Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

Final Project Part 2 - Memory Matching Game Code – Text Based

Game

Requirements

Create a class ‘MemoryMatchGame’, with the various variables, arrays and functions

need to play the game.

Hint: Use functions to divide the code up into specific functional areas

Have the game player select a 1 of the 3 Themes and have 50 words associated with the selected

theme.

  Choose game theme:

1) Theme name 1

2) Theme name 2

3) Theme name 3

Creating the three(3) theme files is part 1 of the final project.

o Words must have a common theme - your choice

o MAX individual word length is 8 characters, no spaces, 1 word per line...

o Examples: Like Periodic Table Elements, or Sports teams, or Types of cars...

o Hint: load, from one of the three files, into a single dim array of string in class (Menu to

select)

Have one Term describing a category you picked. This is the FACE term...

Additional Menu User Interaction:

Level of Play – Use selects at start of game

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

4 x 4 grid (Easy)

6 x 6 grid (Moderate)

8 X 8 grid (Difficult)

Hint: Save as a variable in the class

Speed of Play – At start of game, User selects time interval for User selected term-pair to display

2 seconds (Difficult)

4 seconds (Moderate)

6 seconds (Easy)

Hint: Save as a variable in the class, for MS Window coders – use “sleep” code for delay...

Next, Populate answer Grid with randomly selected Terms from the theme array

At start of game – program places the same face/theme term in ALL visible squares in the visible grid

Real Answers not yet visible, only theme name is displayed in all squares, at start of game.

Program select number of random terms from the 50 available for selected theme (that programmer set up)

o If 4 x 4 grid, randomly pick 8 terms, place each image name twice in 2-Dim array.

o If 6 x 6 grid, randomly pick 18 terns, place each image name twice in 2-Dim array.

o If 8 x 8 grid, randomly pick 32 terms, place each image name twice in 2-Dim array.

Hint: Randomly shuffle theme array and just pick the first 8, or 18 or 32 terms per game player selection

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

Next, display the current game state on screen.

Note: ‘Answer’ array is different from ‘display’ array

During the course of play, the face/theme term in the display grid is replaced by a
    corresponding array terms, when user selects a grid square

Decide on how the user select/chooses a square/cell/location that is displayed... there many
different
methods.

Game Play

1) User selects a FIRST square, the theme/face term in the grid square is replace with
correspond stored term, from the 2-dim answer array

2) User selects a SECOND square, the term theme/face in the second grid square is replace with
the corresponding stored term, from the 2-dim answer array

3) The computer compares the terms for the two selected squares.

If they are the same, the terms remain on the screen and can no longer be selected.

If they are different, the term remain the screen for 2, 4 or 6 seconds, depending on user
selection at the beginning of the game. After that elapse time, those two grid terms are
replaced with the face/theme term.

======================================

The class you write

  A class consists of variables/arrays and functions.

All your variables/arrays and functions are to be encapsulated inside the Memory Match game

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

class you write.

The class will use 1 and 2 dimensional arrays

The class will have several variables

The class will have several functions – clearly named

There will be NO GLOBAL VARIABLES/Arrays or functions declared above int main(). All variables

and arrays and functions will be ENCAPSULATED in the class.

The int main() in your code contain only two lines of code::

```
#include iostream;
using namespace std;
#include string;
#include MemoryMatchGame;
Int main() {
    MemoryMatchGame  Game1;  // first line - declare instance of game
    Game1.start();                              // second line - start game
}
```

Timer (Extra credit) - Create/display a timer that keep track of the number of seconds it took to win a

game.

To receive the most credit, this project must be functional.

Hint:

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

Possible arrays needed:

string theme50Words[50]   - load from file into this arrray

string answerArray[#][#}   - load 8, 18, or 32 words twice into this array, depending on game size

selected

string gamePlayArray[#}[#] – Game play and display array

```
//
//  main.cpp
//  Week 17 Final Project
//
//  Created by Pippo Pesic on 11/28/22.
//

#include "MemoryMatchGame.h"

int main() {
	MemoryMatchGame game1;
	game1.start();
	return 0;
}
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

```
//
//  MemoryMatchGame.h
//  Week 18 Final Project
//
//  Created by Pippo Pesic on 11/28/22.
//

#ifndef MemoryMatchGame_h
#define MemoryMatchGame_h

#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <stdlib.h>
#include <unistd.h>
#include "Board.h"
using namespace std;

class MemoryMatchGame {
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

```
    //Define Board Arrays

    Board board = *new Board();

    int rowGuess1 = -1;

    int columnGuess1 = -1;

    int rowGuess2 = -1;

    int columnGuess2 = -1;


    int difficultyLevel;

    string theme;

    int guessTime;


    const int BASE_TIME = 1000000; //In Milliseconds


    string guessedThemeWords[32];



public:



    void setTheme() {
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

