1. **Követelmény, projekt, funkcionalitás**
   1. **Bevezetés**
      1. **Cél**

A dokumentum célja ismertetni a csapat által elkészítendő szoftver követelményeit, alapvető tulajdonságait, valamint a csapat által alkalmazott segédeszközöket, folyamatokat.

* + 1. **Szakterület**

A szoftver egy játék, nem kifejezett szakterületre készül, hanem általános felhasználára, szórakoztatásra. A játék a sokoban nevű játék egy variációja.

* + 1. **Definíciók, rövidítések**

Eclipse - fejlesztőkörnyezet

funkció - A program egy elvárt, egységként kezelhető működési része

Git - verziókezelő rendszer

GitHub - Git-et használó szolgáltatás, elsősorban forráskódok kezelésére

Google Docs - webalapú szövegszerkesztő

HSZK - Hallgatói Számítógép Központ

instance - adott szoftver futó példánya

integráció - egységesítés, beolvasztás

interfész - a számítógép és a felhasználó érintkezési felülete

Java9 - a Java programozási nyelv 9-es verziója

JDK9 - Java Development Kit, Java programozási nyelvhez használt fejlesztőeszköz

Jenkins - folyamatos integrációs eszköz

JRE9 - Java Runtime Environment, Java nyelven írt alkalmazások futtatásához szükséges program

proto - A program azon állapota amikor már a belső működés megvan, de még nem rendelkezik grafikus megjelenítéssel

standard - szabvány

szkeleton - a program azon állapota amikor a belső felépítése már megvan de még nem működik

szoftver - számítógépre írt program és a hozzá tartozó dokumentáció

use-case - használati eset

WhiteStarUML - modellező eszköz melynek segítségével UML diagramokat lehet készíteni

* + 1. **Hivatkozások**

Feladatkiírás - https://www.iit.bme.hu/targyak/BMEVIIIAB02/feladat

* + 1. **Összefoglalás**

A dokumentum későbbi részeiben a következők találhatók:

* Áttekintés, ami nagyvonalakban bemutatja a szoftvert
* A szoftverrel kapcsolatos követelmények
* Lényegesebb use-case-ek felsorolása, valamint use-case diagram
* Szótár, ami a szoftverrel kapcsolatos nem hétköznapi szavak definícióit tartalmazza
* A projekt kivitelezésének terve
* Projekt napló
  1. **Áttekintés**
     1. **Általános áttekintés**

A szoftver két alapvető részből áll:

* játék logika
* megjelenítés

A megjelenítő feladata egy interfészt adni a felhasználó felé, amivel a játék logikát manipulálni lehet.

A játék logika felel a játék szabályainak betartásáért, a játék vége feltételek figyeléséért, a játék állapotának nyomon követéséért.

A Pálya alrendszer felel a pályák betöltéséért és a játéklogika számára rendelkezésre állítja azokat.

* + 1. **Funkciók**

A program egyfajta logikai játék. A célja, hogy a raktárban lévő ládákat a megfelelő helyre juttassuk, ezzel pontokat gyűjtve. A játékosok a munkások mozgatásával képesek a ládák helyzetét változtatni. A raktári rakodómunkások között folyik verseny, a kékpolós munkás és a piros polós munkás harca ez. A megfelelő ládák, megfelelő sorrendben való mozgatásával kell eljuttatni a lehető legtöbb ládát a célba. A gyors, stratégikus gondolkodás elengedhetetlen. A ládák célba juttatását lehet akadályozni is kapcsolókkal aktivált lyukakkal. A lyukakban a ládák és a játékosok is eltűnnek. Továbbá a pályán találhatóak nem vezérelt lyukak is, ezek mindig ott vannak. Így színes variációt biztosít a játék a stratégiáknak.  
A játék időre megy. Az idő lejártával a játék véget ér, ekkor a magasabb pontszámú játékos nyer. A játék idő előtt is véget érhet, amennyiben az egyik játékos beleesik egy lyukba, vagy az egyik játékos a falnak préselődik és persze fel is adhatja a játékot.

