Buffer Overflow Code Puzzle Game

Pre-Requisite Knowledge and Skills

1. Understand the basic of functional stack in the computer memory
2. Understand the basic of compilation and assembly code

Learning Objective:

1. Understand the concept of buffer overflow
2. Understand the functional code structure in the computer memory
3. Be able to arrange code structure and corresponding code into correct memory location

Recommended Running Environment and Software:

1. Computers Running Windows 7 or Window 10 x64 OS
2. Unity3D Exe files and data folders of Buffer Overflow Code Puzzle Game

Instructional Material:

1. Buffer Overflow Code Puzzle Game
2. In-game Instructions of Gameplay
3. PPT Lecture Slides

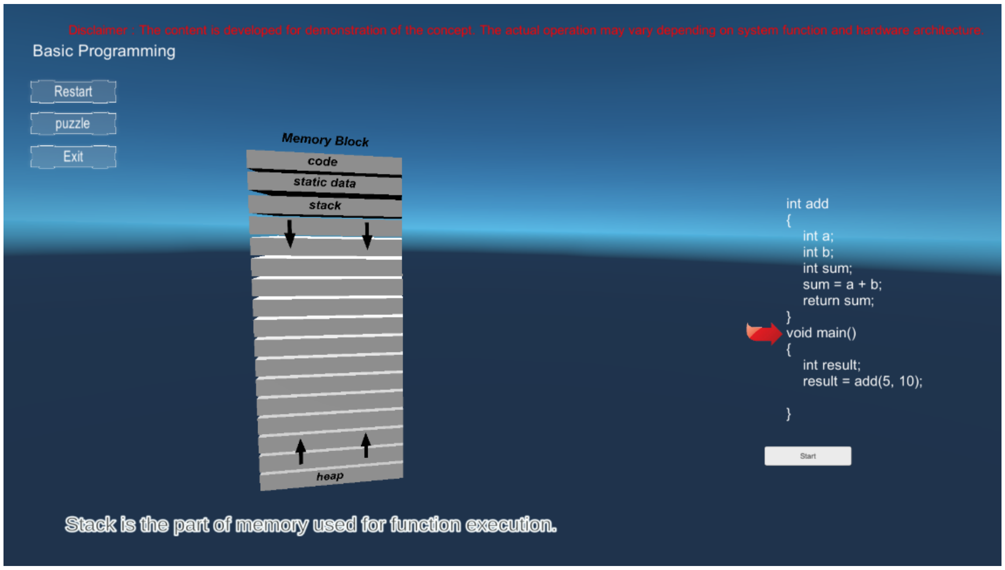
Video Demonstration:

1. to be developed

Lab Assessment:

1. Windows OS

Lab Instructions



Disclaimer : The content is developed for demonstration of the concept. The actual operation may vary depending on system function and hardware architecture.

Basic Programming

```
int add
{
    int a;
    int b;
    int sum;
    sum = a + b;
    return sum;
}
void main()
{
    int result;
    result = add(5, 10);

}
```

Stack is the part of memory used for function execution.

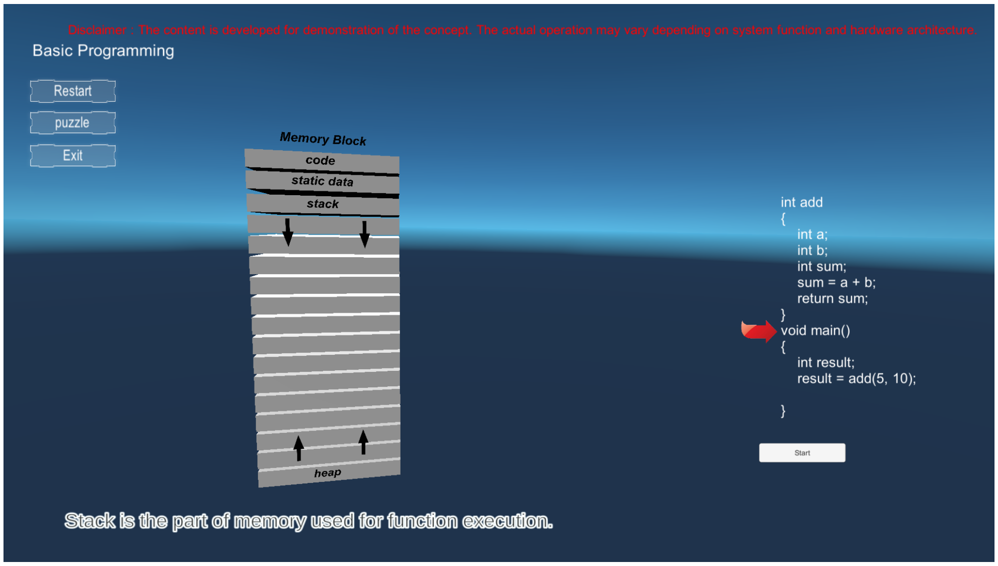
Buffer Overflow Code Puzzle Game Main Menu

A buffer overflow is one of the most malicious software vulnerability, mainly exists in C programs. This happens when a program attempts to load input data that is larger than its original reserved space or buffer. Thus, it will write data outside the boundary to an adjacent memory location.

