

# SPACE VS INVADERS

BY ANDIKVÁRIUM – ANDI PÉTER, GÓGA ANDREA, GYIMESI KRISTÓF

## TARTALOMJEGYZÉK

Space Vs Invaders .....	1
Alapfeladat.....	2
Tematika: .....	2
Célközönség: .....	2
Fogalomtár:.....	2
Megvalósíthatósági elemzés .....	3
Követelmények feltárása és elemzése .....	4
Kutatás .....	4
Osztályozás és priorizálás: .....	4
Játékot megelőzően: .....	5
Játék közben: .....	5
Rendszer architektúra .....	6
Rendszertervezés .....	6
Rendszermodellek: .....	8
Rendszerevolúció: .....	9
Termékkövetelmények: .....	9
Becsült Hardver .....	9
Követelményspecifikáció .....	9
Használati esetek: .....	10
Követelmény validáció .....	10
Felhasználói történetek .....	10

## ALAPFELADAT

[http://mcserep.web.elte.hu/data/education/2019-2020-2\\_SZT\\_Feladat.pdf](http://mcserep.web.elte.hu/data/education/2019-2020-2_SZT_Feladat.pdf)

Adott egy  $n \times m$ -es pálya, amelyeknek egyik oldalán a játékos háza / vára helyezkedik el (jelölés: H), a vele átellenes oldalon pedig az ellenség egységeinek belépési pontjai találhatóak (spawn point, jelölés: S). A játékos célja, hogy épületek és tornyok építésével (jelölés: T) megelőzze, hogy a házat a számítógép által vezérelt ellenség egységei (jelölés: E) elérjék. Az ellenség egységei véletlenszerűen jelennek meg a belépési pontokon, majd egyenes irányba haladnak a játékos háza felé. A tornyok megtámadják (a hatókörükben) lévő ellenséges egységeket, illetve az ellenséges egységek is sebzik az útjukba kerülő tornyokat, azokon átlépni nem tudnak, míg le nem rombolják őket.

A játék valós idejű, a játékos folyamatosan interakcióba léphet a játéktáblával, azon építkezhet. Az ellenséges egységek léptetésére, valamint a harcra időzített módon kerül sor, de az egyes egységek előrehaladási és a tornyok tüzelési sebessége eltérő lehet. A játékos által épített épületek aranyba kerülnek. Egy kevés mennyiségű arany folyamatosan termelődik a játékos számára, emellett aranyra lehet szert tenni Termelő épület építésével és az ellenséges egységek elpusztításával. A játék jelenítse meg a játékos aktuális aranykészletét.

## TEMATIKA:

Az általunk megvalósított játékban a játékos a Föld bolygó védelméért küzd a sötét űrbéli erők ellen. Az idegen civilizáció űrhajókon keresztül támadja a játékost, míg a játékos célja, hogy műholdak pályára helyezésével eloszlassa a fenyegetést.

## CÉLKÖZÖNSÉG:

- Mindenki, akinek van számítógépe. A játék nem igényel internet kapcsolatot, telepítés nélkül, offline játszható.
- A játékot használóknak nincs szükségük korábbi játékokban szerzett esetleges tapasztalatra. Egy stratégiai játékról van szó, mely intuitívan is tökéletesen játszható kialakításának köszönhetően.
- A játék 12 éves kor alatt csak szülői engedéllyel játszható, felső korhatár természetesen nincs.
- A játék témája (űrből érkező ellenségek) korábbi népszerű, széles körben ismert játékokéhoz hasonló, ezért több generációt szólíthatunk meg általa. Az emberiséget régóta foglalkoztatja a Földön kívüli civilizáció létezése, ezért minden létjogosultsága megvan ennek a már-már sci-fibe hajló játéknak.

