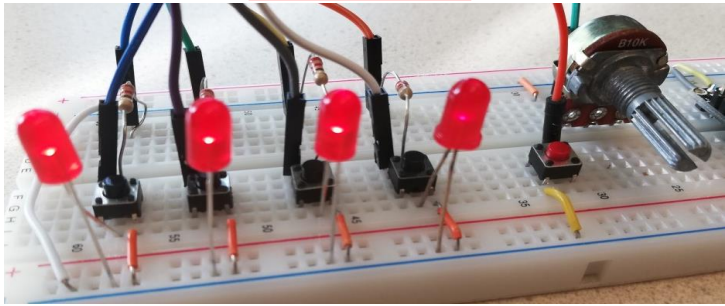
Check if the button fell*

Bouncing

Signal is HIGH (due to pull-up)



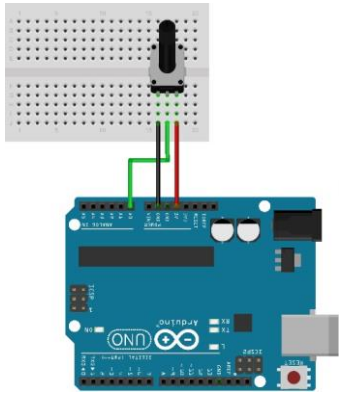
Signal fell (has become LOW)



* Because pull-up resistors were used, we have to check the value to fall (becomes LOW) when the button is pressed.

READING ANALOG VALUE

- Potmeter: variable resistor
- `sensorValue = analogRead(A0);`



```
void loop() {
  // read analog input:
  sensorValue = analogRead(sensorPin);
  // check if value changed from last read:
  if (abs(lastSensorValue-sensorValue)>2 ) {
    lastSensorValue = sensorValue;
    Serial.print("Analog val: ");
    Serial.println(sensorValue);
  }
}
```

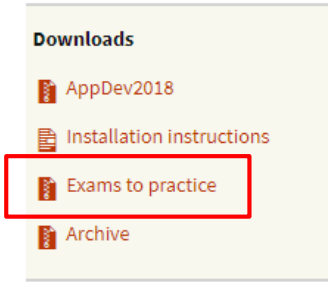
wikipedia.org/wiki/Potentiometer

arduino.cc/reference/en/language/functions/analog-io/analogread/

TESTS (EXAM): PRACTICE TEST

- Two tests to practice
- More info in lecture #8 and ‘live’ practice test-questions
- Exam: Monday July 2nd 8:45-11:45, location to be announced

downloads @ vanslooten.com/appdev:



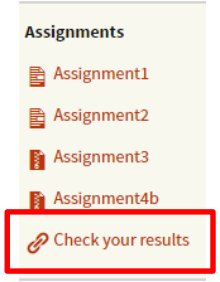
ASSIGNMENT #5

This afternoon: teacher available for help with project

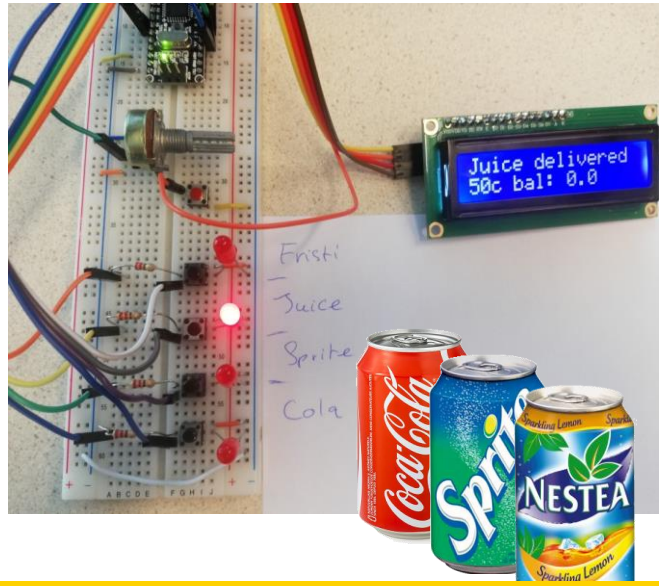
- "Build prototype of soda machine"
- Build electronic circuit with Arduino
- Recommended to do with 2 students:
  - Split tasks: build circuit, program

Checkpoint

Check assignments results:



- Assignment 5b: if you do not have Lego Mindstorms kit/Arduino, or do not want to build Arduino circuit



UNIVERSITY OF TWENTE.

Slides, assignments etc @ vanslooten.com/appdev