Virtual Game Time Travel

Version 1.0

Minecraft Game Design Report

Team 4, Group A&B
Nguyen Dang Khoa (Leader) - 14322
Phuong Hong Nguyen - 14939
Tran Huu Phuc - 14665
Le Hai Yen - 14952

English for Computer Science 4
Foundation Year Studies
Vietnamese-German University
8 April 2019

Nguyen Dang Khoa
132/7A Nguyen Trong Tuyen Street,
Phu Nhuan District, HCMC, Vietnam

8 April 2019

Mr. Richard G. Bradley, MAL, MSM
Vietnamese German University
Foundation Year Studies
Nam Ky Khoi Nghia Street, Hoa Phu District
Binh Duong New City, Binh Duong Province

Dear Mr. Bradley,

Enclosed is our final report entitled "*Virtual Game Time Travel*" submitted to partially fulfill the requirements of the course "English for Computer Science 4", as specified on 18 February 2019.

This report contains the context of an adventure Minecraft Pi game related to time travel, gives a detailed of the software description. The paper also portrays our theme, storyline, the mini-game and description of the design as well as a real-world interface system for the virtual Minecraft game. Additionally, we have attached all the code for the world and a Gantt chart showing the progress of the whole project at the end of the report.

Your consideration of this report is greatly appreciated, and we hope you find it satisfactory. However, if you have any further questions or concerns, please contact us without any kind of hesitation either by telephone 0908268821 or e-mail: khoadangnguyen2012@gmail.com.

Sincerely yours,

Nguyen Dang Khoa
Team Leader

Enclosure: "*Final Report_Team 4_Group A&B*" (1 Copy)

Vietnamese-German University
Foundation Year Studies
English for Computer Science 4

Minecraft Game Design Report
for
Virtual Game Time Travel
Version 1.0

Team 4, Group A&B
Nguyen Dang Khoa (Leader) - 14322
Phuong Hong Nguyen - 14939
Tran Huu Phuc - 14665
Le Hai Yen - 14952
Instructor: Richard G. Bradley, MAL, MSM

Due date: 8 April 2019

Disclaimer

We declare that this report is a product of our own work, unless otherwise referenced. We also declare that all opinions, results, conclusions and recommendations are our own and may not represent the policies or opinions of the Vietnamese-German University.

Nguyen Dang Khoa - 14322

Team Leader

Phuong Hong Nguyen - 14939

Tran Huu Phuc - 14665

Le Hai Yen - 14952

Abstract

Virtual Game Time Travel

Team 1, Group A&B

Nguyen Dang Khoa (Leader) - 14322

Phuong Hong Nguyen - 14939

Tran Huu Phuc - 14665

Le Hai Yen - 14952

This report involves the design and coding of an interactive adventure world in Minecraft Pi running on Raspberry Pi. Minecraft is a popular sandbox open-world building game which the entire world is generated and crafted with blocks. There is available a free version of Minecraft for the Raspberry Pi and it comes with a programming interface, allows us to write commands and scripts in Python syntax to build buildings in the game automatically. The API also allows us to write programs which control and interact with the virtua world - Minecraft. With the guidance of the course instructor, we have successfully completed the interactive adventure Minecraft Pi game related to time travel to meet the requirements at the end of the phase and build a sustainable foundation for us to do the further in the future.

The virtual Minecraft world is a three-dimensional space including a mini-game and a real-world interface. The main character, Steve , had lost his beloved daughter who was sent to the past due to the technical issue of the time machine. Then, in order to save her, Steve decided to complete the machine and came back the time his daughter was sent to. Therefore, the world is divided into two different timelines, one is the future laboratory in the 23rd century while the other is the Notch kingdom with a long history dating back more than a thousand year. The design for real-world interface consists of three tactile buttons called “Play”, “Enter” and “Exit” buttons in order to follow the plot of the game together with the gameplay.

Keywords: Minecraft, Virtual game, programming, Raspberry Pi.

Table of Contents

List of Figures

Acknowledgments

We would like to express the sincere gratitude to our course instructor Mr. Richard G. Bradley at the first place for providing his invaluable guidance, comments and suggestions throughout the course of the project.

We are also grateful to the technicians of the laboratory of the Vietnamese-German University department for their help in offering us the materials during the planning and development of the robot.

This report also could not be completed without the help of our friends and classmates who have willingly helped us out with their abilities.

During this project, tasks were divided up as follows:

Nguyen Dang Khoa wrote the storyline, description of the world, real-world interface, coding and texting a laboratory as well as a real-world interface with on-screen complementary, formatted and checked the report.

Phuong Hong Nguyen wrote software, Gantt chart, coding and testing all the features in the main North kingdom's world.

Tran Huu Phuc wrote gameplay, Gantt chart, coding the laboratory and did the presentation.

Le Hai Yen wrote a letter of transmittal, abstract, acknowledgments, introduction, the context of the project, storyline, drew a map of the world and checked the report.

Team 4

1. Introduction

1.1 Purpose of the Report

This report was created to partially fulfill the requirements of the course “English for Computer Science 4” at the Vietnamese German University, as specified on 18 February 2019. The paper consists of a detailed description of the overall software as well as descriptions of the gameplay and a real-world interface.

1.2 Audience

The intended audience of this document is the course instructor, who will use it as the basis for the determination of a portion of our grade for the class “English for Computer Science 4”.

1.3 Scope of the Report

This report covers a section on the design stage of an adventure game on the small scale in the Minecraft project, indicates an overall instruction portraying how the game will start, gives a detailed software description and a real-world interface. The paper also includes the software code which is written in Python and the Gantt Chart showing planned schedule for work on the project. Not included in this report are any designs for the game with multiple players competing with or against each other to reach the game's goal.

2. Context of the Project

2.1 Requirements and Constraints of the Project

This project is about the design and coding of an interactive adventure world in Minecraft Pi based on a particular theme and runs on Raspberry Pi by using a Raspberry Pi, a small monitor, a keyboard, a mouse, power adapters and an SD card [1].

There are some requirements that the project must involve:

- A Minecraft world that includes a mini-game and real-world interface.
- A short story setting the background for the game.
- A detailed description of the design and interface.
- An explanation of the way the game will be played.
- A map of the design.
- A tour of the world with on-screen commentary, messages and audio responses to user actions.
- A way to remotely influence the play of the game.

Besides, there are some constraints that limit the options for satisfying the requirements above:

- The design must not be copied an existing design or major structure from the Internet as well as not be insulting to any ethnic group, contain offensive material or advocate any political position.
- The coding must be in Python and use the Minecraft API [2].

3. Overview of the Virtual World

3.1. Storyline

In the 23rd century, Steve, along with his colleagues, collaborated on a project generating a time machine in order to carry out time travel. They are still in progress of experimenting the ability as well as the restrictions that machine could be. They use the advances in technology and theoretical physicists in time to develop a way to snare some of the infinitesimally tiny wormholes and then make them billions of times bigger so we could go where and when we want. While the machine is set to be completed, Steve's daughter, who was curious about her father's work, went into the laboratory. After realizing Steve had not locked the laboratory's door during a lunch break, he rushes to the room and at the same time, his daughter screams out loud "Dad, help...". He couldn't reach to his daughter on time, Sarah was dragged into the time gate. Although everybody told him to quit looking for his daughter, he still tried every day to fix the time machine as it was the only way to find his little girl.

Steve Pi has dedicated the rest of his life trying to complete the time machine in order to bring his beloved daughter Sarah back. In 2554, after many efforts try to stabilize the time machine portal, he finally makes it. Steve immediately jumps right into the time machine although he knows that the timer has not completed yet, he adjusts the time to the time his daughter was pulled in. The time machine has successfully brought him to an ancient world through the bridge of time and space. The time machine did not work as he planned, but instead, it sends him back to after 100 years his daughter had arrived. Steve wandered about the old and strange town without any villagers, until he sees Sarah's name on a grave saying she had died 23 years ago due to the war. He also finds a chest with a message from Sarah, without any hesitation, he goes back to the time machine as he knows the time he adjusted was incorrect. Steve has shocked and painful because of Sarah's death, he suddenly passes out and wakes up in his office. Looking at his clock on the table, he sees – the year 2524...

3.2. Description of the world

The world is divided into 2 different timelines, one is the future laboratory in the 23rd century and the Notch kingdom with a long history dating back more than a thousand year.

The Laboratory, one of the most advanced lab in the future was established by Steve Pi and many of the best scientists in the world at that time. This laboratory is equipped with many high-end technologies. They worked secretly, isolated from the world, in order to prevent an evil power from stealing this technology to change the history of Minecraft. The lab includes the main section called "Time machine laboratory" and many rooms where scientists lived.