## FOGALOMTÁR:

1. **Játékos háza/vára/bázisa:** a játéktábla legalsó sorának összes mezője. A játékos célja, hogy megakadályozza, hogy az ellenséges karakterek elérjék ezeknek a mezőknél bármelyikét. Amennyiben ez mégis megtörténne, a játéknak azonnal vége.
2. **Tornyok, jelen esetben műholdak:** olyan épületek, amelyekkel a játékos az ellenség közeledését akadályozza meg, hátráltatja, közvetlenül (támadással) vagy közvetetten (aranytermeléssel, javítással). A tornyok a játéktábla egy fix mezőjén helyezkednek el, mozgásra nem képesek. A játékosnak a saját bázisán kívül lehetősége van a játéktábla bármely mezőjére lerakni a tornyokat.
3. **Ellenség bázisa/ellenség belépési pontjai:** a játéktábla legfelső sora. Az újonnan generált ellenséges egységek minden esetben az ez alatt elhelyezkedő sorból indulnak lefelé, a játékos bázisa felé.
4. **Ellenség/ellenséges egységek:** a számítógép által véletlenszerűen, bizonyos időközönként generált figurák, bábuk. Ezek belépési pontja minden esetben az ellenség bázisa alatti sor. Az ellenségek célja, hogy elérjék a játékos bázisát, emiatt adott időközönként lépnek lefelé.
5. **Torony tüzelése:** amennyiben ugyanabban az oszlopban jön létre ellenség, mint amelyben a torony van, a torony minden esetben támadja az ellenséget, ezáltal csökkenti ennek életerejét, a játékos szempontjából szerencsés esetben el is pusztítja azt.

6. **Ellenség elpusztul:** az ellenség típusától függően előre megadott életerővel rendelkezik, ez nem módosítható. Amennyiben a torony elegendő mennyiségű és erejű támadást adott le rá, megsemmisül: lekerül a játéktábláról, nem támad tovább.
7. **Ellenséghullám:** a játékidő alatt periodikusan jelennek meg ellenségek a játéktéren, az idő elteltével egyre több/erősebb ellenfél közelíti a várat.
8. **Torony lerombolása:** az ellenség a támadás során lenullázza a műhold életerejét, emiatt a műhold megszűnik.
9. **Torony eladása:** a játékos dönthet úgy, hogy arany fejében megszünteti a tornyot. Ennek a cselekvésnek eredményeképpen kevesebb aranyat kap, mint a torony eredeti ára.
10. **Torony fejlesztése:** a kiválasztott tornyot a játékos fejlesztheti, ez vonatkozik az adott torony életerejére és specifikus tulajdonságára (pl. tüzelő torony gyakrabban/sebzőbben tüzel) is.
11. **Különböző tornyok:** a játékban 3 féle torony lerakása közül választhatunk, ezek a következők: tüzelő torony (elsődleges célja: ellenség életerejének csökkentése), termelő torony (elsődleges célja: arany termelése) és gyógyító torony (elsődleges célja: a többi torony életerejének növelése).
12. **Különböző ellenségek:** 3 féle ellenséget különböztetünk meg, ez sebességben, sebzésben és sebezhetőségben (életerőben) különböznek egymástól.

## MEGVALÓSÍTHATÓSÁGI ELEMZÉS

Szükséges erőforrások:

- humán: 3 devops
- hardver: egy fejlesztői számítógép, közepes hardverigény
- szoftver: Linux/Windows, fejlesztőkörnyezet: VSCode, verziókövető: Git, projektvezető oldal: GitLab
- üzemeltetés: nem kell biztosítani. A szoftver nem igényel karbantartást a fejlesztést követően.
- megvalósítás időtartama: 100 emberóra

### KUTATÁS

A feladat megvalósítása hétről hétre ellenőrzött környezetben zajlik. A Scrum Master, aki a gyakorlatvezetőnk jelen esetben, minden héten visszajelzéseket ad, megválaszolja a kérdéseinket és kiegészíti a feltevéseinket. A feladatot az ezeken az alkalmakon lezajlottak szerint valósítjuk meg. Elvárásai velünk szemben a fent említett feladat megvalósítása, kivitelezése.

### OSZTÁLYOZÁS ÉS PRIORIZÁLÁS:

#### MUST HAVE:

- $n \times m$ -es pálya, melynek egyik oldalán az ellenség, a másikon pedig a játékos bázisa található
- a játéktáblára helyezhető épületek, amelyekkel a játékos az ellenségtől védi saját bázisát
- véletlenszerűen megjelenő bábuk/szörnyek az ellenség részéről
- a játékos és az ellenség karakterei küzdenek egymás ellen, gyengítve a másik életerejét
- idő mérése, tekintve, hogy idő alapján tüzelnek a tornyok, jönnek létre új ellenségek; egy adott idő után nyerhető a játék
- egy idő mentén növekvő (adott esetben épületek által termelt) egység (arany), mellyel a játékos újabb épületeket rakhat le, vagy meglévőket fejleszthet

