Assignment/Activity Design

David Bunde[1], Erik Saule[2], Kalpathi Subramanian[2]

[1]Knox College
[2]The University of North Carolina at Charlotte

IPDPS Tutorial, 2026

What makes a good assignment/activity?

- Clearly target some learning outcome.
- Naturally integrate in the class.
- Keep the students engaged.
- Well scoped for the alloted time.

Table of Contents

Engagement and Motivation

- Well understood that student engagement is an important predictor of student achievement.
- Engagement can span many dimensions[1]:
  - skills engagement
  - participation/interaction engagement
  - emotional engagement
  - performance engagement
- Engagement and motivation are closely tied to each other
- How do we motivate and engage students?
- What engagement strategies can we use?

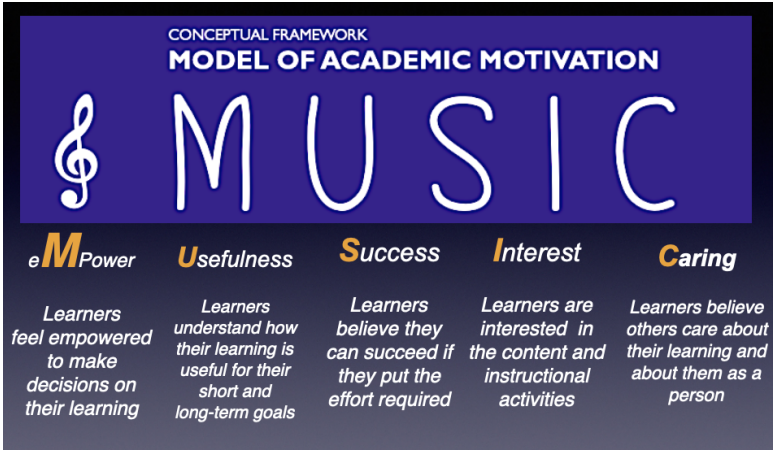
[1] Handelsman et al., A Measure of College Student Course Engagment, Journal of Educ. Res., 2005

Engagement Strategies

- **Active Learning:**
  - Pair Programming
  - Flipped classroom
  - Group work/collaboration/Light Weight Teams
  - Quizzes
- **Content Based**
  - Real world data integrated into curriculum, demonstrate relevance
  - Align with student interests, values, social relevance

BRIDGES focuses on **content based engagement**

The MUSIC Model of Engagement



[2] Jones, B.D, Motivating Students to Engage in Learning: The MUSIC Model of Academic Motivation, Intl. Journal of Teaching and Learning in Higher Ed., 2009

Engaging Students: Experiences from an OOP Course [3]

Two semesters of a project based OOP course, using student reflections after each course module

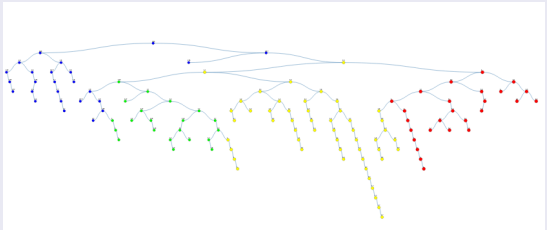
- e**M**powerment: Project choice, freedom to be creative, experimentation and tinkering
- **U**sefulness: Working with real-world data/tools, team environment
- **S**uccess: Assignments with clear instructions, predictability, reflect on personal successes/failures, feedback, challenges (in a good way!)
- **I**nterest: Fun factor, games, real world images used as part of course
- **C**aring: Sensitive to student needs, prompt feedback, deadline flexibility

[3] Subramanian et al., Influence of Course Design on Student Engagement and Motivation in an Online Course, ACM SIGCSE 2020

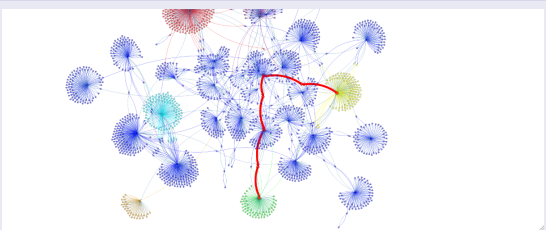
Engagement Using BRIDGES: Visual and Interactive

- BRIDGES generates **visualizations** of data structures (**that students implement!**), algorithm outputs as a mechanism for engaging students.
- Visualizations of classic CS concepts can be helpful in making them real and more meaningful.
- Student feedback has been very positive, appreciating the features of BRIDGES that enables them to **see what they code and produce**.