Notch kingdom, the first land that has been known to exist since Minecraft world was created by gods. This kingdom was famous for wheat agriculture, any mysterious forests located in the North East and the South West of the kingdom. King Notch had built this gorgeous kingdom surrounded with walls and secured carefully with only one gate in the South side of the kingdom to keep his citizens safe. From the main street of the Southern gate, wheat is farmed on the West side of the road next to the mysterious forest while the other side is where villagers live and raise cattle. There is also a market section next to the villagers' houses for trading and holding festivals. A statue of Notch – king of Minecraft was built in the center which divides this kingdom into four main sections. Villager's houses and a beautiful garden are located in the North East section. It also has a pagoda and a mysterious forest at the North East corner next to a graveyard where dead villagers are buried. Across the main road in the West, Notch's castle is surrounded with four forts, two of them are located in the East of the castle, the others are built in front of the castle. This castle where Notch king used to live includes a big library containing all of the historical records of the world and all of the knowledge in that time. There is also a beautiful garden inside the castle and a magnificent palace.

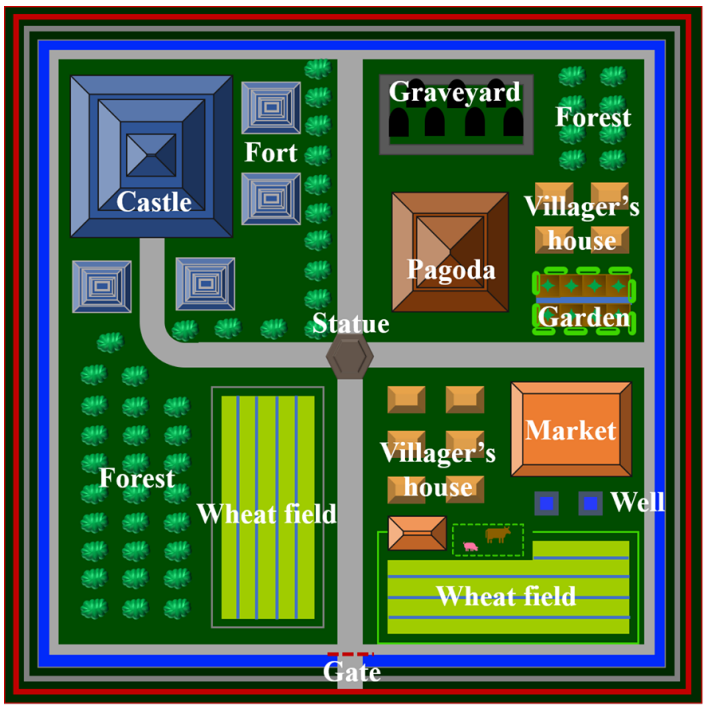


Figure 1 Map of the world

3.3. Gameplay

When the game starts, Steve – the main character is at a his room in a laboratory. Suddenly, he heard his daughter – Sarah screamming in the lab, he quickly run to the time portal in the laboratory where a time machine is developing. While the player is in time machine's experiental room, a mini-game that challenge the player to close a time portal, which Sarah was dragged into, therefore, player must complete this game according to the plot. After 30 years, he finally completed the time machine and began the journey to Notch kingdom in order to save Sarah. Although successful travel to the past, Steve didn't know the time because of the timer had been broken. At this time, the player's position would be at the Southern gate of the Notch kingdom. The player can explore and visit any buildings which had been constructed in this map such as a castle, forests, villagers' houses, pagoda, etc. However, though Steve's main mission to find Sarah, he accidentally found out that Sarah had died 68 years ago, he shocked and passed out, then returns to his room in laboratory and ends the game.

4. Software

4.1. Overall design

Python and Minecraft Pi is one of the most important software of this project. Thanks to the help from the UNIX operating system, the libraries can be easily installed to the Raspberry Pi and Python. Minecraft Pi library and Python 3 programming language are used to develop this project.

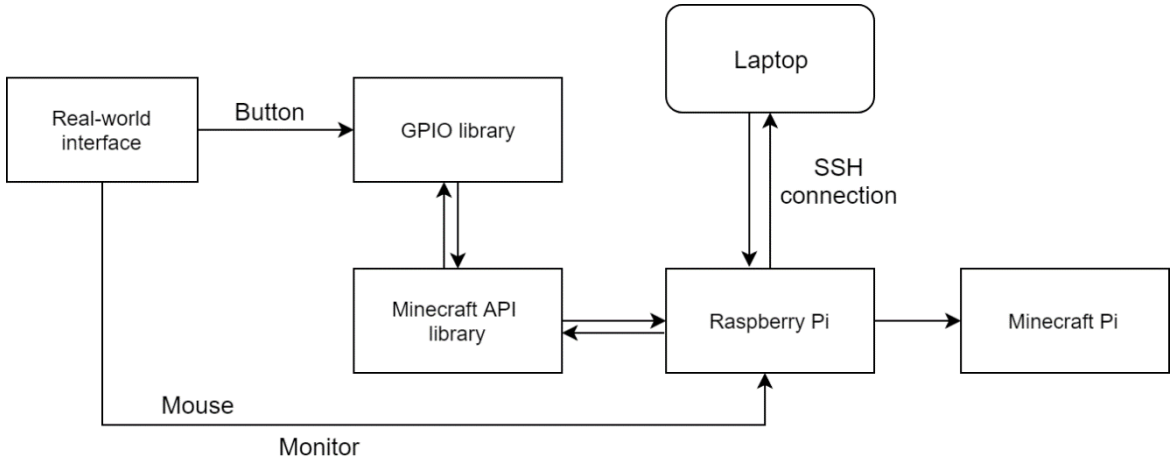


Figure 2 Software Diagram

4.2. Opening module

This project uses Python 3 to create a Minecraft world which is built in three dimensions, interfacing with Minecraft Pi API Programming library. UNIX programming is used to download and install the libraries.

4.3. Subsequent modules

A RealVNC connection has been established in order to check that if there have any mistakes in the program and to avoid breaking down the Raspberry Pi. IDLE (Python 3.7 32-bit) was used to code the world on the computer and then transfer the code to the Raspberry Pi.

5. Real-world Interface

5.1. Overall Design

The design for real-world interface consists of three tactile buttons called “Play”, “Enter” and “Exit” buttons, in order to follow the plot of the game together with the gameplay. This design takes one button for gameplay, while the others are dedicated to a real-world interface which helps the user to interact with the virtual world – Minecraft Pi. Those two buttons work as a shortcut to enter or to exit the buildings and the time machine. The “Play” button is used as a controlling button to play the game at laboratory according to the plot. This system, therefore, was designed to fulfill the given requirements as well as setting up a way for users to interact with the virtual world.

5.2. Sub-system

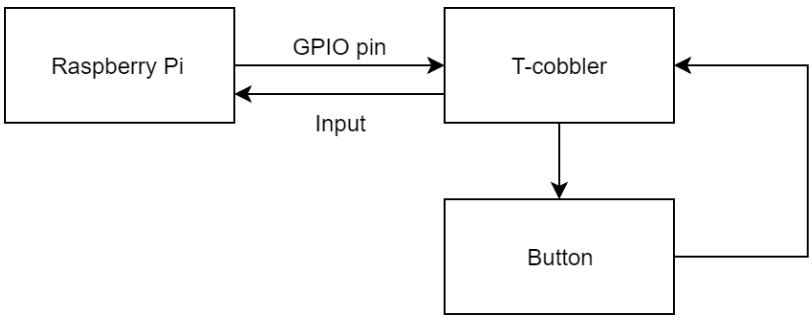


Figure 3 Concept Diagram

The sub-system of this project includes a Raspberry Pi Model 3B+, 3 tactile buttons, a T-cobbler, 10K resistors and jumper wires. One side of the T-cobbler is attached to the Raspberry Pi 3B+ while the other is plugged to the test board for the purpose of protecting the main part of the project – Raspberry Pi 3B+. Three buttons are also plugged to the test board and carefully marked with “Play”, “Enter” and “Exit” label.

5.2.1. Raspberry Pi Model 3B+

Figure 4 Raspberry Pi 3 Model B+

The heart of the project is Raspberry Pi Model B+ which was launched in 2018. This Model B+ is a microcomputer, compact size with capable of running several different operating systems including Linux-based OSes, Window 10,... especially Raspberian which the game of the project is based on. It has 1GB RAM, 1.2GHz quad-core CPU, built-in wireless and Bluetooth connection, multiple USB ports, a micro-SD card reader, Ethernet, HDMI, audio outputs and a video camera connection. Moreover, it also has GPIO that allow you to control electronic components, e.g., buttons or LEDs.

5.2.2. Assembled T-cobbler

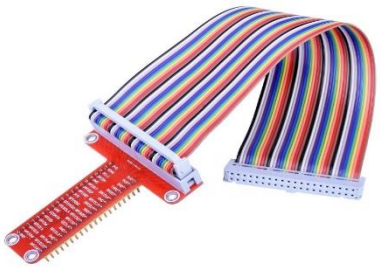


Figure 5 Assembled T-cobblers

Assembled T-cobbler for Raspberry Pi is designed for Raspberry Pi Model B Revision 1 or Revision 2. This kit includes soldered T-cobbler and 40 pin ribbon cable designed for easier connection.

Moreover, besides more convenient for connectivity, it also helps to read the labels easily and to avoid any mistakes could damage the Rasberry Pi.

