



UNIVERSITY OF MALAYA

Fakulti Sains Komputer & Teknologi Maklumat

Assignment Technical Report

WIX 1002

Fundamentals Of Programming

CAT_ARE_AWESOME

Lecturer: Dr. Mohd Hairul Nizam Bin Md Nasir

Group Name: FOP_is_Easy

NAME	MATRIX NUMBER
BOO JIN SIANG(LEADER)	WID190005
ALVIN ANG ZEE QIN	WID190003
NG YONG ZHANG	WID190510
TONG BOON ONN	WID190057

Introduction

The task that is being assigned to us is project 3 which is Cats-Are-Awesome. The task requires us to write a pet cat program that can adopt an unreal cat in the game.

In the program, the cat class should include:

name: String

happiness: Double

hunger: Double

level: Integer/Double

timeOfBirth: Long

TTL (Time to leave): Long

This game includes several features:

1. Able for user log in, sign up and log out.
2. Able to generate different interface according to the condition.
3. Able to generate a map of 20x20 squares.
4. Able to walk in four direction (**UP, DOWN, RIGHT, LEFT**) depending on the obstacles when collection “Munny” and defend against rats and birds.
5. The “Munny”, rats, obstacle are generated randomly in the map.
6. Different “Meow” sounds are generated when talking with cat.
7. Time-to-live is counted automatically.
8. The cat can be levelled up by feeding and playing with it.

The live of cat is depends on how the player take care of it as the player can treats the cat with many different methods.

Our team consists of 4 members, who are:

1. Boo Jin Siang (WID 190005)
2. Alvin Ang Zee Qin (WID 190003)
3. Ng Yong Zhang (WID 190510)
4. Tong Boon Onn (WID 190057)

Approach taken to solve the task

Firstly, we start with the basic features that must be implemented in the program.

1. Authentication

- when running the game, the user able to log in, sign up and log out.
- data is saved before the user exit the game.
- the User need to enter username, email, and password.

2. Add your own cat

- Interface for user to add a new cat is provided
- user able to name their cat by insert the name in the given space.

3.Play with your cat

- The map is made up of JavaFX. (20x20)
- The cat can walk by entering (**UP, DOWN, LEFT, RIGHT**).
- “Munny” and obstacle are generated randomly.
- When touching the “Munny”, the amount of “Munny” will increase by one.
- If there is an obstacle, the cat unable to move to the direction.
- If playing with the cat, the happiness of cat will increase.

4.Feed your cat

- The hunger and happiness are shown in the game.
- If feeding the cat, the happiness will increase but the hunger will decrease.
- If the hunger is maximum, the cat will die.
- The time is counted automatically and the cat need to be deed on time as hunger will increase systematically.

5.Chat with your cat

- “Meow” language is needed to unlock to chat with the cat.
- There is a chatroom for the user and cat to communicate with each other.

6. Level up your cat

- The level of cat is considered in the game and can be increase by feeding and playing with it.
- The cat’s level is calculated systematically.
- When the cats level up, more kinds of “Meow” language is unlocked.

Extra Features

1.Unique cat

- When the user sign in the game, the user able to choose hair colour, eye, and personality given.
- The cat's appearance will be shown according to the choice of user in the game.

2. Cat breeding

- When reach certain level, the cat able to breed with other cats.
- The user can have more than one cat at the same time.

3.Cat fighter

- The map with the size of 10x10 is generated.
- The rats or birds are generated randomly in the map which will move from top to bottom.
- The cat able to move in two direction by entering (**LEFT, RIGHT**).
- When the cat touched the rats or birds, they will be eliminated.

4.Collecting data

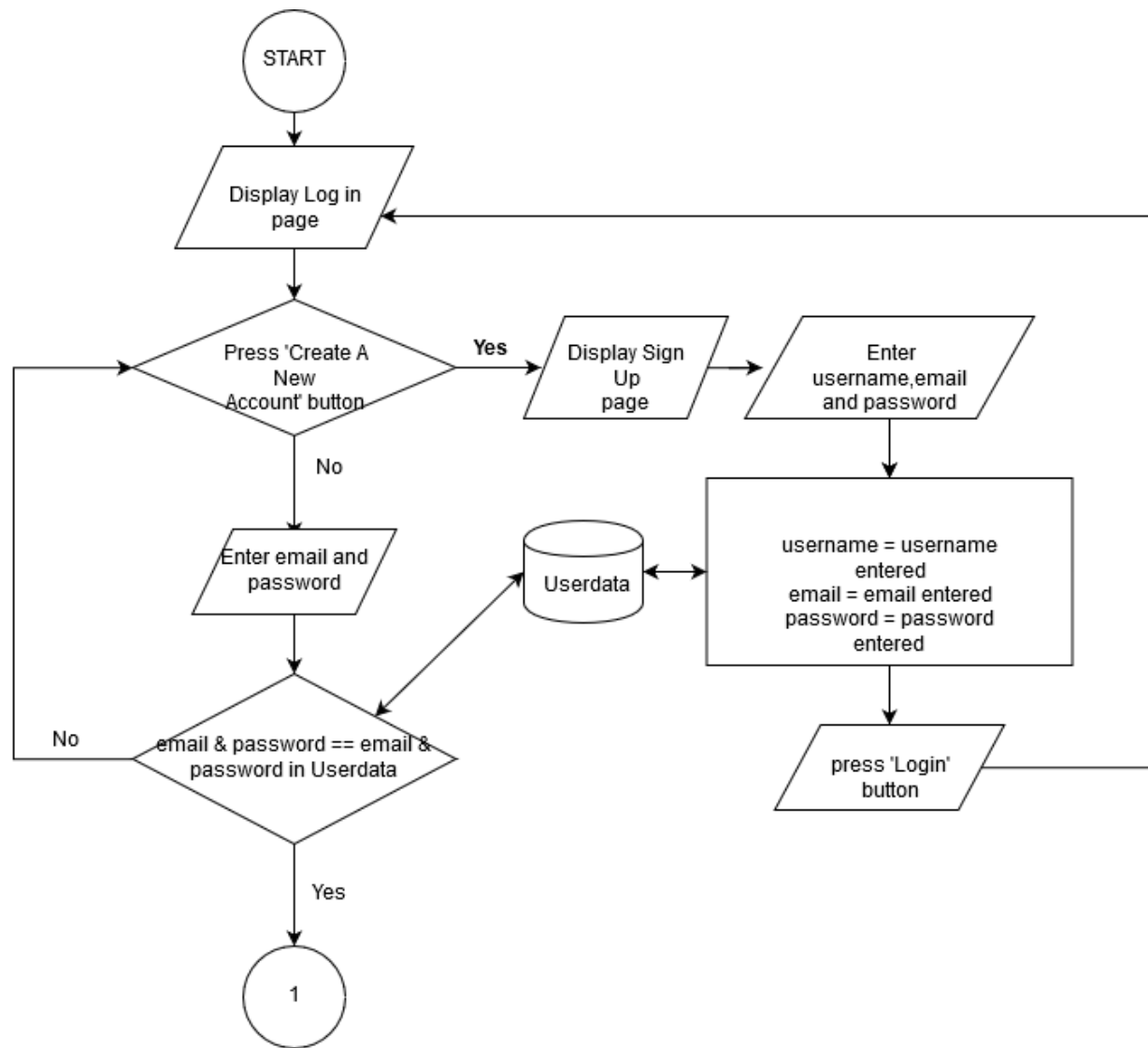
- The data such as graph of feeding rates and growth of happiness from the cat is collected in the game.
- The user can see and analyse the data in the game.