#### SHOULD HAVE:

- **Többféle torony:** 3 féle tornyot különböztetünk meg: offenzív, gyógyító és termelő tornyok. Ezek ebben a sorrendben a következőket végzik: támadják az ellenséget, növelik a többi torony életerejét egy adott hatótávolságon belül és aranyat termelnek a játékos számára.
- **Torony fejlesztés:** egy adott torony specifikus tulajdonságát és életerejét arany fejében egyszerre fejleszthetjük.
- **Többféle ellenség:** háromféle ellenséget különböztetünk meg, mindegyiknek van egy olyan tulajdonsága, amelyben kiemelkedik társai közül, ez történhet sebességben, sebzésben meg életerőben.
- **Ellenség hullámok:** periodikusan, az idő függvényében ellenségek jelennek meg a pályán. Ez lineárisan, az idő elteltével a játék folyamatosan nehezedik, ezekben a hullámokban erősebb/több ellenség érkezik a játékos ellenében.
- **Fejleszthető vár:** a játékos bázisát arany ellenében tudja fejleszteni, hogy több élete legyen.

#### COULD HAVE:

- **Nehézségi fokok:** a játékos új játék indításakor választhat, hogy könnyű, közepes vagy nehéz szinten szeretné a játékot játszani. Az egyes szintek között az ellenségek haladási sebességében, felbukkanásuk gyakoriságában és mennyiségében van eltérés.
- **Katasztrófák:** bizonyos időközönként véletlenszerűen megsemmisülnek mind az ellenség, mind a játékos bábui.

#### WON'T HAVE:

- **Multiplayer:** a játék csak gép ellen, egyénileg játszható.
- **Online játszható játékokra jellemző tulajdonságok:** pl. nincs globális ranglista és arra sincs lehetőség, hogy anyagi juttatás ellenében a játékos könnyítésekben részesüljön.
- **Perzisztencia**

---

#### JÁTÉKOT MEGELŐZŐEN:

- új játék indítása vagy kilépés az alkalmazásból

---

#### JÁTÉK KÖZBEN:

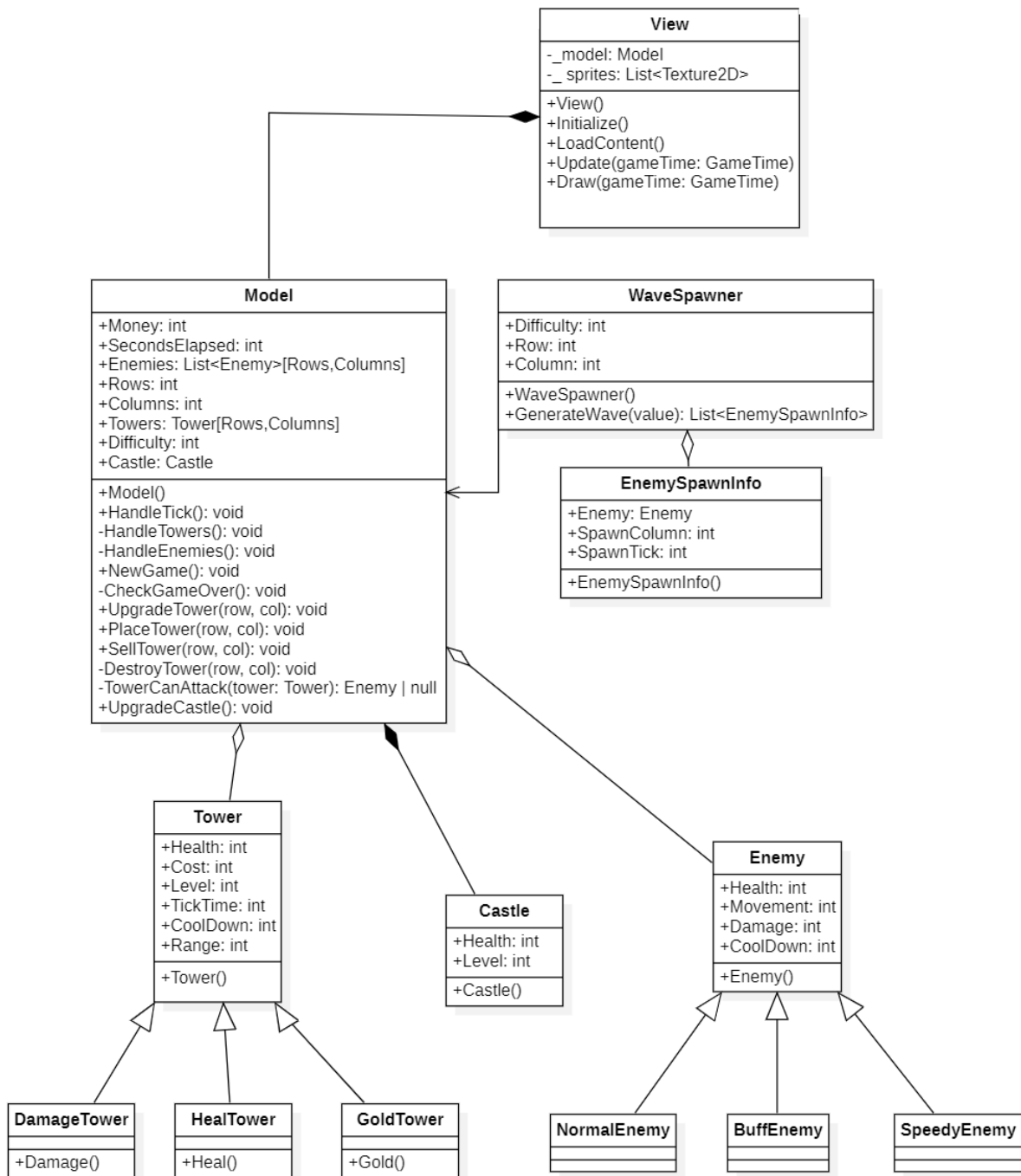
- aktuális játékállás és idő megjelenítése
- lépés a játékban: toronyépítés, -fejlesztés, -rombolás
- játék végének felismerése, játékos pontjainak megjelenítése
- játék végén új játék indítása vagy kilépés az alkalmazásból

