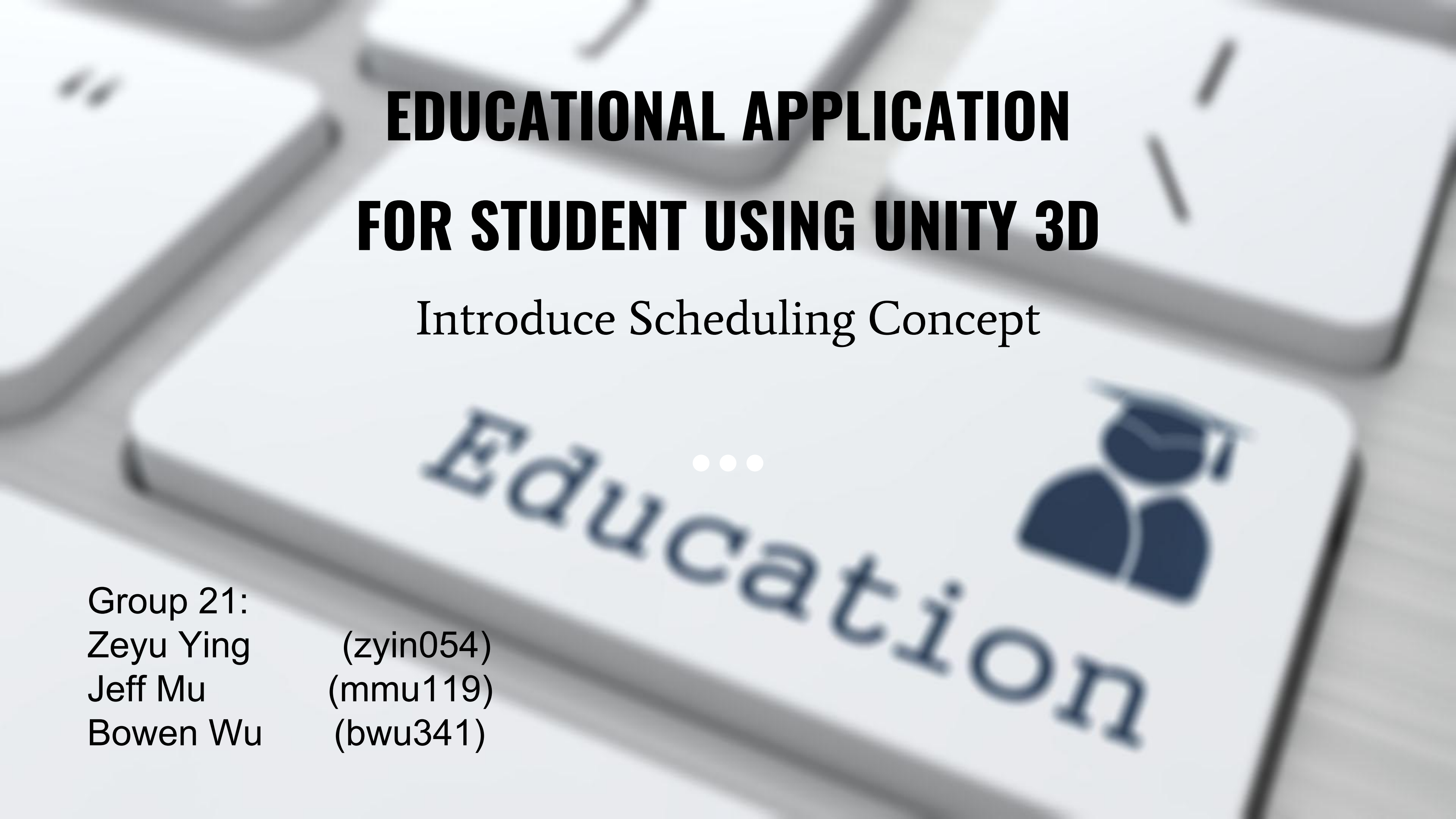


EDUCATIONAL APPLICATION FOR STUDENT USING UNITY 3D

Introduce Scheduling Concept

Group 21:

Zeyu Ying	(zyin054)
Jeff Mu	(mmu119)
Bowen Wu	(bwu341)

Presentation Overview

- Introduction
- Target audience
- Why we choose Scheduling
- Project goal
- Flowchart
- Implementation
- Demo
- Future work
- Question

Introduction

A cross-platform game engine which is primarily used to develop video games and simulations for PC, consoles, mobile devices and websites.

Hierarchical integrated development environment, visual editing, detailed property editor and dynamic application preview.