```
        bool isThemeSelected = false;


        while (!isThemeSelected){
        cout << "Select theme: Cars, Elements, Fruit: [c,e,f] ";
        cin >> theme;
        cout << endl;
        if (theme == "c" or theme == "e" or theme == "f") {
                isThemeSelected = true;
        }
        else {
                cout << endl << "Invalid theme" << endl;
        }
        }
        }


        void setDifficultyLevel() {
        cout << "Enter difficulty level (4, 6, 8): ";
        cin >> difficultyLevel;
        if (difficultyLevel != 4 and difficultyLevel != 6 and difficultyLevel != 8) {
        cout << endl << "Invalid difficulty level" << endl;
        }
```

```
}


void setTimeDelay() {

cout << endl << "Enter time in seconds between guesses (2, 4, 6): ";

cin >> guessTime;

if (guessTime == 2 or guessTime == 4 or guessTime == 6) {

guessTime = guessTime * BASE_TIME;

}

else {

cout << endl << "Invalid time" << endl;

}

}



void gameLoop() {

int guessedWordsCount = 0;

int uniqueWordCount = difficultyLevel * difficultyLevel / 2;


while (guessedWordsCount < uniqueWordCount) {

cout << "Guess 1:" << endl;

cout << "Enter column (0, 1, 2, 3...) ";
```

```
	cin >> columnGuess1;
	cout <<  endl << "Enter row (0, 1, 2, 3...) ";
	cin >> rowGuess1;

	// re-print board
	board.printBoard(guessedThemeWords, rowGuess1, columnGuess1, -1, -1);

	cout << endl << "Guess 2:" << endl;
	cout << "Enter column (0, 1, 2, 3...) ";
	cin >> columnGuess2;
	cout <<  endl << "Enter row (0, 1, 2, 3...) ";
	cin >> rowGuess2;

	if (board.isMatch(rowGuess1, columnGuess1, rowGuess2, columnGuess2)){
		guessedThemeWords[guessedWordsCount] = board.getElement(rowGuess1,
columnGuess1);
		guessedWordsCount++;
	}
	else {
		board.printBoard(guessedThemeWords, rowGuess1, columnGuess1, rowGuess2,
columnGuess2); // print both guesses and delay
```

```
            usleep(guessTime);
        }


        // re-print board
        board.printBoard(guessedThemeWords, -1, -1, -1, -1);
        }
        }


        //Start procedure and guess loop
        void start() {


        setTheme();
        setDifficultyLevel();
        setTimeDelay();


        board.boardSize = difficultyLevel;
        board.theme = theme;
        board.initializeBoard();
        board.printBoard(guessedThemeWords, -1, -1, -1, -1);


        gameLoop();
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

```
        cout << "You won!" << endl;

        cout << "Philip Pesic 12/11/22" << endl;

        }


};




#endif /* MemoryMatchGame_h */


//

//  Board.h

//  Week 17 Final Project

//

//  Created by Pippo Pesic on 11/28/22.

//


#ifndef Board_h

#define Board_h
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

```
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <stdlib.h>
#include <unistd.h>
using namespace std;

class Board {

        string boardData[8][8];
        string allThemeWords[50];

public:

        int boardSize;
        string theme;

        // Randomizes the order of words
        void shuffleArray(string arr[], int size){
        srand(time(0));
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

```
for(int i = 0; i < size; i++){

int randomIndex = rand() % size;

//swap

string temp = arr[i];

arr[i] = arr[randomIndex];

arr[randomIndex] = temp;

}

}


// check if arr contains a value

bool contains(string arr[], int size, string val){

for(int i = 0; i < size; i++){

if(arr[i] == val){

        return true;

}

}

return false;

}



bool isMatch(int row1, int col1, int row2, int col2){
```

```
return boardData[row1][col1] == boardData[row2][col2];

}


string getElement(int row, int col){

return boardData[row][col];

}


void loadAllThemeWords() {

ifstream fin;

string line;


if (theme == "c") {

fin.open("/Users/pippo/Desktop/Week 18 Final Project/Cars.txt");

}

else if (theme == "e") {

fin.open("/Users/pippo/Desktop/Week 18 Final Project/Atoms.txt");

}

else  if (theme == "f") {

fin.open("/Users/pippo/Desktop/Week 18 Final Project/Fruit.txt");

}
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

```
	int i = 0;
	while (!fin.eof()) {
	getline(fin, line);
	allThemeWords[i] = line;
	i++;
	}
	}



	void printBoard(string guessedThemeWords[], int rowGuess1, int columnGuess1, int
rowGuess2, int columnGuess2) {
	int uniqueWordCount = boardSize * boardSize / 2;
	for (int i = 1; i < 100; i++) {
	cout << endl;
	}
	for(int row = 0; row < boardSize; row++){
	for(int column = 0; column < boardSize; column++){
		if (contains(guessedThemeWords, uniqueWordCount, boardData[row][column])){
		cout << setw(10) << boardData[row][column];
		}
```

