

# HF specifikáció

---

## 1. Röviden

Feladatom egy valós idejű stratégiai játék elkészítése. A játék egy hatszögekből álló(~méhsejtes) pályán játszódik. A felhasználó épületeket épít, melyek a saját embereiket automatikusan vezérlik. A játékos feladata hogy elpusztítsa az ellenfelének épületeit.

## 2. A játékos

A játékosnak először játékot indít egy főmenüből, melyben vagy új játékot indít vagy egy már megkezdettet tölt be és folytat.

A játék közben a játékos épületeket épít a pálya szabad mezőire, valamint utakat hoz létre, valamint a meglévő épületek adatait megtekintheti. Ezen kívül lehetősége van a pálya teljes területén navigálni, illetve az egyes mezők saját tulajdonságait megvizsgálni.

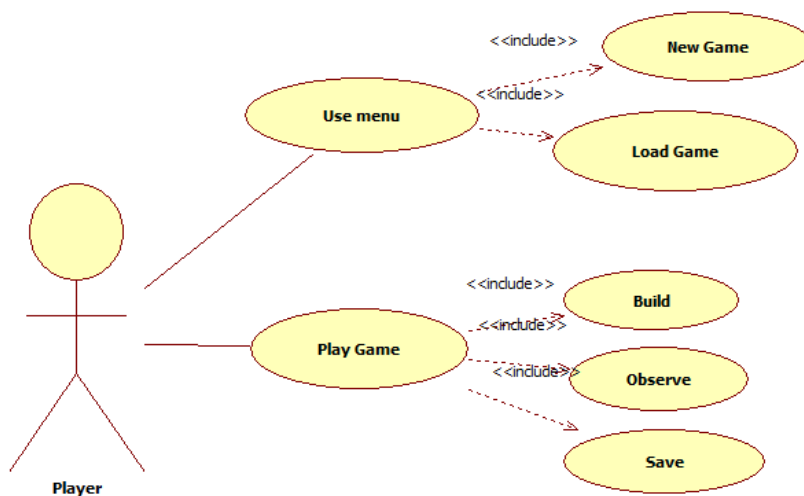
## 3. A játék elemei

- Mező: A pályát lefedő hatszöges mezők, melyekre az utak és épületek kerülnek. Egyes mezők alatt nyersanyagok rejtőzhetnek(itt ebben a konkrét esetben vas), melyeket a felhasználó megtekinthet (megfelelő billentyű lenyomása után ezek a mezők elszíneződnek)
- Épületek: Minden épület felépítése egyes nyersanyagokat igényel (ezek pontos aránya a játék egyensúlyától függ). Miután felépültek minden épület egy saját viselkedést követ (ezekről az egyes épületek leírásánál) Az épületek megadott esetben felépítés után elpusztíthatók.
- Fő épület: A játékos alapvetően egy fő raktár épülettel kezd, ide érkezik minden nyersanyag, valamint innen kérhetnek az épületek a saját építésükre.
- Nyersanyagok: itt elsősorban a különféle épületek építésére szánt anyagok, mint fa és kő, valamint fegyver készítéséhez tartozó vas szerepel.
- Fa: Az egyes mezőkre helyezett fák miután elérték teljes méretüket (ez egy idő függő dolog) véletlen szerűen szaporodhatnak szomszédos mezők felé. Egy favágó segítségével nyersanyag nyerhető ki belőle mely a fa elpusztulásával jár.
- Kő: Meglévő nyersanyag egyes mezőkön található kőhalmok formájában. Kőfejtő-vel nyersanyag(kő) nyerhető belőle. A játékban véges mennyiségben áll rendelkezésre, de egyes mezőkön nagyobb mennyiség is állhat.
- Kereszteződés és út: A kereszteződések között utak húzhatók. Egyes ilyen két elem közötti útszakaszhoz tartozik egy kiszolgáló ember, mely a nyersanyagokat csomópontról csomópontra szállítja, így voltaképp „csatárláncban” juttatja el a szükséges helyre. (Ez feltételezi, hogy a szállítandó anyag képes eldönteni merre kell tovább haladnia.) Egyes kereszteződések között utak futnak(ezek két egymástól nem túl messzi kereszteződésre kattintva hozhatóak létre, illetve nem keresztezhetik egymást)
- Belépési pont: Minden épülethez tartozik egy saját belépési pont, mely egy speciális kereszteződés. Az épületek minden forgalma innen indul ki és ide fut be.