A munkások a ládákat mozgatva képesek egymást eltolni. Amennyiben nincs hely, hogy a megfelelő irányba elmozduljon a játékos, tehát falba vagy ládába ütközik, akkor meghal és a másik játékos nyer. A ládákat azonban bölcsen kell rendezni mivel egymásba nem tolhatóak és így elzárhatják a munkások útját.  
A pontokat a játék során a megfelelő mezőkre tolt ládák számával lehet növelni, így ezek maximalizálva vannak a játék kezdetekor, a raktárban található ládák számával. Így teret adva olyan taktikai meggondolásoknak mint a másik játékos ládáinak megsemmisítése a csapdákkal. A pályán a rakodóterületek, ahova a ládákat el kell juttatni, változatosak. Lehetnek több részletben elszórva, vagy egybefüggően.

A csapdák olyan kapcsolók amik nyomás, azaz súly hatására aktiválódnak és kinyitnak egy lyukat a raktár padlóján. Fontos megjegyezni, hogy a csapdákat a munkások súlya nem tudja aktiválni, ehhez egy láda súlya szükséges. Amennyiben elmozgatjuk a csapdáról a rajta található ládát, akkor a kapcsoló visszaáll és a lyuk bezárul. Érdemes tehát megfontoltan mozogni és megjegyezni a lyukak helyét. A játékban az ilyen kapcsolható csapdákon kívül vannak a pályán permanens lyukak is.

A pályák szerkezete megköveteli az előre gondolkodást és a tervezést, a munkások a pálya egy megszabott területéről indulnak. A játék indulás előtt hagy néhány másodpercet a pálya szemrevételezésére és a munkaterv megalkotására. A játék időtartama alatt a ládák helyzete jelentősen változhat. Ehhez a jó munkásnak alkalmazkodnia kell tudni, a legjobb munkásnak pedig előre látni ezeket a változásokat.

A játékosok a játék indítása utána kiválaszthatják, hogy mely adott sorszámú pályán akarják esetleg összemérni erejüket. A játék végeztével választhatunk másik pályát de akár visszavágót is kérhetünk. A játék időtartama változtatható, így adva még több változatosságot a kihíváshoz. Például a gyakorlott munkások versenyéhez egy rövidebb, izgalmasabb időkorlát adható, a kezdő rakodók pedig állíthatnak több idő a játék kitanulásához illetve hogy hozzászokjanak a raktár tempójához.

* + 1. **Felhasználók**

A felhasználók a játék természetét, a lehetőségeket és az eszközök használatát könnyen elsajátíthatják. A megfelelő rövid leírások elolvasásával, speciális képzettség nélkül játszható, korra való tekintet nélkül.

* + 1. **Korlátozások**

A szoftver elkészítéséhez csak a Java9 Standard könyvtárait lehet felhasználni.

* + 1. **Feltételezések, kapcsolatok**

Feladatleírás - https://www.iit.bme.hu/targyak/BMEVIIIAB02/feladat

* 1. **Követelmények**
     1. **Funkcionális követelmények**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Use-case** | **Komment** |
| 1.1 | A pálya mezőkből áll | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.2 | A mezőn lehet munkás | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.3 | A munkások mozgathatók | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.4 | A munkások meghalhatnak | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.5 | A mezőn lehet lyuk | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.6 | A lyukba bele eshet a játékos | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.7 | A lyukba bele eshet a láda | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.8 | Vannak dobozok a pályán | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.9 | A dobozokat mozgathatja a játékos | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.10 | Doboz tolhat dobozt | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.11 | Doboz tolhat játékost | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.12 | A doboz ha célba ér eltűnik | bemutatás | Alapvető | Csapat | View Map |  |
| 1.13 | A mezőn lehet fal | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.14 | A pálya széle fal | bemutatás | Fontos | Csapat | View Map |  |
| 1.15 | A mezőn lehet kapcsoló | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.16 | A kapcsoló nyitja a lyukat ha aktív | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.17 | A kapcsoló zárja a lyukat ha nem aktív | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.18 | A kapcsolót a doboz aktiválja | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.19 | A kapcsolót a játékos nem aktiválja | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.20 | A kapcsoló visszaáll ha nincs rajta doboz | bemutatás | Alapvető | Csapat | Control Traps |  |
| 1.21 | A játék kezdete előtt a pálya megtekinthető | bemutatás | Fontos | Csapat | View Map |  |