```
                else if ((row == rowGuess1 && column == columnGuess1) || (row == rowGuess2 && column == columnGuess2)){
                cout << setw(10) << boardData[row][column];
                }
                else {
                cout << setw(10) << "X";
                }
        }
        cout << endl;
        }
        }

        void initializeBoard(){
        // select random theme words
        int count = 0;
        int uniqueWordCount = boardSize * boardSize / 2;
        string selectedThemeWords[32];
        srand(time(0));
        loadAllThemeWords();

        while (count < uniqueWordCount) {
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

```
        // pick a random theme word
        int index = rand() % 50;
        if (allThemeWords[index] != "") {
                selectedThemeWords[count] = allThemeWords[index];
                allThemeWords[index] = "";
                count++;
        }
        }

        string doubleSelectedThemeWords[64];
        for(int i = 0; i < uniqueWordCount; i++){
        doubleSelectedThemeWords[i] = selectedThemeWords[i];
        doubleSelectedThemeWords[i+uniqueWordCount] = selectedThemeWords[i];
        }

        shuffleArray(doubleSelectedThemeWords, 16);

        // assign words into board
        int i = 0;
        for(int row = 0; row < boardSize; row++){
        for(int column = 0; column < boardSize; column++){
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project

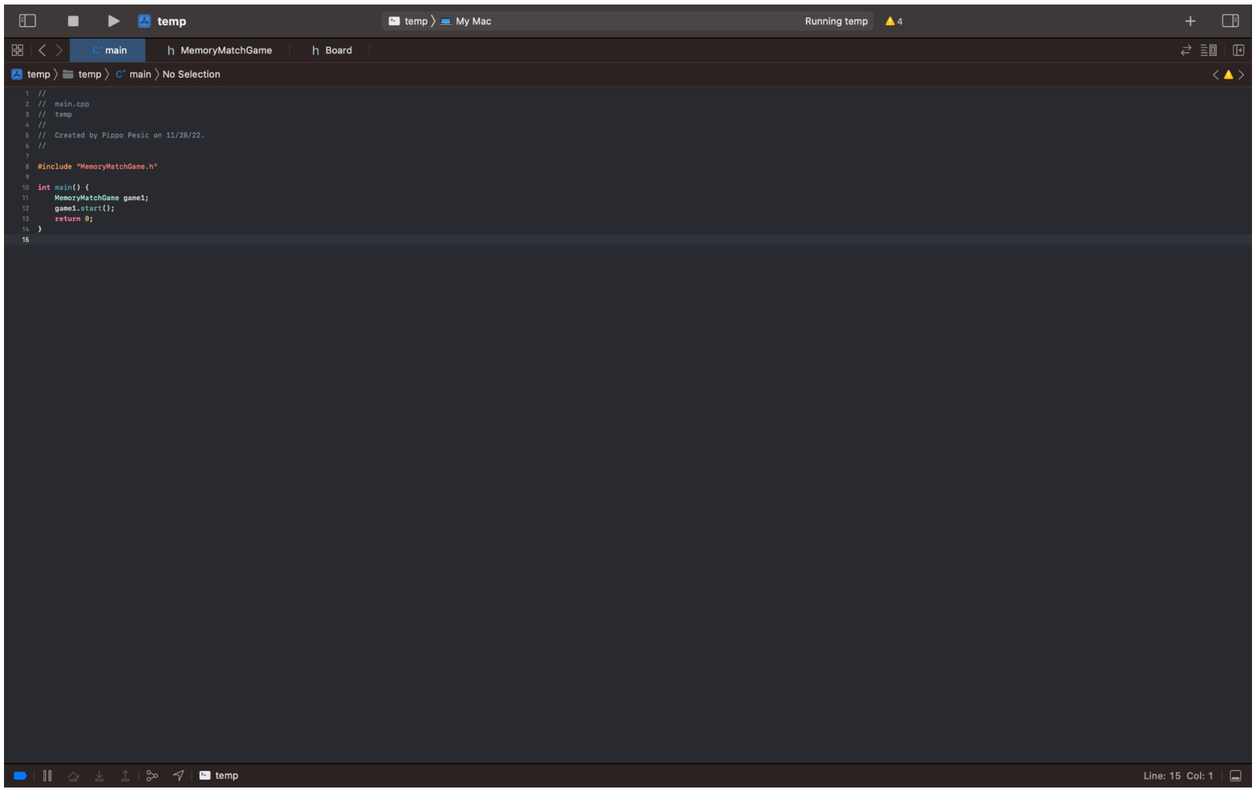
```
                boardData[row][column] = doubleSelectedThemeWords[i];
                i++;
        }
        }
        }
};

