PLTW Computer Science Principles

Nimmi Arunachalam
Project Lead The Way Master Teacher

FETC Firehose
January 25, 2017
Inspiring, Engaging, and Empowering K-12 Students

Computer Science

Engineering

Biomedical Science
A computer science experience with a specific grade band in mind.

PLTW Launch (K-5) taps into students’ exploratory nature, engages them in learning that feels like play helping them understand how computer science relates to the world around them.

PLTW Gateway (6-8) focuses on the positive impact of the application of computer science to other disciplines, fundamental for students as they define their career path opportunities and plans.

PLTW Computer Science (9-12) offers the opportunity for students to explore increasingly complex computational challenges and solve open-ended problems like professionals will do.
PLTW provides a unique opportunity for students to connect learning across disciplines and industries. Students deepen their computational thinking knowledge and skills as they advance through our four year-long course offering.

- Computer Science Essentials
- Computer Science Principles
- Computer Science A
- Cybersecurity(*)

(*) Available Fall 2018
Empowers students to grow from consumers to creators of technology

- Builds on our proven activity-, project-, problem-based learning approach
- Develops computational thinking
- Generates excitement about career paths that utilize computing
- Introduces professional tools that foster creativity and collaboration

PLTW is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP).
Activity-, Project-, Problem-based Learning Approach

Learning in context

• A problem frames each lesson

<table>
<thead>
<tr>
<th>Opportunities to Create</th>
<th>Opportunities to Explore</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Create a Scratch Game or Story</td>
<td>• Digital Property and Forensics</td>
</tr>
<tr>
<td>• Design an App</td>
<td>• Human Computer Interaction</td>
</tr>
<tr>
<td>• Design an Image Algorithm</td>
<td>• Security and Liberty</td>
</tr>
<tr>
<td>• Design a Data-Driven Website</td>
<td>• Big Data and Parallel Processing</td>
</tr>
<tr>
<td>• Investigate with Data</td>
<td>• Computing Impacts All Fields</td>
</tr>
<tr>
<td>• Modify a Simulation</td>
<td></td>
</tr>
</tbody>
</table>

Activity, Project, Problem-based Learning Approach

Activity 1 | 2 | 3 | Project | Problem

Knowledge & Skill Acquisition | Meaning Making & Investigation | Application & Transfer

Well-Defined | Open-Ended
- PLTW provides ready-to-use, activity-, project-, and problem-based curricula.

- Course includes comprehensive teacher plans, cohesive student and teacher materials, and access to PLTW’s full-service support team.

- PLTW Computer Science Principals is structured into 12 lessons grouped in 4 units.
Semester Grade Calculator. This was done using App Inventor. Shown below are the user interface setup, and the programming blocks.
Programming to create an interactive game with Python®

```python
import random

def goguess():
    print "Guess a number between 1 and 20"
    num = random.randint(1,20)
    guess = int(raw_input('Guess '))
    print num
    guesses = 0
    while guess!=num:
        if guess < num:
            print 'Too low... try again'
            guess = int(raw_input('Guess '))
            guesses+=1
        else:
            print 'Too High... try again'
            guess = int(raw_input('Guess '))
            guesses+=1
    print ('You are done in',guesses,'guesses!')
```

Python: Modifying images to give user defined effects

Using Python to create widgets using the Tkinter library
Agent Modeling Library – NetLogo

**Wolf Sheep Predation**

This model explores the stability or predator-prey ecosystems. Both a system is called unstable if tends to result in extinction for one or more species involved. In contrast, a system is stable if tend to maintain itself over time, despite fluctuations in population sizes.

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```

```
turtles set initial-to-energy [energy]
set initial-to-energy [energy]
set initial-to-energy [energy]
```
Prisoner’s Dilemma using GitHub

The actual Python code

```python
1  from __future__ import print_function
2  
3  def play_round(player1, player2, history1, history2, score1, score2):
4      
5      Calls the get_action() function which will get the characters
6      'c' or 'b' for collude or betray for each player.
7      The history is provided in a string, e.g. 'ccb' indicates the player
8      colluded in the first two rounds and betrayed in the most recent round.
9      Returns a 4-tuple with updated histories and scores
10     (history1, history2, score1, score2)
11     
12     RELEASE = 0 # (R) when both players collude
13     TREAT = 100 # (T) when you betray your partner
14     SEVERE_PUNISHMENT = -500 # (S) when your partner betrays you
15     PUNISHMENT = -250 # (P) when both players betray each other
16     # Keep T > R > P > S to be a Prisoner's Dilemma
17     # Keep 2R > T + S to be an Iterative Prisoner's Dilemma
18     
19     #Get the two players' actions and remember them,
20     action1 = get_action(player1, history1, score1, score2)
21     action2 = get_action(player2, history2, score1, score2)
22     
23     # Calculate the scores
24     score1 += 0 if action1 == 'c' and action2 == 'c' else PUNISHMENT
25     score1 += TREAT if action1 == 'b' else 0
26     score2 += 0 if action2 == 'c' and action1 == 'c' else PUNISHMENT
27     score2 += TREAT if action2 == 'b' else 0
28     
29     # Update the histories
30     history1 += action1
31     history2 += action2
32     
33     print(f'Round {i + 1}: {score1}, {score2}')
34     
35     return history1, history2, score1, score2
36 
```

1. PrisonerDilemma.py allows hard-coding different strategies for the Iterative Prisoners Dilemma, the canonical game of game-theory.
2. Each strategy plays 100 to 200 rounds against each other strategy.
3. The results of all previous rounds within a 100-200 round stretch are known to both players.
4. play_tournament() executes the tournament and stores output in tournament.txt
5. Players should each code their strategies in their assigned section of code.
6. Aggregated results are stored in tournament.txt
7. Unpublished work (c)2013 Project Lead The Way
8. CSE Project 1.3.5 Collaborating on a Project
9. Draft, Do Not Distribute
Preparing Teachers to Transform the Classroom Experience
Preparing Educators and Instilling Confidence

- PLTW Professional Development offers dynamic, research-validated learning experiences for educators through powerful and flexible instructional support:
  - tailored to individual needs,
  - live instruction,
  - on-demand resources, and
  - an ongoing professional community.

Through our online PLCs, teachers can connect and share experiences and expertise with fellow educators from across the nation who are teaching CSP.
Girls Build With PLTW Computer Science

Creating Solutions Through Purposeful Programming

The Impact of Computer Science Principles

Everyone’s a Maker
Let’s Continue the Conversation

Find us

http://pltw.org

Like us

Facebook: Project Lead The Way

Follow us

Twitter: @PLTWorg #PLTW

LinkedIn: Project Lead The Way

Contact us at solutioncenter@pltw.org.

Check our blog for more stories.