5.Sound

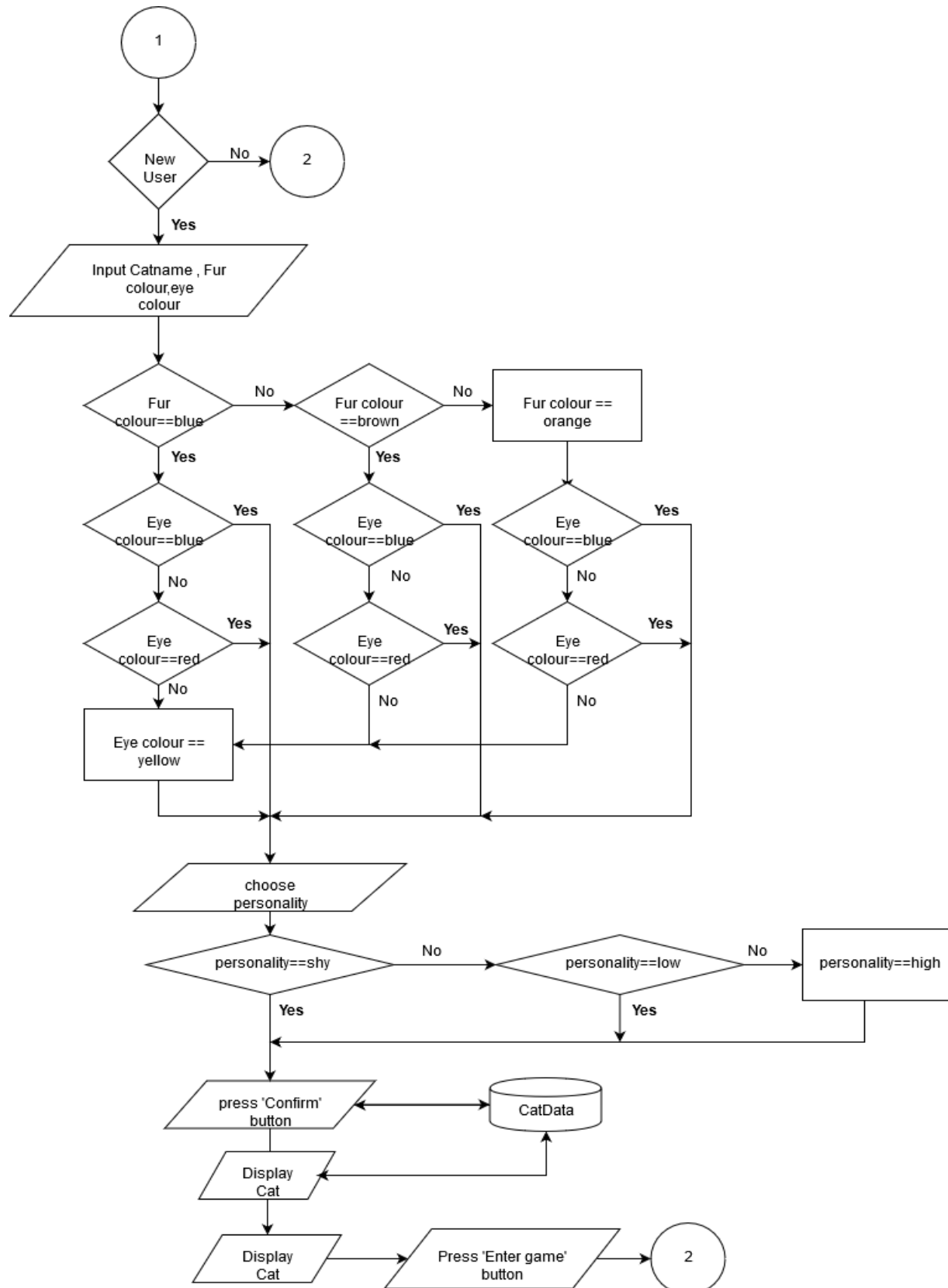
- When enter the game, there is a background song.
- When the user clicked the cat, the cat will response with the "meow" sounds.