#endif /* Board_h */
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project



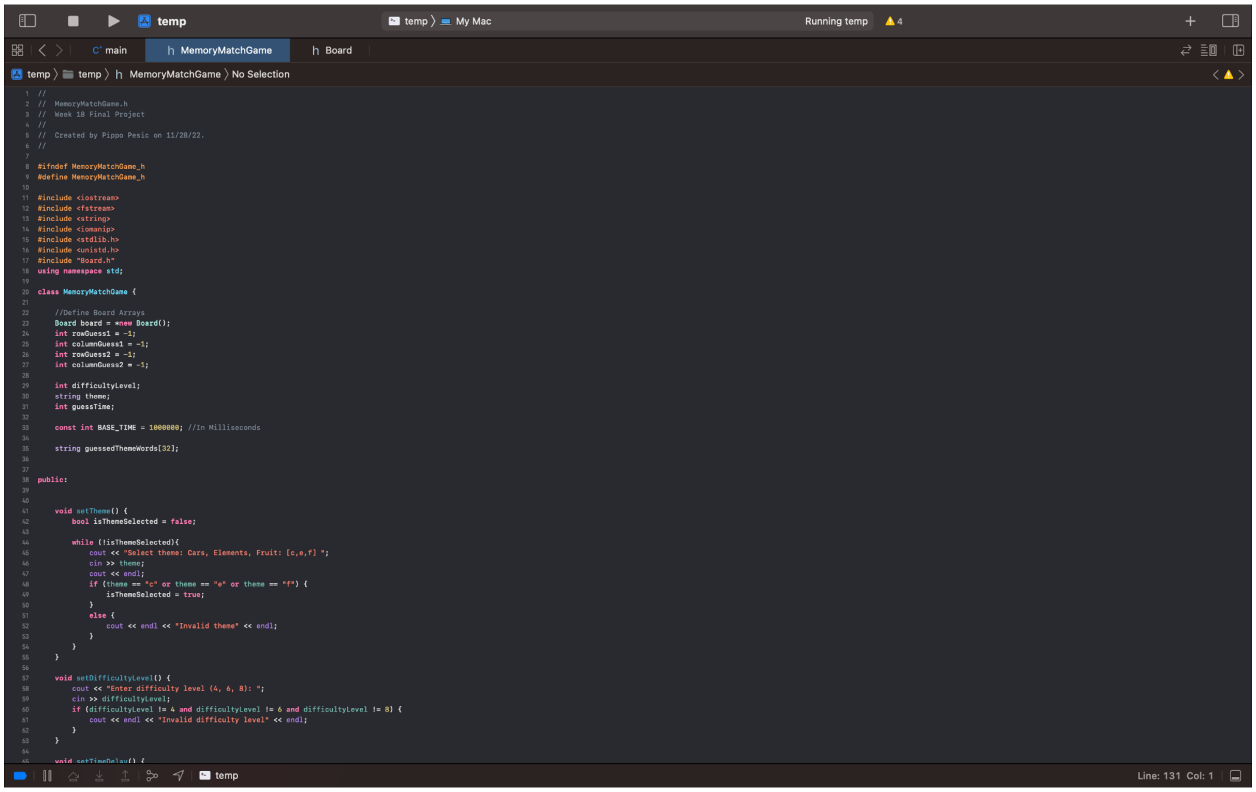
```
//
//  main.cpp
//  temp
//
//  Created by Pippo Pesic on 11/28/22.
//

#include "MemoryMatchGame.h"

int main() {
    MemoryMatchGame game1;
    game1.start();
    return 0;
}
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project



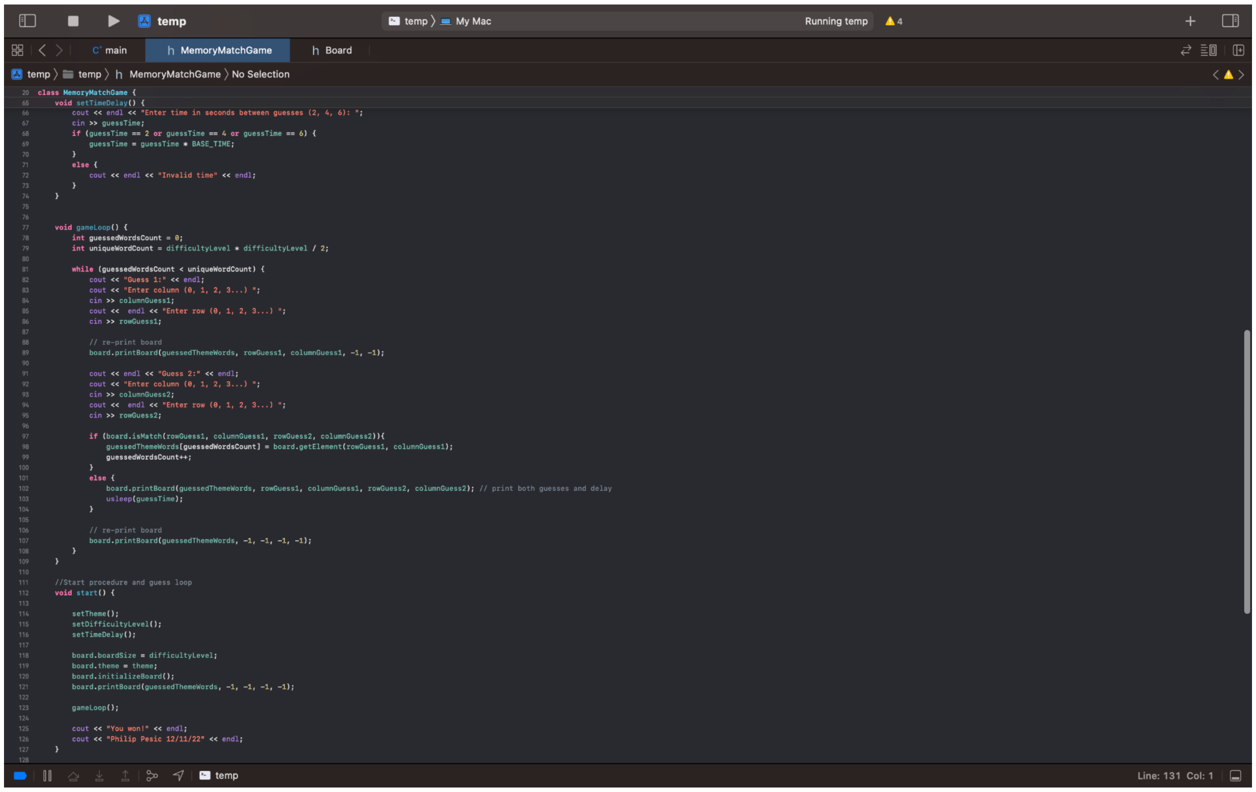
```cpp
//
//  MemoryMatchGame.h
//  Week 18 Final Project
//
//  Created by Pippo Pesic on 11/28/22.
//

