Using Game Based-Learning for “C”Programming

Teaching and learning process has been modified from lecture-based learning to be more interesting and motivating learning environment with the use of modern technologies and variety of technological tools. The educational computer game, which is one of the effective learning tools, has been used to support teaching and to motivate learning in various subjects. With less computer game in higher education, especially on C programming language course, this study developed a mobile game for simulating the compiler working with human daily life situations for the C programming language learning. The developed game was conducted on university students for evaluating the overview of students' C programming language motivation and learning achievement. Moreover, this study examined the differences of motivation and learning achievement between students who had experience and inexperience with educational computer games.

Game based learning is where game characteristics and principles are embedded within learning activities. Here, learning activities promote student engagement and motivation to learn. Components of game-based learning include points systems, badges, leaderboards, discussion boards, quizzes and classroom response systems. Points may come with academic rewards such as having an extra week to submit an assignment once reaching a certain point threshold. Badges can be given if students reach a certain success level while classroom response systems like Kahoot or Top Hat encourage participation through points.

Game based learning is also an active learning technique where games are used to enhance student learning. Here, the learning comes from playing the game and promotes critical thinking and problem solving skills. Game based learning can be accomplished with digital or non-digital games and may include simulations that can allow students to experience the learning firsthand.

Game based learning has begun to impact educational institutions across the world. With the increased availability of mobile devices, educational games have the potential to become a major tool for learning. This Innovation compares and contrasts the different pedagogical approaches and learning styles and discusses how they have been integrated into C Rocks, an Android game that has been developed to teach C programming. Recent technical advancements for mobile devices have enabled opportunities to develop complex games on them.

Commonly, games are often used outside of the classroom therefore game-based teaching has not been successfully integrated into many educational institutions. Academic papers, books, web sites and news reports were used to understand the needs of education in computer science, the current research for game-based learning and the methods used for teaching programming

Game-Based Learning vs. Textbook Learning

The educational system has always been slow to change. On one hand, the fact that change is slow is a good thing, since it helps avoid the pitfalls of adopting education fads before they're fully tested. On the reverse side, it leads to a system that is sluggish to respond to the rapidly changing technological world. The only way to increase that response rate is to drastically change the systemic approach to curriculum, which is no small task.

Big reason why change takes so long? Textbooks themselves. When a textbook is released, it contains all of the current information, educational methods, and theories as of when the textbook was sent to print, which is often at least a year before it arrives in schools. Since textbooks are meant to be used for years, there's a substantial delay before curriculum can be changed and information updated.

Game-based learning is built to be adaptive from the beginning. During the production process the game is tested and adjusted to be a more effective learning tool. As new information comes out in the field or educational approaches are adjusted, the game can be changed to suit them. After the game is released, it can be updated with new information, approaches, and more. Games even have student-monitoring analysis tools which allow teachers to monitor the students so the game can be adjusted in future updates.

Learning to program by playing games

Several research studies have been carried out and corresponding games have been developed that aim to include the aforementioned features to support learning and teaching focusing on the computer programming domain. The majority of these games include a specific scenario that aims to cover a specific computer programming unit, while fewer games cover multiple learning objectives and theory units.

Using games both as learning systems and design objects can be promising and attractive for teachers and students. Games as learning tools motivate students to learn and provide training skills. But development of such games requires qualified software engineers, experienced in game design, and also specialists in the field of education that could prepare learning materials (Fig.1).

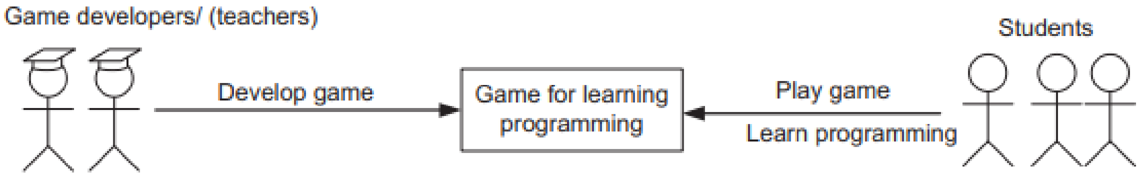


Figure 1: Learning programming through playing games

Using computer games as design objects in teaching the program development process has also several important advantages. The purpose of games development is naturally understood by the majority of the students; many students are familiar with this field and can form adequate requirements for this kind of programming systems. They are motivated to work on a game as they are interested to see the result. Game development also trains team work skills, the setting of project management priorities, planning and conflict resolution.

