PasteTrace: A Single Source Plagiarism Detection Tool For Introductory Programming Courses

Jesse McDonald

Advisors: Anthony Peruma, Henri Casanova, Scott Robertson | MS Plan B / ICS 699, Fall 2024

Introduction

- Intro CS courses lay the groundwork for advanced CS courses
- Intro CS courses are often a students first experience in programming
- Programming can be hard to learn
- Programming assignments are easy to plagiarize (especially now with ChatGPT)
- Students struggling in intro CS courses can see plagiarism as a way out.
- It is really hard to detect plagiarism in a batch of identical intro CS assignments by comparing code

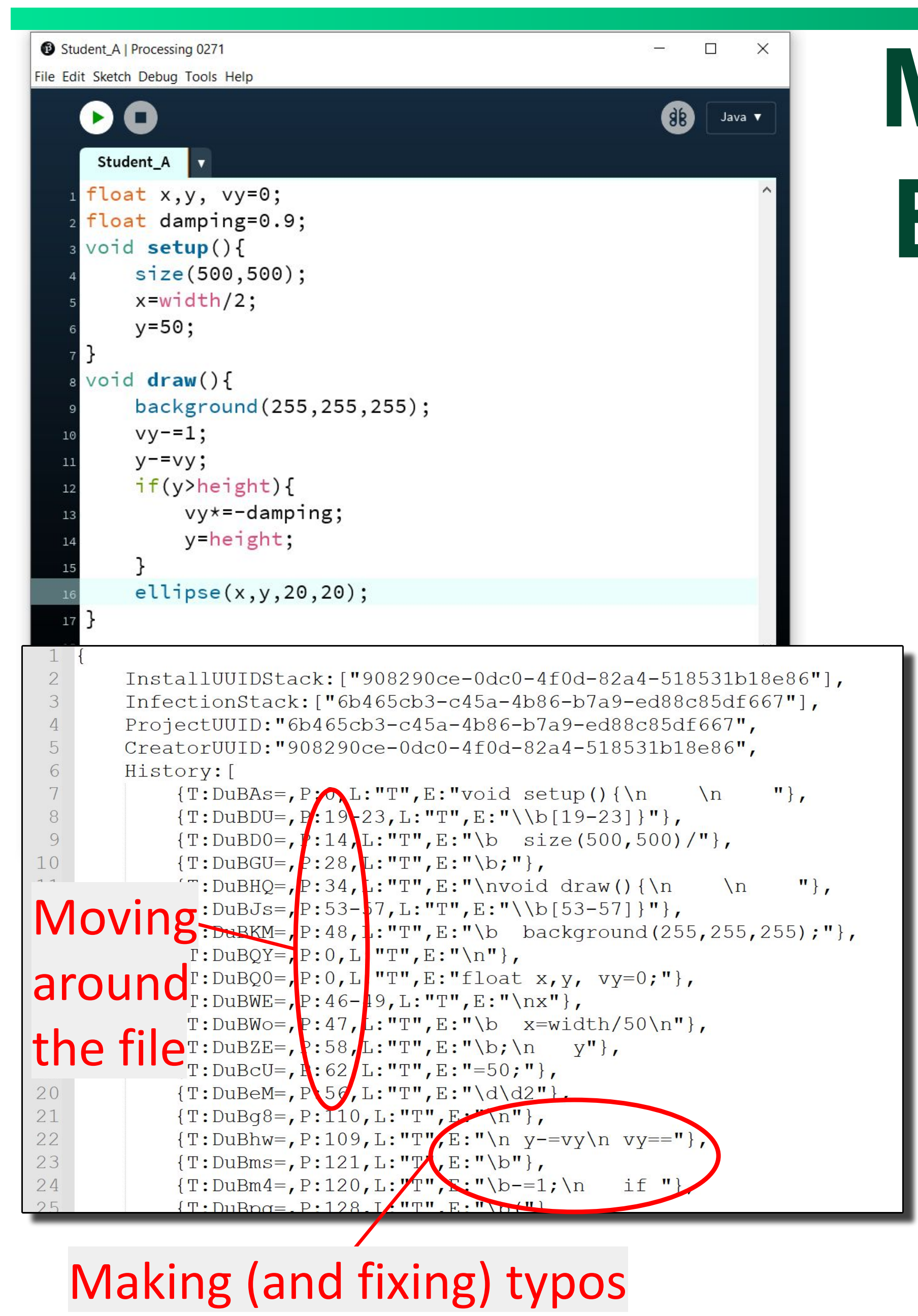
Motivation

- If a student passes an intro class by plagiarizing, then they likely do not understand part or all of the course
- If this student continues to advanced classes without the basics, they must
  - Learn the basics outside of class (more work)
  - Continue to plagiarize assignments (what they are already doing)
- Continuing to plagiarize could lead to graduating without the requisite skill set
- Jobs require actual skill, not plagiarism
- Employers may assume the degree to blame, not plagiarism
  - They may then raise the bar for future “entry level” positions
- Current detection tools compare source code semantics or syntax
  - Frequently falsely detects common code patterns (e.g. nested loops)
  - Frequently confused by minor variations (e.g. renamed variables)
  - Good at catching identical files

Solution

- PasteTrace is an Integrated Development Environment (IDE) for plagiarism detection
- PasteTrace IDE uses special metadata
  - Hidden in comment at the end of a file
  - Unique Identifier (UUID) of computer
    - Generated during IDE install
  - UUID of project
  - Edit event history
    - Typed code
    - Copy events
    - Paste events
    - Deleted code
- PasteTrace hides metadata in copy events
  - All UUIDs included
  - Invisible binary encoding using U+200B (0 width spaces) and regular characters.
  - Paste events categorized based on UUIDs to determined origin of code
- Analysis script can use this metadata to detect plagiarism

Metadata Example



Student_A | Processing 0271

```
float x,y, vy=0;
float damping=0.9;
void setup(){
    size(500,500);
    x=width/2;
    y=50;
}
void draw(){
    background(255,255,255);
    vy-=1;
    y-=vy;
    if(y>height){
        vy*=-damping;
        y=height;
    }
    ellipse(x,y,20,20);
}
```

