**SOFTWARE ENGINEERING**

**RAPOR**

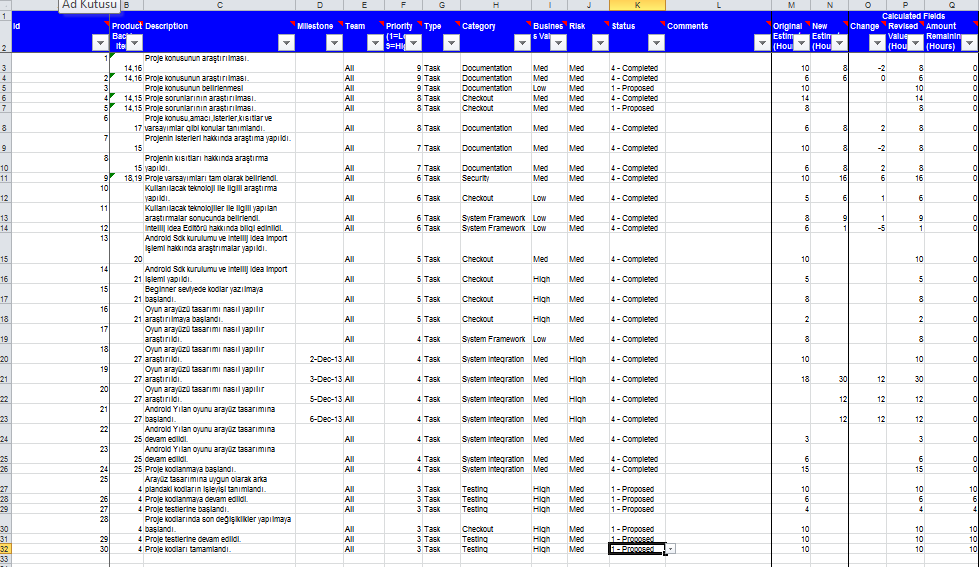
**HİLAL ATICI**

**302020131004**

**OSMANGAZİ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

1. **Proje Scrum Planlaması**



Resim 1.Scrum Planlaması

Proje Scrum planlaması excel kullanılarak yapıldı.Yapılacak işlemler excelde gösterildi.

**Android Programı: SNAKE GAME**

**Giriş:**

Android tabanlı bir oyun kodu yazdım.Snake oyunu arayüzünü ve arka planda çalışan kodları yazdım.Android 4.1.2 sürümüne göre tasarladım.

**Uygulama geliştirdiğim platform:**

Uygulamamı Intellij Idea 2012 editöründe yazdım.

**Arayüz:**

Tasarım kısmını xml ile yaptım.

**Kullandığım diller:**

JAVA, XML

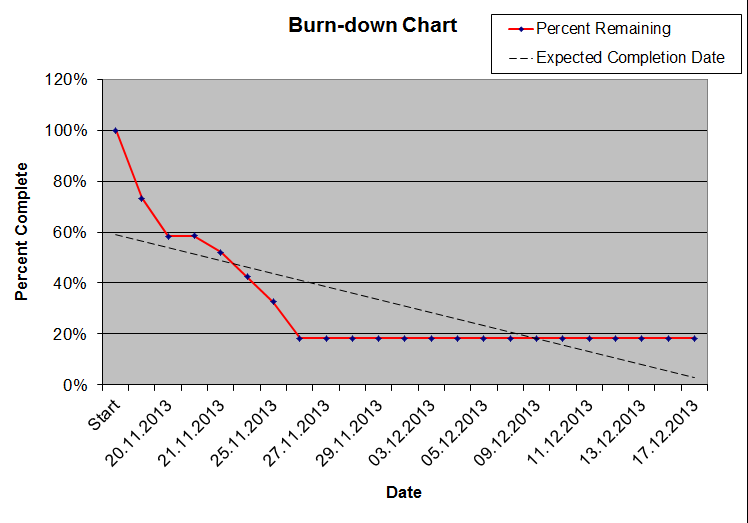
**Projenin içeriği:**

Kullanıcı uygulamayı açtığında ekrana dokunmadan oyun başlamıyor.Ekrana dokunduktan sonra yılan hareket etmeye başlıyor.Yılan her yemi yakaladığında skor bir artıyor.

