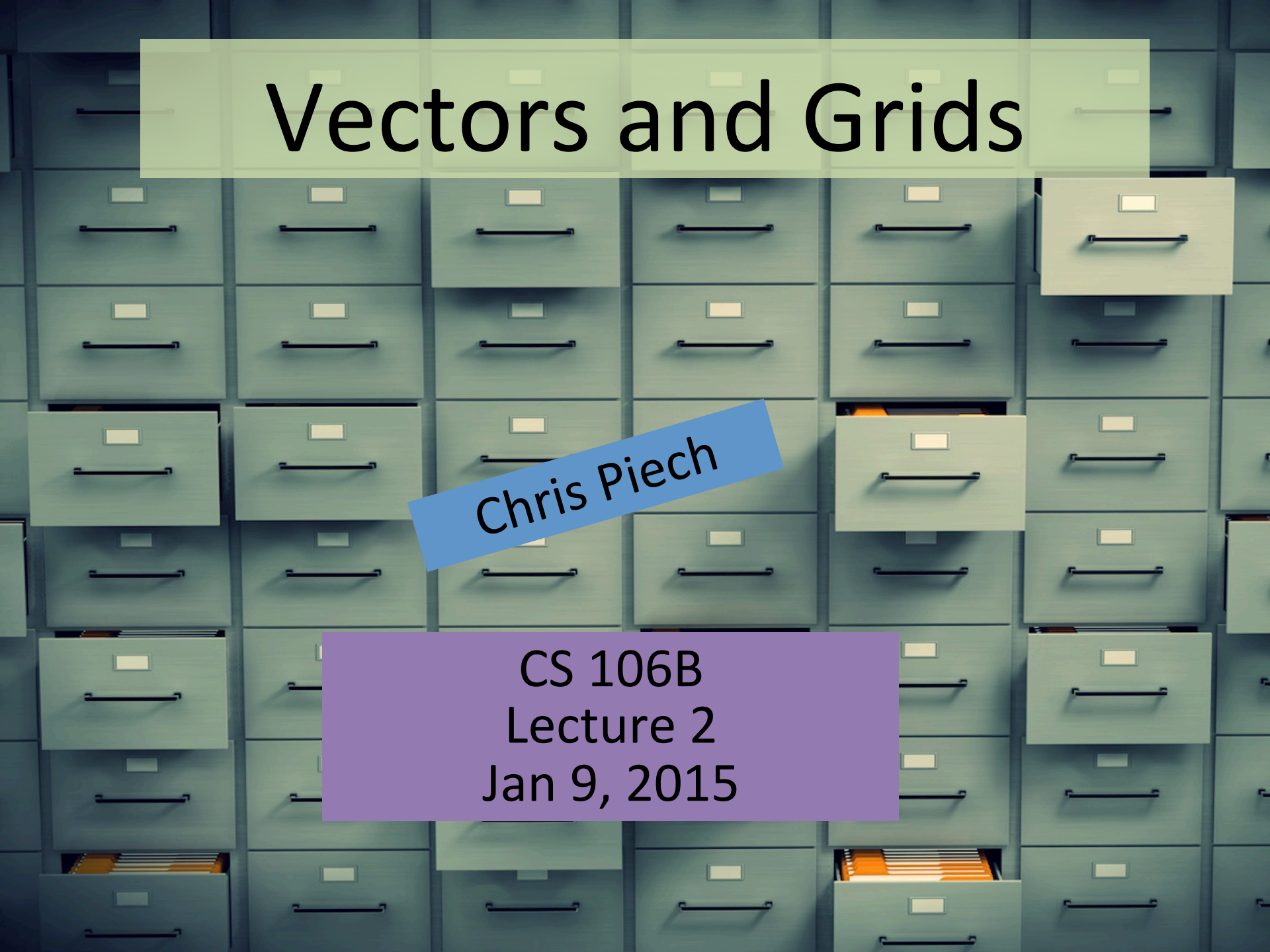


Vectors and Grids

Chris Piech

CS 106B
Lecture 2
Jan 9, 2015

Announcements

- **Honor code!**
- **Help Calendar**

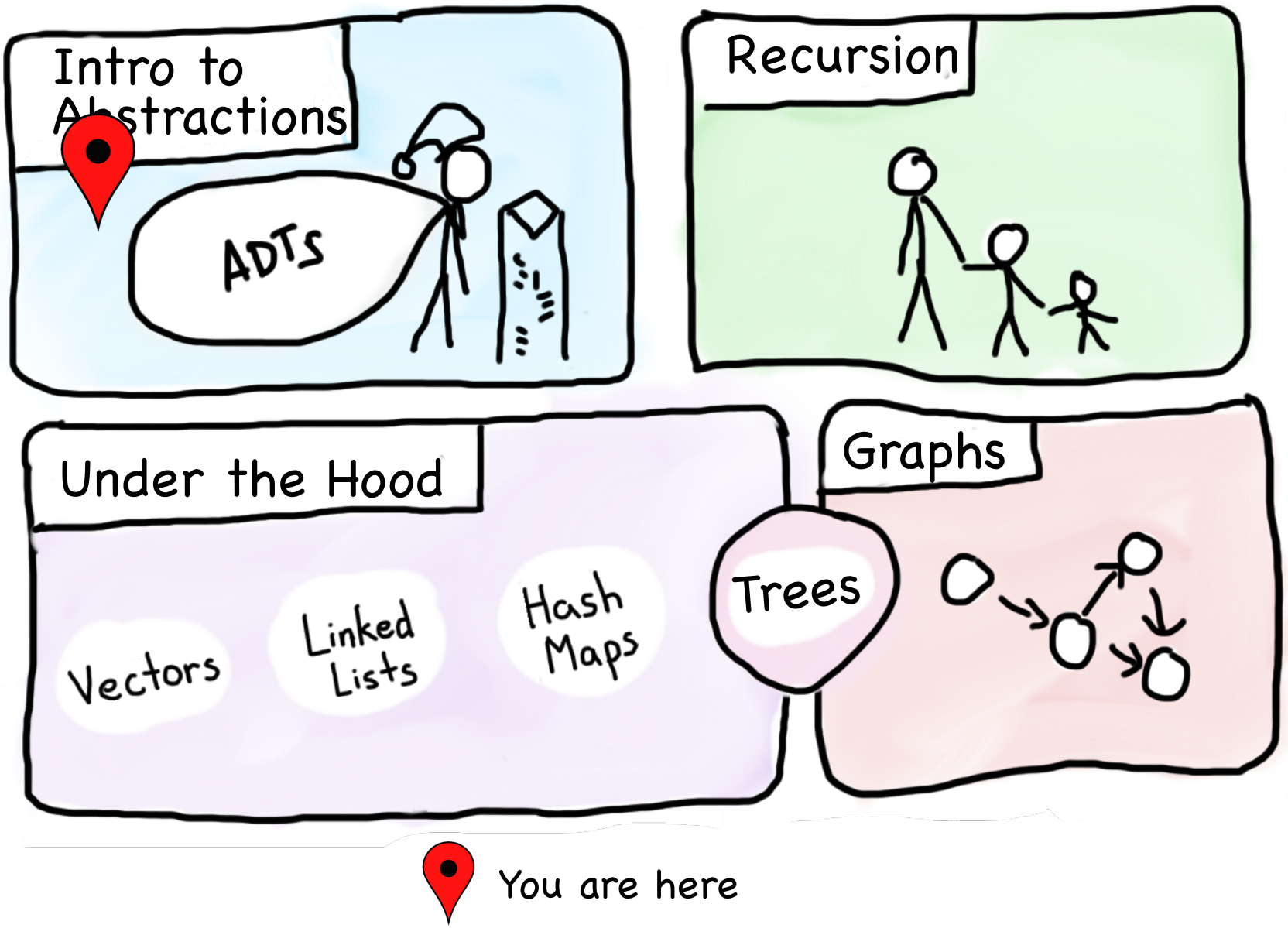
  The Life YEAH session will be Monday 5-6pm in BishopAud

  Megan Special OH on Monday 10-11am in Gates 104

  Chris Special OH today 2:30-3:30pm in Gates 193

  LaIR opens on Sunday
- **Section sign ups are open**
- **Corresponding Handout Today**