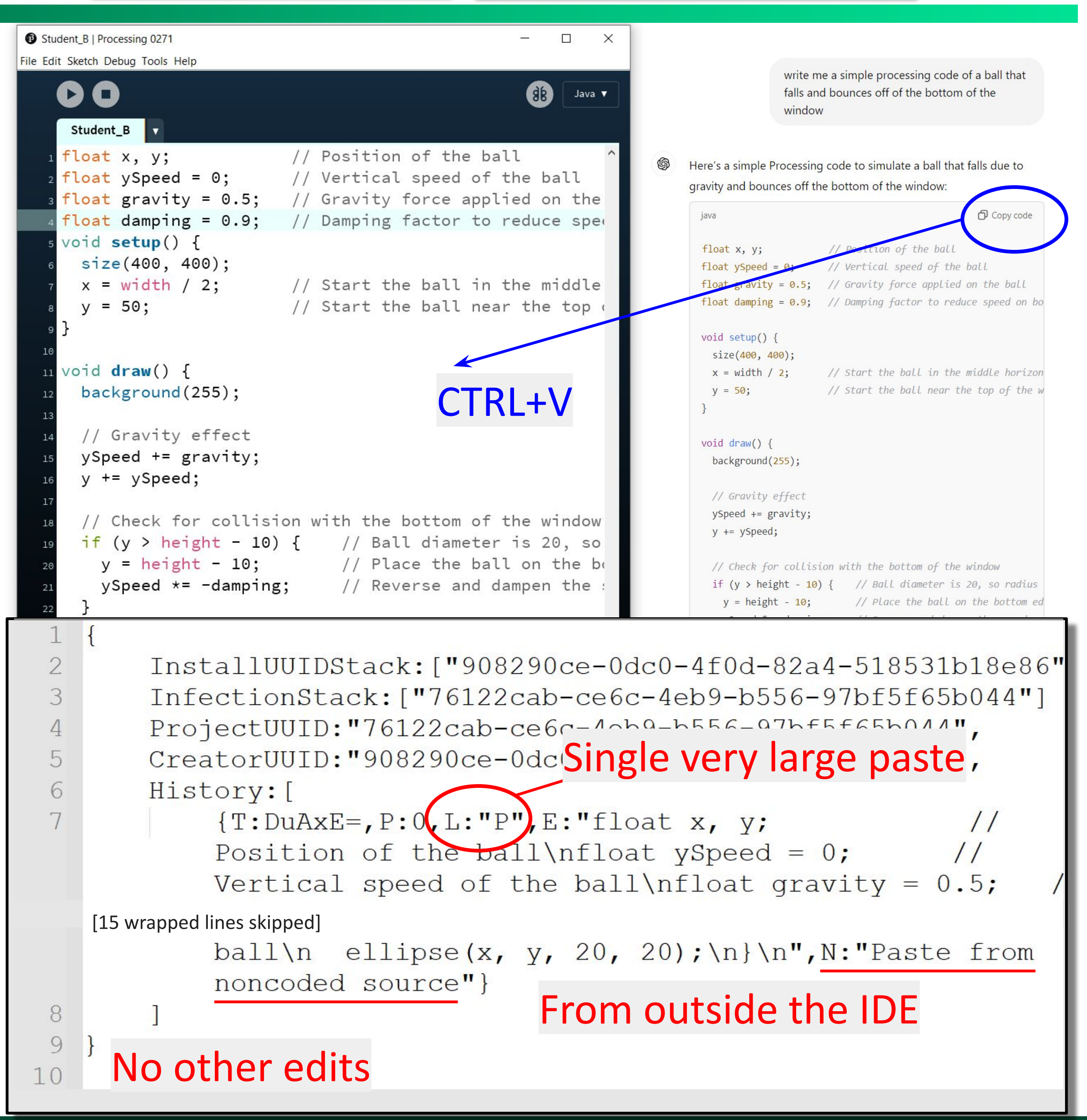
```
{
    InstallUUIDStack:["908290ce-0dc0-4f0d-82a4-518531b18e86"],
    InfectionStack:["6b465cb3-c45a-4b86-b7a9-ed88c85df667"],
    ProjectUUID:"6b465cb3-c45a-4b86-b7a9-ed88c85df667",
    CreatorUUID:"908290ce-0dc0-4f0d-82a4-518531b18e86",
    History:[
        {T:DuBAs=,P:0,L:"T",E:"void setup(){\n    \n    "},
        {T:DuBDU=,P:19-23,L:"T",E:"\\b[19-23]}"},
        {T:DuBD0=,P:14,L:"T",E:"\b  size(500,500)/"},
        {T:DuBGU=,P:28,L:"T",E:"\b;"},
        {T:DuBHQ=,P:34,L:"T",E:"\nvoid draw(){\n    \n    "},
         :DuBJs=,P:53-57,L:"T",E:"\\b[53-57]}"},
         :DuBKM=,P:48,L:"T",E:"\b  background(255,255,255);"},
         T:DuBQY=,P:0,L "T",E:"\n"},
         T:DuBQ0=,P:0,L "T",E:"float x,y, vy=0;"},
         T:DuBWE=,P:46-49,L:"T",E:"\nx"},
         T:DuBWo=,P:47,L:"T",E:"\b  x=width/50\n"},
         T:DuBZE=,P:58,L:"T",E:"\b;\n   y"},
         T:DuBcU=,P:62,L:"T",E:"=50;"},
        {T:DuBeM=,P:56,L:"T",E:"\d\d2"},
        {T:DuBg8=,P:110,L:"T",E:"\n"},
        {T:DuBhw=,P:109,L:"T",E:"\n y-=vy\n vy=="},
        {T:DuBms=,P:121,L:"T",E:"\b"},
        {T:DuBm4=,P:120,L:"T",E:"\b-=1;\n   if "},
```

Moving around the file

Making (and fixing) typos

Student A (left) completed assignment without plagiarism.