* + 1. **Erőforrásokkal kapcsolatos követelmények**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| 2.1 | Git | nincs | alapvető | csapat | elolsztott verziókezelő |
| 2.2 | GitHub | nincs | alapvető | csapat | Hosztolt Git repo |
| 2.3 | Eclipse | nincs | alapvető | nincs | Integrált fejlesztőkörnyezet |
| 2.4 | Billentyűzet | nincs | alapvető | csapat |  |
| 2.5 | Egér | nincs | alapvető | csapat |  |
| 2.6 | Monitor | nincs | alapvető | csapat |  |
| 2.7 | BME HSZK-ban található PC | bemutatás | alapvető | megrendelő |  |
| 2.8 | WhiteStarUML | nincs | alapvető | csapat | UML modellező program |
| 2.9 | Google Docs | nincs | alapvető | csapat | Szövegszerkesztő |
| 2.10 | JRE9 & JDK9 | nincs | alapvető | csapat | Futtatókörnyezet |
| 2.11 | Jenkins | nincs | opcionális | csapat | Continous Integration |

* + 1. **Átadással kapcsolatos követelmények**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| 3.1 | Szkeleton beadás | bemutatás | alapvető | megrendelő | március 19. |
| 3.2 | Proto beadás | bemutatás | alapvető | megrendelő | április 23. |
| 3.3 | Grafikus beadás | bemutatás | alapvető | megrendelő | május 1 |
| 3.4 | Segítség nélkül, útmutató alapján telepíthető | bemutatás | alapvető | megrendelő |  |
| 3.5 | Bemutatás a HSZK-ban | bemutatás | alapvető | megrendelő |  |

* + 1. **Egyéb nem funkcionális követelmények**

Nincs ilyen.

* 1. **Lényeges use-case-ek**

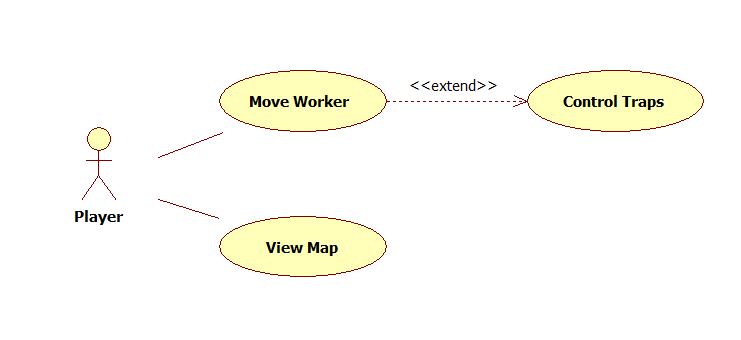
**Use-case leírások**

|  |  |
| --- | --- |
| **Use-case neve** | Move Worker |
| **Rövid leírás** | A játékos irányítja a munkást. |
| **Aktorok** | Player |
| **Forgatókönyv** | 1.A játékos felfelé ,lefelé balra,jobbra mozgathatja a munkást. |
| **Alternatív Forgatókönyv** | 1.A.1.Ha a munkás lyukra lép,akkor meghal,ekkor a játék véget ér. |
| **Alternatív Forgatókönyv** | 1.B.1.Ha a munkásra ládát tolnak akkor egy mezőt arrébb lép |
| **Alternatív Forgatókönyv** | 1.B.2.Ha a munkásra ládát tolnak,de az nem tud arrébb mozdulni akkor meghal. |

|  |  |
| --- | --- |
| **Use-case neve** | View Map |
| **Rövid leírás** | A játékos megtekinti a pályát. |
| **Aktorok** | Player |
| **Forgatókönyv** | 1.A rendszer kirajzolja a pálya aktuális állapotát.  2.A játékos megtekinti a pálya aktuális állapotát. |

|  |  |
| --- | --- |
| **Use-case neve** | Control Traps |
| **Rövid leírás** | A rendszer a padlókon lévő csapóajtókat nyitja vagy zárja. |
| **Aktorok** | Player |
| **Forgatókönyv** | 1.A csapóajtóra láda kerül és aktiválja.  2.A csapóajtóról lekerül a láda és deaktiválódik. |