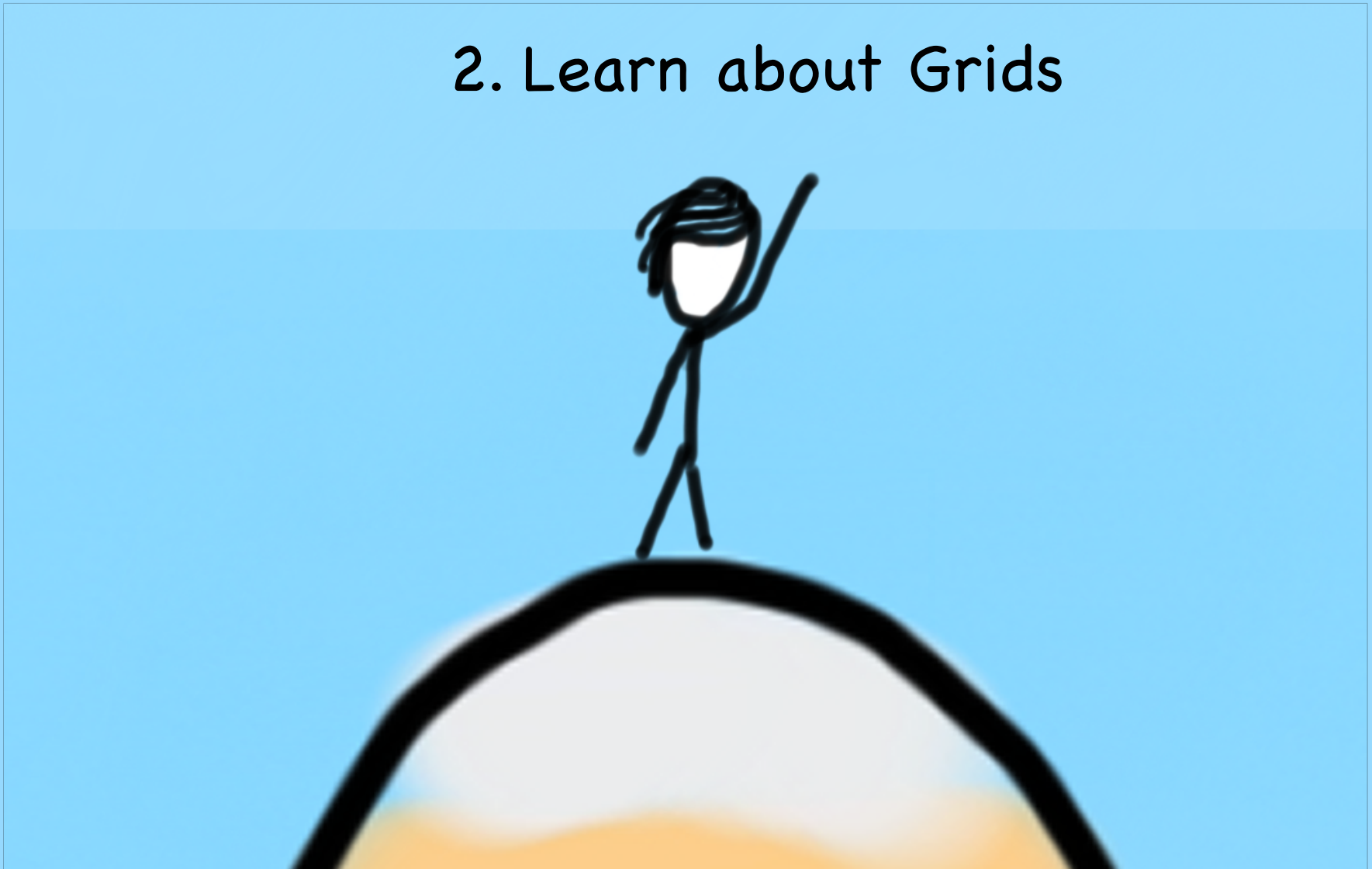
Course Syllabus



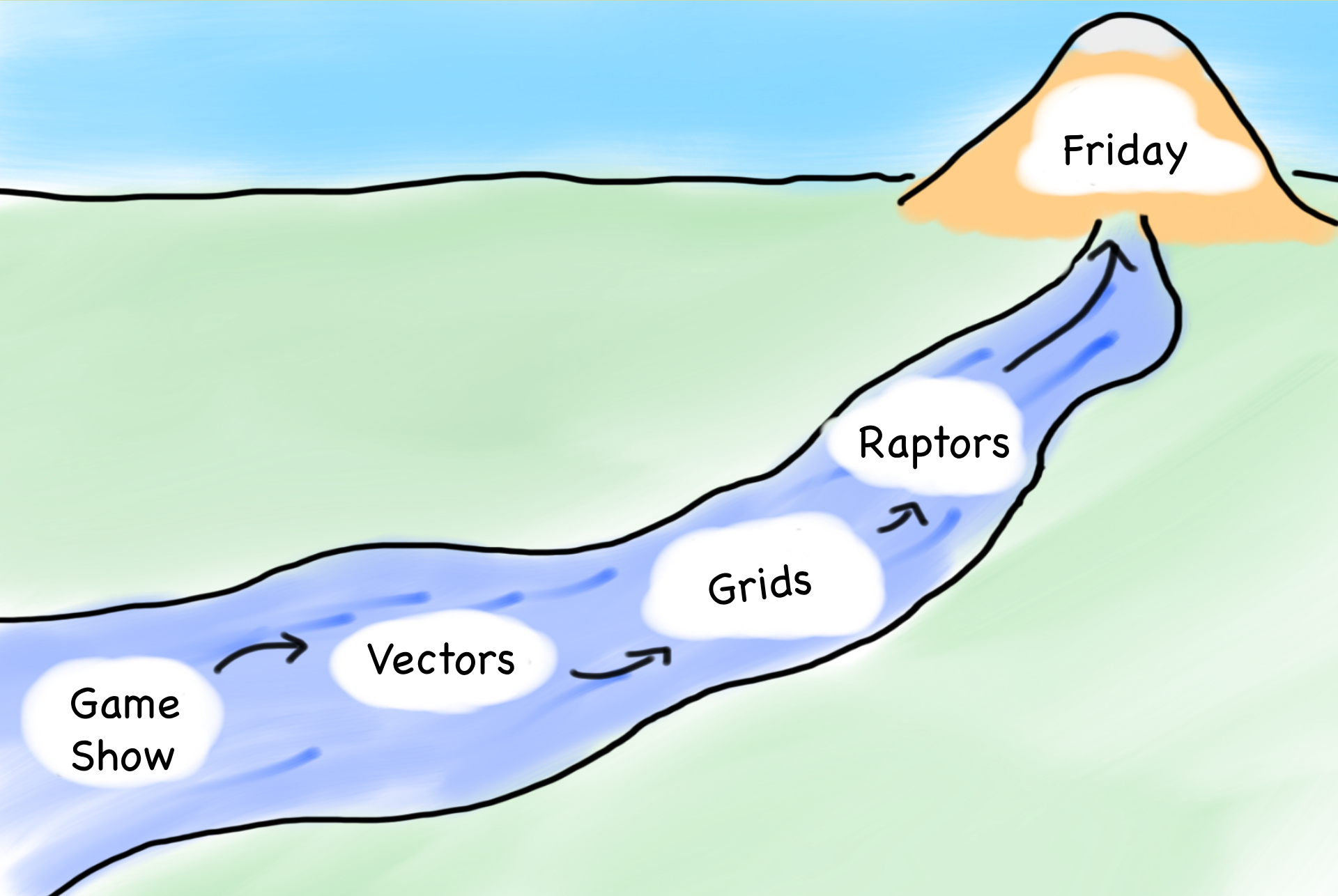
Today's Goals

1. Learn about Vectors

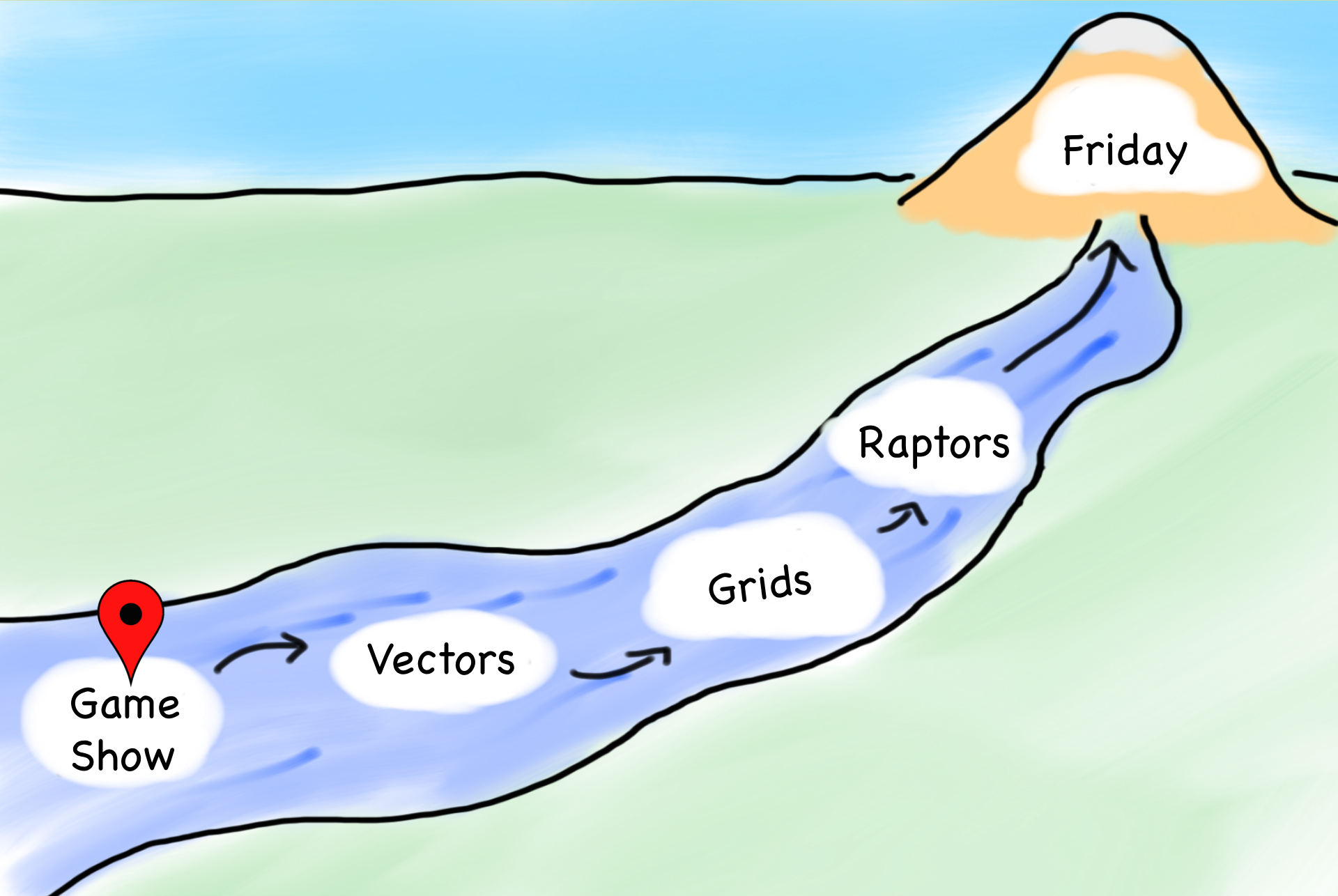
2. Learn about Grids



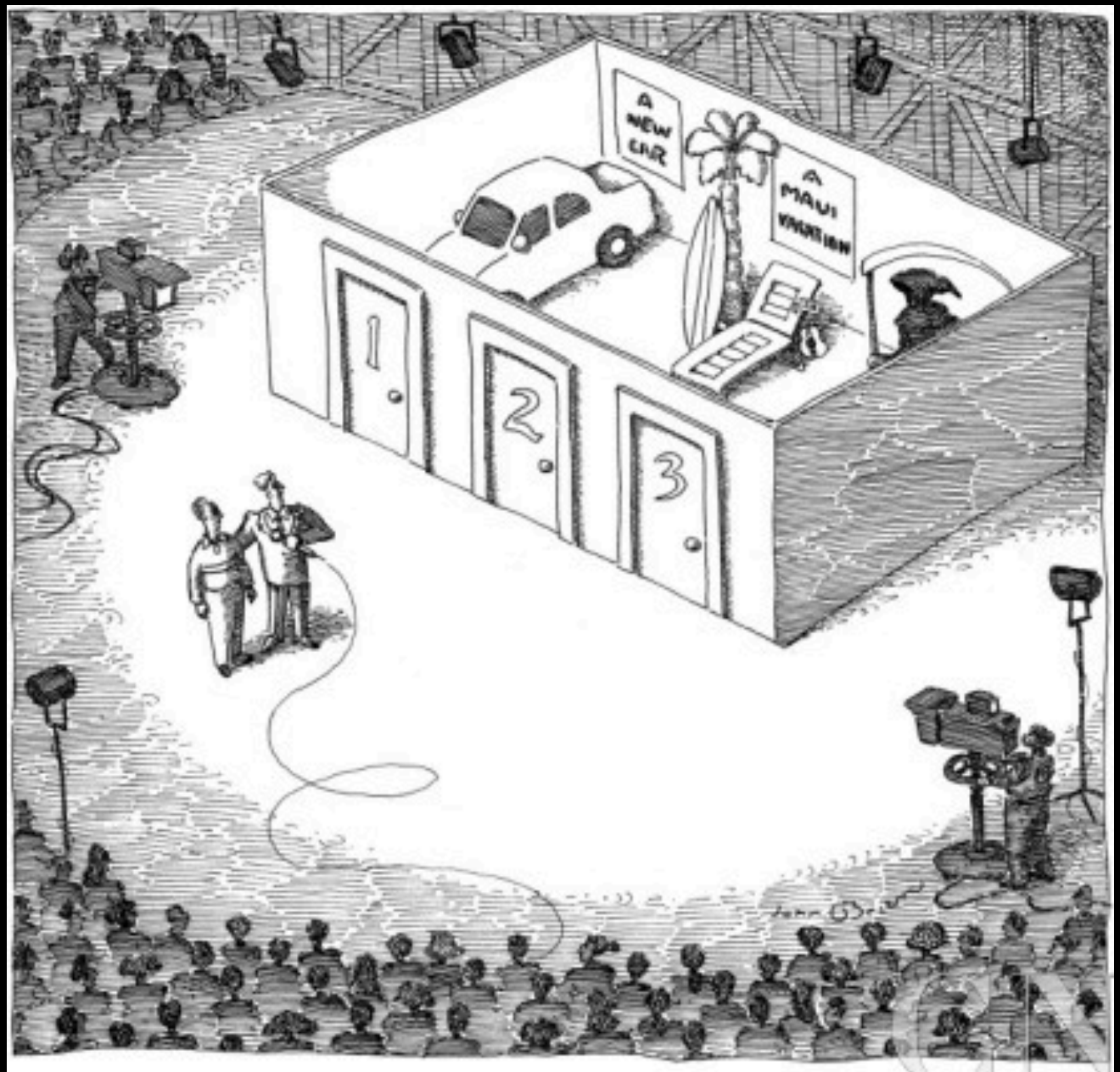
Today's Goals



Today's Goals



CS106B Game Show



Helper Function

```
int getChoice() {
    string prompt = "Which door would you like to open (1
    while(true) {
        int choice = getInteger(prompt);
        if(choice < 1 || choice > 3) {
            cout << "Illegal door. Try again." << endl;
        } else {
            return choice;
        }
    }
}
```