Step 1: Tutorial

- **The game will automatically play in tutorial mode. Click on the "start" button from the bottom**

  The concept of buffer overflow requires basic knowledge of functional stack, code compilation and assembly code. In the automatic tutorial mode, students will quickly go over these basics.



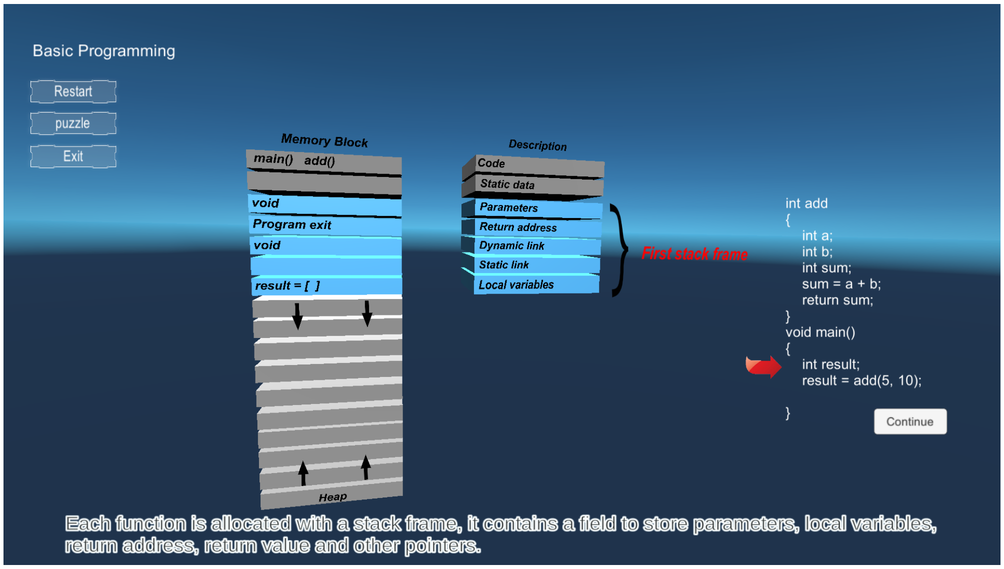
Disclaimer : The content is developed for demonstration of the concept. The actual operation may vary depending on system function and hardware architecture.

```
int add
{
    int a;
    int b;
    int sum;
    sum = a + b;
    return sum;
}
void main()
{
    int result;
    result = add(5, 10);

}
```

Stack is the part of memory used for function execution.

1. The tutorial mode will play in each section. After each section student needs to click "Continue" button from the bottom to play next segment.



```
int add
{
    int a;
    int b;
    int sum;
    sum = a + b;
    return sum;
}
void main()
{
    int result;
    result = add(5, 10);

}
```

Each function is allocated with a stack frame, it contains a field to store parameters, local variables, return address, return value and other pointers.

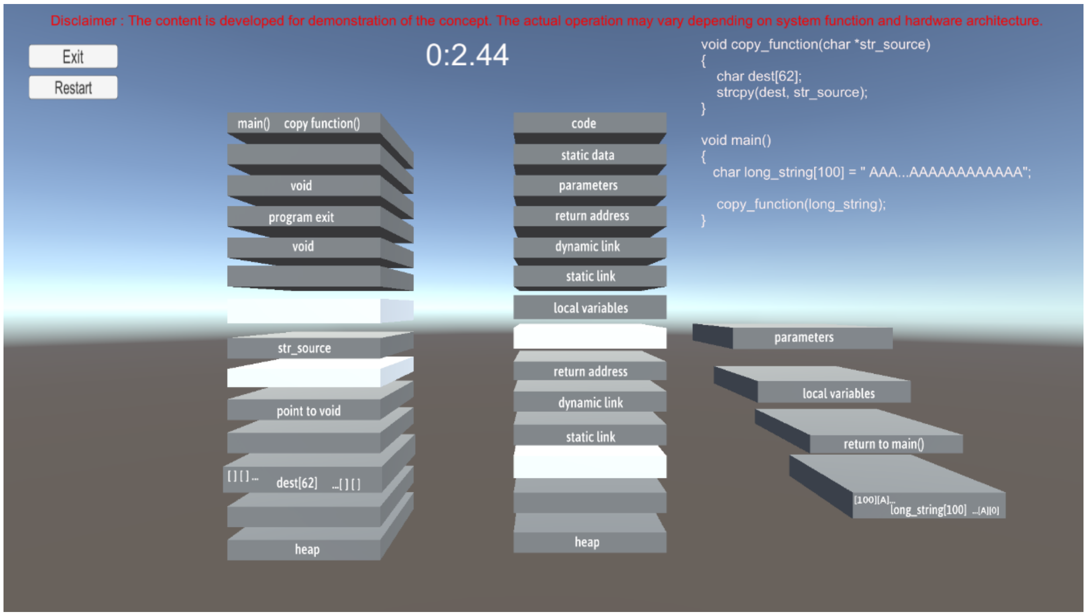
2. After the tutorial compled playing, student can click "Restart" to re play the tutorial,