Student B (right) just copied from ChatGPT



Student_B | Processing 0271

```
float x, y;              // Position of the ball
float ySpeed = 0;        // Vertical speed of the ball
float gravity = 0.5;     // Gravity force applied on the
float damping = 0.9;     // Damping factor to reduce spe
void setup() {
  size(400, 400);
  x = width / 2;         // Start the ball in the middle
  y = 50;                // Start the ball near the top
}

void draw() {
  background(255);

  // Gravity effect
  ySpeed += gravity;
  y += ySpeed;

  // Check for collision with the bottom of the window
  if (y > height - 10) {    // Ball diameter is 20, so
    y = height - 10;        // Place the ball on the bo
    ySpeed *= -damping;     // Reverse and dampen the
  }
```

write me a simple processing code of a ball that falls and bounces off of the bottom of the window

Here's a simple Processing code to simulate a ball that falls due to gravity and bounces off the bottom of the window:

```
float x, y;              // Position of the ball
float ySpeed = 0;        // Vertical speed of the ball
float gravity = 0.5;     // Gravity force applied on the ball
float damping = 0.9;     // Damping factor to reduce speed on bo

void setup() {
  size(400, 400);
  x = width / 2;         // Start the ball in the middle horizon
  y = 50;                // Start the ball near the top of the w
}

void draw() {
  background(255);

  // Gravity effect
  ySpeed += gravity;
  y += ySpeed;

  // Check for collision with the bottom of the window
  if (y > height - 10) {    // Ball diameter is 20, so radius
    y = height - 10;        // Place the ball on the bottom ed
```

CTRL+V

```
{
    InstallUUIDStack:["908290ce-0dc0-4f0d-82a4-518531b18e86"
    InfectionStack:["76122cab-ce6c-4eb9-b556-97bf5f65b044"]
    ProjectUUID:"76122cab-ce6c-4eb9-b556-97bf5f65b044",
    CreatorUUID:"908290ce-0dc
    History:[
        {T:DuAxE=,P:0,L:"P",E:"float x, y;              //
        Position of the ball\nfloat ySpeed = 0;        //
        Vertical speed of the ball\nfloat gravity = 0.5;    /
[15 wrapped lines skipped]
        ball\n  ellipse(x, y, 20, 20);\n}\n",N:"Paste from
        noncoded source"}
    ]
}
```

