|  |
| --- |
| A black and white image of a building  Description automatically generated with medium confidence  Budapesti Műszaki és Gazdaságtudományi Egyetem  Villamosmérnöki és Informatikai Kar |
| Kígyó játék tervdokumentáció |
| Beágyazott rendszerek szoftvertechnológiája házi feladat |
|  |
|  |
| Bilibok Bence – VRERYK |
|  |

BUDAPEST, 2023

|  |
| --- |
|  |

## Funkcionális elvárások

A játék néhány kritériumot teljesítenie kell, hogy a projekt teljesértékű legyen. Ezen funkcionális elvárások segítségével a játék funkciói és felhasználói élménye meghatározható, és az ehhez szükséges fejlesztési feladatok is könnyebben áttekinthetők. A funkcionális elvárások segítenek az összes érintett félnek (fejlesztők, megrendelő, stakeholderek) egyetértésre jutni a játék tervezésével és fejlesztésével kapcsolatban.

A Snake multiplayer játék funkcionális elvárásai a következők lehetnek:

1. A játék funkcionálisan működjön UDP kapcsolaton keresztül.
2. Az online játék működjön a szerveren, ahol a játékosok csatlakozhatnak és egymással játszhatnak.
3. A játék lehetőséget ad a játékosoknak, hogy kígyójukat mozgassák a pályán, gyűjtsék a pontokat és kerüljék el a falakat és más játékosokat.
4. A játékosok kígyói meghalnak, ha falba vagy más kígyóba ütköznek.
5. A játékosok kígyója hosszabbodik, amikor az pontot gyűjt fel.
6. A játék folyamatosan frissüljön a játékosok adatbázisával, hogy a kígyók mozgása valós időben legyen látható.
7. A játék képes legyen kezelni a játékosok késleltetését, hogy biztosítsa az egyenlő játékot.
8. A játékosoknak pontot kell számolni aszerint, hogy hány zavazatot gyűjtöttek össze.

## User Interface

A játék UI a Java-ban használható AWT (Abstract Window Toolkit) és Swing toolkit által nyújtott funkciók által lesz megvalósítva, ami egy standard API a felhasználók vizuális élményei biztosítására.

Elsősorban egy képernyő fog megjelenni a program indítása után, ki lesz írva a játék indításáig hátralevő másodpercek száma. Itt a még csatlakozni vágyó játékosokat várjuk, és amikor mindenki megérkezett, indul a játék

Ha valaki magába ütközik a kígyóval, annak a képernyőjén meg lesz jelenítve az elért pontszámai, ezek alapján lehet a győztes hirdetni.

## Tesztelés

A játék egyszerű jellege miatt a tesztelés nem von maga után túl összetett folyamatokat. Elsősorban a kapcsolat sikeres létrejövetele a tesztelés legfontosabb feladata, tud-e a kliens és a szerver egymás között üzeneteket váltani. Ezt a legegyszerűbb, System.out.println() metódust alkalmazva vizsgálom meg.

A játékidőben tesztelésre kerülő mozzanatok azt foglalják magukba, hogy ellenőrizzük a kapcsolat úgy valósult-e meg, hogy egyik félnél sem késik a kígyó irányítása, és nincsenek-e olyan lag-ok a játékban, ami miatt több képkocka kiesik és így a játékos könnyen magának ütközhet. Emellett játékidőben szükséges tesztelni, hogy a különböző forgatókönyvű játékkimenetekre megfelelően végződik-e a játék, valamint hogy a megevett almák után megfelelően növekednek-e a kígyók.

## Technikai specifikációk

A játékot a JAVA objektum-orientált programozói nyelv segítségével fogom lefejleszteni, ennek nagy előnye az, hogy több platformon is futtatható. A nyelv hozzátártozója több olyan könyvtár, ami a segítségemre lehet a játék megvalósításában.

Az egyik ilyen csomag az AWT API, amely a GUI, Graphical User Interface, egyablakós applikációk fejleszéséhez segítségül alkalmas. Itt olyan komponensek jellenek meg, mint a Label, TextArea, Checkbox, Choice. Ez a könyvtár továbbá tartalmazza a .eventet, amivel a billentyűlenyomásokat és az egérkattintásokat, vagy akár a kurzor helyét követhetjük.

A másik csomag, aminek a hasznát vettem az ugyancsak a JAVA-ba egy beépített könyvtár, az a Util könyvtár, amely adatstruktúrákat, mint a List, Collection, fájlkezelést, random szám generálást tartalmaz.

Az UDP kapcsolat megvalósításához a java.net csomag UDP specifikus osztályai kerültek használatra.

## UPD kapcsolat

+-------------+ broad Port(=5656) +--------------+  
|***Game***|---------------------------->|Client||Handler|<--------------------------- |\_\_\_\_\_\_\_|  
+-----------------+ inputPort +------------------+  
***|  
| starts the game*V+--------------+ clientPort +--------------+|*Game|---------------------> |*Client*|  
|*Manager*|<--------------------- |***|  
+---------------+ gamePort +---------------+

## Elvi rajz

### A játék inicializálása

A játék inicializálásának a folyamata, block diagram rajza az alábbi 1. ábrán látható.

A screenshot of a diagram

Description automatically generated with low confidence

*Ábra 1. A játék inicializálásának folyamata, a szerver elindítása és a kliens csatlakoztatása*

### A szerver által küldött üzenetek a klienseknek

A 2. ábra a szerver által küldött üzeneteket mutatja be, időrendi sorrenben a „Game was initialized” pillanat után.

A picture containing text, screenshot, font, line

Description automatically generated

*Ábra 2. A szerver üzenetei a kliensnek a játék folyamán*