or go to the Puzzle mode to solve the interactive puzzle.

Step 2: Puzzle

- **Click on the "puzzle" button from left menu**

  The puzzle game will be played in timed mode. Student needs to drag 4 code blocks in the right side and move them into appropriate spot in the computer memory heap in light grey color.



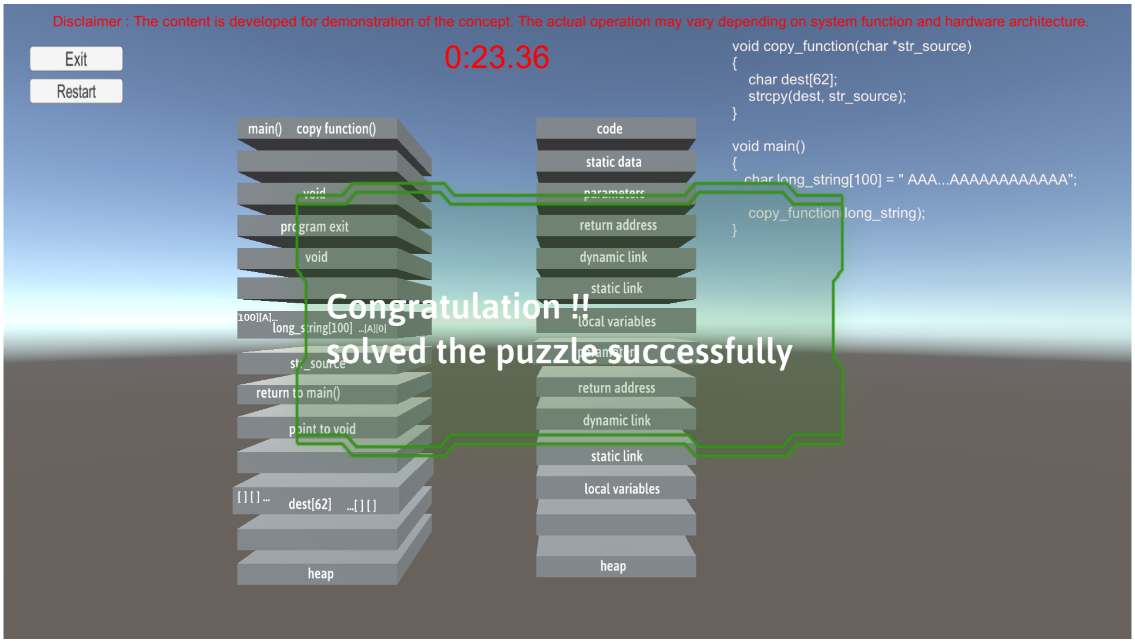
Disclaimer : The content is developed for demonstration of the concept. The actual operation may vary depending on system function and hardware architecture.

0:2.44

```
void copy_function(char *str_source)
{
   char dest[62];
   strcpy(dest, str_source);
}
void main()
{
  char long_string[100] = " AAA...AAAAAAAAAAAA";

   copy_function(long_string);
}
```

1. The "parameter" and "local variables" blocks should be arranged into the "code" heap.
2. The "return to main()" and "[100][A]… long_string[100] ..[A][0]" blocks should be arranged into "main() copy fuction" heap.



Disclaimer : The content is developed for demonstration of the concept. The actual operation may vary depending on system function and hardware architecture.

0:23.36

```
void copy_function(char *str_source)
{
   char dest[62];
   strcpy(dest, str_source);
}
void main()
{
  char long_string[100] = " AAA...AAAAAAAAAAAA";

   copy_function(long_string);
}
```

Congratulation !!
solved the puzzle successfully

3. If the student correctly placed all blocks into their corresponding spots, a message will pop up to show “congratulation!! solved the puzzle successfully”

4. After completing the puzzle game, the student who completed in the shortest amount of time will be the winner.

Discussion

- **How buffer overflow vulnerability will occur in the memory heap?**
- **What is the risk of buffer overflow and how to avoid it?**