Helper Function

Function that returns an integer

```
int getChoice() {
    string prompt = "Which door would you like to open (1
    while(true) {
        int choice = getInteger(prompt);
        if(choice < 1 || choice > 3) {
            cout << "Illegal door. Try again." << endl;
        } else {
            return choice;
        }
    }
}
```

Helper Function

```
int getChoice() {
    string prompt = "Which door would you like to open (1
    while(true) {
        int choice = getInteger(prompt);
        if(choice < 1 || choice > 3) {
            cout << "Illegal door. Try again." << endl;
        } else {
            return choice;
        }
    }
}
```

Helper Function

```
int getChoice() {
    string prompt = "Which door would you like to open (1
    while(true) {
        int choice = getInteger(prompt);
        if(choice < 1 || choice > 3) {
            cout << "Illegal door. Try again." << endl;
        } else {
            return choice;
        }
    }
}
```

Useful Stanford Library function

Helper Function

```
int getChoice() {
    string prompt = "Which door would you like to open (1
    while(true) {
        int choice = getInteger(prompt);
        if(choice < 1 || choice > 3) {
            cout << "Illegal door. Try again." << endl;
        } else {
            return choice;
        }
    }
}
```

Welcome Message in a File

welcome.txt

```
6
Welcome to the CS106B game show!
You stand infront of three doors
and behind each door is a special
prize.
Will you be brave?
Will you be wise?
Step right up and try your luck.
```

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Creates a file stream variable

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Opens the file "welcome.txt"

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Declares a string

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Puts the next line in the file into the string

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Interprets the first line as an int

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Loop numLines times

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Each time read another line from the file and cout it

Another Helper Function

```
void setUpGame() {
    ifstream fileStream;
    openFile(fileStream, "welcome.txt");

    // get first line
    string numLinesStr;
    getline(fileStream, numLinesStr);
    int numLines = stringToInteger(numLinesStr);

    // output the welcome message
    for(int i = 0; i < numLines; i++) {
        string line;
        getline(fileStream, line);
        cout << line << endl;
    }
}
```

Another Helper Function

```
void suspense() {
    cout << endl << "Dumroll!" << endl;
    for(int i = 0; i < 10; i++) {
        string line = "";
        for(int j = 0; j < (10 - i); j++) {
            line += ".";
        }
        cout << line << endl;
        pause(200);
    }
}
```

CS106B Game Show

```
int main() {
    setUpGame();
    string prize = "some candy";

    int choice = getChoice();
    if(choice == 1) {
        doorOne(prize);
    } else if(choice == 2) {
        doorTwo(prize);
    } else if(choice == 3) {
        doorThree(prize);
    }

    suspense();
    cout << "You win " << prize << endl;
    return 0;
}
```

The Doors

```
void doorOne(string & prize) {
    int dollars = 1 / 5 * 100;
    prize = "$" + integerToString(dollars);
}

void doorTwo(string prize) {
    prize = "a Maasai rungu";
}

void doorThree(string & prize) {
    prize = "a pineapple";
}
```

Volunteer

The Doors

```
void doorOne(string & prize) {
    int dollars = 1 / 5 * 100;
    prize = "$" + integerToString(dollars);
}

void doorTwo(string prize) {
    prize = "a Maasai rungu";
}

void doorThree(string & prize) {
    prize = "a pineapple";
}
```

The Doors

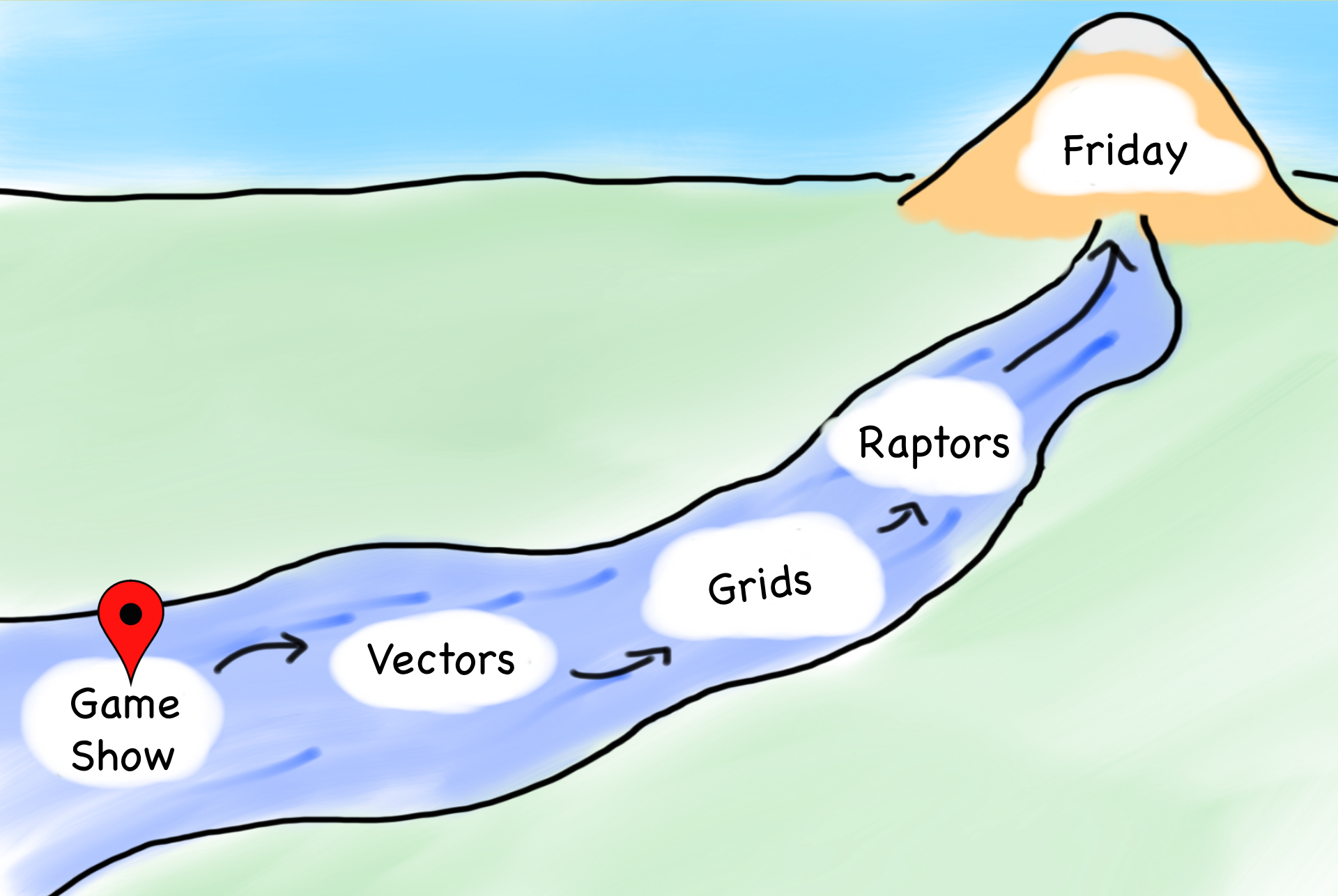
```
void doorOne(string & prize) {
    int dollars = 1 / 5 * 100;
    prize = "$" + integerToString(
}

void doorTwo(string prize) {
    prize = "a Maasai rungu";
}

void doorThree(string & prize) {
    prize = "a pineapple";
}
```

Integer divided by an integer results in an integer… which is floored

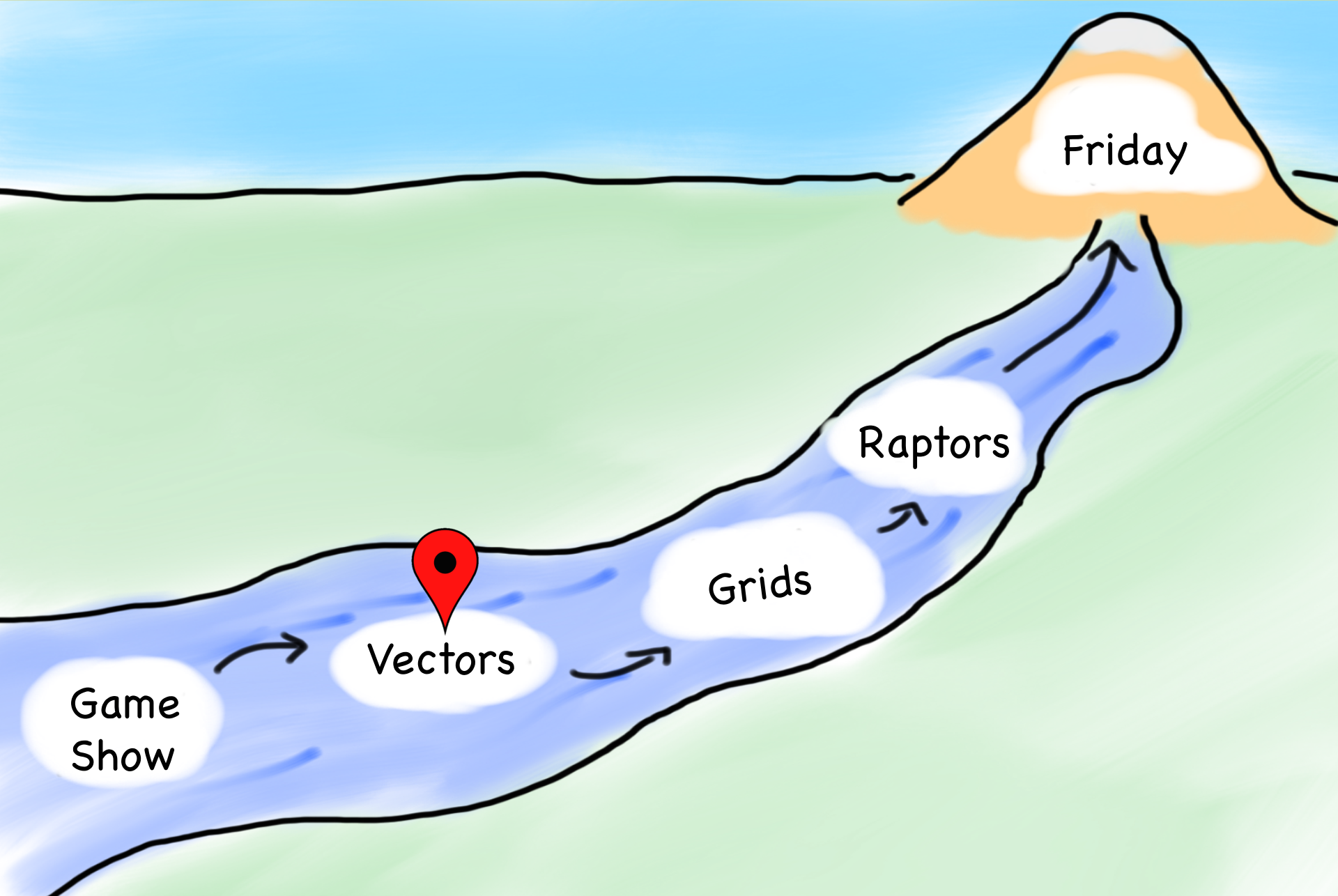
Not passed by reference. Changes don't persist.