#ifndef MemoryMatchGame_h
#define MemoryMatchGame_h

#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <stdlib.h>
#include <unistd.h>
#include "Board.h"
using namespace std;

class MemoryMatchGame {

    //Define Board Arrays
    Board board = *new Board();
    int rowGuess1 = -1;
    int columnGuess1 = -1;
    int rowGuess2 = -1;
    int columnGuess2 = -1;

    int difficultyLevel;
    string theme;
    int guessTime;

    const int BASE_TIME = 1000000; //In Milliseconds

    string guessedThemeWords[32];


public:


    void setTheme() {
        bool isThemeSelected = false;

        while (!isThemeSelected){
            cout << "Select theme: Cars, Elements, Fruit: [c,e,f] ";
            cin >> theme;
            cout << endl;
            if (theme == "c" or theme == "e" or theme == "f") {
                isThemeSelected = true;
            }
            else {
                cout << endl << "Invalid theme" << endl;
            }
        }
    }

    void setDifficultyLevel() {
        cout << "Enter difficulty level (4, 6, 8): ";
        cin >> difficultyLevel;
        if (difficultyLevel != 4 and difficultyLevel != 6 and difficultyLevel != 8) {
            cout << endl << "Invalid difficulty level" << endl;
        }
    }

    void setTimeDelay() {
```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project



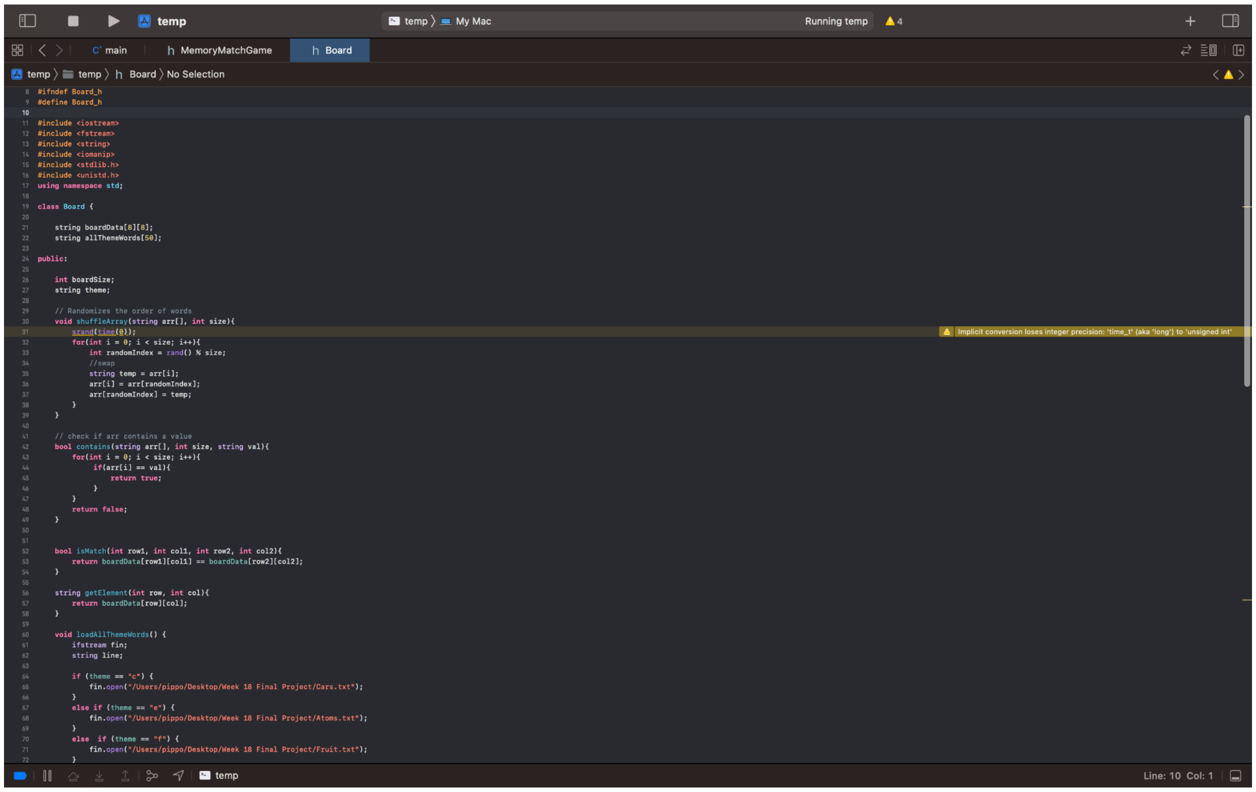
```cpp
class MemoryMatchGame {
    void setTimeDelay() {
        cout << endl << "Enter time in seconds between guesses (2, 4, 6): ";
        cin >> guessTime;
        if (guessTime == 2 or guessTime == 4 or guessTime == 6) {
            guessTime = guessTime * BASE_TIME;
        }
        else {
            cout << endl << "Invalid time" << endl;
        }
    }


    void gameLoop() {
        int guessedWordsCount = 0;
        int uniqueWordCount = difficultyLevel * difficultyLevel / 2;

        while (guessedWordsCount < uniqueWordCount) {
            cout << "Guess 1:" << endl;
            cout << "Enter column (0, 1, 2, 3...) ";
            cin >> columnGuess1;