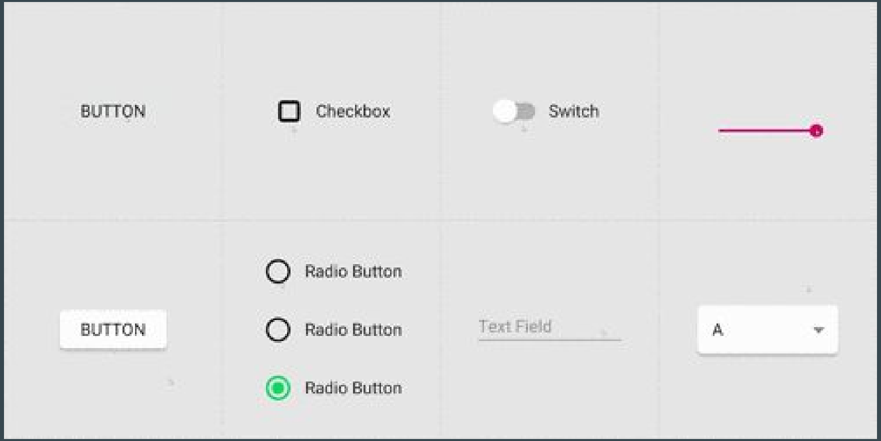
What does it provide?...

Introduction

Create visually-appealing games and applications with MaterialUI for

Unity, a meticulously-crafted tool

that allows you to easily utilize the bold and modern components of Google's Material Design.



materialUI

Target Audience

Students

- Close to us
  - We are fresh to the concepts
  - What are they learning
  - What’s important for them
  - Students need more interactive way to learn
- Scope fits our project

Why we choose Scheduling

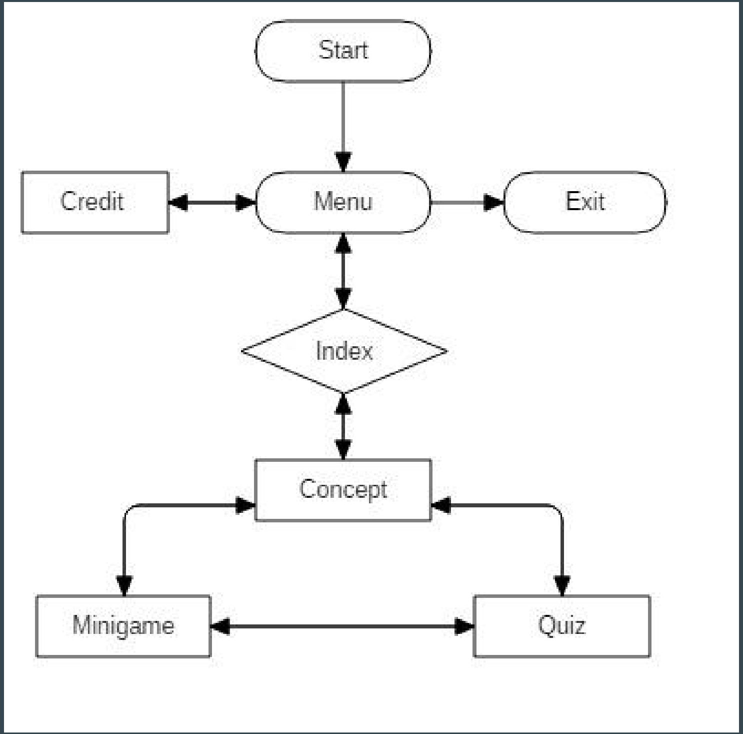
- It is a fundamental concept for parallel programming.
- Suitable for interaction.
- Suitable for visualization.
- Enable user to understand the concept more intuitive.

Project Goal

An educational application helps student to learn scheduling concept

- Adequate theory content
- Intuitive presentation
- Engaging interaction

FLowchart



Scheduling

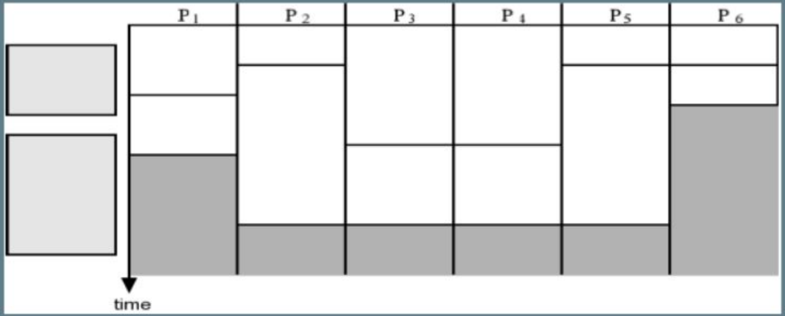
- What is scheduling
- Two kinds of scheduling
    - Task scheduling
        - Thoery
        - Game
    - Loop scheduling
        - Thoery
        - Quiz

Task Scheduling

- Introduce what is Task scheduling.