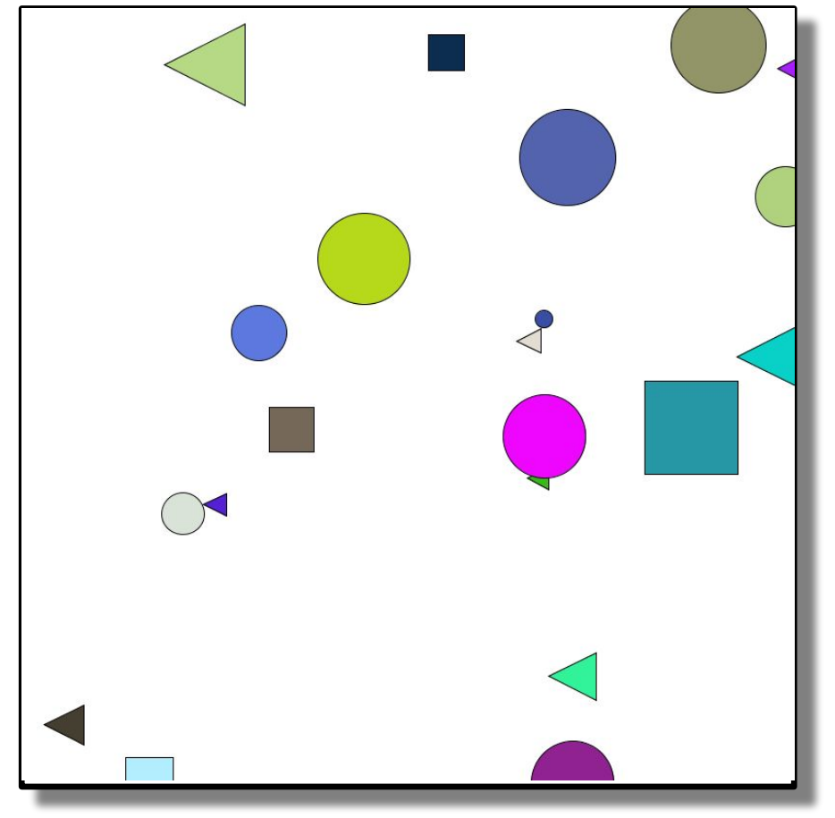
Single very large paste

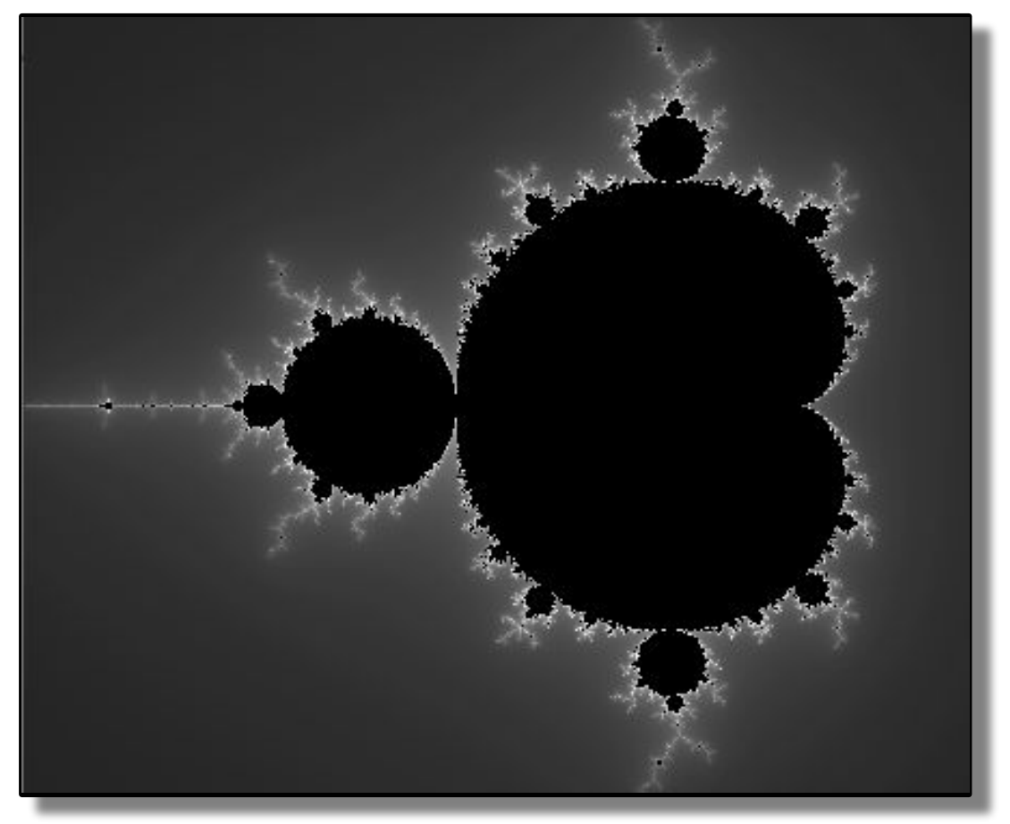
From outside the IDE

No other edits

Case Studies

- Participants selected from SP2023 ICS 111 and ICS 211 classes
  - 5 participants from ICS 111
  - 18 participants from ICS 211
- Participants given identical USB drive with Paste Trace IDE and difficult assignment
  - Polymorphic bouncing shapes (ICS 111)
  - Mandelbrot Fractal (ICS 211)
- Participants encourage to plagiarize this assignment and describe how they completed the assignment
- Submissions were anonymous via USB
- Solution to each “seeded” several places
  - GitHub
  - Assignment website





Results

- Submissions given to Stanford MOSS plagiarism detection tool as well
  - MOSS designed to detect code sharing
  - Frequently detects common code
- Very few participants shared code directly with another participant
- Some participants copied same online source independently
- Not everyone who cheated completed the assignment
- One Participant manually copied video
- Most plagiarized with single large paste

Feature of Interest	Count
Participants who completed assignment (correctly)	15
Participants who plagiarized	15
Participants who used ChatGPT	3
Participants who found seeded	1
Participants who used any other online source	12
Participants who used same source as another participant	5
Participants who shared code	2
Assignments detected without automated tools	4
Assignments detected using PasteTrace metadata	14
MOSS correct detections	5
MOSS incorrect detections	7

Future Work

- Detect copying video by “linear typing”
- Detect Collaboration using timestamps
- Use PasteTrace to research efficacy of methods to reduce plagiarism

Information and Computer Sciences Department, University of Hawai’i at Mānoa