## RENDSZER ARCHITEKTÚRA

A játék grafikus része és a játék logikája is komoly tervezést és háttérmunkákat igényel. Ennek megfelelően a View-Model architektúrát választjuk a megvalósításhoz, hogy megfelelően elkülönüljön a nézet a logikától, így moduláris, egyszerű és könnyen tesztelhető a kód.

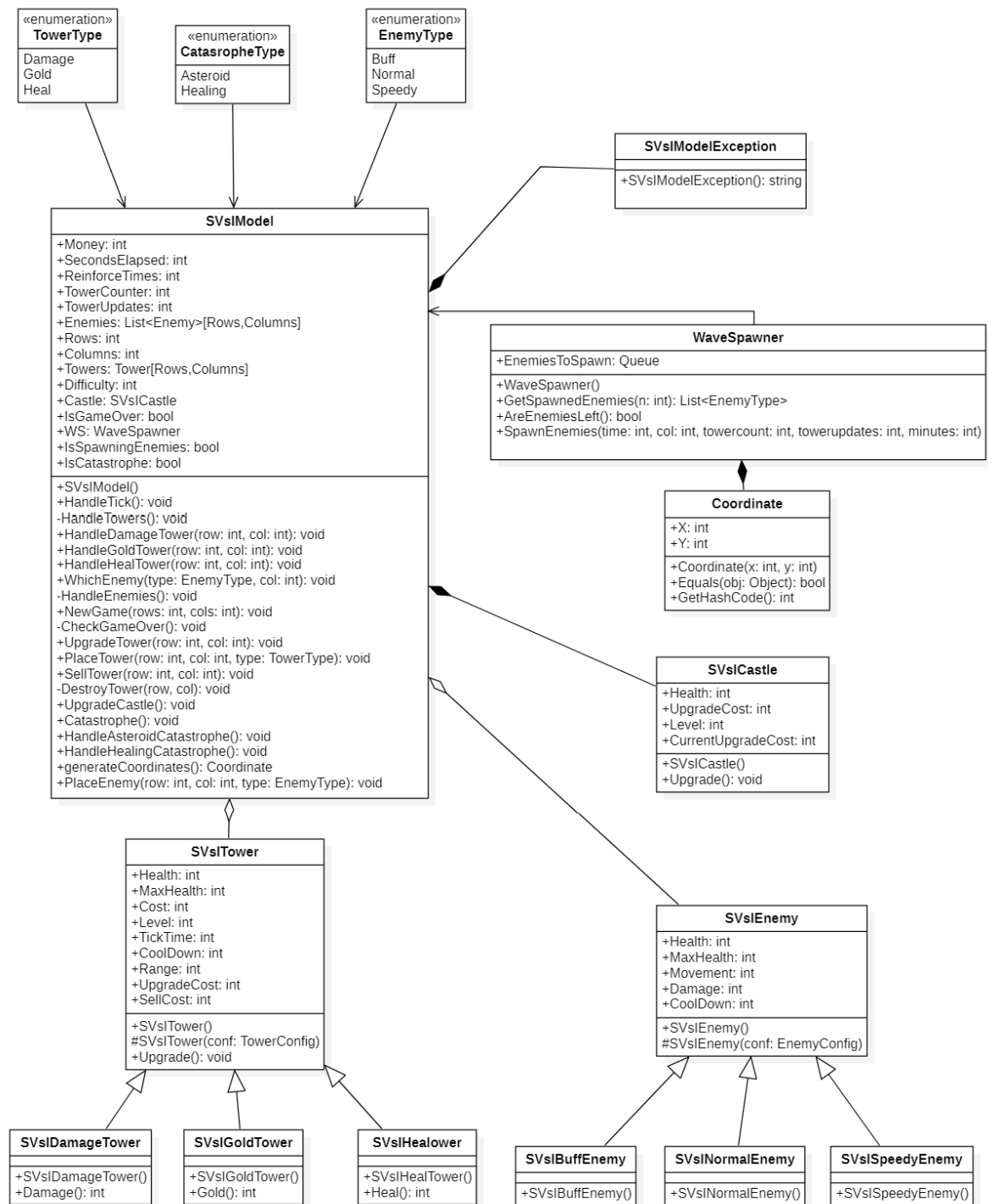
## RENDSZERTERVEZÉS

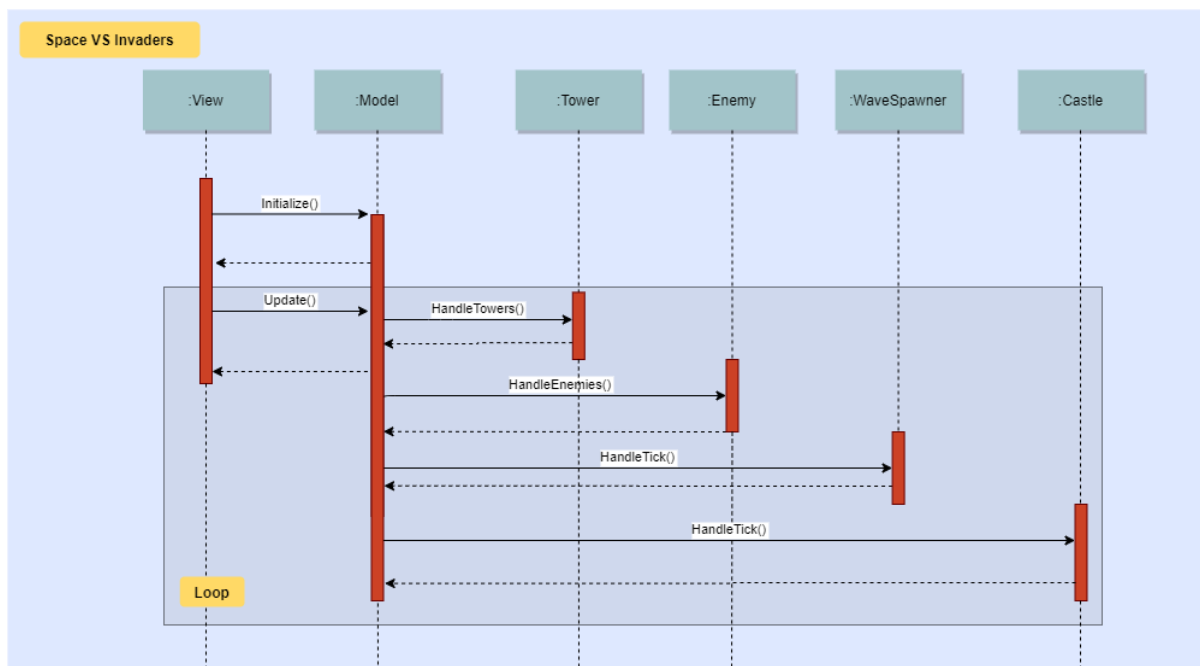
A statikus tervezést UML-diagrammal valósítottuk meg.