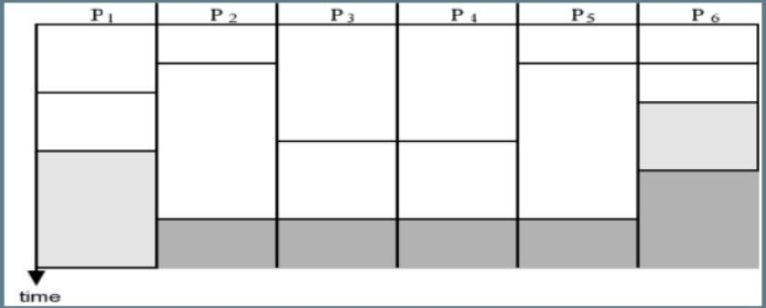
- Introduce two kinds of task scheduling
  - Task scheduling -- without dependence
  - Task scheduling -- with dependence

Task scheduling without dependence



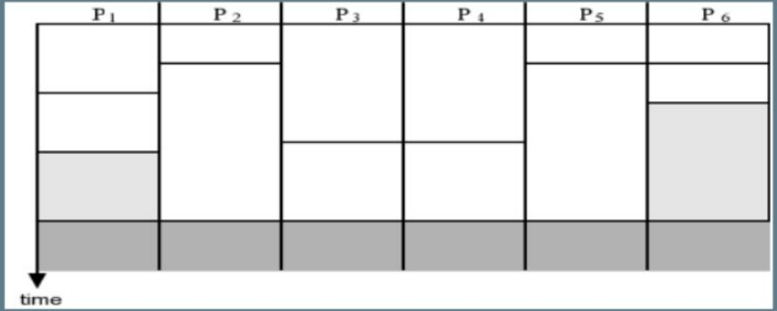
Given a set of independent tasks, where each task has an execution time, and several processors.

Task scheduling without dependence



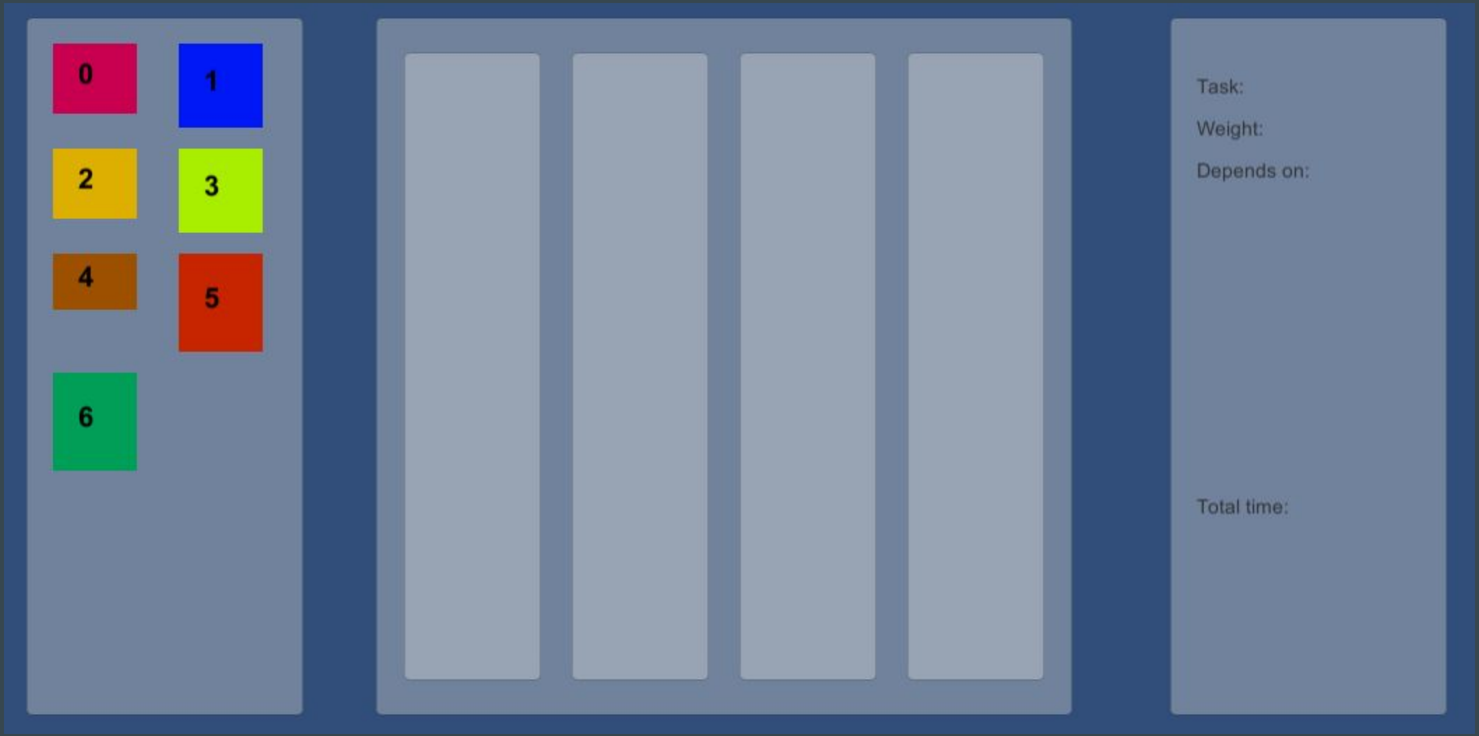
Scheduling eash task on processors.