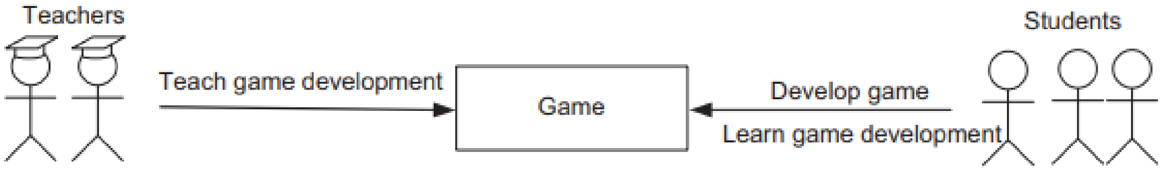


Figure 2: Learning programming through game development

But even more challenging is a combination of the two approaches if students develop not just regular games but games for learning programming (Fig.3). In this case they have to cooperate with teachers familiar with programming languages in order to create learning content and to find a way of its integration to a game context. They have also to test their games on other students studying programming languages. In such a way the students study all the phases of the program development process. As a result of their work the teachers get learning games that can be used for teaching the next generation of students.

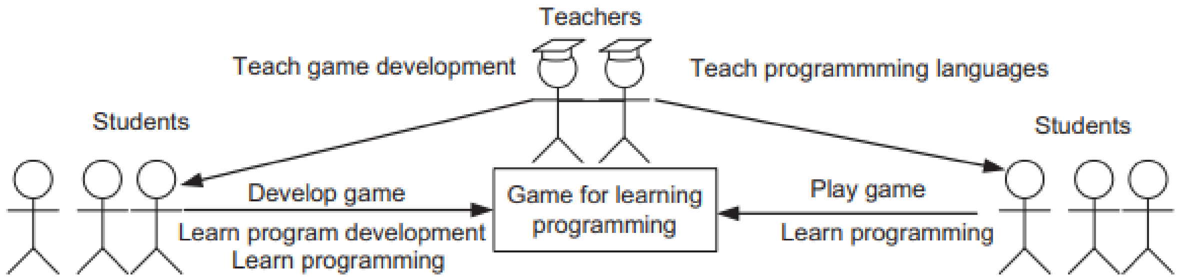


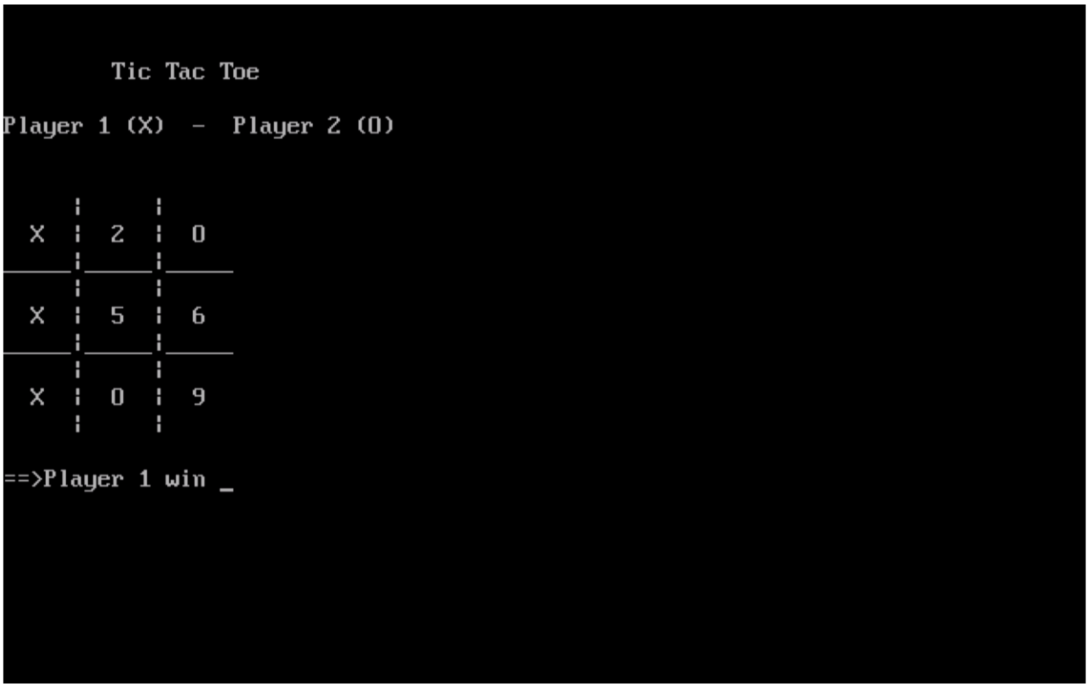
Figure 3: Learning programming through the development of games for learning programming