* + 1. **Use-case diagram**



* 1. **Szótár**

csapda, lyuk – egy mezőnyi hely a pályán melyet egy kapcsoló irányít. Két állapota van: ha zárva van padlóként azaz semlegesen viselkedik, ha nyitva akkor eltünteti a rajta tartózkodó ládát vagy munkást

fal – a pálya szélét reprezentáló objektum

kapcsoló – a munkások által állítható kétállapotú rendszer mely egy hozzá kapcsolt csapdát irányít

kék – a második játékos által irányított munkás megkülönböztető színe

játék vége – akkor következik be, ha valamelyik munkás meghal vagy ha lejár az idő, az a munkás nyer, aki életben maradt, vagy ha mindketten élnek akkor az, aki több pontot szerzett

láda – munkások által tolással mozgatható objektum, a cél, hogy a ládát a rakodóterületre juttassuk

meghal – egy munkás akkor hal meg ha nyitott csapdán tartózkodik vagy ha egy másik munkás egy ládával falnak vagy oszlopnak nyomja

oszlop – falként viselkedő egységnyi terület azonban a pályán bárhol lehet, ahol nincs csapda vagy rakodóterület

piros – az első játékos által irányított munkás megkülönböztető színe

pont – egy adott munkás teljesítményét jellemző szám, ha egy munkás egy ládát egy rakodóterületre tol akkor eggyel növeli ezt a számot

(rakodó)munkás – a játékosok által irányított szereplők

rakodóterület – az az egységnyi hely, ahova egy ládát eljuttatva egy munkás pontot kap

raktár – a játék pályája

## Projekt terv

### 2.6.1 Csapat

Csapatunk 5 főből áll, a felelősségeket a következőképp osztottuk ki a tagok közt, annak érdekében, hogy mindenki nagyjából egyformán legyen terhelve:

|  |  |
| --- | --- |
| **Név** | **Felelősségek** |
| Pünkösd Marcell | Kód és dokumentáció |
| Torma Kristóf | Menedzsment és kód |
| Tóth Dániel Péter | Kód és dokumentáció |
| Tóth Vince | Kód és dokumentáció |
| Veres Csaba | UML és dokumentáció |

### 2.6.2 Kommunikáció

**Telegram** A csapatnak létrehoztunk egy supergroup-ot telegrammon, ahol sürgős csapatot érintő dolgokat azonnal el tudjuk intézni. Továbbra ezt használjuk a kérdéses ügyek megbeszélésére is, amely nem igényel személyes találkozót.

**Személyes megbeszélések** Eseti alkalmakkor, ha úgy érezzük, szükséges valamit közösen megbeszélni, akkor előre egyeztetünk időpontot a találkozásra. Ezen kívül minden szerdán 8:15-kor találkozunk a konzultáción.

### 2.6.4 Csapatmunkát támogató eszközök

**Verziókezelés** A verziókezeléshez a Git nevű verziókezelő rendszert használjuk. Ezzel könnyebbé téve a csapatnak, hogy ugyanazon a kódbázison dolgozzunk mindannyian. A kódunkat a GitHub központi tárhelyén tároljuk. Ennek segítségével könnyen követhetjük, hogy ki mivel foglalkozott, és hogy kinek a hibája, ha valami nem működik. :)

**Időkövetés** Írtunk python-ban egy telegram botot, amellyel könnyen követni lehet, hogy ki mikor mennyit dolgozott a projekten. Illetve erről kiértékelést is tudunk készíteni. Ez többek közt segíti a napló megírását.

**Tesztelés** A tesztelés automatizálására a Jenkins nevű szoftvert használjuk, ami automatikusan lefuttatja a tesztelést, és erről részletes jelentést készít. Onnan tudjuk, hogy rendben van a szoftver, hogy minden “zöld”.

### 

### 2.6.3. Használt programok

**Verziókezelés** A verziókezeléshez a fent említett okokból a Git programot használjuk.

**Dokumentáció** A Dokumentációt a Google Docs szöveg szerkesztőjével írjuk, a benne megtalálható UML diagrammok pedig WhiteStarUML-ben készülnek. Azért a Google Docs-ra esett a választás, mert könnyű benne több embernek ugyanazon a dokumentumon dolgozni, és egyből látjuk a mások változtatásait.