Task scheduling without dependence



Compare with the previous one see the difference. The objective is to schedule each task on processor that allows its earliest start time.

Interactive Minigame

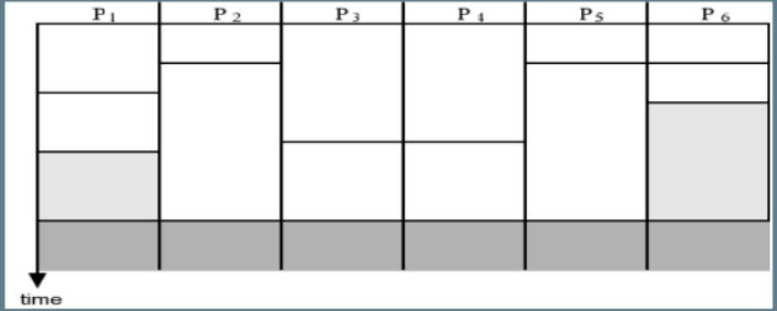


Loop Scheduling

- Introduce what is loop scheduling.

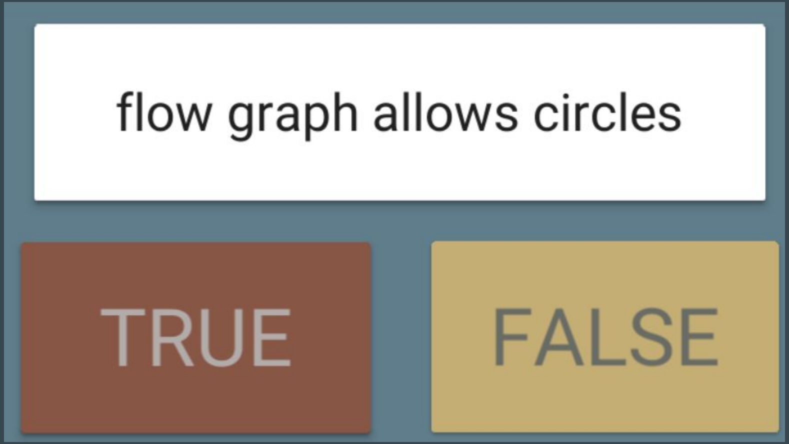
- Introduce two kinds of loop scheduling
    - Loop scheduling -- without dependence
    - Loop scheduling -- with dependence
        1. Difference between loop scheduling with dependence and task scheduling
        2. Introduce some existing techniques: Unrolling/loop body scheduling/software pipelining/loop shifting

Loop scheduling

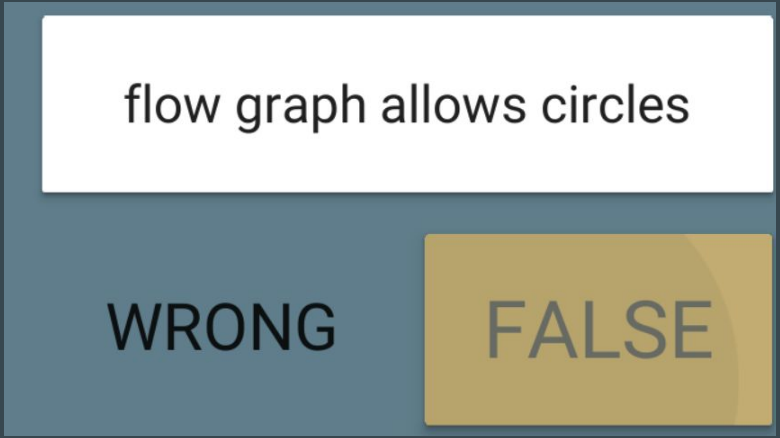


Compare with the previous one see the difference. The objective is to schedule each task on processor that allows its earliest start time.