C tic tac toe game code:

```
#include <stdio.h>
#include <conio.h>

char square[10] = { 'o', '1', '2', '3', '4', '5', '6', '7', '8', '9' };
int checkwin();
void board();
int main()
{
   int player = 1, i, choice;
   char mark;
   do
   {
      board();
      player = (player % 2) ? 1 : 2;


      printf("Player %d, enter a number:  ", player);
      scanf("%d", &choice);
      mark = (player == 1) ? 'X' : 'O';
      if (choice == 1 && square[1] == '1')
         square[1] = mark;
      else if (choice == 2 && square[2] == '2')
         square[2] = mark;
      else if (choice == 3 && square[3] == '3')
         square[3] = mark;
      else if (choice == 4 && square[4] == '4')
         square[4] = mark;
      else if (choice == 5 && square[5] == '5')
```

```
            square[5] = mark;
        else if (choice == 6 && square[6] == '6')
            square[6] = mark;
        else if (choice == 7 && square[7] == '7')
            square[7] = mark;
        else if (choice == 8 && square[8] == '8')
            square[8] = mark;
        else if (choice == 9 && square[9] == '9')
            square[9] = mark;
        else
        {
            printf("Invalid move ");
            player--;
            getch();
        }
        i = checkwin();
        player++;
    }while (i ==  - 1);
    board();
    if (i == 1)
        printf("==>\aPlayer %d win ", --player);
    else
        printf("==>\aGame draw");
    getch();
    return 0;
}
/*********************************************
FUNCTION TO RETURN GAME STATUS
1 FOR GAME IS OVER WITH RESULT
-1 FOR GAME IS IN PROGRESS
O GAME IS OVER AND NO RESULT
 **********************************************/
```

```
int checkwin()
{
    if (square[1] == square[2] && square[2] == square[3])
        return 1;
    else if (square[4] == square[5] && square[5] == square[6])
        return 1;
    else if (square[7] == square[8] && square[8] == square[9])
        return 1;
    else if (square[1] == square[4] && square[4] == square[7])
        return 1;
    else if (square[2] == square[5] && square[5] == square[8])
        return 1;
    else if (square[3] == square[6] && square[6] == square[9])
        return 1;
    else if (square[1] == square[5] && square[5] == square[9])
        return 1;
    else if (square[3] == square[5] && square[5] == square[7])
        return 1;
    else if (square[1] != '1' && square[2] != '2' && square[3] != '3' &&
        square[4] != '4' && square[5] != '5' && square[6] != '6' && square[7]
        != '7' && square[8] != '8' && square[9] != '9')
        return 0;
    else
        return  - 1;
}

/*******************************************************************
FUNCTION TO DRAW BOARD OF TIC TAC TOE WITH PLAYERS MARK
 ********************************************************************/
void board()
{
    system("cls");
    printf("\n\n\tTic Tac Toe\n\n");
    printf("Player 1 (X)  -  Player 2 (O)\n\n\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c \n", square[1], square[2], square[3]);
    printf("_____|_____|_____\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c \n", square[4], square[5], square[6]);
    printf("_____|_____|_____\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c \n", square[7], square[8], square[9]);
    printf("     |     |     \n\n");
}
```



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  2  |  O
_____|_____|_____
     |     |
  X  |  5  |  6
_____|_____|_____
     |     |
  X  |  O  |  9
     |     |

==>Player 1 win _
```

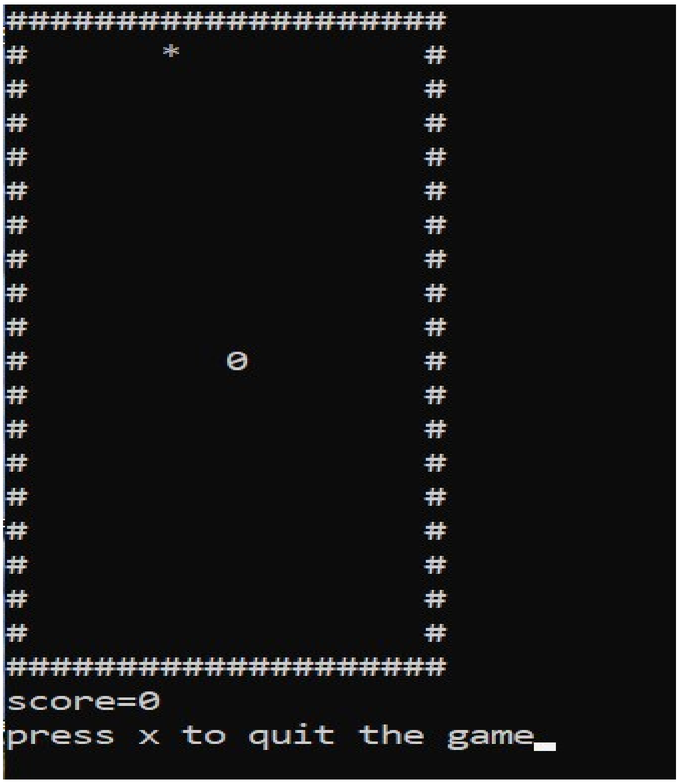
// C program to build the complete snake game//