Indexing USGS Earthquake



Bacon Number [IMDB Data]

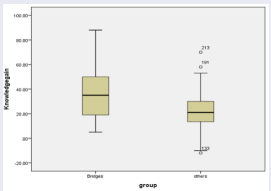


Engagement Using BRIDGES: Use Real-World Data

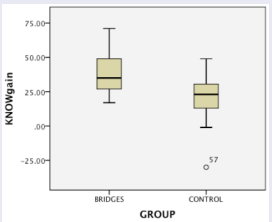
- Using **real-world data** in course work is an important engagement tool
- Students respond to working with data from real-world scenarios/data: weather/climate, maps, medical, census, books, music, videos, games
- Data is everywhere, the **harder part is**
  - Accessing data in a **ready-to-use form** for course work
  - Mapping the right data to course work to **meet learning objectives**.
- BRIDGES supports a number of datasets ready to use in early CS courses.

BRIDGES strategy works

Students in BRIDGES sections gained more knowledge

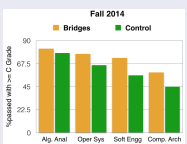


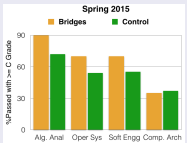
Fall 2014

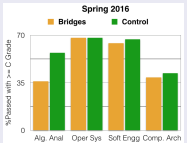


Spring 2015

Students in BRIDGES sections progressed faster in CS





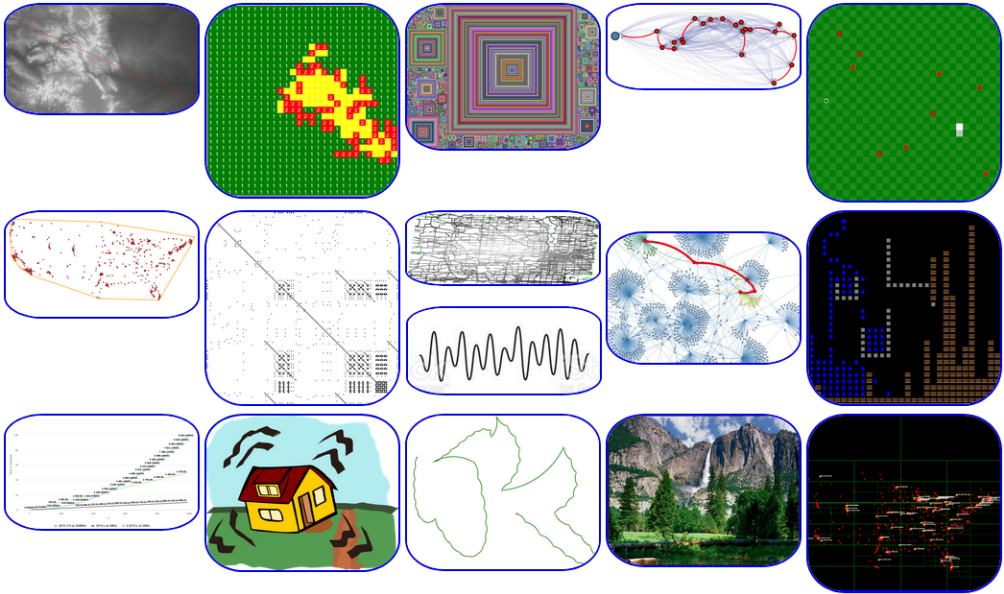


Comparing long-term student achievement between students who used the BRIDGES toolkit in the Data Structures course vs. Control group. The evaluation was performed with 3 cohorts of students (Fall 14, Spring 15, Spring 16).

Analysis performed Spring 2019.

Table of Contents

BRIDGES Gallery



Key technique: Scaffolding

Definition

Any form support designed to enable student to solve a smaller problem calibrated for the student's level and the concept they are trying to learn.
Often with the idea that the scaffolding will be removed once they know how to solve these problems.