Pineapples are delicious and healthy.

Today's Goals

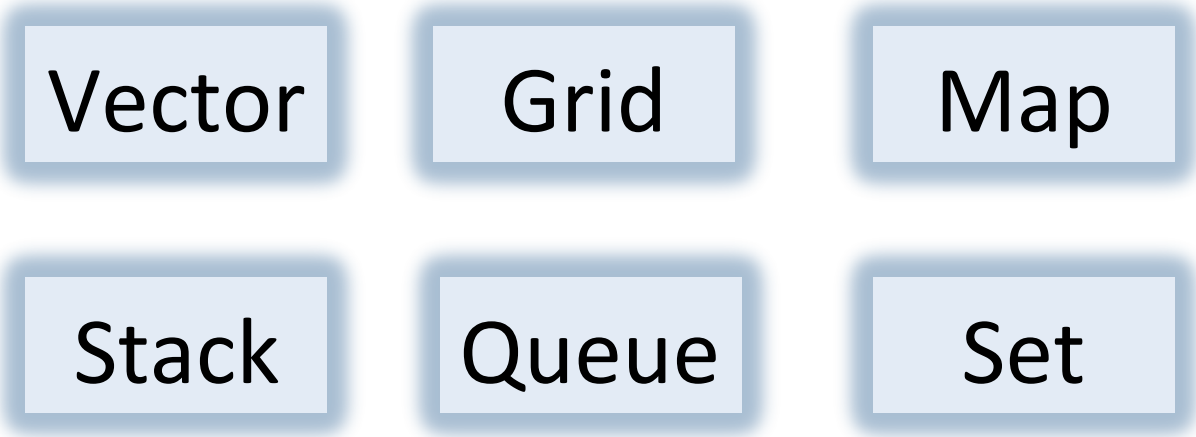


Today's Goals



Collections Lecture 1

Collections



Collections

Vector

Grid

Map

Stack

Queue

Set

Collections

Vector

Vector<type>

What is it?

- ArrayList<*type*>
- A list of elements that can grow and shrink. Each element has a place (or index) in the list.
- Advanced array.

Important Details

- Constructor creates an empty list.
- Indexed by 0.
- Bounds checks.
- Knows its size.

Why not use arrays?

Vector Creation

```
Vector<int> vec;
```

or

```
Vector<int> vec();
```

Vector Methods

`vec.size()` Returns the number of elements in the vector.
`vec.isEmpty()` Returns `true` if the vector is empty.
`vec[i]` Selects the i^{th} element of the vector.
`vec.add(value)` Adds a new element to the end of the vector.
`vec.insert(index, value)` Inserts the value before the specified index position.
`vec.remove(index)` Removes the element at the specified index.
`vec.clear()` Removes all elements from the vector.

For the exhaustive list check out http://stanford.edu/~stepp/cppdoc/Vector-class.html

Vector Example

```
Vector<int> magic;
magic.add(4);
magic.add(8);
magic.add(15);
magic.add(16);
cout << magic[2] << endl;
```

Vector Example

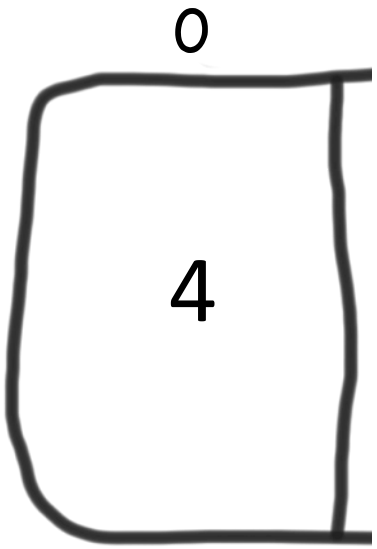
```
Vector<int> magic;
magic.add(4);
magic.add(8);
magic.add(15);
magic.add(16);
cout << magic[2] << endl;
```

magic:

Vector Example

```
Vector<int> magic;
magic.add(4);
magic.add(8);
magic.add(15);
magic.add(16);
cout << magic[2] << endl;
```

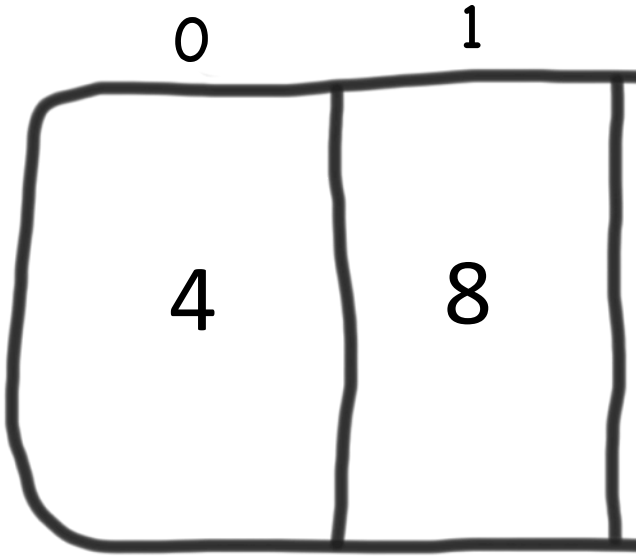
magic:



Vector Example

```
Vector<int> magic;
magic.add(4);
magic.add(8);
magic.add(15);
magic.add(16);
cout << magic[2] << endl;
```

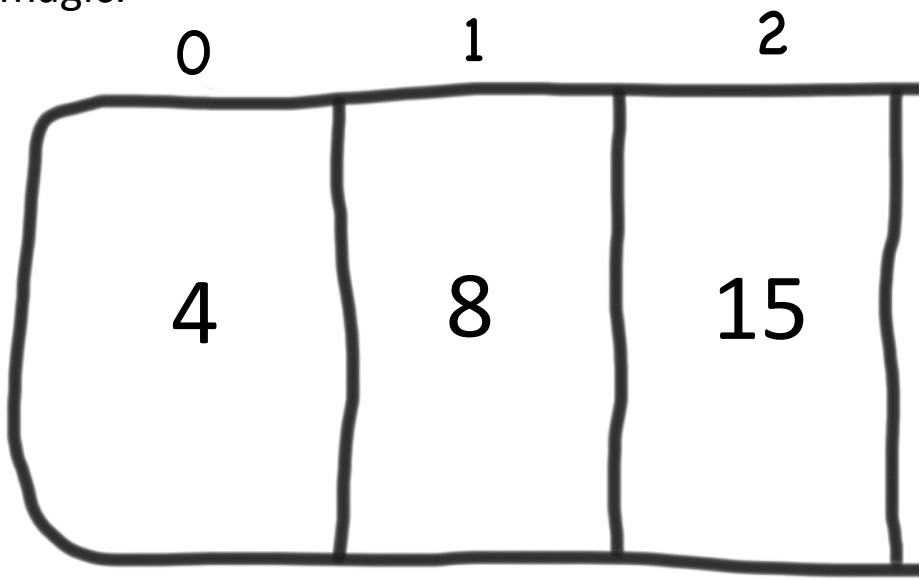
magic:



Vector Example

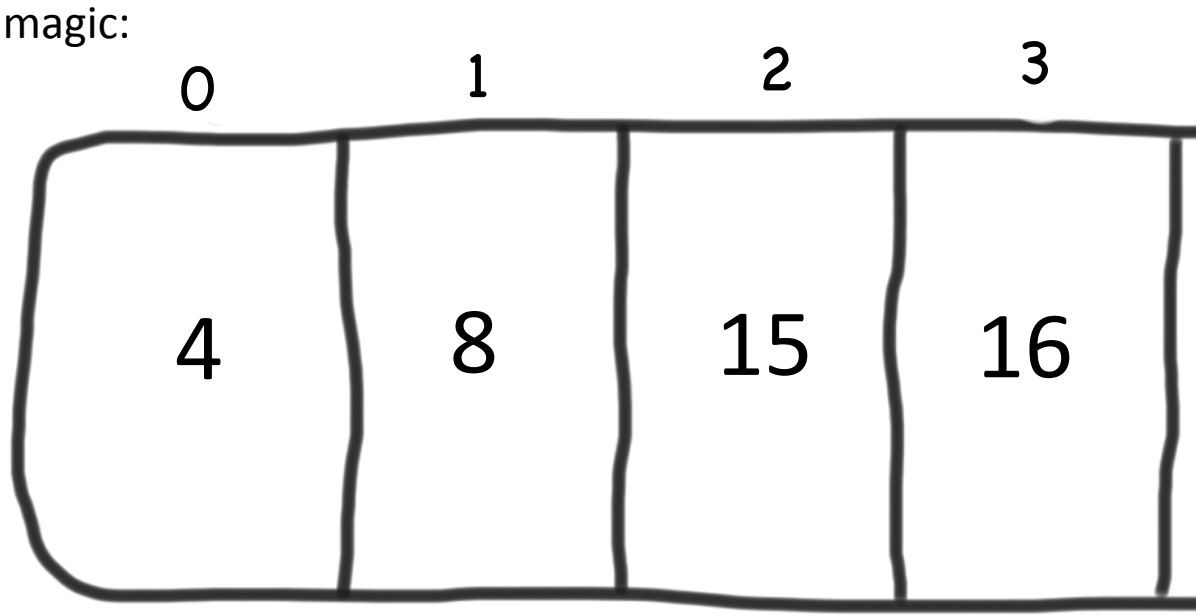
```
Vector<int> magic;
magic.add(4);
magic.add(8);
magic.add(15);
magic.add(16);
cout << magic[2] << endl;
```

magic:



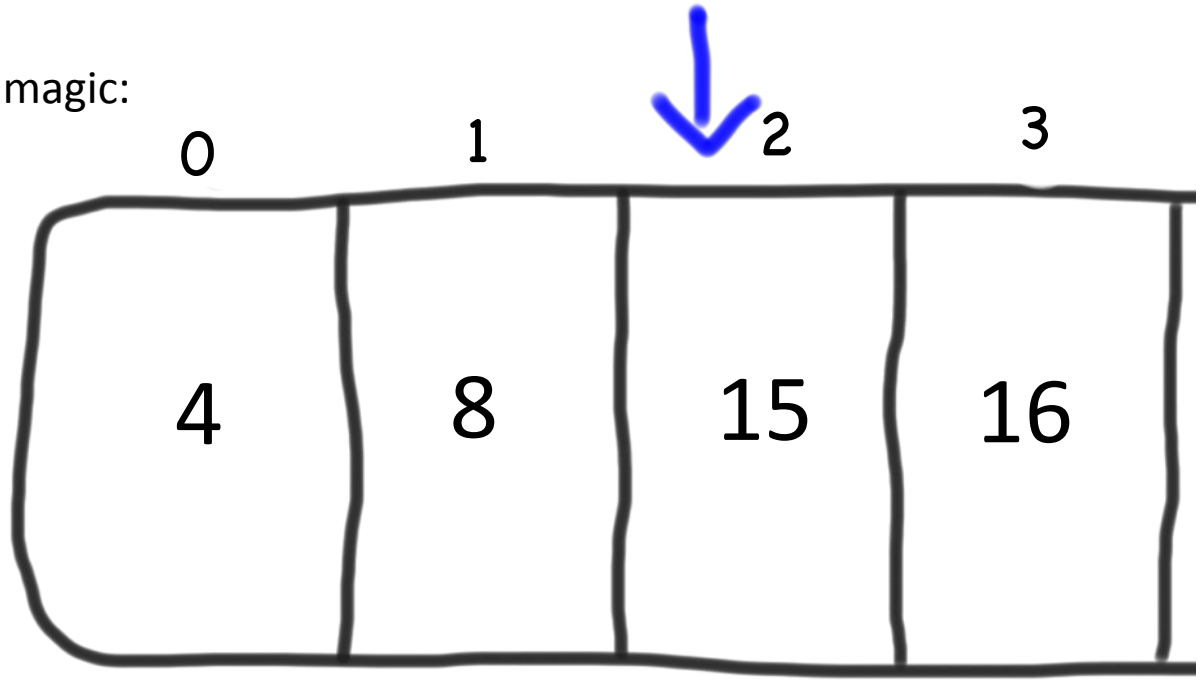
Vector Example

```
Vector<int> magic;
magic.add(4);
magic.add(8);
magic.add(15);
magic.add(16);
cout << magic[2] << endl;
```