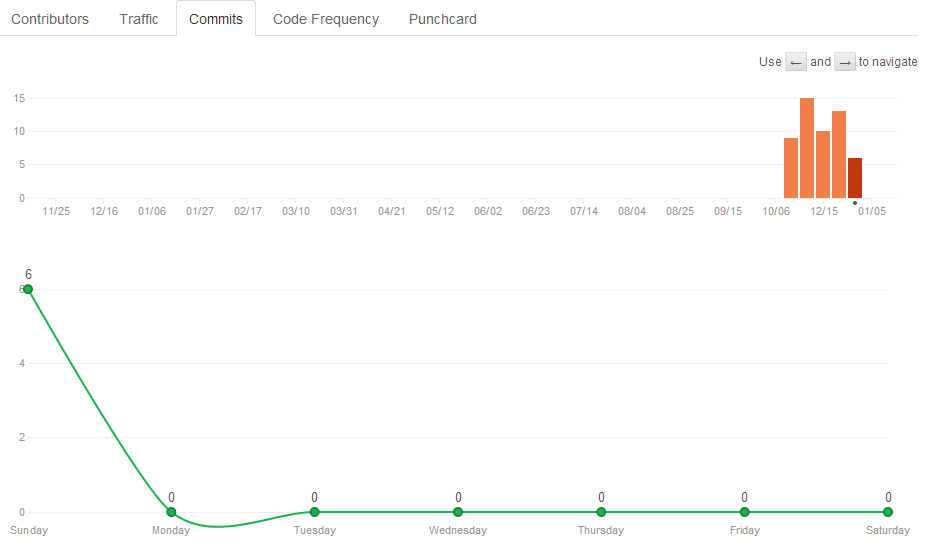
**Proje Zamanı:**

**30 Eylül- 5 Ocak 2013** tarihleri arasında proje hazırlandı.

**2-)BurnDown Grafiği**

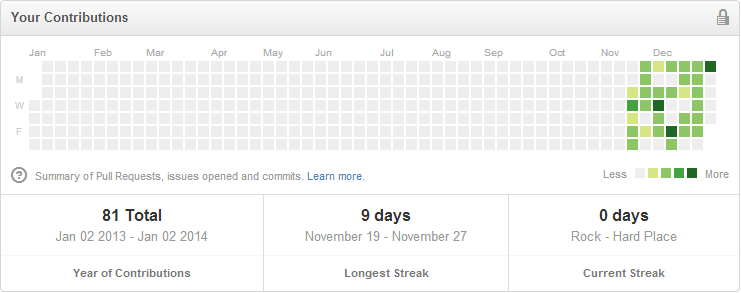


Resim 2.BurnDown Chart



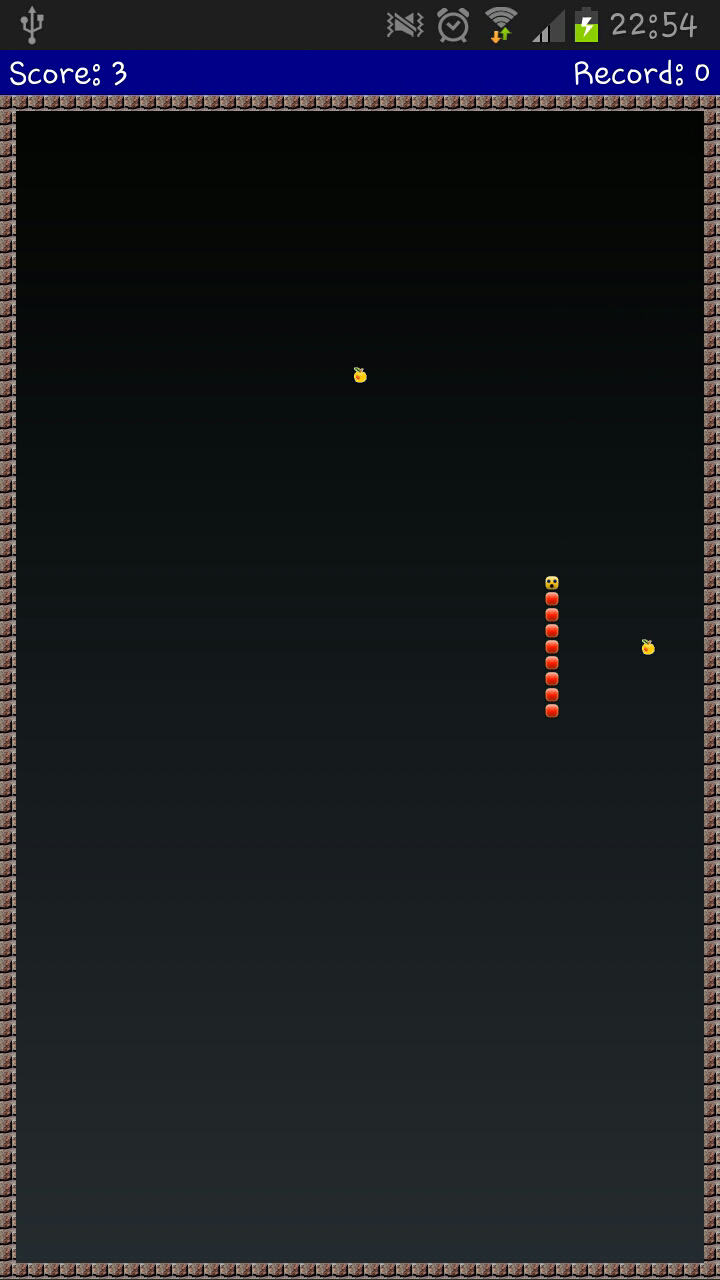
Resim 3.Github Commit Grafiği

**3-)Proje Github Commit Logu**



Resim 4.Github Commit Logu

**4.Proje Ekran Çıktısı**

****

Resim 5.Uygulama Ekran Çıktısı

**Software Engineering Special :**

**Kaynak Kod:**DosyadanOku.java

import java.io.\*;

import java.util.Scanner;

/\*\*

\* Created with IntelliJ IDEA.

\* User: Hilal

\* Date: 26.12.2013

\* Time: 17:20

\* To change this template use File | Settings | File Templates.

\*/

public class DosyadanOku {

public static void main(String [] args)

{

System.out.println("Thread sayisini giriniz..");

Scanner sc = new Scanner(System.in);

int i = sc.nextInt();

A threadA = new A("file1.txt",0,3,5,i);

B threadB = new B("file2.txt",0,3,5,i);

C threadC = new C("file3.txt",0,3,5,i);

D threadD = new D("file4.txt",0,3,5,i);

E threadE = new E("file5.txt",0,3,5,i);

System.out.println("Started Thread A");

threadA.start();

System.out.println("Started Thread B");

threadB.start();

System.out.println("Started Thread C");

threadC.start();

System.out.println("Started Thread D");

threadD.start();

System.out.println("Started Thread E");

threadE.start();

System.out.println("End of main thread");

} }

class A extends Thread {

String msg;

private int startIdx, nThreads, maxIdx,k;

public A(String ms,int s, int n, int m,int k)

{

this.msg=ms;

this.startIdx = s;

this.nThreads = n;

this.maxIdx = m;

this.k=k;

}

public void run() {

System.out.println("Thread A started");

String content = null;

File file = new File(msg);

String string="";

//reading

try{

InputStream ips=new FileInputStream(file);

InputStreamReader ipsr=new InputStreamReader(ips);

BufferedReader br=new BufferedReader(ipsr);

String line;

for(int i = this.startIdx; i < maxIdx; i += this.nThreads)

{

//System.out.println("[ID " + this.getId() + "] " + i);

while ((line=br.readLine())!=null){

int a=line.length(),z;

String y="";

for(z=a-1;z>=0;z--){

y+=line.charAt(z);