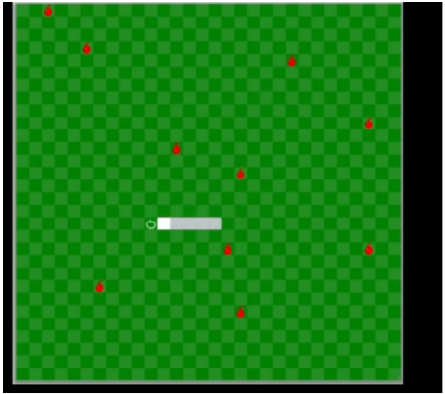
Instructional Scaffolding

Breaking the problem in piece meal tasks and steps to enable the student to come to the solution by themselves.

Code Scaffolding

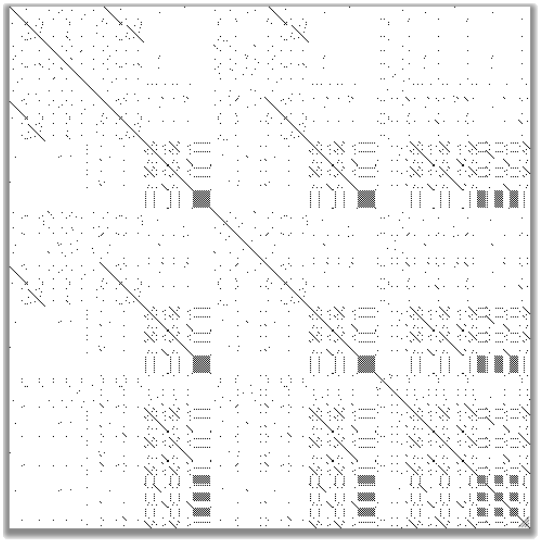
- Code skeleton
- Libraries
- Unit tests

Game API in BRIDGES



```
// this function sets the direction changes based on key press and current direction
void handleInput () {
        if (keyLeft() && dir != East && lastDir != East) {
                dir = West;
        }
        else if (keyUp() && dir != South && lastDir != South) {
                dir = North;
        }
        else if (keyDown() && dir != North && lastDir != North) {
                dir = South;
        }
        else if (keyRight() && dir != West && lastDir != West) {
                dir = East;
        }
}


// redraw
void paint() {
        // set background color
        for (int i = 0; i < numCols; ++i) {
                for (int j = 0; j < numRows; ++j) {
                        if (i % 2 == j % 2)
                                setBGColor(j, i, bg);
                        else
                                setBGColor(j, i, bc);
                }
        }

        // set snake head color
        setBGColor(head->y, head->x, hc);

        // draw the apple
        drawSymbol(apple.y, apple.x, NamedSymbol::apple, NamedColor::white);

        // the draw the bombs
        for(int i = 0; i < 5; i++) {
                drawSymbol(bomb[i].y, bomb[i].x, NamedSymbol::bomb, NamedColor::red);
        }

        // draw the snake
        Block *current = head->next;
        while (current != nullptr) {
                setBGColor(current->y, current->x, fg);
                current = current->next;
        }
}
```

Lyrics + ColorGrid in BRIDGES



```
int main(int argc, char* argv[]) {

  std::string author = "Daft Punk";
  std::string title = "Harder Faster Better Stronger";


  //Get Song data
  DataSource ds;
  Song s = ds.getSong(title, author);


  //tokenize lyrics
  auto words = lyrics_tokenize(s.getLyrics());

  int wordCount = words.size();

  //Build an empty grid
  ColorGrid grid (wordCount, wordCount);

  Color matchColor (0, 0, 0, 255);
  Color mismatchColor (255, 255, 255, 255);

  //Build repetition matrix
  for (int i = 0; i < wordCount; ++i) {
    for (int j = 0; j < wordCount; ++j) {
      if (words[i].compare(words[j]) == 0)
        grid.set(i, j, matchColor);
      else
        grid.set(i, j, mismatchColor);
    }
  }

  return 0;
}
```

Table of Contents

Exercise: PDC assignment

For the course you worked on before, design an assignment to teach some PDC in it.

- What are the learning objective for the assignment?
- What is the overall assignment?
- What are the engagement strategies?
- What gets scaffolded?
- What do the students do?