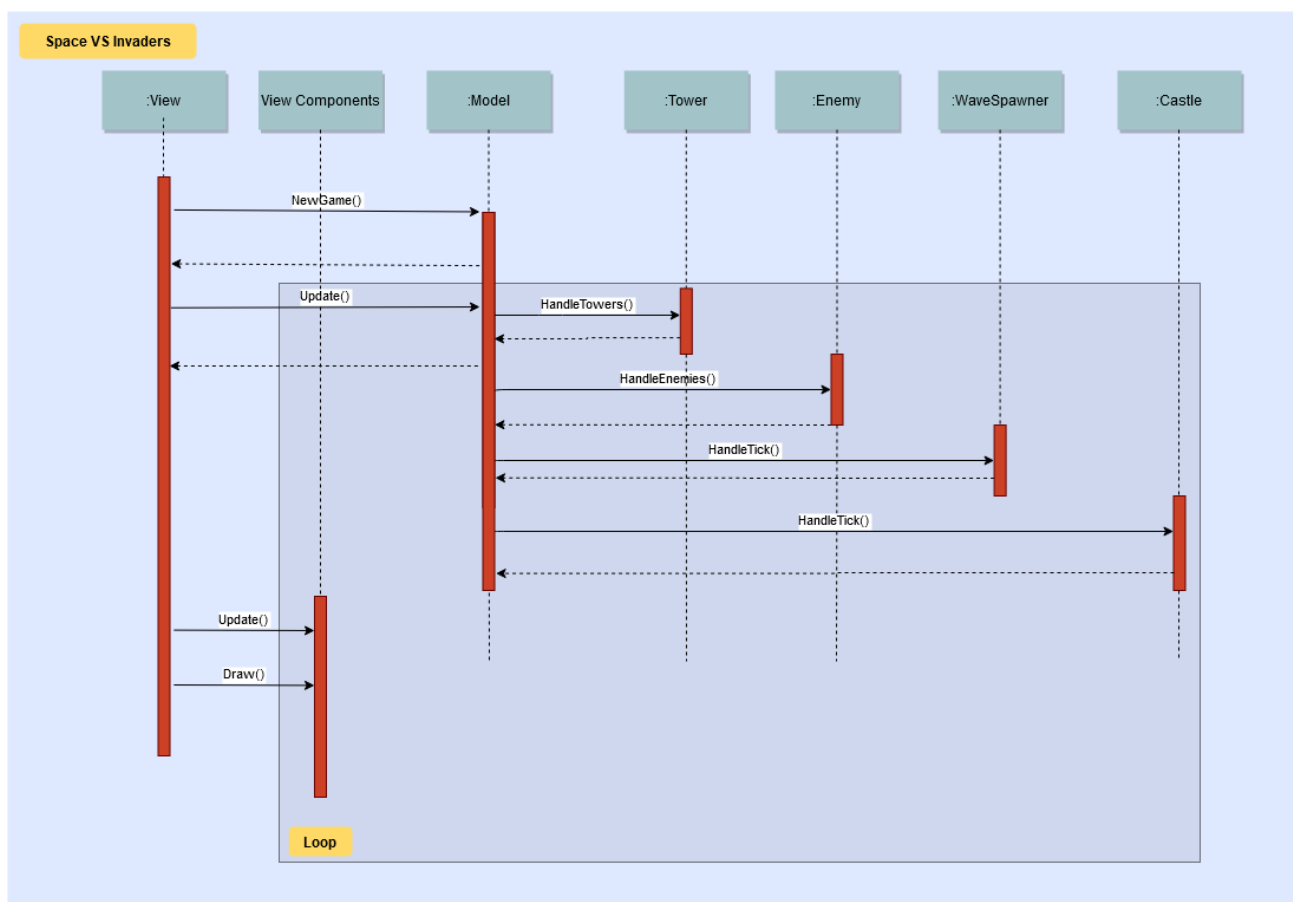
A dinamikus tervezést, amelyben az objektumok kommunikációját ábrázoljuk, szekvenciadiagrammal valósítottuk meg.