FLOW CHART OF THE PROGRAM

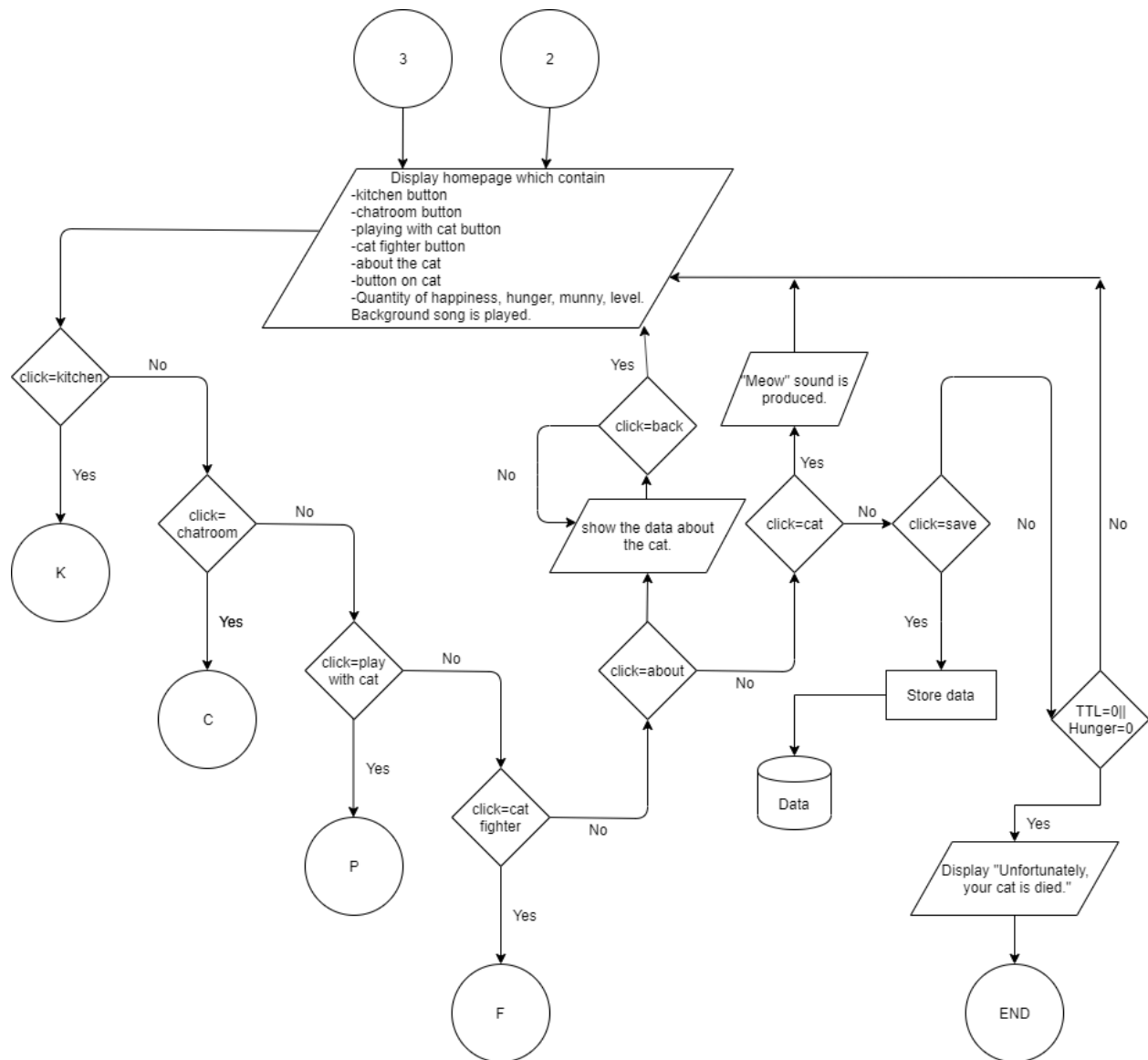
-Log in & Sign up



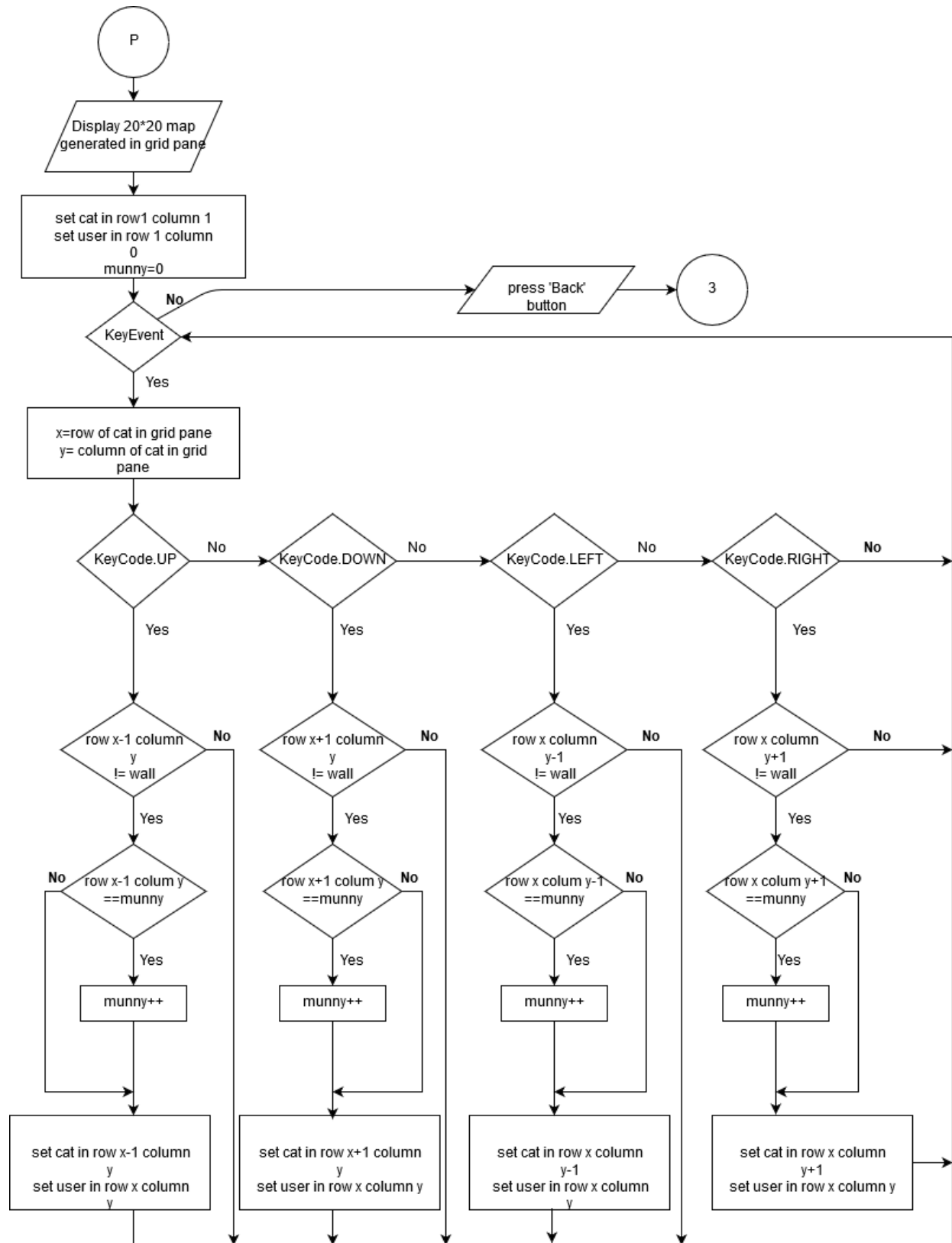
-Design t Cat



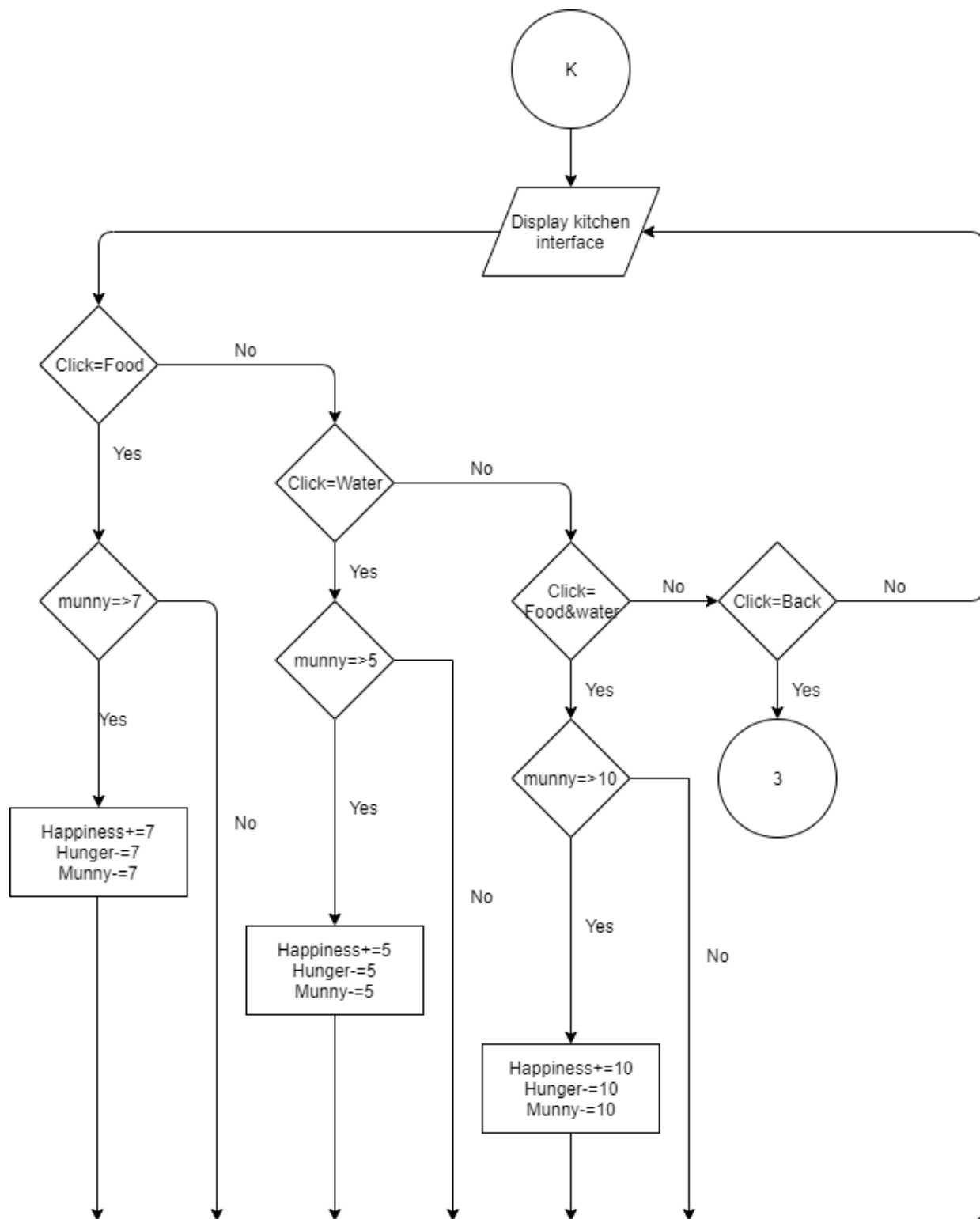
-Homepage



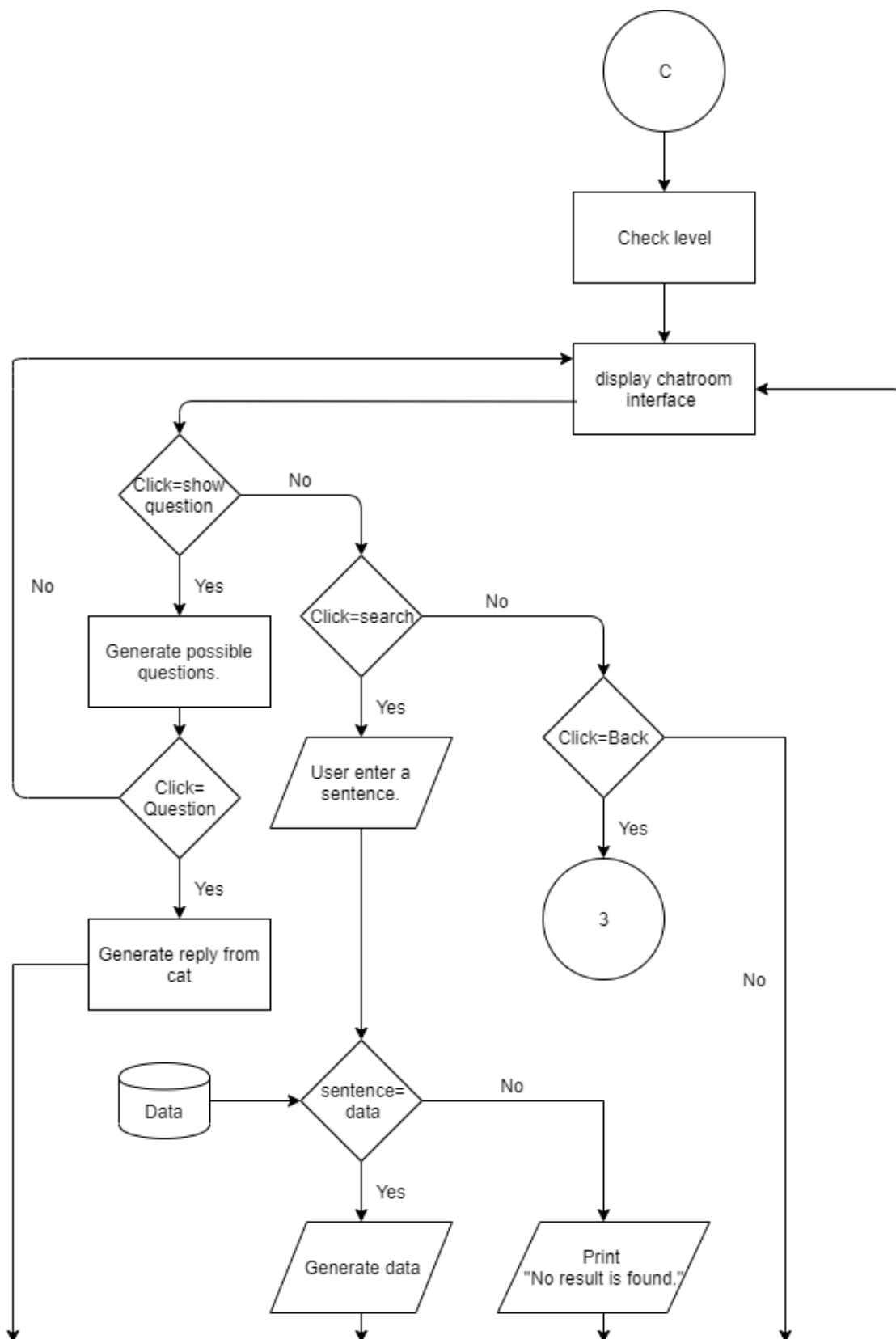
-Play with cat



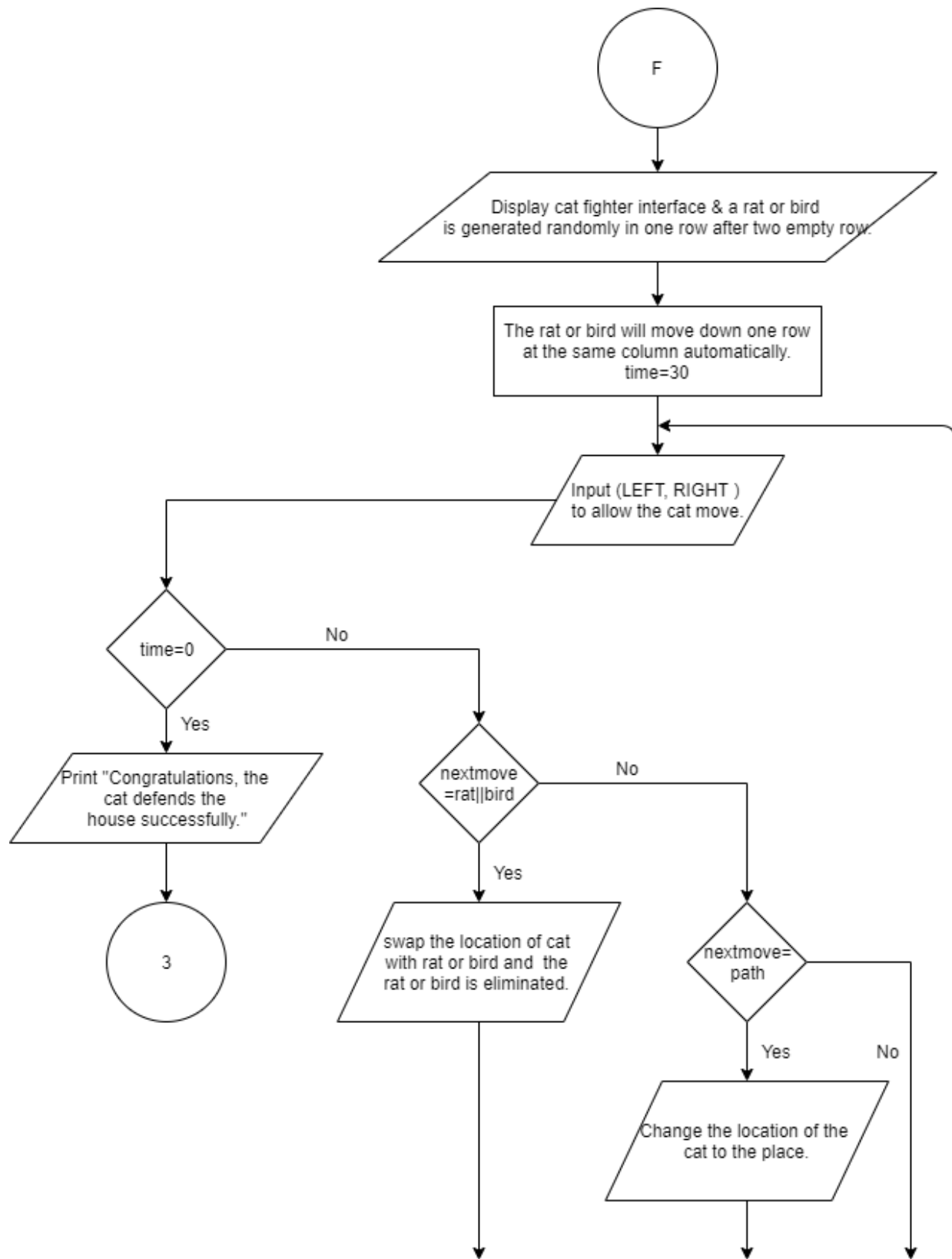
-Kitchen



-Chatroom

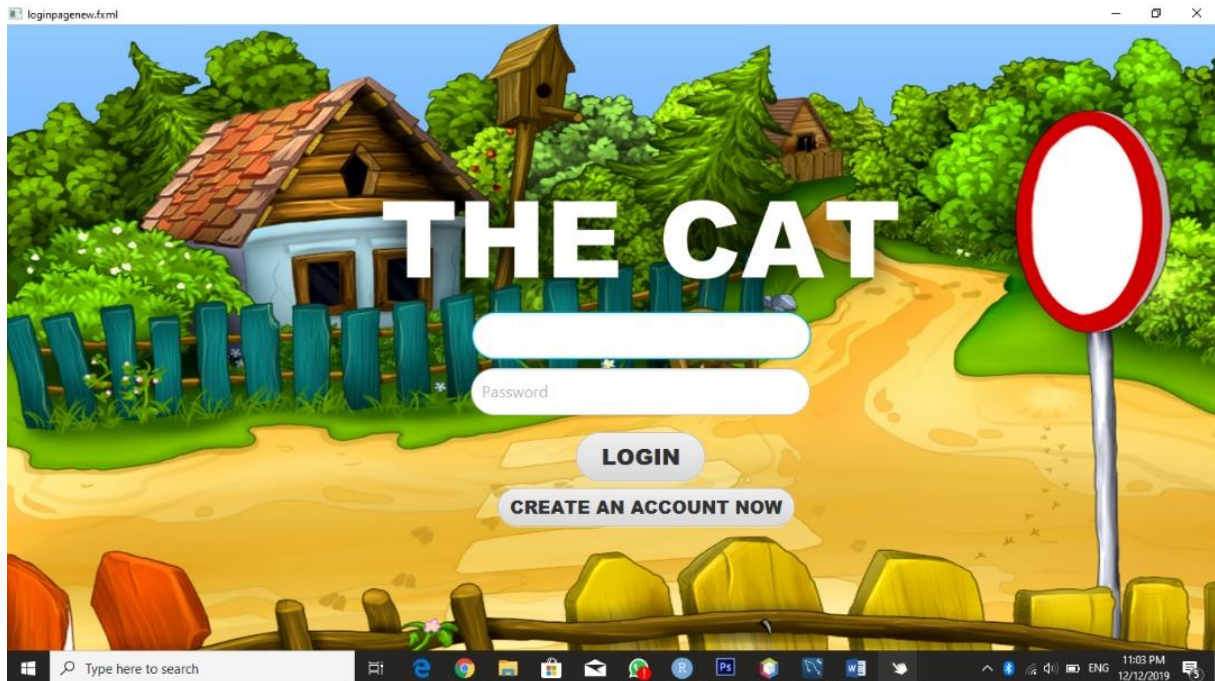


-Cat Fighter



Output Program

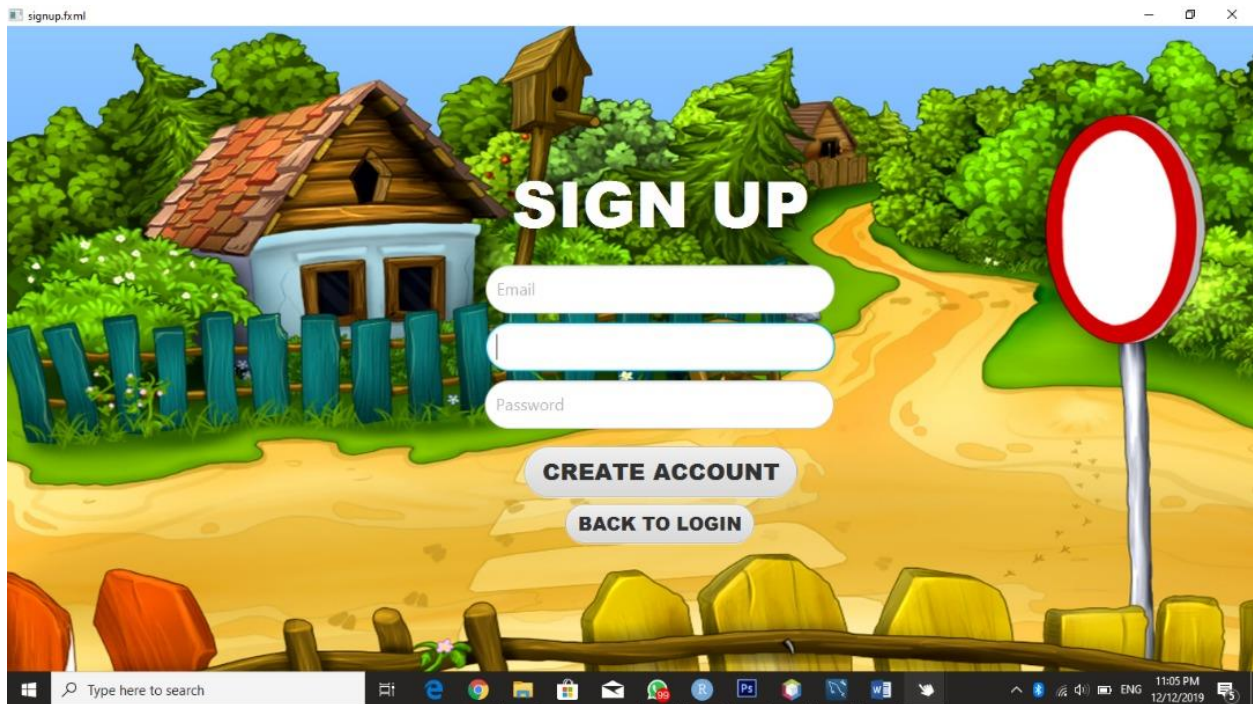
-Login Page



Compare with UserData

```
}  
  
for(Users user : users){  
    if(user.getPassword().equalsIgnoreCase(password)) {  
        List<Cat> cats = CatData.getInstance().getCat(user.getUsername());  