5.2.3. Buttons and resistors

Three tactile switches and 10K resistors are attached to the test board. Buttons are marked carefully to distinguish from each other. 10K resistors are also plugged to the test board in order to reduce the current that could damage the Raspberry Pi.

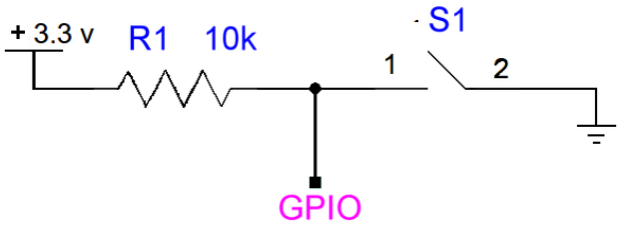


Figure 6 Button Scheme

Figure 7 Tactile button

Figure 8 10K resistors

6. Conclusion

To sum up, the project works as planned, achieved the given requirements which involves writing a plot, designing a virtual world and coding a mini-game according to the story. The game is programmed by Python 3.7, running on Raspberry Pi 3 microcomputer and interacting with the user through GPIO as buttons, monitor and mouse.

7. References

[1] R. G. Bradley, English for Computer Science Lecture Notes, Topic: “Requirements for Minecraft World Design Class”, Vietnamese German University, Binh Duong Province, Apr. 2015

[2] Stuff About Code, Minecraft API, [Online], Available: https://www.stuffaboutcode.com/p/minecraft-api-reference.html

[3] Raspberry Pi Organization, RASPBERRY PI SPECS AND BENCHMARKS: 3A+, 3B+, ZERO W, [Online], Available: https://www.raspberrypi.org/magpi/raspberry-pi-specs-benchmarks/

Appendix A: Glossary of Terms

GPIO - General Purpose Input/Output

HDMI - High-Definition Multimedia Interface

CPU - Central Processing Unit

USB - Universal Serial Bus

IDLE - Intergrated Development Environment (Python)

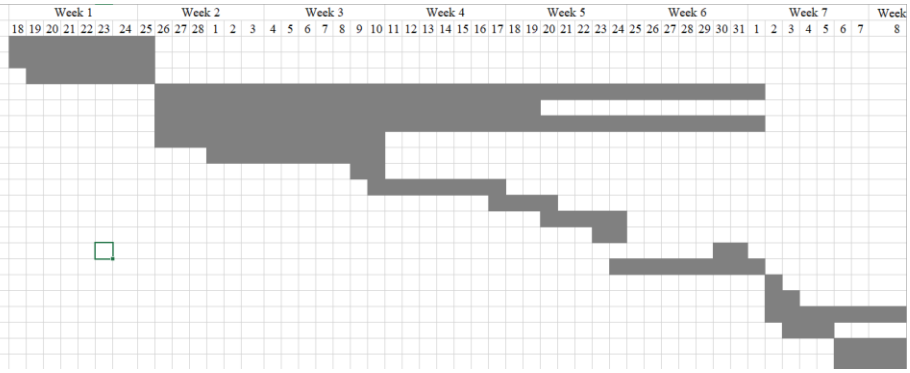
LED - Light Emitting Diode

PC - Personal Computer

API - Application Program Interface

Appendix B: Gantt Chart

TASK	START	END
STAGE 1	18/2/2019	25/2/2019
Borrow equipments	18/2/2019	25/2/2019
Analyse the requirements	19/2/2019	25/2/2019
STAGE 2	26/2/2019	1/4/2019
Search code	26/2/2019	20/3/2019
Apply code	28/2/2019	1/4/2019
Determine storyline and mini-game	26/2/2019	10/3/2019
Sketch the map	1/3/2019	10/3/2019
Presentation on design	9/3/2019	10/3/2019
Build houses, garden, well and gate	10/3/2019	17/3/2019
Build forest, wheat field and statue	17/3/2019	20/3/2019
Build graveyard, lab and market	21/3/2019	24/3/2019
Progress Reports 1	23/3/2019	24/3/2019
Progress Reports 2	30/3/2019	31/3/2019
Build pagoda, fort and castle	24/3/2019	1/4/2019
STAGE 3	2/4/2019	8/4/2019
Interface with real world	2/4/2019	2/4/2019
Start making the mini-game	2/4/2019	3/4/2019
Complete the world and apply the mini-game	2/4/2019	5/4/2019
Final Presentation Preparation	6/4/2019	8/4/2019
Final Report	6/4/2019	8/4/2019



Appendix C: Program Listing

Castle.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import Minecraft
mc=minecraft.Minecraft.create()
#######################################
#                Variables            #
#######################################
x=-112
z=-112
#######################################
#                Functions                    #
#######################################
def buildCastle(x,z):
        mc.setBlocks(x,4,z,x+80,45,z+80,0)
        mc.setBlocks(x,4,z,x+78,13,z+78,98)
        mc.setBlocks(x+4,4,z+4,x+74,13,z+74,0)
        for i in range(0,79,2):
           mc.setBlock(x,14,z+i,98)
           mc.setBlock(x+i,14,z,98)
           mc.setBlock(x+78,14,z+i,98)
           mc.setBlock(x+i,14,z+78,98)
        for i in range(0,73,2):
           mc.setBlock(x+3,14,z+3+i,98)
           mc.setBlock(x+3+i,14,z+3,98)
           mc.setBlock(x+75,14,z+3+i,98)
           mc.setBlock(x+3+i,14,z+75,98)
```

```
mc.setBlocks(x+37,4,z+75,x+41,9,z+78,0) #Gate
mc.setBlocks(x+38,10,z+75,x+40,10,z+78,0)
mc.setBlocks(x+39,11,z+75,x+39,11,z+78,0)
mc.setBlocks(x+15,4,z+15,x+63,21,z+63,4) #1st floor
#Make gold inside the castle
mc.setBlocks(x+16,5,z+16,x+62,20,z+62,41)
mc.setBlocks(x+17,6,z+17,x+61,19,z+61,0)
mc.setBlocks(x+37,4,z+63,x+41,9,z+62,0)          #Castle's gate
mc.setBlocks(x+38,10,z+63,x+40,10,z+62,0)
mc.setBlocks(x+39,11,z+63,x+39,11,z+62,0)
for i in range(16,63,9):                          #1st floor window
    mc.setBlocks(x+i,15,z+62,x+i+1,19,z+63,20)
    mc.setBlocks(x+15,15,z+i,x+16,19,z+1+i,20)
    mc.setBlocks(x+62,15,z+i,x+63,19,z+1+i,20)
    mc.setBlocks(x+i,15,z+15,x+i+1,19,z+16,20)
for i in range(15,65,2):
    mc.setBlock(x+15,22,z+i,4)
    mc.setBlock(x+i,22,z+15,4)
    mc.setBlock(x+63,22,z+i,4)
    mc.setBlock(x+i,22,z+63,4)
mc.setBlocks(x+20,22,z+20,x+58,35,z+58,1)
mc.setBlocks(x+21,20,z+21,x+57,34,z+57,0)
mc.setBlocks(x+21,21,z+21,x+57,21,z+57,20)
for i in range(20,60,2):
    mc.setBlock(x+20,36,z+i,1)
    mc.setBlock(x+i,36,z+20,1)
    mc.setBlock(x+58,36,z+i,1)
    mc.setBlock(x+i,36,z+58,1)
mc.setBlocks(x+25,35,z+25,x+53,35,z+53,20)
mc.setBlock(x+39,35,z+39,1)
for i in range(25,56,9):
```

```
    mc.setBlocks(x+i,29,z+58,x+i+1,32,z+58,20)
    mc.setBlocks(x+20,29,z+i,x+20,32,z+1+i,20)
    mc.setBlocks(x+58,29,z+i,x+58,32,z+1+i,20)
    mc.setBlocks(x+i,29,z+20,x+i+1,32,z+20,20)
#Flag
mc.setBlocks(x+39,36,z+39,x+39,45,z+39,85)
mc.setBlocks(x+37,43,z+37,x+37,45,z+37,35,14)
mc.setBlocks(x+38,43,z+38,x+38,45,z+38,35,14)
mc.setBlocks(x+39,43,z+39,x+39,45,z+39,35,14)

mc.setBlocks(x+37,4,z+20,x+41,5,z+77,35,14)        #Red cape
mc.setBlocks(x+37,5,z+78,x+41,5,z+78,67,3)
mc.setBlocks(x+37,4,z+79,x+41,4,z+79,67,3)
mc.setBlocks(x+36,6,z+20,x+42,9,z+23,35,15)#King chair
mc.setBlocks(x+37,6,z+20,x+41,9,z+23,41)
mc.setBlocks(x+38,7,z+22,x+40,9,z+23,0)
for i in range(2):
    mc.setBlocks(x+39-i,11-i,z+21,x+39+i,11-i,z+21,41)
for i in range(3):
    mc.setBlocks(x+39-i,12-i,z+20,x+39+i,12-i,z+20,35,15)
for i in range(23,55,2):
    mc.setBlocks(x+i,22,z+23,x+i,25,z+50,47)#Books
mc.setBlocks(-55,22,-55,-91,25,-55,47)#Books
for i in range(0,17):
    mc.setBlocks(x+54,5+i,z+36+i,x+57,5+i+2,z+36+i,1)
    mc.setBlocks(x+55,5+i,z+36+i,x+56,5+i,z+36+i,67,2)#Stair
    mc.setBlocks(x+55,5+i+1,z+36+i,x+56,5+i+4,z+36+i,0)
mc.setBlock(x+17,6,z+17,50,5)        #Torch
mc.setBlock(x+61,6,z+17,50,5)
mc.setBlock(x+17,6,z+61,50,5)
mc.setBlock(x+61,6,z+61,50,5)
```