            cout <<  endl << "Enter row (0, 1, 2, 3...) ";
            cin >> rowGuess1;

            // re-print board
            board.printBoard(guessedThemeWords, rowGuess1, columnGuess1, -1, -1);

            cout << endl << "Guess 2:" << endl;
            cout << "Enter column (0, 1, 2, 3...) ";
            cin >> columnGuess2;
            cout <<  endl << "Enter row (0, 1, 2, 3...) ";
            cin >> rowGuess2;

            if (board.isMatch(rowGuess1, columnGuess1, rowGuess2, columnGuess2)){
                guessedThemeWords[guessedWordsCount] = board.getElement(rowGuess1, columnGuess1);
                guessedWordsCount++;
            }
            else {
                board.printBoard(guessedThemeWords, rowGuess1, columnGuess1, rowGuess2, columnGuess2); // print both guesses and delay
                usleep(guessTime);
            }

            // re-print board
            board.printBoard(guessedThemeWords, -1, -1, -1, -1);
        }
    }

    //Start procedure and guess loop
    void start() {

        setTheme();
        setDifficultyLevel();
        setTimeDelay();

        board.boardSize = difficultyLevel;
        board.theme = theme;
        board.initializeBoard();
        board.printBoard(guessedThemeWords, -1, -1, -1, -1);

        gameLoop();

        cout << "You won!" << endl;
        cout << "Philip Pesic 12/11/22" << endl;
    }

```

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project



```cpp
#ifndef Board_h
#define Board_h

#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <stdlib.h>
#include <unistd.h>
using namespace std;

class Board {

    string boardData[8][8];
    string allThemeWords[50];

public:

    int boardSize;
    string theme;

    // Randomizes the order of words
    void shuffleArray(string arr[], int size){
        srand(time(0));
        for(int i = 0; i < size; i++){
            int randomIndex = rand() % size;
            //swap
            string temp = arr[i];
            arr[i] = arr[randomIndex];
            arr[randomIndex] = temp;
        }
    }