Quiz

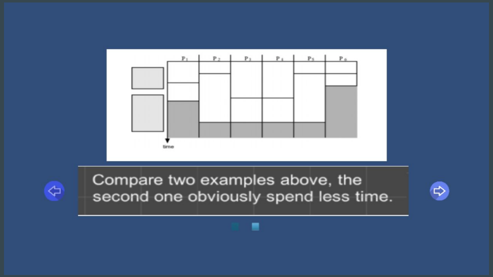


flow graph allows circles

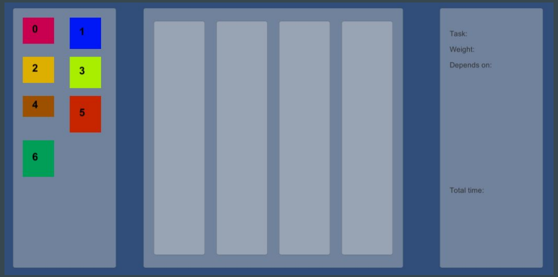


flow graph allows circles

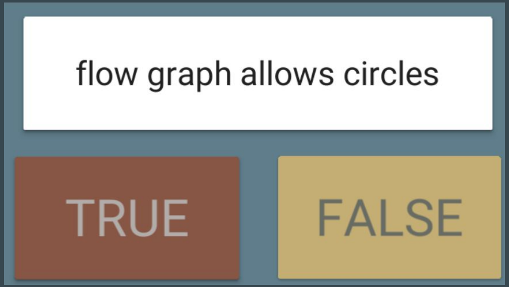
Implementation


Compare two examples above, the second one obviously spend less time.