```
        for j in range(25,60,3):          #Tree
            mc.setBlock(x+43,6,z+j,17)
            mc.setBlock(x+35,6,z+j,17)
            mc.setBlocks(x+35,7,z+j,x+35,8,z+j,18)
            mc.setBlocks(x+43,7,z+j,x+43,8,z+j,18)
buildCastle(x,z)
```

clearland.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc = minecraft.Minecraft.create()
#Clearland
mc.postToChat("Clearing land...")
mc.setBlocks(-128,0,128,128,64,-128,0)
mc.postToChat("Clear land complete")
mc.postToChat("Placing grassblock to the world...")
mc.setBlocks(-128,0,128,128,3,-128,2)
mc.postToChat("Done")
mc.postToChat("The world is ready")
```

farm.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc = minecraft.Minecraft.create()
```

```
#######################################
#                Functions            #
#######################################
def wheat(x,z,a,b):
    p=mc.player.getPos()
    y=4
    mc.setBlocks(x,y-1,z,x+a,y-1,z+b,17,0)#gỗ
    mc.setBlocks(x,y,z,x,y,z+b,85)#hàng rào
    mc.setBlocks(x,y,z,x+a,y,z,85)
    mc.setBlocks(x+a,y,z,x+a,y,z+b,85)
    mc.setBlocks(x,y,z+b,x+a,y,z+b,85)
    for i in range(1,a-1,5):
        if a-1-i<5:
            k=a-1-i
            mc.setBlocks(x+i,y-1,z+1,x+i+k,y-1,z+b-1,60,7)
            mc.setBlocks(x+i,y,z+1,x+i+k,y,z+b-1,59,7)
        else:
            mc.setBlocks(x+i,y-1,z+1,x+i+3,y-1,z+b-1,60,7)
            mc.setBlocks(x+i,y,z+1,x+i+3,y,z+b-1,59,7)
            mc.setBlocks(x+i+4,y-1,z+1,x+i+4,y-1,z+b-1,9)
        print(i)
#######################################
#                Indicator            #
#######################################
mc.postToChat("Building the southeast wheat field...")
wheat(-55,19,45,90)
mc.postToChat("Done!")
mc.postToChat("Building the southwest wheat field...")
wheat(9,60,100,50)
mc.postToChat("Done!")
mc.postToChat("Complete to build the farm.")
```

forest.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
import random
from mcpi import minecraft
mc = minecraft.Minecraft.create()
########################################
#           Functions              #
########################################
def tree(x,z):
    y=4
    mc.setBlocks(x-1,y+5,z-1,x+1,y+6,z+1,18) #Top-leave
    mc.setBlocks(x-2,y+3,z-2,x+2,y+4,z+2,18) #Bottom-leave
    mc.setBlocks(x,y,z,x,y+5,z,17,0)        #Log
#l : Initial x
#m : Initial y
#a : Length of x
#b : Length of y
def forest(l,m,a,b):
    #Clear area
    mc.setBlocks(l-3,4,m-3,l+3+a,10,m+3+b,0)
    for i in range(0,a,5):
        for j in range(0,b,5):
            print(i,j)
            x=random.randint(l+i,l+i+2)
            z=random.randint(m+j,m+j+2)
            tree(x,z)
#For main program
forest(-112,10,55,100)
```

```
forest(77,-110,35,40)
```

fort.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc=minecraft.Minecraft.create()
########################################
#                Functions             #
########################################
def fort(x,z):
    mc.setBlocks(x,4,z,x+7,14,z+7,1)
    mc.setBlocks(x-3,15,z-3,x+10,18,z+10,1)
    mc.setBlocks(x-2,16,z-2,x+9,18,z+9,0)
    for i in range(-3,10,2):
        mc.setBlock(x+i,19,z-3,1)
        mc.setBlock(x-3,19,z+i,1)
        mc.setBlock(x+10,19,z+i+1,1)
        mc.setBlock(x+i+1,19,z+10,1)
########################################
#                Main                  #
########################################
fort(-66,-20)
fort(-87,-20)
fort(-28,-45)
fort(-28,-102)
```

game.py

```
#Vietnamese German University
```

```
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)  #Establish input
GPIO.setup(13, GPIO.IN) #Play button
mc=minecraft.Minecraft.create()
#m=0 Face to the west
#m=-1 Face to the east
#gameover=True Build the minigame
#gameover=False Play the game
########################################
#           Functions              #
########################################
def Minigame(x,y,z,v,m,gameover=True): #v is the hard of the game
    p=0 #Point
    k=y #Store value of y
    mc.setBlocks(x+1+m,k-2,z,x+1+m,k+5,z+1,42) #Background
    mc.setBlocks(x+1+m,k+4,z,x+1+m,k+4,z+1,35,14) #Back redline
    mc.setBlocks(x-m,k-2,z,x-m,k-1,z+1,89) #Glowstone
    mc.setBlocks(x-m,k,z,x-m,k,z+1,35,14) #Front redline
    mc.setBlocks(x-m,k+1,z,x-m,k+5,z+1,20) #Glass
    if not gameover:
        mc.postToChat('Help Steve to stable the energy column.')
        time.sleep(3)
        mc.postToChat('Keep the energy level under control.')
        time.sleep(3)
        mc.postToChat('Please hold the play button')
        time.sleep(3)
```

```
    mc.postToChat('So let begin in...')
    time.sleep(1)
    mc.postToChat('1')
    time.sleep(1)
    mc.postToChat('2')
    time.sleep(1)
    mc.postToChat('3')
    time.sleep(0.5)
    mc.postToChat('Let goooo..')
while not gameover:
    if(GPIO.input(13) == False):   # Check input
        if k>=y:
            print("pressed")
            k-=1
            mc.setBlocks(x-m,y-2,z,x-m,y+5,z+1,20) # Reset the glass
            mc.setBlocks(x-m,y-2,z,x-m,k-1,z+1,89) # Reset the glow
            mc.setBlocks(x-m,k,z,x-m,k,z+1,35,14)  # Put the red bar
    time.sleep(v) # Raise the bar after v second(s)
    k+=1
    mc.setBlocks(x-m,y-2,z,x-m,y+5,z+1,20)  # Reset the glass
    mc.setBlocks(x-m,y-2,z,x-m,k-1,z+1,89)  # Reset the glow
    mc.setBlocks(x-m,k,z,x-m,k,z+1,35,14)   # Put the red bar

    if k==y+5: # Lose condition
        gameover=True
        mc.postToChat('YOU LOSE')
        mc.postToChat("Let's play again!")
        return Minigame(x,y,z,v,m,False) #Restart the game
    elif p==10: # Win condition
        mc.postToChat("YOU'RE SAFE")
        gameover=True
```

```
        else:
            p+=1
#######################################
#             Main                    #
#######################################
Minigame(10,-5,37,0.5,0,True)
Minigame(10,-5,48,0.5,0,True)
Minigame(-10,-5,37,0.5,-1,True)
Minigame(-10,-5,48,0.5,-1,True)
```

Garden.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc = minecraft.Minecraft.create()
x = 69
y = 4
z = -27
a = 20
b = 40
#######################################
#           Functions                 #
#######################################
def buildGarden(x,z):
        #
        mc.setBlocks(x,y,z,x+b,y,z+a,17) #Build surrouding log
        mc.setBlocks(x,y+1,z,x+b,y+6,z+a,85) #Place fences
        mc.setBlocks(x+1,y+1,z,x+b-1,y+5,z+a,0) #Cave inside
        mc.setBlocks(x,y+1,z+1,x+b,y+5,z+a-1,0) #Cave inside
```

```
        mc.setBlocks(x,y+7,z,x+b,y+7,z+a,18) #Set leave to the top
        for i in range(1,a,2):
                if i < a-1:
                        mc.setBlocks(x+1,y,z+i,x+b-1,y,z+i,12) #Set sand
                        mc.setBlocks(x+1,y,z+i+1,x+b-1,y,z+i+1,9) #Set water
                        #Set leave & log
                        mc.setBlocks(x-1,y+4,z+i+1,x-1,y+7,z+i+1,18)
                        mc.setBlocks(x+b+1,y+4,z+i+1,x+b+1,y+7,z+i+1,18)
                        mc.setBlocks(x,y+7,z+i+1,x+b,y+7,z+i+1,17)
                        #Set sugarcane
                        mc.setBlocks(x+1,y+1,z+i,x+b-1,y+4,z+i,83)
                else:
                        mc.setBlocks(x+1,y,z+i,x+b-1,y,z+i,12) #Set sand
                        mc.setBlocks(x+1,y+1,z+i,x+b-1,y+4,z+i,83) #Set sugarcane
########################################
#                       Main                        #
########################################
buildGarden(x,z)
```

graveyard.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft APIimport random
from mcpi import minecraft
mc = minecraft.Minecraft.create()
########################################
#                       Variables               #
########################################
x = 13
z = -73
```