Vector Example

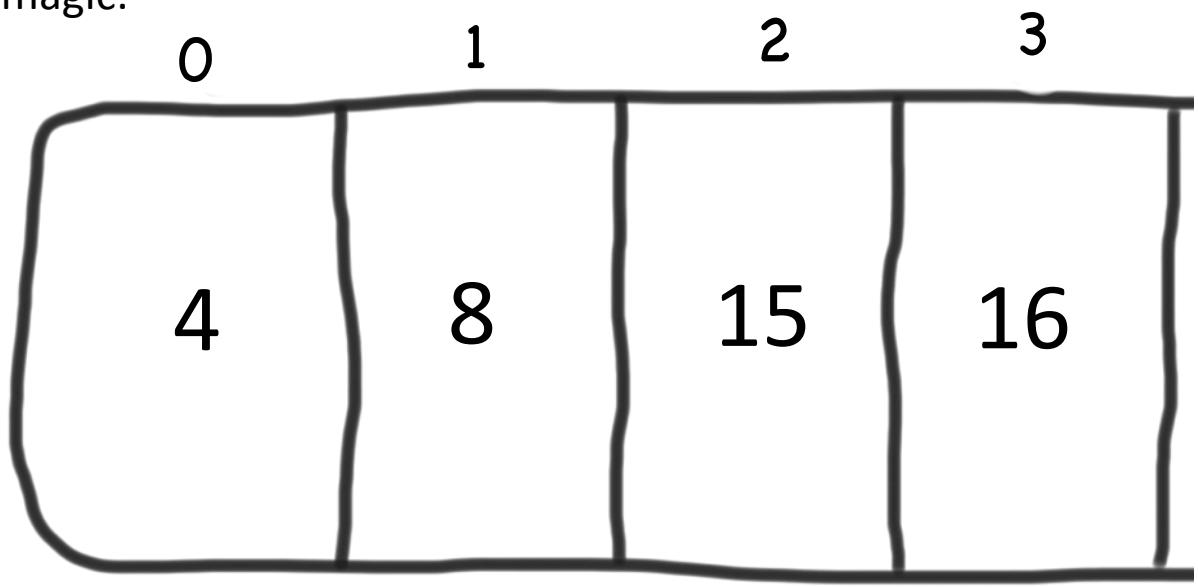
```
Vector<int> magic;
magic.add(4);
magic.add(8);
magic.add(15);
magic.add(16);
cout << magic[2] << endl;
```



Vector Example

```
for(int i = 0; i < magic.length(); i++) {
    cout << magic[i]
}
```

magic:

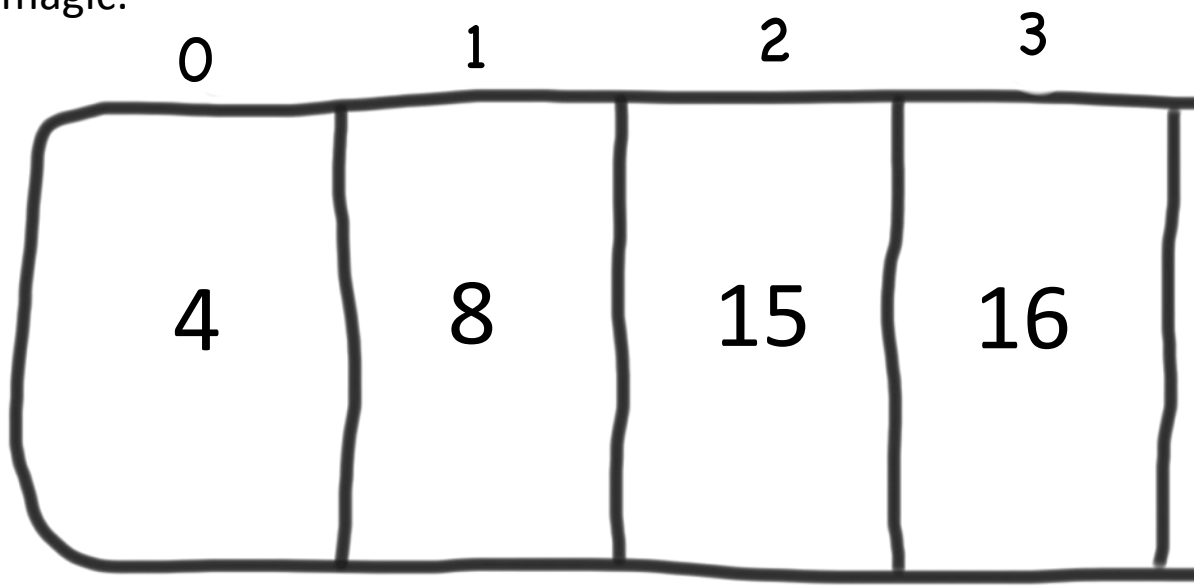


Vector Example

4

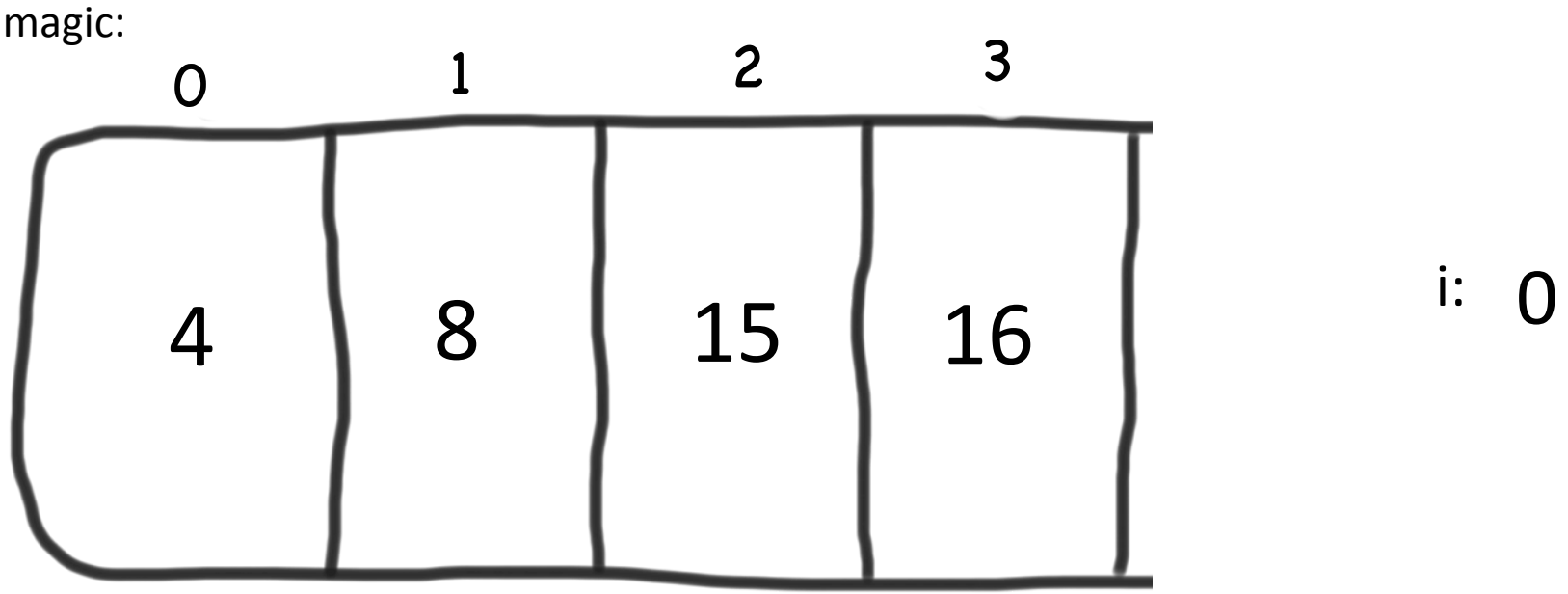
```
for(int i = 0; i < magic.length(); i++) {
    cout << magic[i]
}
```

magic:



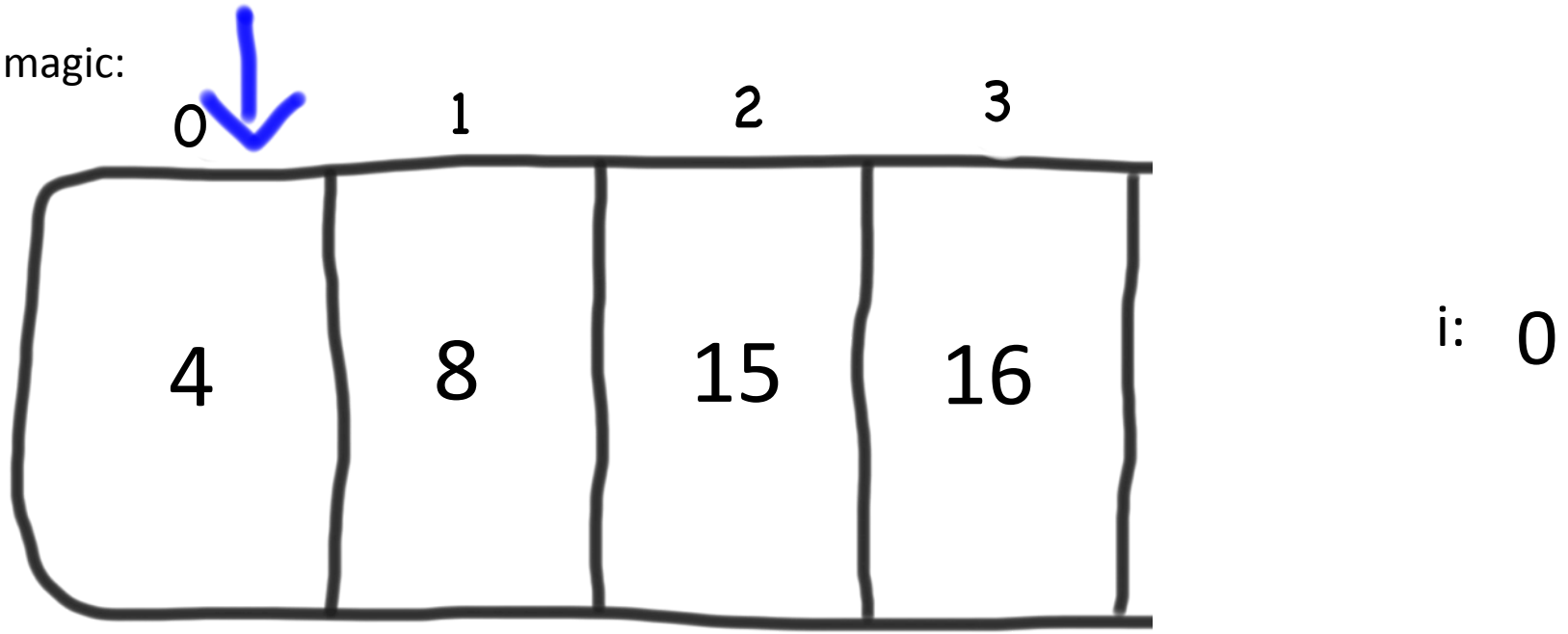
Vector Example

```
for(int i = 0; i < magic.length(); i++) {
    cout << magic[i]
}
```