Theory



Mini game

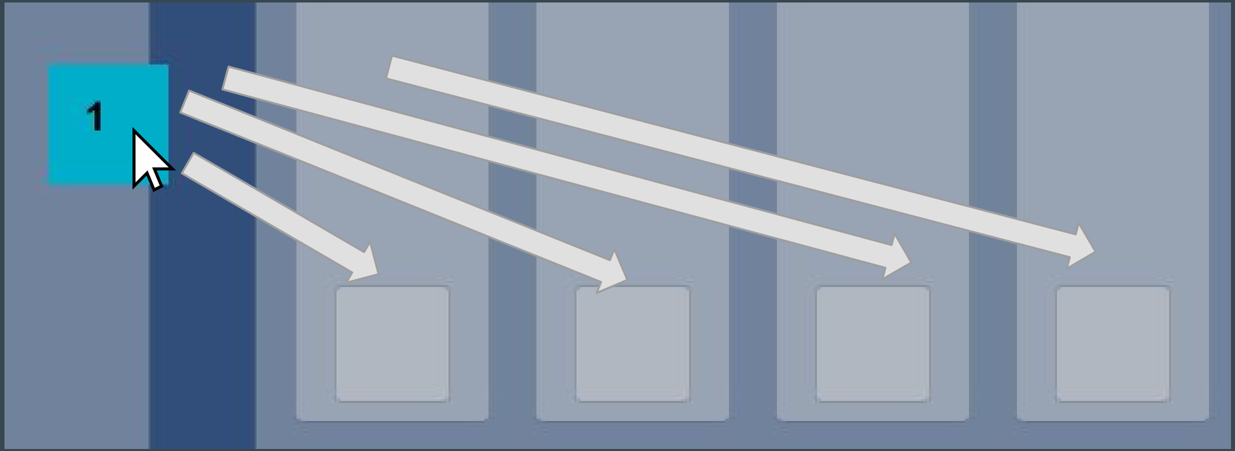

flow graph allows circles

Quiz

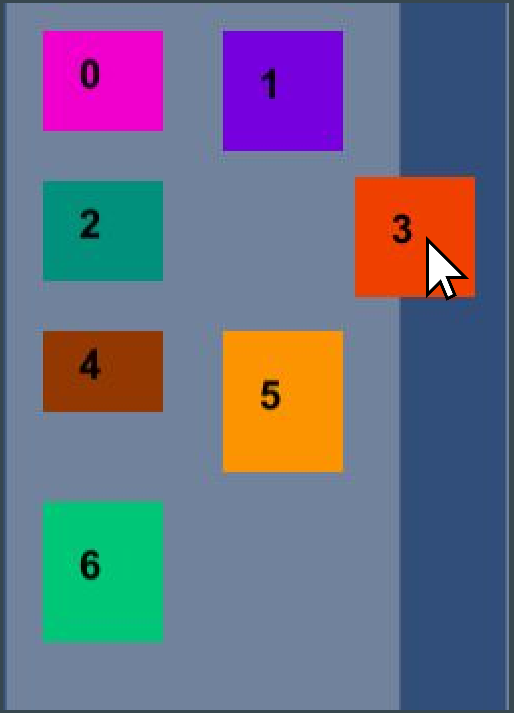
Implementation

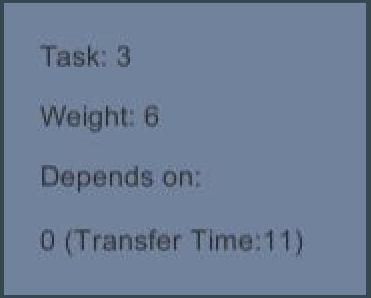
- Intuitive
- Visible
- Extensible

Drag and drop



Visiability



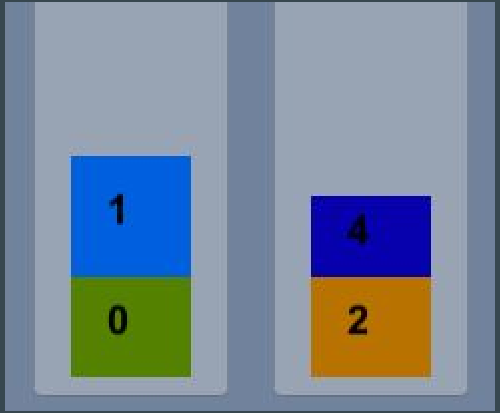

Task: 3

Weight: 6

Depends on:

0 (Transfer Time:11)