```
#include <conio.h>
#include <stdio.h>
```

```
#include <stdlib.h>
#include <unistd.h>

int i, j, height = 20, width = 20;
int gameover, score;
int x, y, fruitx, fruity, flag;

// Function to generate the fruit
// within the boundary
void setup()
{
    gameover = 0;

    // Stores height and width
    x = height / 2;
    y = width / 2;
label1:
    fruitx = rand() % 20;
    if (fruitx == 0)
        goto label1;
label2:
    fruity = rand() % 20;
    if (fruity == 0)
        goto label2;
    score = 0;
}

// Function to draw the boundaries
void draw()
{
    system("cls");
    for (i = 0; i < height; i++) {
        for (j = 0; j < width; j++) {
            if (i == 0 || i == width - 1
                || j == 0
                || j == height - 1) {
                printf("#");
            }
            else {
                if (i == x && j == y)
                    printf("0");
                else if (i == fruitx
                        && j == fruity)
                    printf("*");
                else
                    printf(" ");
            }
        }
        printf("\n");
    }
    // Print the score after the
    // game ends
    printf("score = %d", score);
    printf("\n");
    printf("press X to quit the game");
}

// Function to take the input
void input()
{
    if (kbhit()) {
        switch (getch()) {
        case 'a':
            flag = 1;
            break;
        case 's':
            flag = 2;
            break;
        case 'd':
            flag = 3;
            break;
        case 'w':
            flag = 4;
            break;
        case 'x':
            gameover = 1;
            break;
        }
    }
}

// Function for the logic behind
// each movement
void logic()
{
    sleep(0.01);
    switch (flag) {
    case 1:
        y--;
        break;
    case 2:
        x++;
        break;
    case 3:
        y++;
        break;
    case 4:
        x--;
        break;
```

```
	default:
		break;
	}

	// If the game is over
	if (x < 0 || x > height
		|| y < 0 || y > width)
		gameover = 1;

	// If snake reaches the fruit
	// then update the score
	if (x == fruitx && y == fruity) {
	label3:
		fruitx = rand() % 20;
		if (fruitx == 0)
			goto label3;

	// After eating the above fruit
	// generate new fruit
	label4:
		fruity = rand() % 20;
		if (fruity == 0)
			goto label4;
		score += 10;
	}
}

// Driver Code
void main()
{
	int m, n;

	// Generate boundary
	setup();

	// Until the game is over
	while (!gameover) {

		// Function Call
		draw();
		input();
		logic();
	}
}
```

Output:



```
####################
#      *           #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#         0        #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
#                  #
####################
score=0
press x to quit the game_
```

Result Analysis

Game-based learning is about more than giving students educational games like The Oregon Trail. It’s about changing the approach of students towards learning and the approach of learning towards students. The goal is for students to enjoy the process of learning itself.

As a student learns through game-based learning, they gain much more ownership of the material, which improves retention. The combination of different approaches and goal-based learning addresses all of the learning styles at the same time. No matter what a student’s primary, secondary, or tertiary learning style, a game can address them. Games can even combine multiple subjects into a single game, creating a versatile learning tool.

Learning games also provide students a safe environment for failure. It can be hard for students, especially adolescents, to fail in a public setting like a classroom. Games give them a chance to try out new things. If they fail in the environment of the game, then they can simply try again and learn from their mistakes. Instead of education based on rote memorization, students learn through experimentation and trial and error.

Students that participated in experiment and developed educational games expressed their opinion about the Innovation. They consider game development as a very inspiring thing to do. Development of educational games helped them a lot in practicing skills they learned during studying. Also, most of students noticed that they were motivated to learn many new things that are not included in the standard curriculum.

Conclusion

Games have been used as a learning tool for centuries. Chess was used to teach strategic thinking as far back as the Middle Ages, and the game of Kreigsspiel was invented in 1812 specifically to teach Prussian officers' strategy. Beyond military strategy, the genesis of kindergarten in the mid-1800s was Friedrich Fröbel's ideas of learning through play.

The core concept behind game-based learning is teaching through repetition, failure and the accomplishment of goals. Video games are built on this principle. The player starts off slow and gains in skill until they're able to skillfully navigate the most difficult levels. Games that are planned and designed well will offer enough difficulty to keep it challenging while still being easy enough for the player to win.

Game-based learning takes this same concept and applies it to teaching a curriculum. Students work toward a goal, choosing actions and experiencing the consequences of those actions. They actively learn and practice the right way to do things. The result is active learning instead of passive learning.