- Favágó: Megépítése után automatikusan egy adott távon belül keres felnőtt fákat, majd ezeket kivágja és a szerzett anyagot visszajuttatja az épülethez, ami később a főépületbe küldi.
- Kőfejtő: Megépítése után automatikusan egy adott távon belül keres kőhalmokat, melyekből kivesz egy „kő” nyersanyagot majd visszajuttatja az épülethez, illetve onnan a főépületbe.
- Bánya: A vasat rejtő mezőkre építve adott ütemben nyersanyagot vesz ki belőlük. Ez a nyersanyag elfogyhat. A kibányászott nyersanyag ezután a főépületbe kerül.
- Fegyverkovács: Vasat kér a főépulettől, melyből fegyvert készít, majd a főépületbe juttatja.
- Vár: A játékos várakat épít melyekben ha fegyver kerül beléjük egy maximum katona számig katonákat készít. Amennyiben van katona, úgy a közelében lévő ellenséges objektumokat támadja meg, melynek módja, hogy útjára indít egy katonát felé. Ezek az épületek saját védelmüktől függően vagy elpusztítják a támadót, vagy maguk pusztulnak el.
- Ellenséges Vár: Ezek a pályán már a játék indításakor szerepelnek. A játék véget ér amikor minden ilyen épület megsemmisül. A játékos váraihoz hasonlóan ezek is rátámadhatnak a túl közeli épületekre (kivéve várakra), melyeket így elpusztíthatják. Ezek is rendelkeznek saját katonákkal, melyek száma minden őket ért támadással csökken. Amikor elfogy akkor a vár megsemmisül.
- Emberek: Egyszerű cél felé mozgó katona, favágó, kőfejtő. Amennyiben a célját nem sikerül elérnie visszatér a kiindulóépületébe.

A játék nehézsége, hogy a játékosnak véges nyersanyagokból kell úgy építkeznie, hogy el tudja pusztítani az ellenfeleit.

#### 4. Use case



Cím	Use menü
-----	----------

Leírás	A játékos a főmenüt használja a játék elindításához
Aktorok	Player
Főforgatókönyv	A játék kirajzolja a new Game s Loda game gombokat melyekből választ a játékos
Cím	New Game
Leírás	A játékos új játékot indít
Aktorok	Player
Főforgatókönyv	A játék a New Game-ra kattintva elindít egy új játékot, melye betölti a pályát. A játékosnak nincs módja a pálya paramétereinek kiválasztásához

Cím	Load Game
Leírás	A játékos betölt egy mentett játékot
Aktorok	Player
Főforgatókönyv	<ul style="list-style-type: none"> <li>• A játék a Load Game-ra kattintás után kilistázza az eddig elmentett játékokat</li> <li>• A játékos kiválasztja a betöltendő játékot</li> <li>• A játék betöltődik</li> </ul>

Cím	Save
Leírás	A játékos elmenti a játékot
Aktorok	Player
Főforgatókönyv	<ul style="list-style-type: none"> <li>• A játékos a save gombra kattint.</li> <li>• A megjelenő menüben nevet ad a játéknak</li> </ul>

Cím	Play game
Leírás	A játékos a játék indítása után hozzáfér a felhasználói felülethez.
Aktorok	Player
Főforgatókönyv	<ul style="list-style-type: none"> <li>• A játékos kiválasztja a mezőt</li> <li>• A megjelenő elemek közül kiválasztja az építendő elemet</li> </ul>

Cím	Build
Leírás	A játékos játék közben egy kiválasztott mezőre épít
Aktorok	Player
Főforgatókönyv	<ul style="list-style-type: none"> <li>• A játékos kiválasztja a mezőt</li> <li>• A megjelenő elemek közül kiválasztja az építendő elemet</li> <li>• (út esetén kiválasztja a következő csomópontot)</li> </ul>

## 5. Megoldási terv:

A megjelenítéshez és a bemenetek kezeléséhez javafx grafikus könyvtárat használok. A játék közben megjelenő elemeket első sorban ImageView-ekkel kezelem, mivel ez habár nem túl gyors, de egyszerű megoldást jelent, lehetőséget adva effektek hozzáadására és kattintások kezelésére. A körökhöz a javafx-ben erre a célra készített időzítőjét használok. A képernyőn navigáláshoz KeyListener-ekkel rögzítem a billentyű lenyomásokat.

A menüben egyszerű Buttonok, míg a betöltésnél ComboBox szerű elemből választható ki az adott játék.

Futás közben a játék egy „World” objektumban tárolja az adatait. Ezek a mezők, illetve a Tick-elő és látható objektumok. (A mezők egy kétdimenziós tömbben helyezkednek el, így könnyen címezhetők, míg a többi lista a dinamikus változások kezelésére). A játék minden körben meghívja az összes Tick-elni képes elem függvényét melyek lépnek, majd megjeleníti a játék aktuális állapotát. A több szálon dolgozó elemek mind a „World” objektumra szinkronizálnak rá, így kerülve el az inkonzisztens állapotokat.

Az objektumok egymás közötti útkereséseit szélességi bejárást alkalmazok, illetve a megfelelő objektumok megtalálását Visitor minta szerint biztosítom.

A főépületbe érkező kérések egy lista végére kerülnek, melyből minden körben a lehetséges kérések végrehajtásra kerülnek.

Mentésre a beépített serializálást használok, mivel ez a központit tárolóval együtt jól használható.