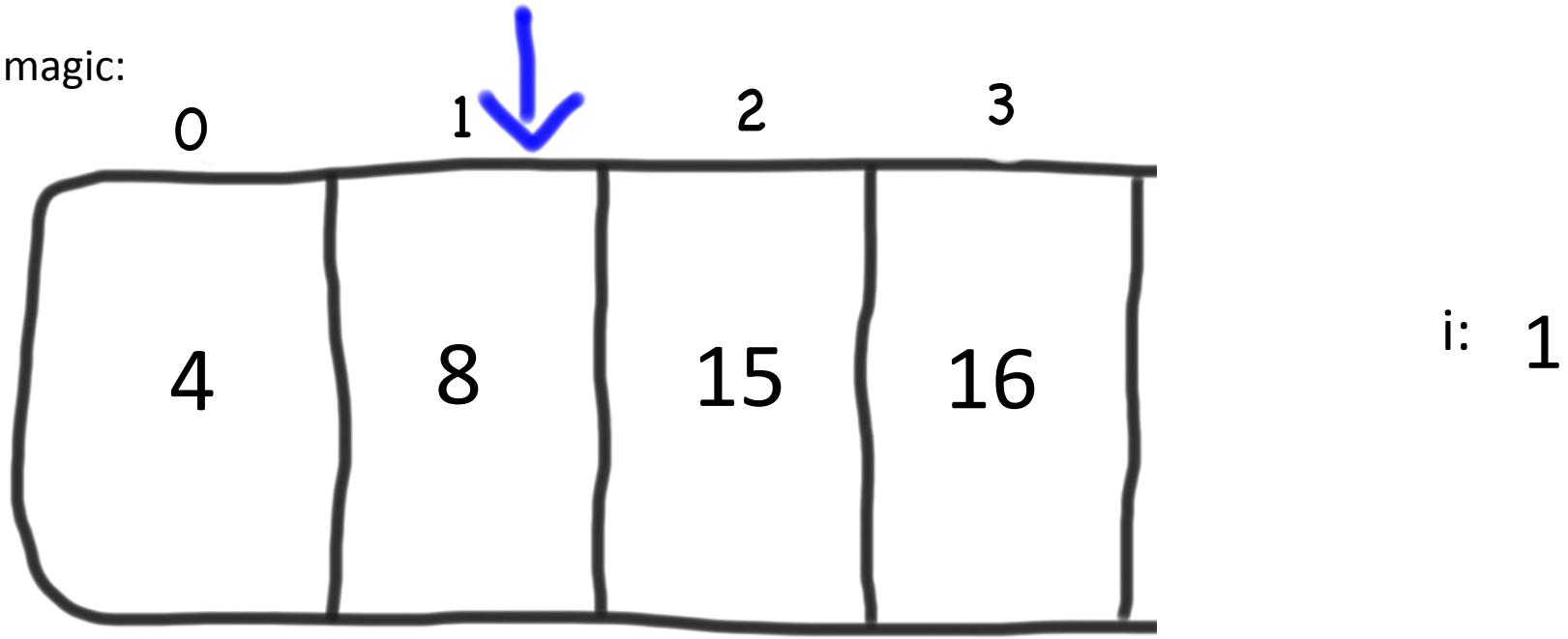
Vector Example

```
for(int i = 0; i < magic.length(); i++) {
    cout << magic[i]
}
```



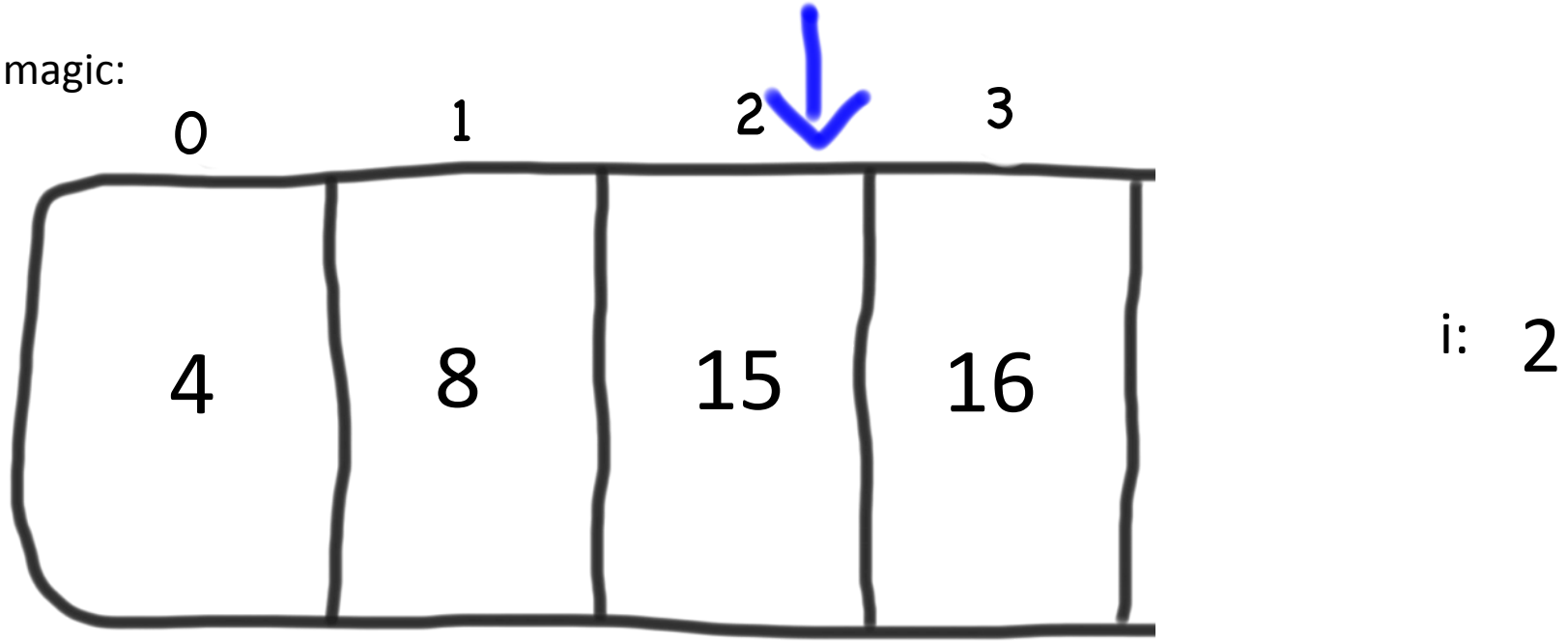
Vector Example

```
for(int i = 0; i < magic.length(); i++) {
    cout << magic[i]
}
```



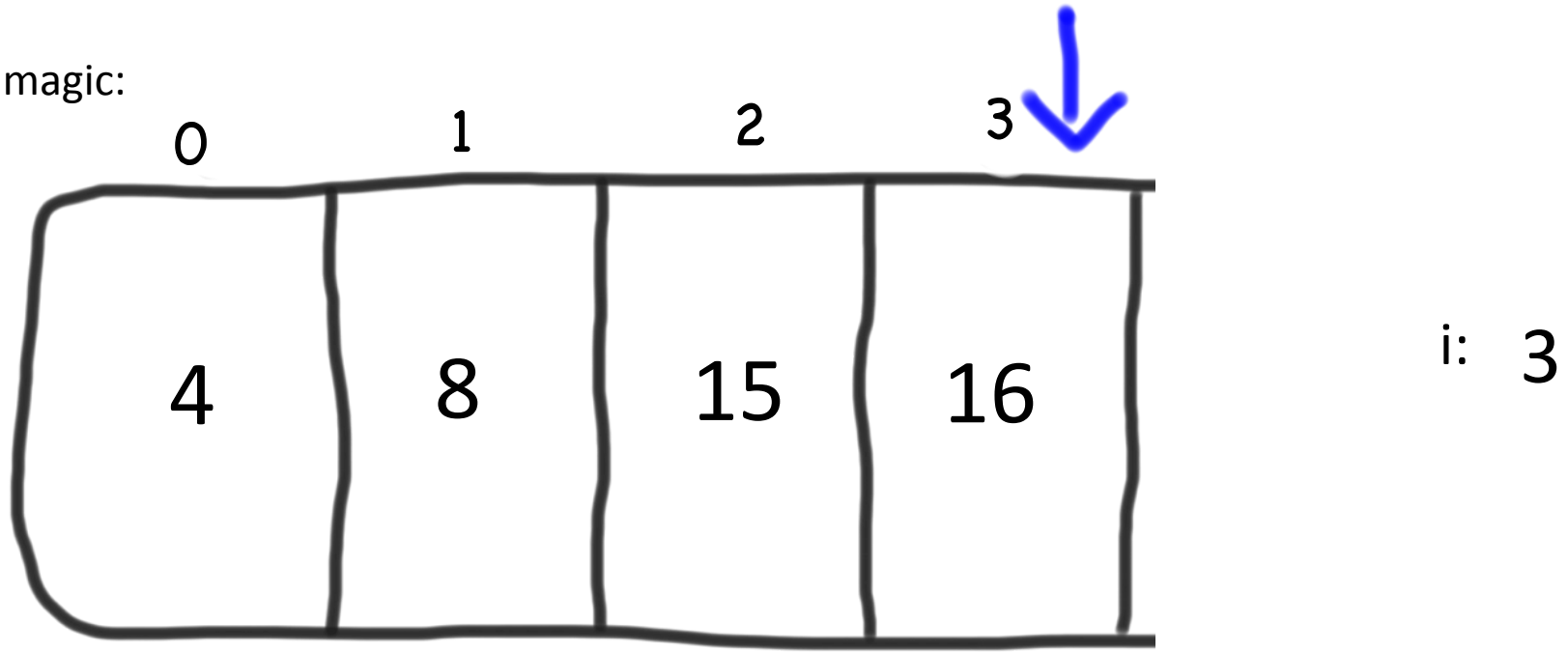
Vector Example

```
for(int i = 0; i < magic.length(); i++) {
    cout << magic[i]
}
```