        if(cats.isEmpty()){  
            Global.setUsername(user.getUsername());  
            Parent root = FXMLLoader.load(getClass().getResource("/sample/views/choosecharacter.fxml"));  
            Stage stage= (Stage) ((Node) event.getSource()).getScene().getWindow();  
            Scene scene = new Scene(root);  
            stage.setScene(scene);  
        }else{  
            for(Cat cat: cats){  
                Global.setCat(cat);  
            }  
            Parent root = FXMLLoader.load(getClass().getResource("/sample/views/main.fxml"));  
            Stage stage= (Stage) ((Node) event.getSource()).getScene().getWindow();  
            Scene scene = new Scene(root);  
            stage.setScene(scene);  
        }  
    }else{  
        isConnected.setText("Password wrong sorry please try again");  
        Parent root = FXMLLoader.load(getClass().getResource("/sample/views/loginpage.fxml"));  
        Stage stage= (Stage) ((Node) event.getSource()).getScene().getWindow();  
        Scene scene = new Scene(root);  
        stage.setScene(scene);  
    }  
}
```

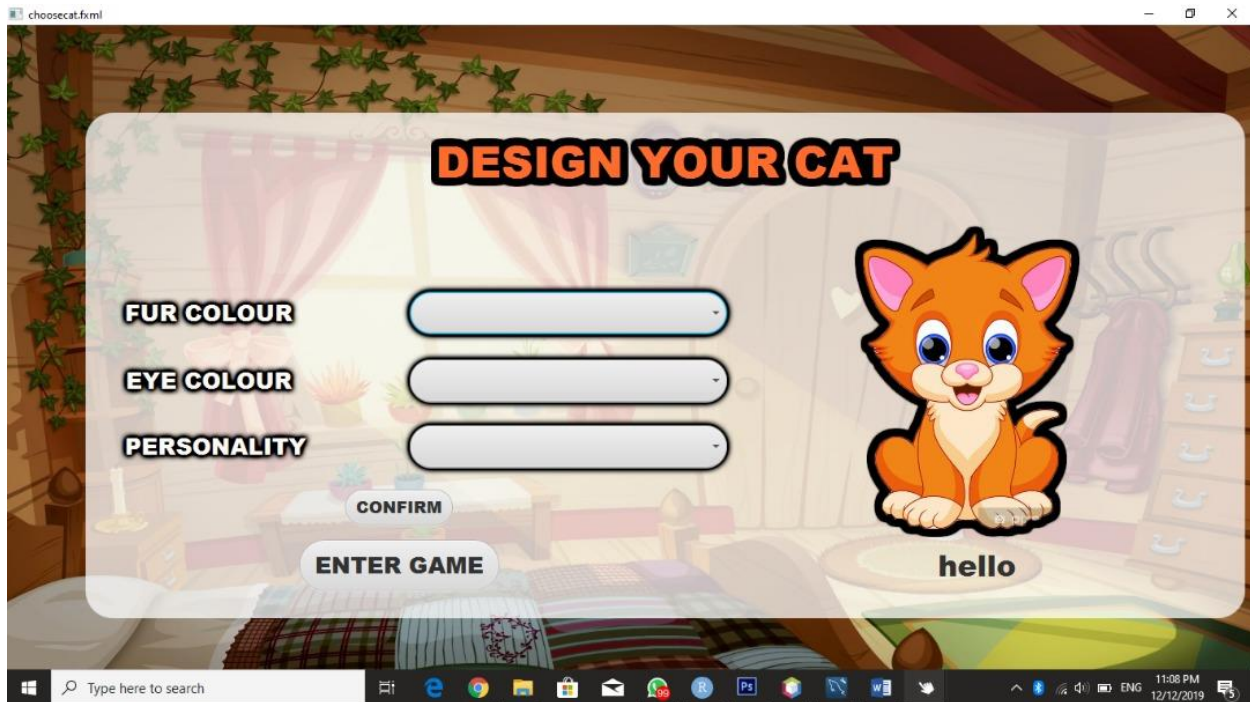
-Sign Up page



Store the User data in UserData

```
@FXML
void signup(MouseEvent event){
    String username = txtUsername.getText();
    String email = txtEmail.getText();
    String password = txtPassword.getText();
    Munny userMunny = new Munny();
    Global.setMunny(userMunny);
    UserData.getInstance().open();
    if (!UserData.getInstance().getUserLogin(email).isEmpty()){
        isConnected.setText("You are registered. Please login...");
    }else if(UserData.getInstance().getUserLogin(email).isEmpty()){
        UserData.getInstance().setInsertUser(username,email, password,userMunny);
        isConnected.setText("Is registered. Please login the account");
    }
    UserData.getInstance().close();
}
}
```