Visiability

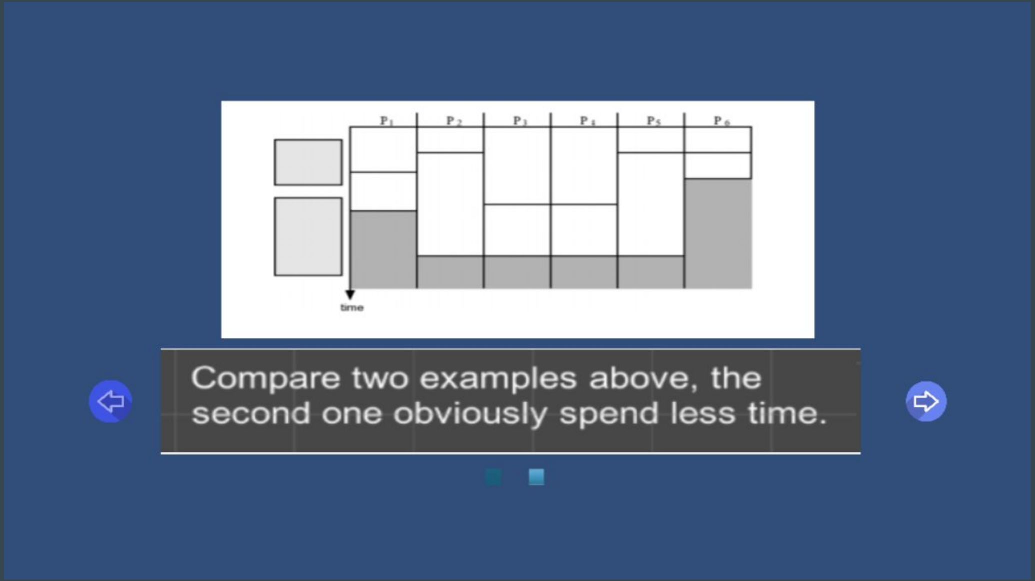


Total time: 11

Extensibility

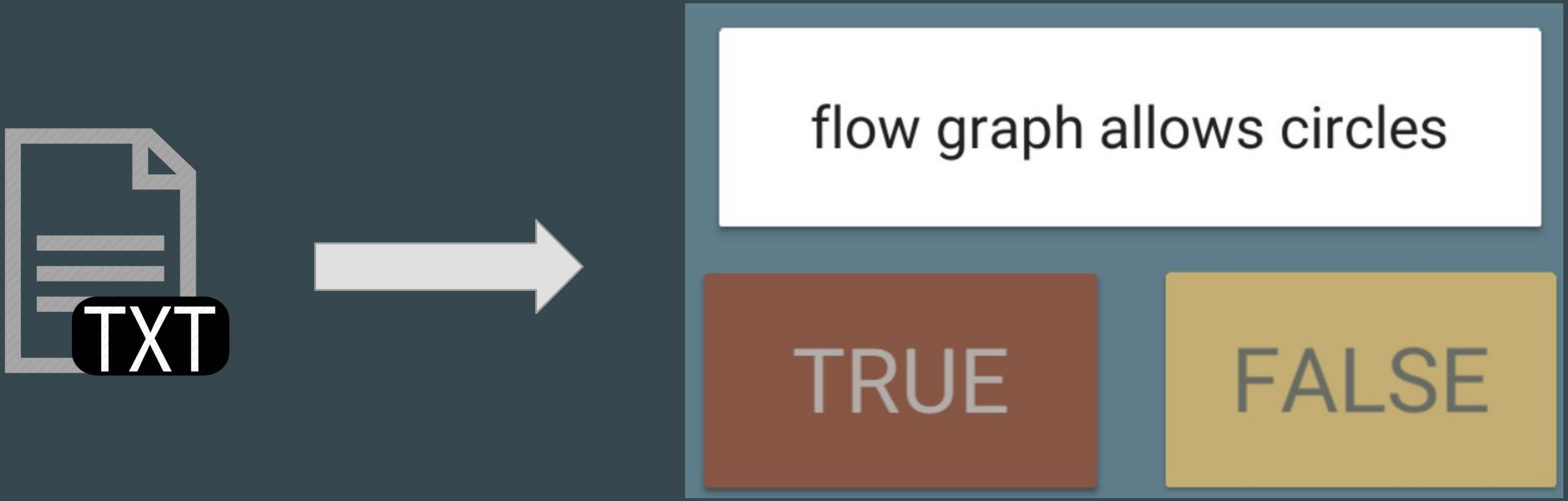
img

TXT



Compare two examples above, the second one obviously spend less time.