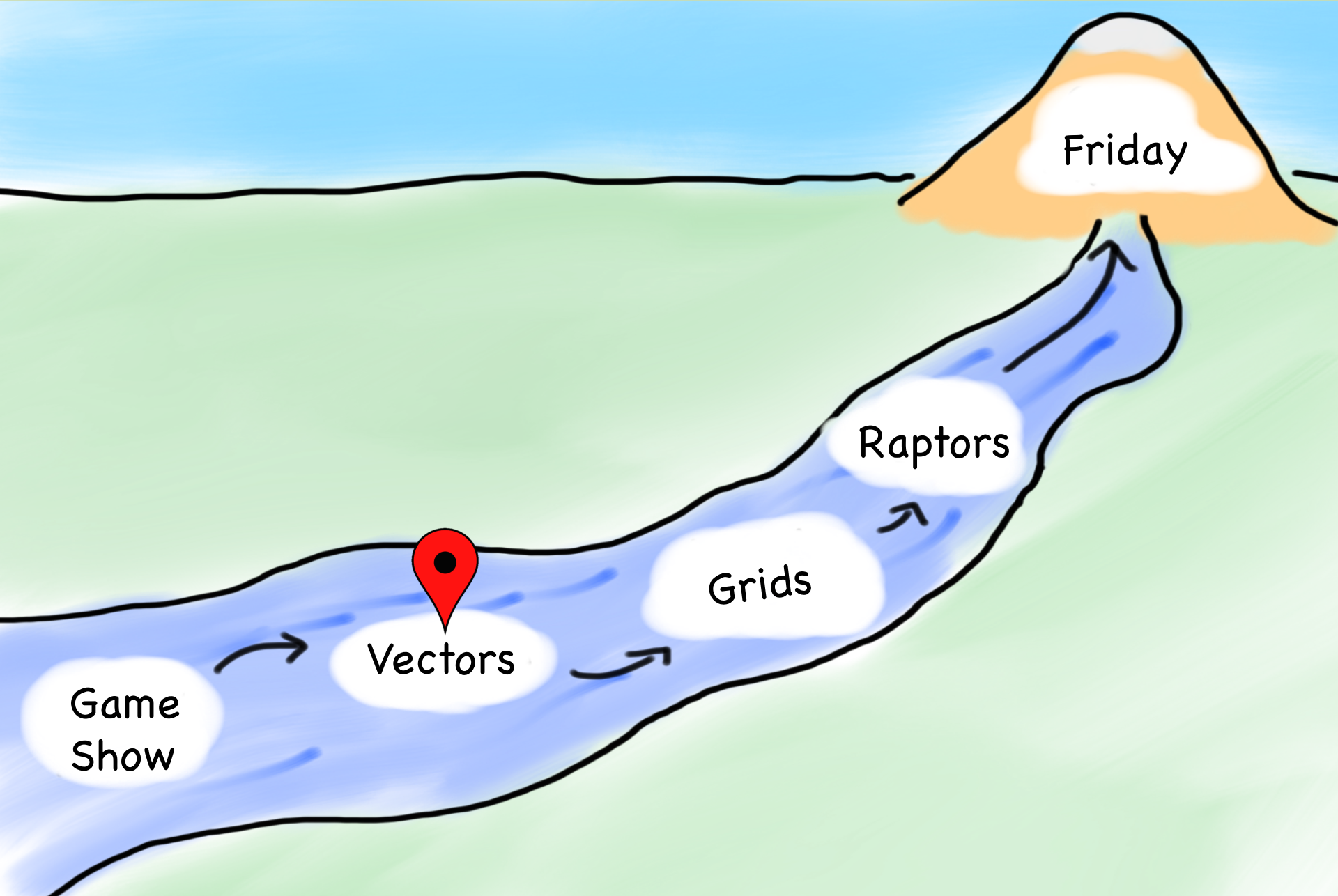


Vector Example

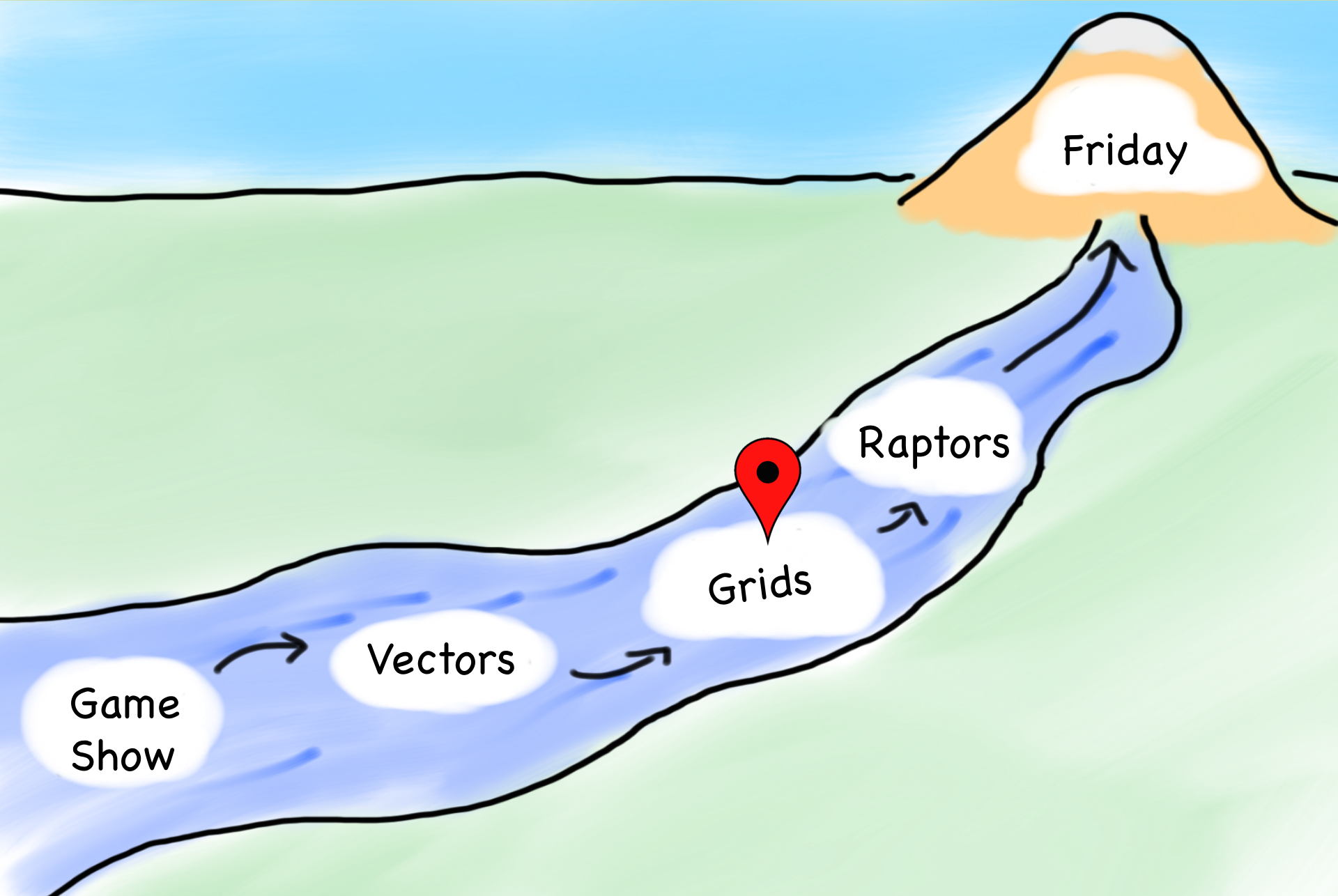
```
for(int i = 0; i < magic.length(); i++) {
    cout << magic[i]
}
```



Today's Goals



Today's Goals



Collections

Vector

Grid

Map

Stack

Queue

Set

Collections

Grid

Grid<type>

Grid<type>


$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}$
WELCOME ..... TO
THE MATRIX!!!!!!

Grid Overview

What is it?

- Advanced 2D array.
- Think spread sheets, game boards

Important Details

- Default constructor makes a grid of size 0
- Doesn’t support “ragged right”.
- Bounds checks
- Knows its size.

Grid Creation

```
Grid<string> grid;
```

or

```
Grid<string> grid(3, 4);
```

Grid Methods

`grid.numRows()` Returns the number of rows in the grid.
`grid.numCols()` Returns the number of columns in the grid.
`grid[i][j]` Selects the element in the `i`th row and `j`th column.
`grid.resize(rows, cols)` Changes the dimensions of the grid and clears any previous contents.
`grid.inBounds(row, col)` Returns `true` if the specified row , column position is within the grid.

Collections

1. Defined as Classes
   This means they have constructors and member functions

2. Templatized
   They have a mechanism for collecting different variable types

3. Deep copy assignment
   Often pass them by reference!

Common Pitfalls 1

```
Vector numbers;
```

Common Pitfalls 1

```
Vector<int> numbers;
```

Common Pitfalls 2

```
Vector<Vector<int>> numbers;
```

Common Pitfalls 2

```
Vector<Vector<int> > numbers;
```

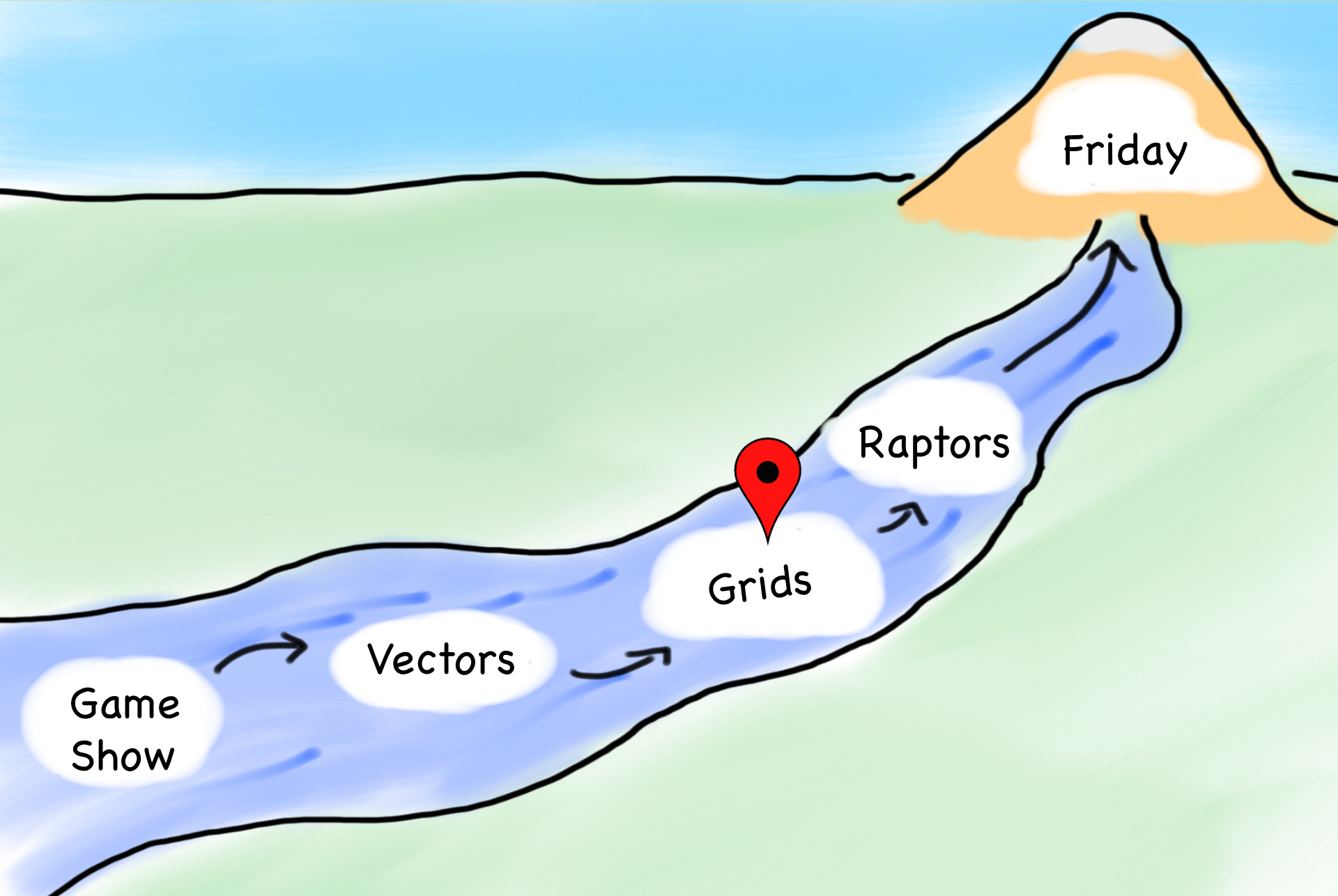
Common Pitfalls 3

```
void myFunction(Grid<bool> gridParam);
```