```
a = 60
b = 30
#######################################
#                Functions                #
#######################################
def buildGraveYard(x,z):
    mc.setBlocks(x,4,z,x+a,4,z-b,85) #Fence
    mc.setBlocks(x,4+1,z,x+a,4+1,z-b,44)
    mc.setBlocks(x+1,4,z-1,x+59,4+1,z-29,0)

    mc.setBlocks(x,4,z-13,x,4+5,z-17,17) #Gate
    mc.setBlocks(x,4,z-14,x,4+4,z-16,0)

    mc.setBlocks(x,4-1,z-14,x+a-1,4-1,z-16,13) #Gravel road

    for i in range(0,int(b/2-5),6):
        for j in range(0,a,5):
            mc.setBlock(x+3+j,4,z-3-i,4)
            mc.setBlock(x+3+j,4+1,z-3-i,44)
            mc.setBlock(x+3+j,4,z-6-i,random.randint(37,38))
            mc.setBlocks(x+3+j,4,z-4-i,x+3+j,4,z-5-i,44)

    for i in range(0,int(b/2-5),6):
        for j in range(0,a,5):
            mc.setBlock(x+3+j,4,z-21-i,4)
            mc.setBlock(x+3+j,4+1,z-21-i,44)
            mc.setBlock(x+3+j,4,z-18-i,random.randint(37,38))
            mc.setBlocks(x+3+j,4,z-19-i,x+3+j,4,z-20-i,44)
#######################################
#                Main                #
#######################################
```

buildGraveYard(x,z)

house.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft APIimport random
from mcpi import minecraft
mc = minecraft.Minecraft.create()
def house(x,z):
    air=0
    plank=5
    cobblestone=4
    stair=53
    wood=17
    door=64
    l=10
    w=7
    x-=1
    y=4
    z-=1
    mc.setBlocks(x+1,y,z+1,x+l,y+7,z+w,5)#basic
    mc.setBlocks(x+5,y+1,z+1,x+6,y+4,z+1,0)#cửa
    mc.setBlocks(x+2,y+1,z+2,x+l-1,y+8,z+w-1,0)
    mc.setBlocks(x+1,y+7,z+1,x+l,y+7,z+1,0)
    mc.setBlocks(x+1,y+7,z+w,x+l,y+7,z+w,0)
    for i in range(0,4):#cột gỗ
        mc.setBlocks(x+1+3*i,y,z+1,x+1+3*i,y+5,z+1,17,0)
        mc.setBlocks(x+1+3*i,y,z+w,x+1+3*i,y+5,z+w,17,0)
    mc.setBlocks(x+l-1,y+2,z+1,x+l-2,y+3,z+1,20)#cửa sổ
    mc.setBlocks(x+2,y+2,z+1,x+3,y+3,z+1,20)
```

```
    mc.setBlocks(x+1,y+2,z+3,x+1,y+3,z+5,20)
    mc.setBlocks(x+l,y+2,z+3,x+l,y+3,z+5,20)
    mc.setBlocks(x+3,y+1,z+w,x+3,y+4,z+w,20)
    mc.setBlocks(x+l-2,y+1,z+w,x+l-2,y+4,z+w,20)
    mc.setBlocks(x+4,y+1,z+w-1,x+7,y+3,z+w-1,4)
    mc.setBlocks(x+5,y+1,z+w-1,x+6,y+12,z+w-1,4)
    mc.setBlocks(x+5,y+1,z+w-1,x+6,y+2,z-w-1,0)
    mc.setBlock(x+4,y+1,z+w-2,67,2)
    mc.setBlock(x+7,y+1,z+w-2,67,2)
    mc.setBlock(x+4,y+3,z+w-1,67,0)
    mc.setBlock(x+7,y+3,z+w-1,67,1)
    #Bed
    mc.setBlock(x+2,y+1,z+w-1,26,8)
    mc.setBlock(x+2,y+1,z+w-2,26,0)
    mc.setBlock(x+l-1,y+1,z+w-1,26,8)
    mc.setBlock(x+l-1,y+1,z+w-2,26,0)
    mc.setBlock(x+2,y+1,z+2,54,5)
    mc.setBlock(x+l-1,y+1,z+2,54,4)
    mc.setBlock(x+2,y+1,z+3,58)
    mc.setBlock(x+l-1,y+1,z+3,58)
    #Roof
    for i in range(0,4):
        mc.setBlocks(x,y+5+i,z+i,x+l+1,y+5+i,z+i,53,2)
        mc.setBlocks(x,y+5+i,z+w+1-i,x+l+1,y+5+i,z+w+1-i,53,3)
    mc.setBlocks(x,y+8,z+4,x+l+1,y+8,z+4,5)#mái
    mc.setBlocks(x+2,y,z+2,x+l-1,y,z+w-1,4)#nền
    mc.setBlocks(x+5,y,z+1,x+6,y,z+1,53,2)#trym
house(9,50)
house(9,36)
house(9,22)
house(24,50)
```

```
house(24,36)
house(24,22)
house(39,50)
house(39,36)
house(39,22)
#Northeast section
house(73,-42)
house(73,-54)
house(73,-66)
house(88,-42)
house(88,-54)
house(88,-66)
```

lab.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc = minecraft.Minecraft.create()
x = -15
y = -6
z = -1
########################################
#                Functions                #
########################################
def buildTimePortal(x,y,z):
        #Time portal
        mc.setBlocks(x,y,z+1,x+5,y+5,z+1,89)
        mc.setBlocks(x+1,y,z+1,x+4,y+4,z+1,0)
        mc.setBlocks(x+1,y+5,z+1,x+4,y+5,z+1,9)
```

```
        mc.setBlocks(x,y,z,x+5,y+5,z,42)
        mc.setBlocks(x+1,y+1,z,x+4,y+4,z,0)
#####################################
#                Functions                #
#####################################
def buildTimeMachine(x,y,z):
        #Iron foudation
        mc.setBlocks(x+1,y,z,x+2,y,z+3,42)
        #Bottom-Side
        mc.setBlocks(x,y,z,x,y,z+3,67,4)
        mc.setBlocks(x+3,y,z,x+3,y,z+3,67,5)
        #Top-side
        mc.setBlocks(x,y+1,z,x,y+1,z+3,44)
        mc.setBlocks(x+3,y+1,z,x+3,y+1,z+3,44)
        #Control panelminigame.
        mc.setBlocks(x+1,y+1,z+3,x+2,y+1,z+3,42)
        #Chair
        mc.setBlocks(x+1,y+1,z+1,x+2,y+1,z+1,67,3)
#####################################
#                Functions                #
#####################################
def tree(x,y,z):
        mc.setBlock(x,y,z,17)
        mc.setBlocks(x,y+1,z,x,y+2,z,18)
#####################################
#                Functions                #
#####################################
def buildLaboratory(x,z):
        mc.setBlocks(x,y,z,x+31,y+7,z+60,42)            #Create a big cube
        mc.setBlocks(x+14,y,z+35,x+17,y+7,z+56,89)      #Lighting main lab
        mc.setBlocks(x+14,y,z+2,x+17,y+7,z+31,89)       #Lighting hall
```

```
mc.setBlocks(x+12,y+1,z+1,x+19,y+6,z+32,0)        #Make room
#Entrance to main lab
mc.setBlocks(x+15,y+1,z+33,x+16,y+3,z+33,0)
mc.setBlocks(x+15,y+4,z+33,x+16,y+4,z+33,44)
for i in range(0,45,11):
        #Lighting system
        mc.setBlocks(x+2,y,z+i+2,x+9,y+7,z+i+9,89)
        #Make room
        mc.setBlocks(x+1,y+1,z+i+1,x+10,y+6,z+i+10,0)
        #Make door
        mc.setBlocks(x+11,y+1,z+i+5,x+11,y+2,z+i+6,0)
        mc.setBlocks(x+11,y+3,z+i+5,x+11,y+3,z+i+6,44)
        #Table and chair
        mc.setBlocks(x+3,y+1,z+i+4,x+6,y+1,z+i+7,42)
        mc.setBlocks(x+4,y+1,z+i+4,x+4,y+1,z+i+7,0)
        mc.setBlocks(x+3,y+1,z+i+5,x+3,y+1,z+i+6,67,1)
        tree(x+1,y+1,z+i+1)             #Tree for decoration
        tree(x+1,y+1,z+i+10)  #Tree for decoration
        tree(x+10,y+1,z+i+1)  #Tree for decoration
        tree(x+10,y+1,z+i+10)           #Tree for decoration
        mc.setBlock(x+5,y+1,z+i+10,58)      #Crafting table
        mc.setBlock(x+6,y+1,z+i+10,54)      #Chest
        #Lighting system
        mc.setBlocks(x+22,y,z+i+2,x+29,y+7,z+i+9,89)
        #Make room
        mc.setBlocks(x+21,y+1,z+i+1,x+30,y+6,z+i+10,0)
        #Make door
        mc.setBlocks(x+20,y+1,z+i+5,x+20,y+2,z+i+6,0)
        mc.setBlocks(x+20,y+3,z+i+5,x+20,y+3,z+i+6,44)
        #Table and chair
        mc.setBlocks(x+25,y+1,z+i+4,x+28,y+1,z+i+7,42)
```