    // check if arr contains a value
    bool contains(string arr[], int size, string val){
        for(int i = 0; i < size; i++){
             if(arr[i] == val){
                 return true;
             }
        }
        return false;
    }


    bool isMatch(int row1, int col1, int row2, int col2){
        return boardData[row1][col1] == boardData[row2][col2];
    }

    string getElement(int row, int col){
        return boardData[row][col];
    }

    void loadAllThemeWords() {
        ifstream fin;
        string line;

        if (theme == "c") {
            fin.open("/Users/pippo/Desktop/Week 18 Final Project/Cars.txt");
        }
        else if (theme == "e") {
            fin.open("/Users/pippo/Desktop/Week 18 Final Project/Atoms.txt");
        }
        else  if (theme == "f") {
            fin.open("/Users/pippo/Desktop/Week 18 Final Project/Fruit.txt");
        }
```

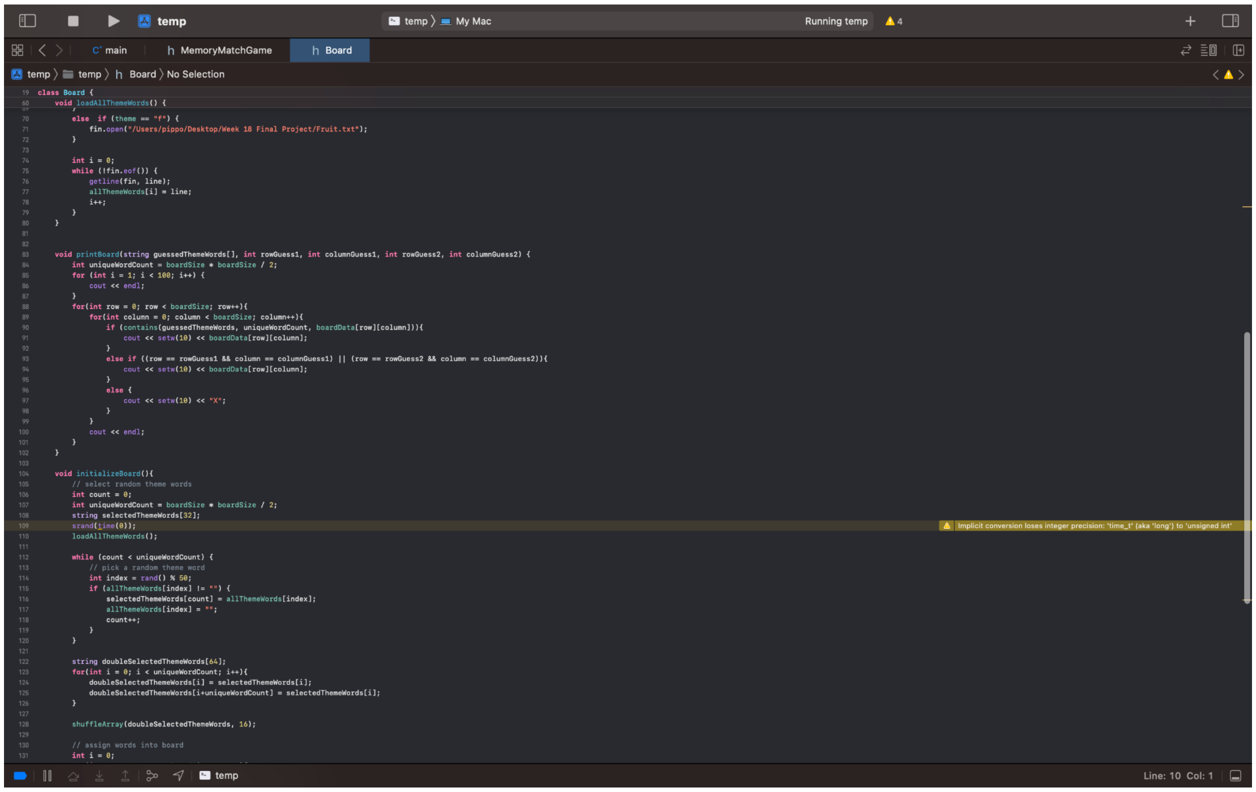
Implicit conversion loses integer precision: 'time_t' (aka 'long') to 'unsigned int'

Philip Pesic

Week 17

December 11 2022

Week 17 Final Project