-Design Cat



Cat designing and save in CatData

```
public void confirmClicked(ActionEvent event) {
    String catname = CatName.getText();
    String fur = (String) furcolour.getSelectionModel().getSelectedItem();
    String eye = (String) eyecolour.getSelectionModel().getSelectedItem();
    Image catImage = Global.getInstance().setCatImage(fur,eye);
    showcat.setImage(catImage);
    character.setText(personality.getValue().toString());
    name.setText(catname);
}

@FXML
void create(MouseEvent event) throws IOException{
    String name = CatName.getText();
    String furColour = (String) furcolour.getSelectionModel().getSelectedItem();
    String eyeColour = (String) eyecolour.getSelectionModel().getSelectedItem();
    String setpersonality = (String) personality.getSelectionModel().getSelectedItem();
    Global.setFurColour(furColour);
    Global.setEyeColour(eyeColour);
    Global.setPersonality(setpersonality);
    CatData.getInstance().open();
    Cat cat = new Cat(name);
    Global.setCat(cat);
    String username = Global.getUsername();
    CatData.getInstance().setInsertCat(name,cat.getHappiness(),cat.getHunger(),
        cat.getLevel(),cat.getTimeOfBirth(),cat.getTimeToLive(), username,furColour,eyeColour,setpersonality);
    Parent root = FXMLLoader.load(getClass().getResource("/sample/views/main.fxml"));
    Stage stage = (Stage) ((Node)event.getSource()).getScene().getWindow();
    Scene scene = new Scene(root);
    stage.setScene(scene);
    CatData.getInstance().close();
}
```


-Homepage



Switch to other interfaces

```
@FXML
public void playGame(MouseEvent event )throws IOException {
    gameController gameController = new gameController();
    final Stage stage = (Stage) ((Node)event.getSource()).getScene().getWindow();
    gameController.start(stage);
}

@FXML
private void chatCat(MouseEvent event )throws IOException {
    Parent root = FXMLLoader.load(getClass().getResource("/sample/views/chat.fxml"));
    Stage stage = (Stage) ((Node)event.getSource()).getScene().getWindow();
    Scene scene = new Scene(root);
    stage.setScene(scene);
}

@FXML
private void feedCat(MouseEvent event )throws IOException {
    Parent root = FXMLLoader.load(getClass().getResource("/sample/views/feed.fxml"));
    Stage stage = (Stage) ((Node)event.getSource()).getScene().getWindow();
    Scene scene = new Scene(root);
    stage.setScene(scene);
}
```

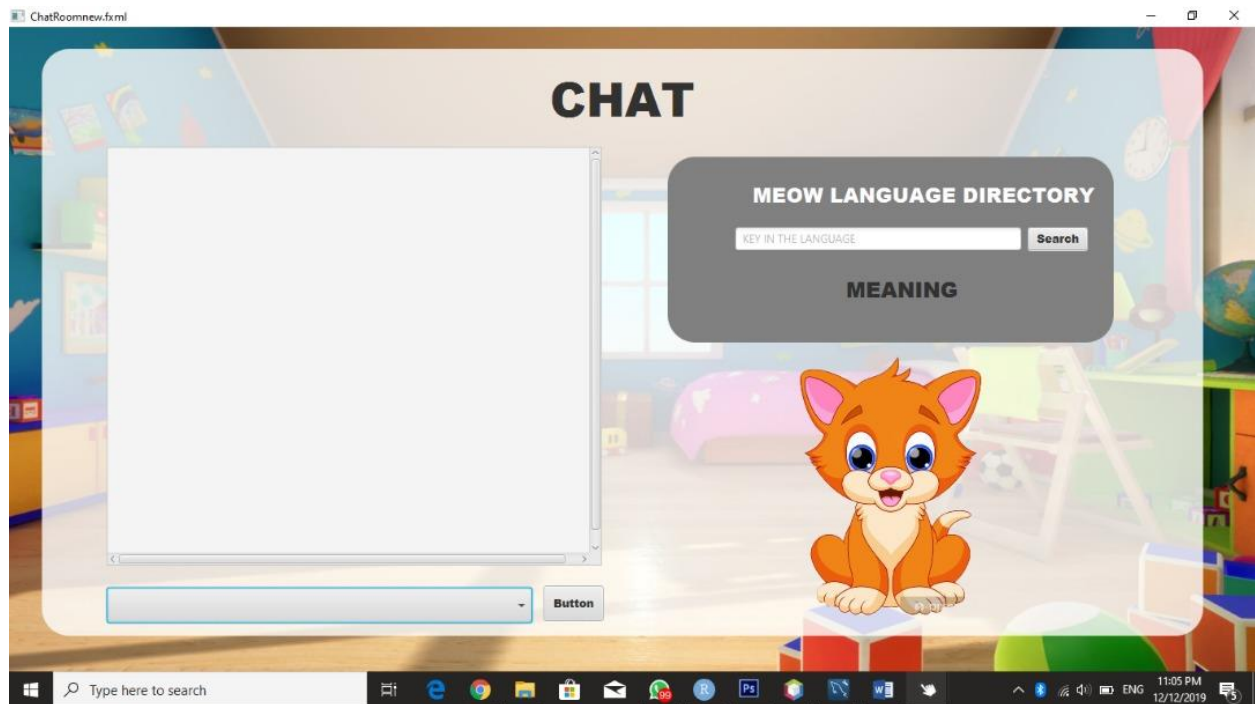
Time To Live

```
@Override
public void initialize(URL url, ResourceBundle rb) {
    timeToLive = Global.cat.getTimeToLive();
    TTL ttl = Global.cat.getTtl();
    second = (int) timeToLive /1000;
    timer = new Timer();
    int delay = 1000;
    int period =1000;
    TimerTask task = new TimerTask() {
        @Override
        public void run() {
            timeToLive-=1;
            ttl.addTtl(timeToLive*1000);
            System.out.println(timeToLive);
            displayTimeToLive.setText(Long.toString(timeToLive));
            setTime();
        }
    };
    timer.scheduleAtFixedRate(task,delay,period);

}

private static final void setTime(){
    if (Global.cat.getTimeToLive() == 1){
        timer.cancel();
    }
}
```

-Chatroom



Import reply form text file

```
while (myrs.next()) {
    chats.setMeowlanguage(myrs.getString("meowlanguage"));
    chats.setMeowlanguage2(myrs.getString("meowlanguage2"));
}
Random r = new Random();
int num = r.nextInt(2);
TextField textfield1 = new TextField();
if (num == 1) {
    textfield1.setText(chats.getMeowlanguage());
    textmessage.getChildren().add(textfield1);
} else {
    textfield1.setText(chats.getMeowlanguage2());
    textmessage.getChildren().add(textfield1);
}
messageinput.setValue(null);
}
```