```
                mc.setBlocks(x+27,y+1,z+i+4,x+27,y+1,z+i+7,0)
                mc.setBlocks(x+28,y+1,z+i+5,x+28,y+1,z+i+6,67,0)
                tree(x+30,y+1,z+i+1) #Tree for decoration
                tree(x+30,y+1,z+i+10)          #Tree for decoration
                tree(x+21,y+1,z+i+1) #Tree for decoration
                tree(x+21,y+1,z+i+10)          #Tree for decoration
                mc.setBlock(x+25,y+1,z+i+10,58)    #Crafting table
                mc.setBlock(x+26,y+1,z+i+10,54)    #Chest
        mc.setBlocks(x+1,y+1,z+34,x+30,y+6,z+59,0) #Clear main lab
        #Time portal
        buildTimePortal(x+13,y+1,z+58)
        #Time machine
        buildTimeMachine(x+14,y+1,z+51)
buildLaboratory(x,z)
```

main.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
import clearland
import wallandroad
import tree
import forest
import farm
import house
import market
import well
import garden
import graveyard
import pagoda
```

```
import castle
import fort
import statue
import lab

market.py
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
import random
from mcpi import minecraft
mc = minecraft.Minecraft.create()
x = 64
y = 4
z = 8
a = 45
b = 45
topcolor = [14,15,11,7]
def buildVendor(x,z):
        mc.setBlocks(x,y-1,z,x+4,y-1,z+3,4) #Build foundation

        mc.setBlocks(x,y,z,x+4,y,z+3,5)
        mc.setBlocks(x+1,y,z+1,x+3,y,z+2,0)

        mc.setBlocks(x,y+3,z-1,x+4,y+4,z+4,35,0) #Build top
        color = random.choice(topcolor) #Random color
        for i in range(0,6,2):
                mc.setBlocks(x+i,y+3,z-1,x+i,y+4,z+4,35,color)
        mc.setBlocks(x,y+4,z-1,x+5,y+4,z-1,0) #Delete 2 left-over
        mc.setBlocks(x,y+4,z+4,x+5,y+4,z+4,0)
```

```
        mc.setBlocks(x,y+1,z,x+4,y+3,z+3,85) #Build fence
        mc.setBlocks(x+1,y+1,z,x+3,y+3,z+3,0)
        mc.setBlocks(x,y+1,z+1,x+5,y+3,z+2,0)

for j in range(0,b,10):
        for i in range(0,a,10):
                buildVendor(x+i,z+j)
```

minigame.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)  #Establish input
GPIO.setup(13, GPIO.IN) #Play button
mc=minecraft.Minecraft.create()
#m=0 Face to the west
#m=-1 Face to the east
#gameover=True Build the minigame
#gameover=False Play the game
def minigame(x,y,z,v,m,gameover=True): #v is the hard of the game
    p=0 #Point
    k=y #Store value of y
    mc.setBlocks(x+1+m,k-2,z,x+1+m,k+5,z+1,42) #Background
    mc.setBlocks(x+1+m,k+4,z,x+1+m,k+4,z+1,35,14) #Back redline
    mc.setBlocks(x-m,k-2,z,x-m,k-1,z+1,89) #Glowstone
    mc.setBlocks(x-m,k,z,x-m,k,z+1,35,14) #Front redline
```

```
mc.setBlocks(x-m,k+1,z,x-m,k+5,z+1,20) #Glass
if not gameover:
    mc.postToChat('Help Steve to stablizing the energy column.')
    time.sleep(3)
    mc.postToChat('Keep the energy level under control.')
    time.sleep(3)
    mc.postToChat('So let begin in...')
    time.sleep(1)
    mc.postToChat('1')
    time.sleep(1)
    mc.postToChat('2')
    time.sleep(1)
    mc.postToChat('3')
    time.sleep(0.5)
    mc.postToChat('Let goooo..')
while not gameover:
    if(GPIO.input(13) == False):   # Check input
        if k>=y:
            print("pressed")
            k-=1
            mc.setBlocks(x-m,y-2,z,x-m,y+5,z+1,20) # Reset the glass
            mc.setBlocks(x-m,y-2,z,x-m,k-1,z+1,89) # Reset the glow
            mc.setBlocks(x-m,k,z,x-m,k,z+1,35,14)  # Put the red bar
    time.sleep(v) # Raise the bar after v second(s)
    k+=1
    mc.setBlocks(x-m,y-2,z,x-m,y+5,z+1,20)  # Reset the glass
    mc.setBlocks(x-m,y-2,z,x-m,k-1,z+1,89)  # Reset the glow
    mc.setBlocks(x-m,k,z,x-m,k,z+1,35,14)   # Put the red bar

    if k==y+5: # Lose condition
        gameover=True
```

```
            mc.postToChat('YOU LOSE')
            mc.postToChat("Let's play again!")
            minigame(x,y,z,v,m,False) #Restart the game
        elif p==10: # Win condition
            mc.postToChat('YOU WIN')
            gameover=True
        else:
            p+=1
```

pagoda.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc=minecraft.Minecraft.create()
x=19
z=-64
def denlong(x,y,z):
    mc.setBlock(x,y,z,85)
    mc.setBlock(x,y-1,z,89)
def tree(x,z):
    y=4
    mc.setBlocks(x-1,y+5,z-1,x+1,y+6,z+1,18)#lá
    mc.setBlocks(x-2,y+3,z-2,x+2,y+4,z+2,18)#lá
    mc.setBlocks(x,y,z,x,y+5,z,17,0)#thân
mc.setBlocks(x,4,z,x+38,6,z+38,4)#thềm đá
for i in range(3):#bậc tam cấp
    mc.setBlocks(x+17,4,z+41-i,x+21,4+i,z+41-i,4)
    mc.setBlocks(x+18,4,z+41-i,x+20,4+i,z+41-i,67,3)
for i in range(0,3,2):#nóc tầng trệt
```

```
    mc.setBlocks(x+4+i,13+i/2,z+4+i,x+34-i,13+i/2,z+34-i,44,2)#ván gỗ
    mc.setBlocks(x+5+i,13+i/2,z+5+i,x+33-i,13+i/2,z+33-i,4)#đá
mc.setBlocks(x+5,7,z+5,x+33,12,z+33,17)#tàng trệt
mc.setBlocks(x+6,8,z+5,x+15,11,z+33,35)#tường trắng tầng trệt
mc.setBlocks(x+23,8,z+5,x+32,11,z+33,35)
mc.setBlocks(x+5,8,z+6,x+33,11,z+15,35)
mc.setBlocks(x+5,8,z+23,x+33,11,z+32,35)
mc.setBlocks(x+5,8,z+17,x+33,11,z+21,35)
mc.setBlocks(x+17,8,z+5,x+21,11,z+33,35)
mc.setBlocks(x+6,7,z+6,x+32,12,z+32,0)
denlong(x+6,12,z+6);denlong(x+6,12,z+32);denlong(x+32,12,z+6);denlong(x+32,12,z+32)
mc.setBlocks(x+17,7,z+33,x+21,11,z+33,0)#cửa
mc.setBlocks(x+6,6,z+6,x+32,6,z+32,89)#đá sáng
mc.setBlocks(x+7,15,z+7,x+31,15,z+31,85)#hàng rào tầng 1
mc.setBlocks(x+8,15,z+8,x+30,15,z+30,0)
for i in range(0,3,2):#nóc tầng 1
    mc.setBlocks(x+9+i,21+i/2,z+9+i,x+29-i,21+i/2,z+29-i,44,2)#ván gỗ
    mc.setBlocks(x+10+i,21+i/2,z+10+i,x+28-i,21+i/2,z+28-i,4)#đá
mc.setBlocks(x+11,15,z+11,x+27,21,z+27,17)#tầng 1
mc.setBlocks(x+11,15,z+12,x+27,21,z+26,35)#tường trắng tầng 1
mc.setBlocks(x+12,15,z+11,x+16,21,z+27,35)
mc.setBlocks(x+22,15,z+11,x+26,21,z+27,35)
mc.setBlocks(x+17,15,z+11,x+21,21,z+11,35)
mc.setBlocks(x+18,15,z+27,x+20,21,z+27,0)#cửa
mc.setBlocks(x+12,15,z+12,x+26,21,z+26,0)
denlong(x+12,21,z+12);denlong(x+12,21,z+26);denlong(x+26,21,z+12);denlong(x+26,21,z+26)
mc.setBlocks(x+12,14,z+12,x+26,14,z+26,89)#đá sáng
mc.setBlocks(x+13,23,z+13,x+25,23,z+25,44,2)
for i in range(0,18,7):
    tree(x+i,z+44)
    tree(x+i,z+51)
```

```
    tree(x+24+i,z+44)
    tree(x+24+i,z+51)
```