**Fejlesztő környezet** A forráskódotEclipse IDE-ben készítjük.

### 2.6.4. Mérföldkövek és határidők

|  |  |  |
| --- | --- | --- |
| **Dátum** | **Leírás** | **Ellenőrzés** |
| 2018. 02. 19. | Követelmény, projekt, funkcionalitás | beadás |
| 2018. 02. 26. | Analízis modell kidolgozása 1. | beadás |
| 2018. 03. 05. | Analízis modell kidolgozása 2. | beadás |
| 2018. 03. 12. | Szkeleton tervezése | beadás |
| 2018. 03. 19. | Szkeleton | beadás |
| 2018. 03. 21. | **Szkeleton** | bemutatás |
| 2018. 03. 26. | Prototípus koncepciója | beadás |
| 2018. 04. 09. | Részletes tervek | beadás |
| 2018. 04. 23. | Prototípus | beadás |
| 2018. 04. 25. | **Prototípus** | bemutatás |
| 2018. 05. 02. | Grafikus felület specifikációja | beadás |
| 2018. 05. 14. | Grafikus változat | beadás |
| 2018. 05. 16. | **Grafikus változat** | bemutatás |
| 2018. 05. 18. | Összefoglalás | beadás |

A feladatot 3 lépcsőben oldjuk meg, a lépcsőket vastagon szedve jelöltük. Ezek bővebben kifejtve:

A **szkeleton** változat célja annak bizonyítása, hogy az objektum és dinamikus modellek a definiált feladat egy modelljét alkotják. A szkeleton egy program, amelyben már valamennyi, a végső rendszerben is szereplő business objektum szerepel. Az objektumoknak csak az interfésze definiált. Valamennyi metódus az indulás pillanatában az ernyőre szöveges változatban kiírja a saját nevét, majd meghívja azon metódusokat, amelyeket a szolgáltatás végrehajtása érdekében meg kell hívnia. Amennyiben a metódusból valamely feltétel fennállása esetén hívunk meg más metódusokat, akkor a feltételre vonatkozó kérdést interaktívan az ernyőn fel kell tenni és a kapott válasz alapján kell a továbbiakban eljárni. A szkeletonnak alkalmasnak kell lenni arra, hogy a különböző forgatókönyvek és szekvencia diagramok ellenőrizhetők legyenek. Csak karakteres ernyőkezelés fogadható el, mert ez biztosítja a rendszer egyszerűségét.  
  
A **prototípus** program célja annak demonstrálása, hogy a program elkészült, helyesen működik, valamennyi feladatát teljesíti. A prototípus változat egy elkészült program kivéve a kifejlett grafikus interfészt. Ez a program is parancssorból futtatható és karakteres ernyőkezelést alkalmaz. A változat tervezési szempontból elkészült, az ütemezés, az aktív objektumok kezelése megoldott. A business objektumok - a megjelenítésre vonatkozó részeket kivéve - valamennyi metódusa a végleges algoritmusokat tartalmazza. A megjelenítés és működtetés egy alfanumerikus ernyőn vezérelhető és követhető, ugyanakkor a vezérlés fájlból is történhet és a megjelenítés fájlba is logolható, ezzel megteremtve a rendszer tesztelésének lehetőségét. Különös figyelmet kell fordítani a parancssori interfész logikájára, felépítésére, valamint arra, hogy az mennyiben tükrözi és teszi láthatóvá a program működését, a beavatkozások hatásait.  
  
A teljes ( **grafikus** ) változat a prototípustól elvileg csak a kezelői felület minőségében különbözhet. Ennek változatnak az értékelésekor a hangsúlyt sokkal inkább a megvalósítás belső szerkezetére, semmint a külalakra kell helyezni.

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.02.12 19:00 | 0,5 óra | Pünkösd  Tóth V  Tóth D  Veres  Torma | Értekezlet.  Döntés: Pünkösd elkészíti a 2.6-os részt, Tóth V elkészíti a 2.5-ös részt, Tóth D elkészíti a 2.4es részt, Veres elkészíti a 2.3-as részt, Torma elkészíti a 2.1 és 2.2-es részeket |
| 2018.02.12. 20:00 | 2 óra | Torma | Tevékenység: Torma elkészítette a 2.1-es és 2.2-es részeket |
| 2018.02.13. 17:00 | 2 óra | Veres | Tevékenység: Veres elkészítette a 2.3-as részt. |
| 2018.02.14. 21:00 | 2 óra | Pünkösd | Tevékenység: Pünkösd elkészítette a 2.6-os részt. |
| 2018.02.15. 21:00 | 2 óra | Tóth V | Tevékenység: Tóth V elkészítette a 2.5-ös részt. |
| 2018.02.16. 21:00 | 2 óra | Tóth D | Tevékenység: Tóth D elkészítette a 2.4-es részt. |