```cpp
class Board {
    void loadAllThemeWords() {
        else  if (theme == "f") {
            fin.open("/Users/pippo/Desktop/Week 18 Final Project/Fruit.txt");
        }

        int i = 0;
        while (!fin.eof()) {
            getline(fin, line);
            allThemeWords[i] = line;
            i++;
        }
    }


    void printBoard(string guessedThemeWords[], int rowGuess1, int columnGuess1, int rowGuess2, int columnGuess2) {
        int uniqueWordCount = boardSize * boardSize / 2;
        for (int i = 1; i < 100; i++) {
            cout << endl;
        }
        for(int row = 0; row < boardSize; row++){
            for(int column = 0; column < boardSize; column++){
                if (contains(guessedThemeWords, uniqueWordCount, boardData[row][column])){
                    cout << setw(10) << boardData[row][column];
                }
                else if ((row == rowGuess1 && column == columnGuess1) || (row == rowGuess2 && column == columnGuess2)){
                    cout << setw(10) << boardData[row][column];
                }
                else {
                    cout << setw(10) << "X";
                }
            }
            cout << endl;
        }
    }

    void initializeBoard(){
        // select random theme words
        int count = 0;
        int uniqueWordCount = boardSize * boardSize / 2;
        string selectedThemeWords[32];
        srand(time(0));
        loadAllThemeWords();

        while (count < uniqueWordCount) {
            // pick a random theme word
            int index = rand() % 50;
            if (allThemeWords[index] != "") {
                selectedThemeWords[count] = allThemeWords[index];
                allThemeWords[index] = "";
                count++;
            }
        }

        string doubleSelectedThemeWords[64];
        for(int i = 0; i < uniqueWordCount; i++){
            doubleSelectedThemeWords[i] = selectedThemeWords[i];
            doubleSelectedThemeWords[i+uniqueWordCount] = selectedThemeWords[i];
        }

        shuffleArray(doubleSelectedThemeWords, 16);

        // assign words into board
        int i = 0;
```

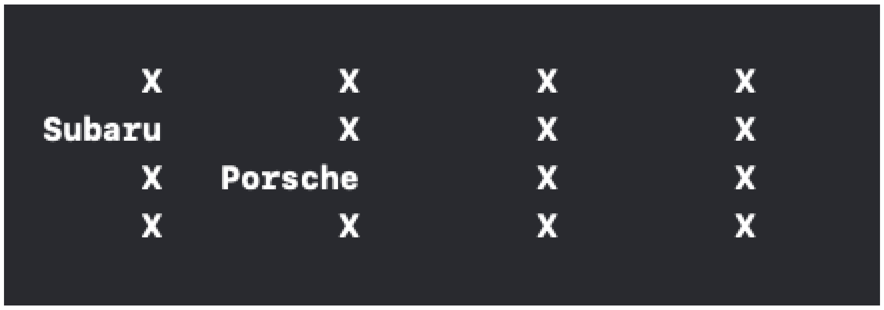
Implicit conversion loses integer precision: 'time_t' (aka 'long') to 'unsigned int'

Philip Pesic

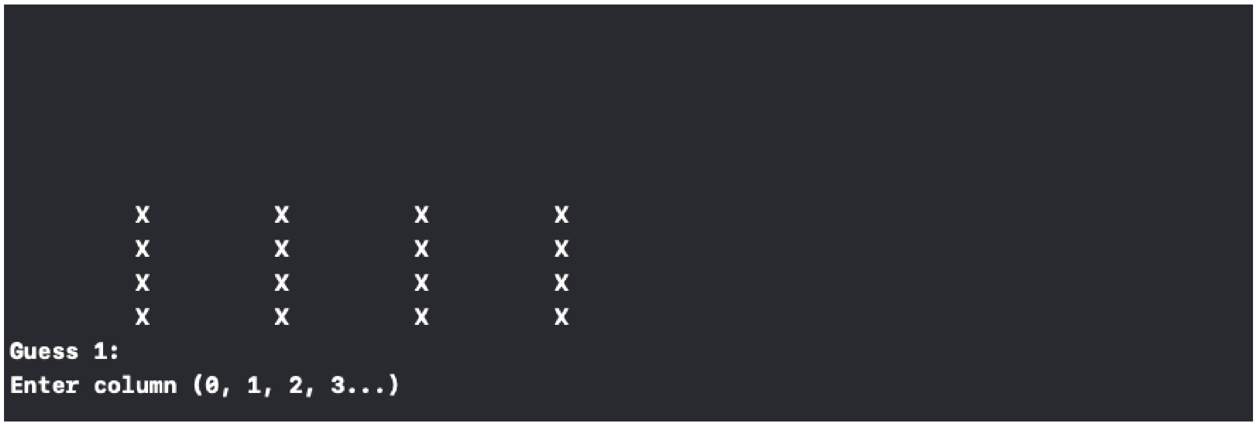
Week 17

December 11 2022

Week 17 Final Project



```
     X         X         X         X
Subaru         X         X         X
     X   Porsche         X         X
     X         X         X         X
```



```
      X      X      X      X
      X      X      X      X
      X      X      X      X
      X      X      X      X
Guess 1:
Enter column (0, 1, 2, 3...)
```

I learned: how to use OOP principles to create a memory match game with file io, recursion, and class elements