statue.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc = minecraft.Minecraft.create()
def statue(x,z):
    p=mc.player.getPos()
    y= 4
    mc.setBlocks(x,y,z,x+8,y,z+8,44,5)#thềm
    mc.setBlocks(x+1,y,z+1,x+7,y,z+7,89)
    mc.setBlocks(x+2,y,z+2,x+6,y,z+6,5)
    mc.setBlocks(x+4,y,z+4,x+5,y+9,z+4,4)#người
    mc.setBlocks(x+4,y+8,z+4,x+5,y+9,z+5,98)    #Head
    mc.setBlocks(x+4,y+5,z+4,x+5,y+7,z+4,42)    #Body
    mc.setBlocks(x+4,y+1,z+4,x+5,y+1,z+4,42)    #Shoe
    mc.setBlocks(x+6,y+5,z+4,x+6,y+7,z+4,98)#tay trái
    mc.setBlocks(x+6,y+5,z+5,x+6,y+5,z+8,85)
    mc.setBlocks(x+3,y+7,z+4,x+3,y+7,z+6,98)#tay phải
    mc.setBlocks(x+3,y+8,z+6,x+3,y+13,z+6,85)#cột cờ
    mc.setBlocks(x+2,y+11,z+6,x+2,y+13,z+6,35,14)#cờ
    mc.setBlocks(x+1,y+11,z+5,x+1,y+13,z+5,35,14)#cờ
    mc.setBlocks(x,y+11,z+4,x,y+13,z+4,35,14)#cờ
statue(-4,-4)
```

tree.py

```
#Vietnamese German University
#By Team 4
```

```
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc = minecraft.Minecraft.create()
def growTree(x,z,direction):
        y = 4
        #Bottom
        mc.setBlocks(x-1,y+3,z,x+1,y+3,z,18)
        mc.setBlocks(x,y+3,z-1,x,y+3,z+1,18)
        #Top
        mc.setBlocks(x,y+6,z-1,x,y+6,z+1,18)
        mc.setBlocks(x-1,y+6,z,x+1,y+6,z,18)
        mc.setBlocks(x+1,y+4,z+1,x-1,y+5,z-1,18)
        #Log
        mc.setBlocks(x,y,z,x,y+5,z,17)
        #Torch
        if direction:
                #For north-south direction
                mc.setBlock(x+1,y+2,z,50,1)
                mc.setBlock(x-1,y+2,z,50,2)
        else:
                #For east-west direction
                mc.setBlock(x,y+2,z-1,50,4)
                mc.setBlock(x,y+2,z+1,50,3)
#West
mc.postToChat("West")
for i in range(0,110,11):
        for j in range(0,9,8):
                growTree(-10-i,4-j,False)
#East
mc.postToChat("East")
```

```
for i in range(0,110,11):
        for j in range(0,9,8):
                growTree(109-i,4-j,False)
#South
mc.postToChat("South")
for i in range(0,110,11):
        for j in range(0,9,8):
                growTree(-4+j,109-i,True)
#North
mc.postToChat("North")
for i in range(0,110,11):
        for j in range(0,9,8):
                growTree(-4+j,-10-i,True)
```

wallandroad.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc = minecraft.Minecraft.create()
x = -3
y = 13
z = 119
#Roadmap
mc.postToChat("Setting road and pathway...")
mc.setBlocks(-2,3,-128,2,3,128,13)   #Set gravel south-north
mc.setBlocks(-128,3,-2,128,3,2,13)   #Set gravel west-east
mc.setBlocks(-10,3,10,10,3,-10,13)   #Set roundabout gravel
mc.setBlocks(3,3,31,48,3,33,13)      #Top villager's house path
mc.setBlocks(2,3,45,48,3,47,13)      #Bottom villager's house path
```

```
mc.setBlocks(-75,3,-3,-71,3,-33,13) #Castle's path
mc.setBlocks(12,3,-89,3,3,-87,13)     #Gravelyard's path
mc.setBlocks(39,3,-22,37,3,-3,13)     #Pagoda's path
mc.setBlocks(61,3,-63,65,3,-3,13)     #North-east villager main path
mc.setBlocks(98,3,-46,66,3,-44,13)   #Top villager's house path
mc.setBlocks(98,3,-56,66,3,-54,13)
mc.postToChat("Done!")
#Set stone wall
mc.postToChat("Building wall...")
mc.setBlocks(119,4,119,-119,12,-118,1)
mc.setBlocks(116,4,116,-116,12,-115,0)
mc.postToChat("Done!")
#Set gate
mc.postToChat("Building gate...")
for i in range(0,6,1):
        if i >= 3: k = 3
        else: k = i
        mc.setBlocks(x+k,y+k,z,x+6-k,y+k,z-2,1)

for i in range(0,3,1):
        mc.setBlocks(x+i+1,y+i-1,z,x+5-i,y+i-1,z-2,0)
mc.setBlocks(x+1,4,z,x+5,y-1,z-2,0)
for i in range(0,3,1):
        mc.setBlocks(x+i+1,y+i-1,z-1,x+5-i,y+i-1,z-1,85)
mc.setBlocks(x+1,y-2,z-1,x+5,y-5,z-1,85,3)
mc.postToChat("Done!")
mc.postToChat("Roadmap and wall have been completed.")
```

well.py

```
from mcpi import minecraft
mc=minecraft.Minecraft.create()
```

```
def buildWell(x,z):
        mc.setBlocks(x,4,z,x+4,4,z+4,4)
        mc.setBlocks(x+1,4,z+1,x+3,4,z+3,9)
        mc.setBlocks(x,5,z,x+4,8,z+4,85)
        mc.setBlocks(x,5,z+1,x+4,8,z+3,0)
        mc.setBlocks(x+1,5,z,x+3,8,z+4,0)
        mc.setBlocks(x,9,z,x+4,9,z+4,44,2)
buildWell(74,54)
buildWell(94,54)
```