}

System.out.println(y);

string+=line+"\n";

}

}

br.close();

}

catch (Exception e){

System.out.println(e.toString());

}

System.out.println("Exit from A Thread");

}

}

class B extends Thread {

String msg;

private int startIdx, nThreads, maxIdx,k;

public B(String ms,int s, int n, int m,int k)

{

this.msg=ms;

this.startIdx = s;

this.nThreads = n;

this.maxIdx = m;

this.k=k;

}

public void run() {

System.out.println("Thread B started");

String content = null;

File file = new File(msg);

String string="";

//reading

try{

InputStream ips=new FileInputStream(file);

InputStreamReader ipsr=new InputStreamReader(ips);

BufferedReader br=new BufferedReader(ipsr);

String line;

for(int i = this.startIdx; i < maxIdx; i += this.nThreads)

{

// System.out.println("[ID " + this.getId() + "] " + i);

while ((line=br.readLine())!=null){

int a=line.length(),z;

String y="";

for(z=a-1;z>=0;z--){

y+=line.charAt(z);

}

System.out.println(y);

string+=line+"\n";

}

}

br.close();

}

catch (Exception e){

System.out.println(e.toString());

}

System.out.println("Exit from B");

}

}

class C extends Thread {

String msg;

private int startIdx, nThreads, maxIdx,k;

public C(String ms,int s, int n, int m,int k)

{

this.msg=ms;

this.startIdx = s;

this.nThreads = n;

this.maxIdx = m;

this.k=k;

}

public void run() {

System.out.println("Thread C started");

String content = null;

File file = new File(msg);

String string="";

//reading

try{

InputStream ips=new FileInputStream(file);

InputStreamReader ipsr=new InputStreamReader(ips);

BufferedReader br=new BufferedReader(ipsr);

String line;

for(int i = this.startIdx; i < maxIdx; i += this.nThreads)

{

// System.out.println("[ID " + this.getId() + "] " + i);

while ((line=br.readLine())!=null){

int a=line.length(),z;

String y="";

for(z=a-1;z>=0;z--){

y+=line.charAt(z);

}

System.out.println(y);

string+=line+"\n";

}

}

br.close();

}

catch (Exception e){

System.out.println(e.toString());

}

System.out.println("Exit from C");

}

}

class D extends Thread {

String msg;

private int startIdx, nThreads, maxIdx,k;

public D(String ms,int s, int n, int m,int k)

{

this.msg=ms;

this.startIdx = s;

this.nThreads = n;

this.maxIdx = m;

this.k=k;

}

public void run() {

System.out.println("Thread D started");

String content = null;

File file = new File(msg);

String string="";

//reading

try{

InputStream ips=new FileInputStream(file);

InputStreamReader ipsr=new InputStreamReader(ips);

BufferedReader br=new BufferedReader(ipsr);

String line;

for(int i = this.startIdx; i < maxIdx; i += this.nThreads)

{

//System.out.println("[ID " + this.getId() + "] " + i);

while ((line=br.readLine())!=null){

int a=line.length(),z;

String y="";

for(z=a-1;z>=0;z--){

y+=line.charAt(z);

}

System.out.println(y);

string+=line+"\n";

}

}

br.close();

}

catch (Exception e){

System.out.println(e.toString());

}

System.out.println("Exit from D");

}

}

class E extends Thread {

String msg;

private int startIdx, nThreads, maxIdx,k;

public E(String ms,int s, int n, int m,int k)

{

this.msg=ms;

this.startIdx = s;

this.nThreads = n;

this.maxIdx = m;

this.k=k;

}

public void run() {

System.out.println("Thread E started");

String content = null;

File file = new File(msg);

String string="";

//reading

try{

InputStream ips=new FileInputStream(file);

InputStreamReader ipsr=new InputStreamReader(ips);

BufferedReader br=new BufferedReader(ipsr);

String line;

for(int i = this.startIdx; i < maxIdx; i += this.nThreads)

{

//System.out.println("[ID " + this.getId() + "] " + i);

while ((line=br.readLine())!=null){

int a=line.length(),z;