-Kitchen



Choose food for eat, increase happiness and decrease hunger

```
EventHandler<MouseEvent> foodOnMouseReleasedEventHandler =
    new EventHandler<MouseEvent>() {
        @Override
        public void handle(MouseEvent event) {
            orgLayoutX = ((ImageView)(event.getSource())).getLayoutX();
            orgLayoutY = ((ImageView)(event.getSource())).getLayoutY();
            double newX = (orgLayoutX + newTranslateX)-catImgX;
            double newY = (orgLayoutY + newTranslateY)-catImgY;
            if (Math.abs(newX)<40 || Math.abs(newY)<5){
                ((ImageView)(event.getSource())).setTranslateY(0);
                ((ImageView)(event.getSource())).setTranslateX(0);
                if(((ImageView)(event.getSource())) == imgFish){
                    eatFood(fish);
                    System.out.println("cat eat fish...");
                }else if (((ImageView)(event.getSource())) == imgWater){
                    eatFood(water);
                    System.out.println("cat drink water...");
                }else if (((ImageView)(event.getSource())) == imgWaterFish){
                    eatFood(fishAndWater);
                    System.out.println("cat eat fish and drink water...");
                }
            }
        }
    };
```

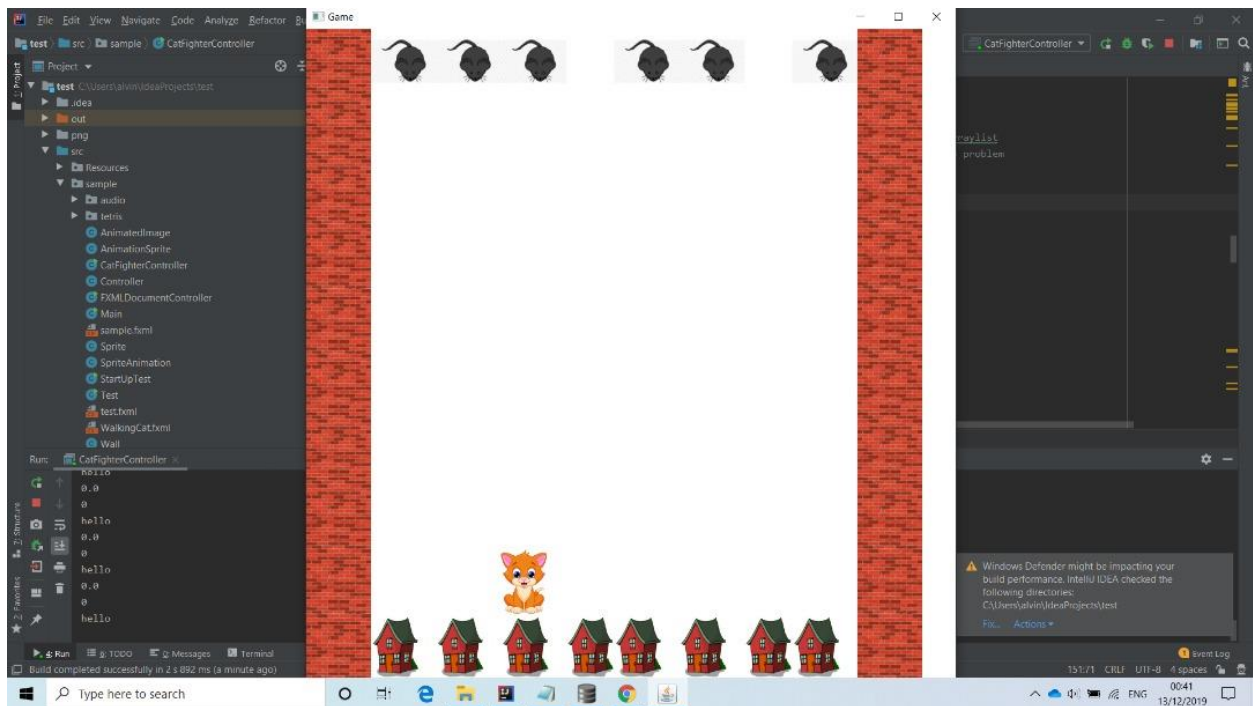
-Play with Cat



The Cat move up, down, left, right

```
EventHandler<KeyEvent> setOnTheKeyPressed =  
    new EventHandler<KeyEvent>() {  
  
        @Override  
        public void handle(KeyEvent event)  
        {  
            int orgRowCat =GridPane.getRowIndex(cat);  
            int orgColumnCat =GridPane.getColumnIndex(cat);  
  
            if ( event.getCode()== KeyCode.UP ){  
                if(wallArray[orgRowCat-1][orgColumnCat]){}else{  
                    GridPane.setRowIndex(cat,orgRowCat-1);  
                    GridPane.setRowIndex(ren,orgRowCat);  
                    GridPane.setColumnIndex(ren,orgColumnCat);} }  
            if ( event.getCode()== KeyCode.DOWN ){  
                if(wallArray[orgRowCat+1][orgColumnCat]){ }else{  
                    GridPane.setRowIndex(cat,orgRowCat+1);  
                    GridPane.setRowIndex(ren,orgRowCat);  
                    GridPane.setColumnIndex(ren,orgColumnCat);}}  
            if ( event.getCode()== KeyCode.RIGHT ){  
                if(wallArray[orgRowCat][orgColumnCat+1]){}else{  
                    GridPane.setColumnIndex(cat,orgColumnCat+1);  
                    GridPane.setRowIndex(ren,orgRowCat);  
                    GridPane.setColumnIndex(ren,orgColumnCat); } }  
            if ( event.getCode()== KeyCode.LEFT ){  
                if(wallArray[orgRowCat][orgColumnCat-1]){}else{  
                    GridPane.setColumnIndex(cat,orgColumnCat-1);  
                    GridPane.setRowIndex(ren,orgRowCat);  
                    GridPane.setColumnIndex(ren,orgColumnCat);}}}}};
```

-Cat Fighter



The Cat Move Left and Right

```
{  
  
    int orgRowCat =GridPane.getRowIndex(cat);  
    int orgColumnCat =GridPane.getColumnIndex(cat);  
  
    if ( event.getCode()== KeyCode.RIGHT ){  
        if(orgColumnCat+1 != 9){  
            GridPane.setColumnIndex(cat, value: orgColumnCat+1);  
        }  
    }  
  
    if ( event.getCode()== KeyCode.LEFT ){  
        if (orgColumnCat-1 != 0){  
            GridPane.setColumnIndex(cat, value: orgColumnCat-1);  
        }  
    }  
}  
}
```