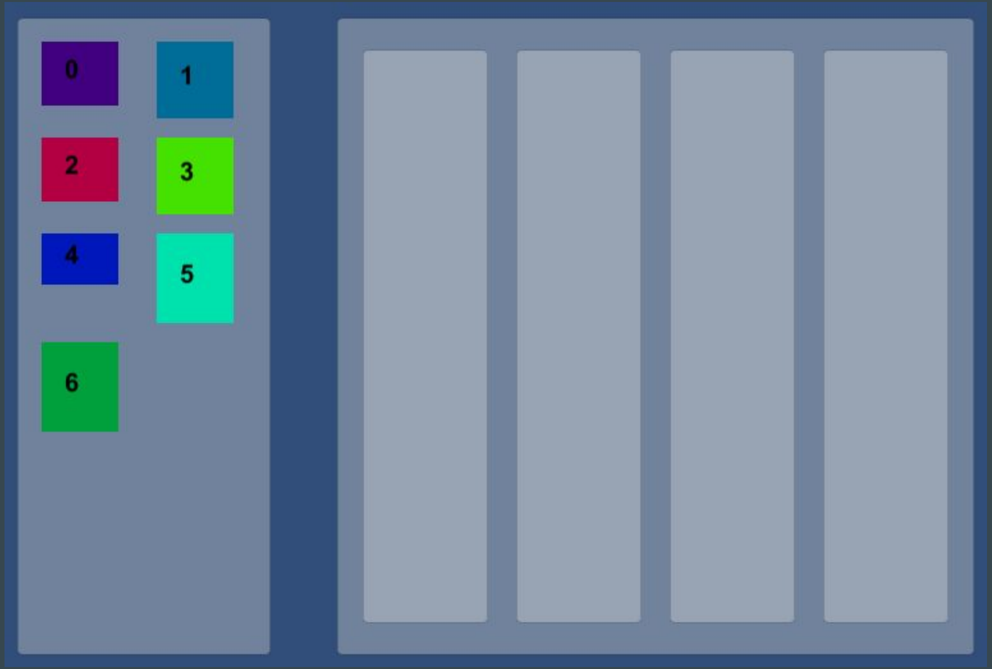
Extensibility



flow graph allows circles

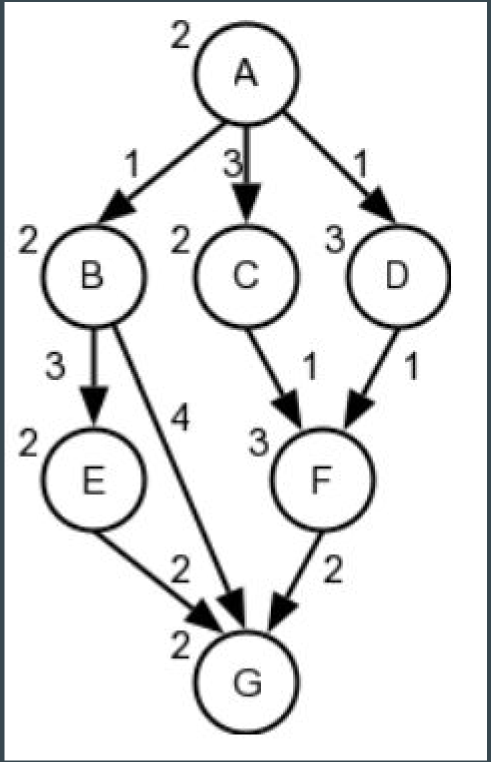
Extensibility

```
Processor   4
0       [Weight=5];
1       [Weight=6];
0 -> 1          [Weight=15];
2       [Weight=5];
0 -> 2          [Weight=11];
3       [Weight=6];
0 -> 3          [Weight=11];
4       [Weight=4];
1 -> 4          [Weight=19];
5       [Weight=7];
1 -> 5          [Weight=4];
6       [Weight=7];
1 -> 6          [Weight=21];
```



Extensibility

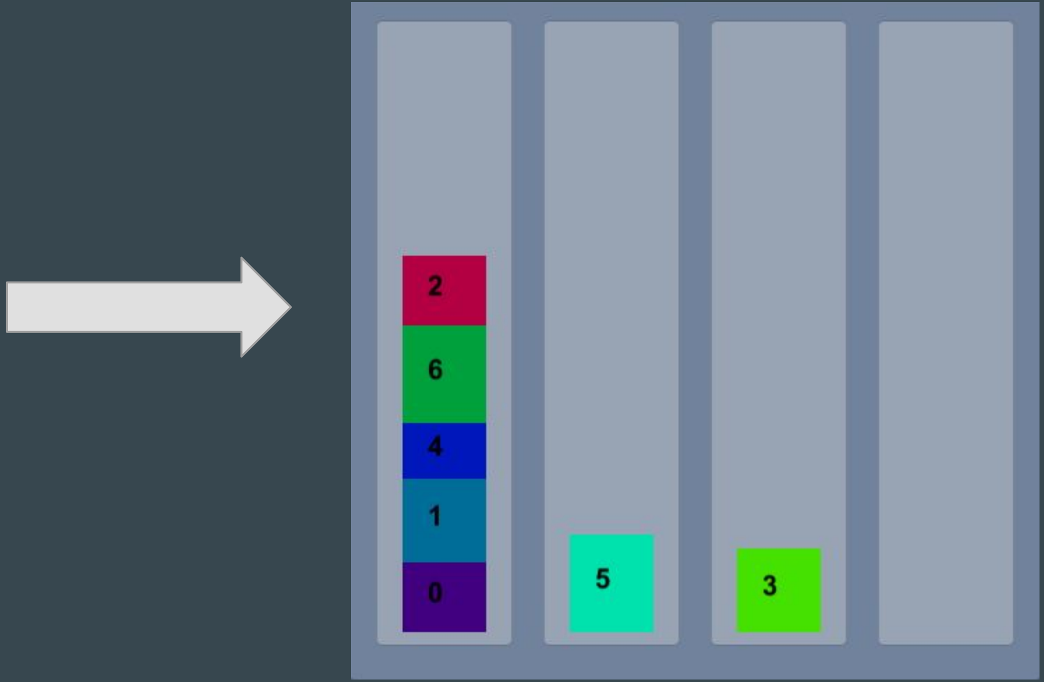
```
Processor   4
0       [Weight=5];
1       [Weight=6];
0 -> 1          [Weight=15];
2       [Weight=5];
0 -> 2          [Weight=11];
3       [Weight=6];
0 -> 3          [Weight=11];
4       [Weight=4];
1 -> 4          [Weight=19];
5       [Weight=7];
1 -> 5          [Weight=4];
6       [Weight=7];
1 -> 6          [Weight=21];
```



In progress

Extensibility

```
0 [Weight=5, Start=0, Processor=1];
1 [Weight=6, Start=5, Processor=1];
2 [Weight=5, Start=22, Processor=1];
3 [Weight=6, Start=16, Processor=3];
4 [Weight=4, Start=11, Processor=1];
5 [Weight=7, Start=15, Processor=2];
6 [Weight=7, Start=15, Processor=1];
```



Demo

Future work

- Finish implementation
- Improve UI
- Enrich theory contents and quiz bank

Question?