Common Pitfalls 3

```
void myFunction(Grid<bool> & gridParam);
```

Today's Goals

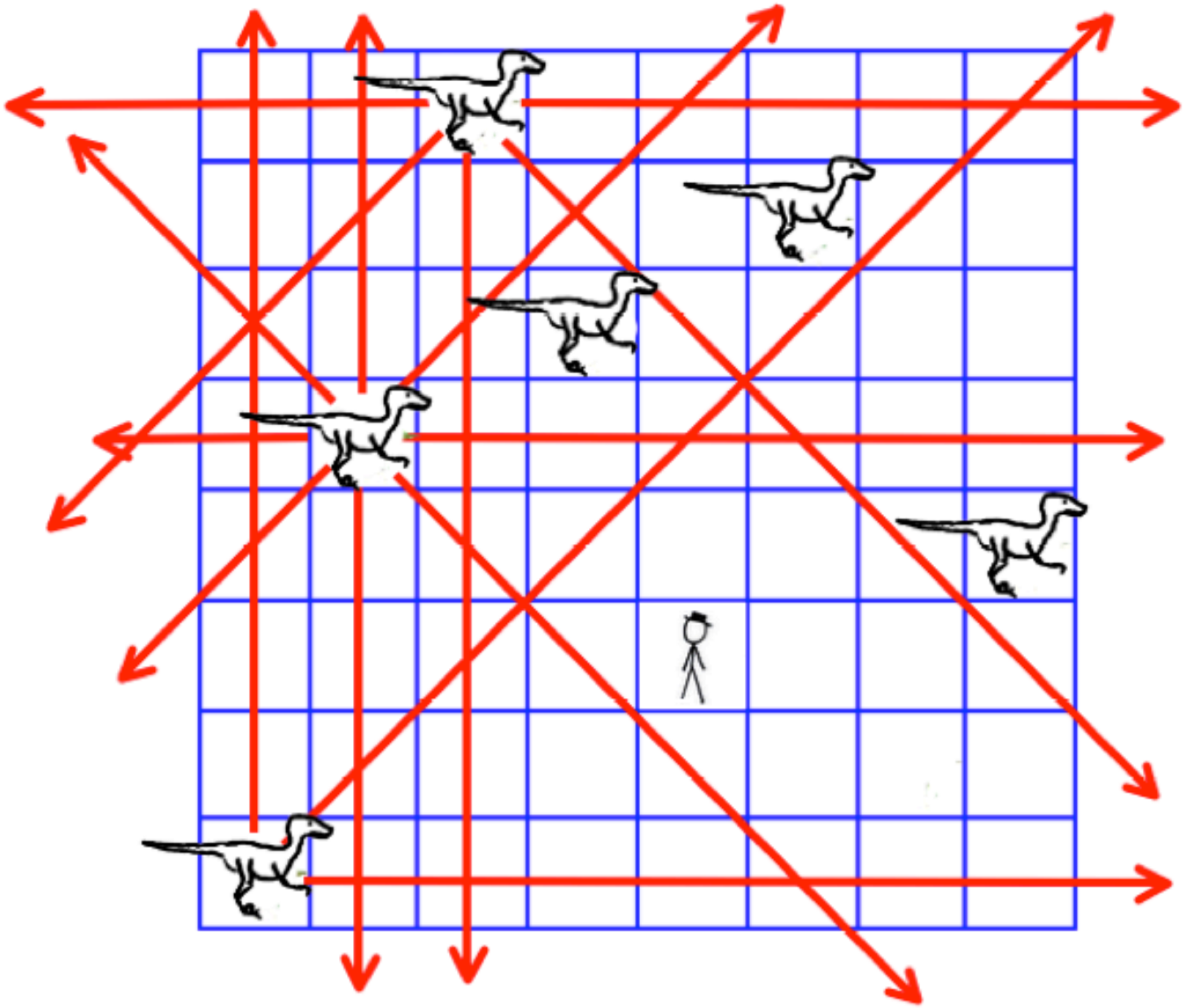


Today's Goals

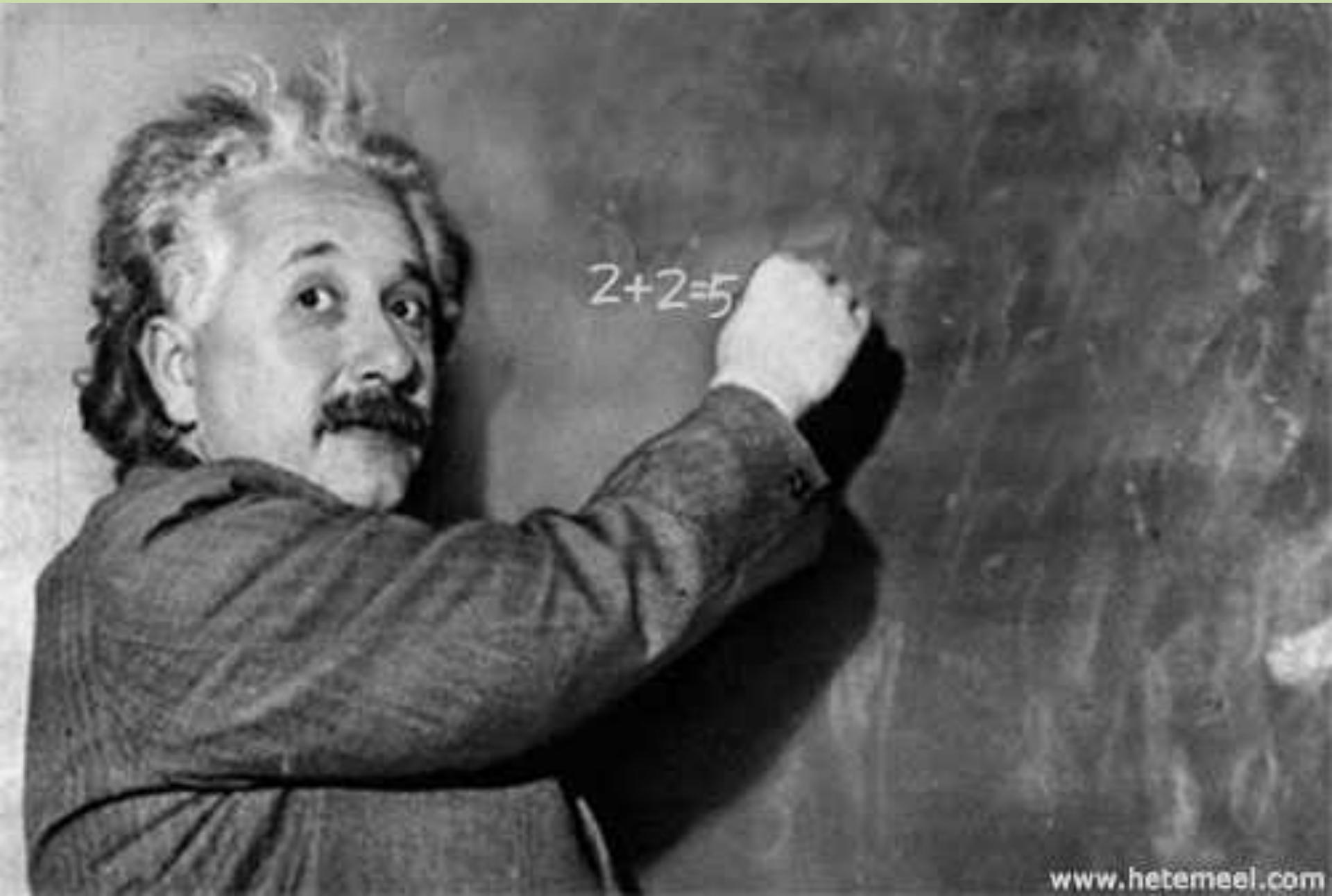
IF THE MATRIX WAS GOOD AND JURRASIC PARK WAS GOOD

WHY NOT HAVE JURRASIC PARK IN THE MATRIX?

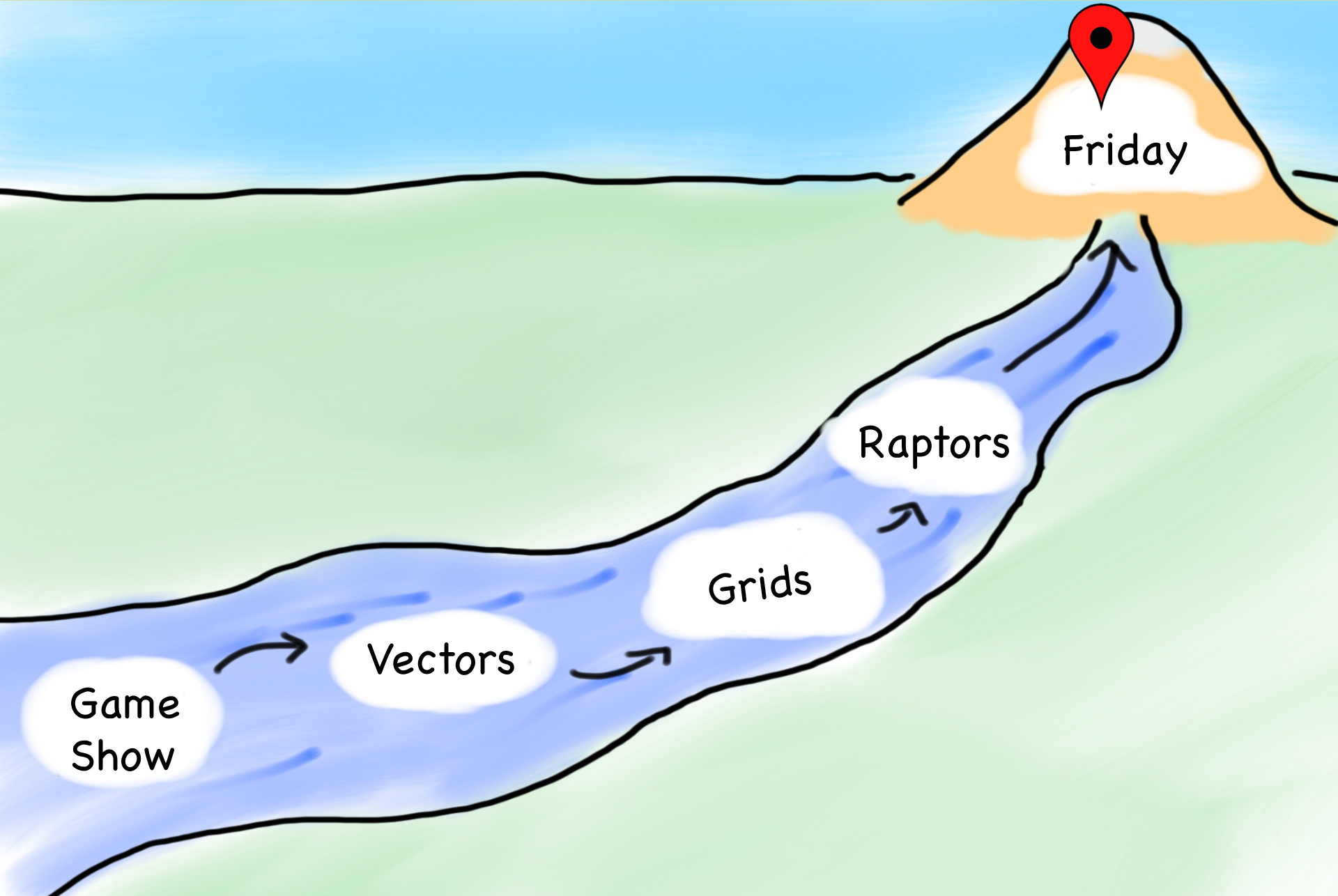
Velociraptor Safety



Life Skills on the Board



Today's Goals



Today's Goals

1. Learn about Vectors

2. Learn about Grids