String y="";

for(z=a-1;z>=0;z--){

y+=line.charAt(z);

}

System.out.println(y);

string+=line+"\n";

}

}

br.close();

}

catch (Exception e){

System.out.println(e.toString());

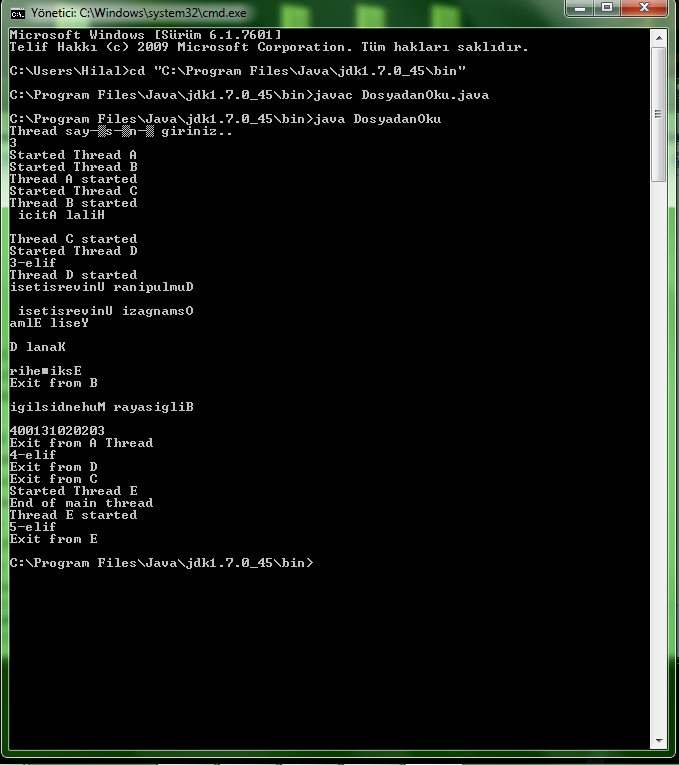
}

System.out.println("Exit from E");

}

}

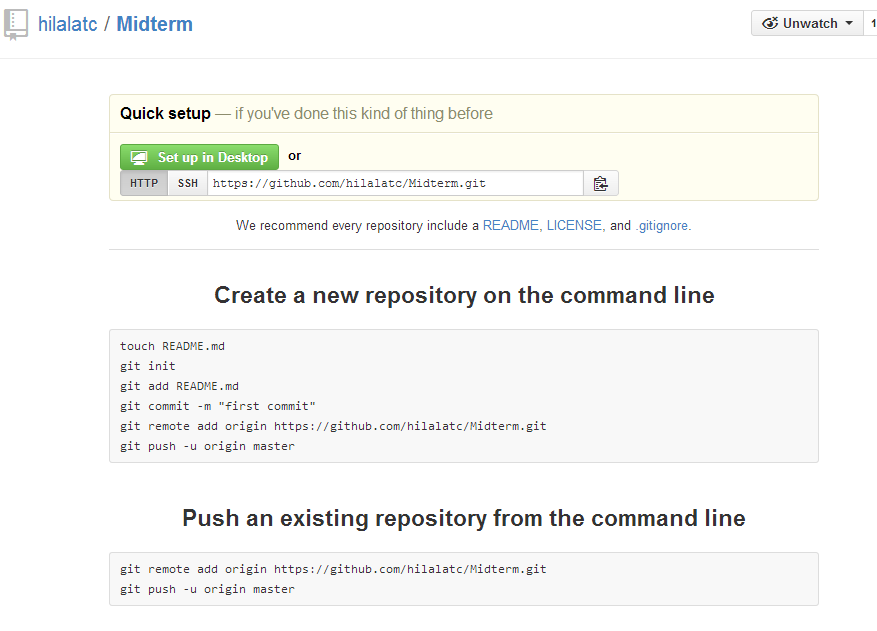
**Ekran Çıktısı:**



Resim 6.Ekran Çıktısı

**Bonus:**

Midterm tagını oluşturdum.



Resim 7.Bonus