tour.py

```
#Vietnamese German University
#By Team 4
#Building function for castle
#Connections used: Minecraft API
from mcpi import minecraft
mc = minecraft.Minecraft.create()
from lab import buildTimeMachine
import time
import game
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)  #Establish input
GPIO.setup(6,GPIO.IN) #Enter button
GPIO.setup(5,GPIO.IN) #Exit button
def Move(x1,y1,z1,x2,y2,z2):
        t = 0.04
        if (x2 - x1 > 0):
                while x1 < x2:
                        mc.player.setPos(x1,y1,z1)
                        x1 = x1 + 0.2
                        time.sleep(t)
```

```
    elif (x2 - x1 < 0):
        while x1 > x2:
            mc.player.setPos(x1,y1,z1)
            x1 = x1 - 0.2
            time.sleep(t)
    if (z2 - z1 > 0):
        while z1 < z2:
            mc.player.setPos(x1,y1,z1)
            z1 = z1 + 0.2
            time.sleep(t)
    elif (z2 - z1 < 0):
        while z1 > z2:
            mc.player.setPos(x1,y1,z1)
            z1 = z1 - 0.2
            time.sleep(t)
#SCENCE 1 - LABORATORY
mc.postToChat("Let's take a tour around this laboratory")
Move(0.5,-5,0.5,0.5,-5,4.5)
Move(0.5,-5,4.5,7.5,-5,4.5)
mc.postToChat("This is a typical room for scientist")
time.sleep(3)
Move(7.5,-5,4.5,0.5,-5,4.5)
Move(0.5,-5,4.5,0.5,-5,27.5)
mc.postToChat("And... this is my room, Steve's room")
time.sleep(3)
Move(0.5,-5,27.5,-6.5,-5,27.5)
Move(-6.5,-5,27.5,-6.5,-5,23.5)
Move(-6.5,-5,23.5,-10.5,-5,23.5)
Move(-10.5,-5,23.5,-10.5,-5,27.5)
mc.postToChat("I have been working since last night")
time.sleep(5)
```

```
mc.postToChat("I need to have a break")
mc.player.setPos(-13.5,-5.5,21.5)
time.sleep(5)
mc.player.setPos(-10.5,-5,27.5)
time.sleep(5)
mc.postToChat("Did i just take a nap ?")
time.sleep(5)
mc.postToChat("Dad help meeee...")
time.sleep(3)
mc.postToChat("Sarah ?")
Move(-10.5,-5,27.5,0.5,-5,27.5)
Move(0.5,-5,27.5,0.5,-5,37.5)
mc.postToChat("Sarahh... Oh no...")
time.sleep(3)
mc.postToChat("I need to close the portal now")
time.sleep(3)
Move(0.5,-5,37.5,7.5,-5,37.5)
mc.postToChat("Level 1")
game.Minigame(10,-5,37,1,0,False)
time.sleep(3)
Move(7.5,-5,37.5,7.5,-5,48.5)
mc.postToChat("Level 2")
game.Minigame(10,-5,48,1,0,False)
time.sleep(3)
Move(7.5,-5,48.5,-5.5,-5,48.5)
mc.postToChat("Level 3")
game.Minigame(-10,-5,48,1,-1,False)
time.sleep(3)
Move(-5.5,-5,48.5,-5.5,-5,37.5)
mc.postToChat("Level 4")
game.Minigame(-10,-5,37,1,-1,False)
```

```
time.sleep(3)
mc.postToChat("I have to get back to my room now...")
time.sleep(3)
Move(-5.5,-5,37.5,0.5,-5,37.5)
Move(0.5,-5,37.5,0.5,-5,27.5)
Move(0.5,-5,27.5,-6.5,-5,27.5)
Move(-6.5,-5,27.5,-6.5,-5,23.5)
Move(-6.5,-5,23.5,-10.5,-5,23.5)
Move(-10.5,-5,23.5,-10.5,-5,27.5)
mc.postToChat("I'm so tired")
time.sleep(5)
mc.player.setPos(-13.5,-5.5,21.5)
time.sleep(8)
mc.postToChat("30 years later...")
time.sleep(8)
mc.postToChat("Argg...")
mc.player.setPos(-10.5,-5,27.5)
time.sleep(3)
Move(-10.5,-5,27.5,-10.5,-5,23.5)
Move(-10.5,-5,23.5,-6.5,-5,23.5)
Move(-6.5,-5,23.5,-6.5,-5,27.5)
Move(-6.5,-5,27.5,0.5,-5,27.5)
mc.postToChat("Finally, the time machine has been completed")
time.sleep(3)
mc.postToChat("I'm not quite sure whether if it works or not...")
time.sleep(3)
mc.postToChat("But i want Sarah back!")
time.sleep(3)
Move(0.5,-5,27.5,0.5,-5,48.5)
while (GPIO.input(6) == True):
        mc.postToChat("Hold enter button to enter the time machine")
```

```
        time.sleep(2)
mc.player.setPos(0.5,-4,52.5)
mc.postToChat("Time machine: ready")
time.sleep(3)
mc.postToChat("Time portal: openning...")
time.sleep(3)
mc.postToChat("Time portal: ready")
time.sleep(3)
mc.postToChat("Jump!")
mc.player.setPos(-13.5,-5.5,21.5)
#SCENCE 2 - NOTCH'S KINGDOM
buildTimeMachine(-1,4,121) #Time machine arrive the city
mc.player.setPos(0.5,5,123.5) #On time machine
mc.postToChat("Ouchh, my head")
time.sleep(3)
mc.postToChat("What year is it ?")
time.sleep(3)
mc.postToChat("Oh gosh, the timer has broken")
time.sleep(3)
mc.postToChat("Anyway, let's take a look then...")
time.sleep(3)
Move(0.5,4,123.5,0.5,4,109.5)
mc.postToChat("Wow, look at those wheats, they're looked well")
time.sleep(3)
Move(0.5,4,109.5,0.5,4,88.5)
mc.postToChat("Are that villagers' houses ?")
time.sleep(3)
Move(0.5,4,88.5,0.5,4,72.5)
mc.postToChat("Hello ?...")
time.sleep(3)
mc.postToChat("Seem to be nobody here...")
```

```
time.sleep(3)
Move(0.5,4,72.5,0.3,4,46.5)
mc.postToChat("Let's take a look at the house, shall we ?")
time.sleep(3)
Move(0.3,4,46.5,13.5,4,46.5)
while (GPIO.input(6) == True):
        mc.postToChat("Hold enter button to enter the house")
        time.sleep(3)
mc.player.setPos(13.5,5,52.5)
mc.postToChat("What a nice house !")
time.sleep(8)
while (GPIO.input(5) == True):
        mc.postToChat("Hold exit button to exit the house")
        time.sleep(3)
mc.player.setPos(13.5,4,46.5)
Move(13.5,4,46.5,0.5,4,46.5)
mc.postToChat("This place looks strange!")
time.sleep(3)
Move(0.5,4,46.5,0.5,4,32.5)
mc.postToChat("Anyway, let's go then...")
time.sleep(3)
Move(0.5,4,32.5,0.5,4,7.5)
mc.postToChat("That's Notch statue - a legend of Minecraft world")
time.sleep(3)
Move(0.5,4,7.5,7.5,4,7.5)
Move(7.5,4,7.5,7.5,4,0.5)
Move(7.5,4,0.5,38.5,4,0.5)
mc.postToChat("What a big pagoda !")
time.sleep(3)
Move(38.5,4,0.5,38.5,4,-20.5)
while (GPIO.input(6) == True):
```

```
        mc.postToChat("Hold enter button to enter pagoda")
        time.sleep(3)
mc.player.setPos(38.5,7,-30)
Move(38.5,7,-30,38.5,7,-43)
mc.postToChat("Why's nothing in here ?")
time.sleep(3)
while (GPIO.input(5) == True):
        mc.postToChat("Hold exit button to exit the pagoda")
        time.sleep(3)
mc.player.setPos(38.5,4,-20.5)
Move(38.5,4,-20.5,38.5,4,0.5)
Move(38.5,4,0.5,82.5,4,0.5)
mc.postToChat("Look at market on the right side")
time.sleep(3)
mc.postToChat("So colorful !!!")
time.sleep(3)
mc.postToChat("Hmm...")
time.sleep(3)
mc.postToChat("Are those bamboos ?")
time.sleep(3)
while (GPIO.input(5) == True):
        mc.postToChat("Hold exit button get back to the north statue")
        time.sleep(3)
mc.player.setPos(0.5,4,7.5)
Move(0.5,4,7.5,-6.5,4,7.5)
Move(-6.5,4,7.5,-6.5,4,0.5)
Move(-6.5,4,0.5,-72.5,4,0.5)
Move(-72.5,4,0.5,-72.5,4,7.5)
mc.postToChat("Urgg, I shouldn't go there, so scary") #Play some horror music ?
time.sleep(3)
Move(-72.5,4,7.5,-72.5,4,0.5)
```

```
Move(-72.5,4,0.5,-72.5,4,-30.5)
while (GPIO.input(6) == True):
        mc.postToChat("Hold enter button to enter the castle")
        time.sleep(3)
mc.player.setPos(-72.5,6,-49.5)
time.sleep(3)
Move(-72.5,6,-49.5,-72.5,6,-81.5)
mc.postToChat("That's look gorgeous")
time.sleep(3)
Move(-72.5,6,-81.5,-55.5,6,-81.5)
while (GPIO.input(6) == True):
        mc.postToChat("Hold enter button to enter the library")
        time.sleep(3)
mc.player.setPos(-55.5,22,-57.5)
mc.postToChat("Wow, that's a big library...")
time.sleep(3)
Move(-55.5,22,-57.5,-65.5,22,-57.5)
mc.postToChat("This place seems to be not my place")
time.sleep(3)
mc.postToChat("Because I only see historical books")
time.sleep(3)
Move(-65.5,22,-57.5,-75.5,22,-57.5)
while (GPIO.input(5) == True):
        mc.postToChat("Hold exit button to exit the castle")
        time.sleep(3)
mc.player.setPos(-72.5,4,-30.5)
Move(-72.5,4,-30.5,-72.5,4,0.5)
Move(-72.5,4,0.5,-6.5,4,0.5)
Move(-6.5,4,0.5,-6.5,4,-6.5)
Move(-6.5,4,-6.5,0.5,4,-87.5)
mc.postToChat("That was a long way...")
```

```
time.sleep(3)
mc.postToChat("Oh, graveyard then... Why this weird tree stand here ?")
time.sleep(3)
Move(0.5,4,-87.5,13.5,4,-87.5)
mc.postToChat("That's scary. I'm not feeling comfortable now...")
time.sleep(3)
Move(13.5,4,-87.5,26.5,4,-87.5)
time.sleep(3)
mc.postToChat("That's alot of dead people")
time.sleep(3)
Move(26.5,4,-87.5,56.5,4,-87.5)
mc.postToChat("Wait, What ?")
time.sleep(3)
#Pause
Move(56.5,4,-87.5,56.5,4,-89.5)
mc.postToChat("Sarah ?")
time.sleep(3)
mc.postToChat("Sarah - 68 years old")
time.sleep(3)
#Pause
mc.postToChat("Oh no, I have to...")
time.sleep(3)
mc.player.setPos(56.5,2,-89.5)
time.sleep(5)
mc.player.setPos(-10.5,-5,27.5)
mc.postToChat("Did i just take a nap ?")
time.sleep(5)
mc.postToChat("Dad help meeee...")
time.sleep(3)
mc.player.setPos(-13.5,-5.5,21.5)
mc.postToChat("End.")
```