## Végleges UML:





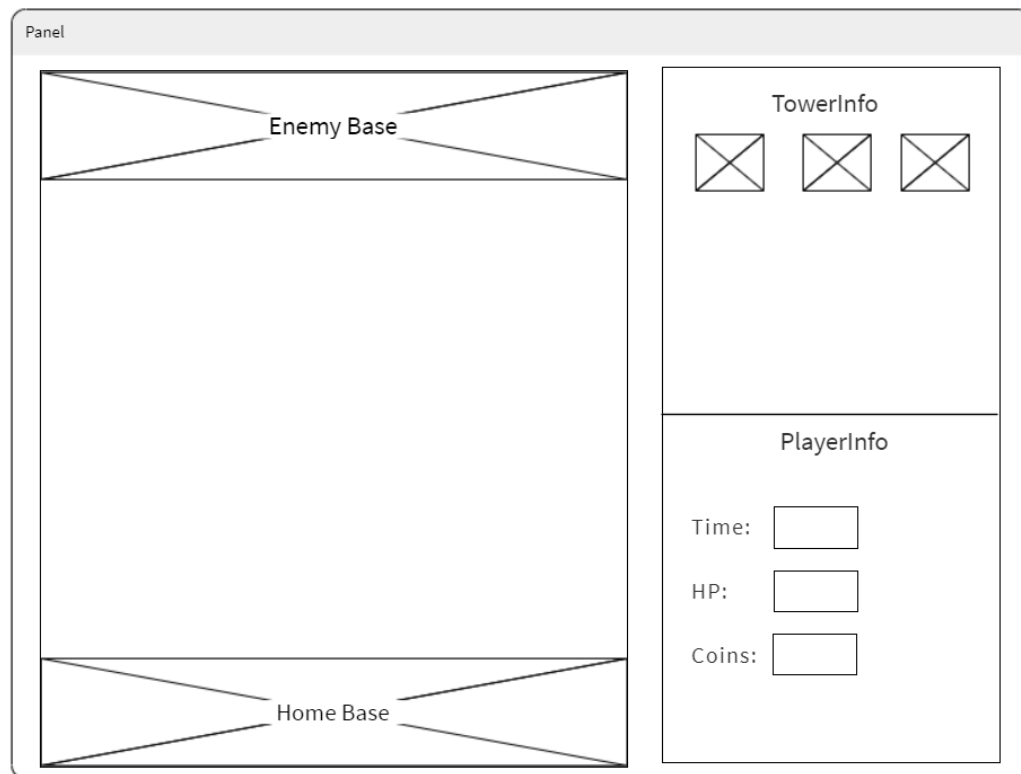
Végleges szekvencia diagramm:



RENDSZERMODELLEK:



A grafikus felhasználói felület (user interface) a következőképpen fog kinézni:



#### RENDSZEREVOLÚCIÓ:

Nem várhatók fejlesztések.

#### TERMÉKKÖVETELMÉNYEK:

- hatékonyság: optimalizált
- megbízhatóság: nem fordul elő hibajelenség
- biztonság: nem releváns
- hordozhatóság: telepítést nem igényel
- felhasználhatóság: a program bővíthető, minden környezetben
- környezeti: nem működik együtt semmilyen külső szoftverrel, szolgáltatással
- működési: a hardverkövetelményeknek megfelelő gépeken hibamentesen futtatható
- fejlesztési: C# nyelv, VSCode környezet, objektumorientált paradigma

#### BECSÜLT HARDVER

Felhasználói oldalról közepes hardverspecifikációjú számítógép javasolt a problémamentes játékelmény elérése érdekében.

#### KÖVETELMÉNYSPECIFIKÁCIÓ

Lásd lent, felhasználói történetek

## HASZNÁLATI ESETEK:

Aktor: játékos (felhasználó)

Funkciók:

- új játék indítása
- lépés a játékban (toronyépítés, -fejlesztés, -rombolás)
- játék vége
- kilépés

## KÖVETELMÉNY VALIDÁCIÓ

Ez az alpont a következő ellenőrző pont követelményében szerepel, akkor lesz részletezve.

## FELHASZNÁLÓI TÖRTÉNETEK

AS A		Játékos
I WANT TO		új játékot indítani
1	GIVEN	főmenüben vagyunk
	WHEN	új játék gomba rákattintunk
	THEN	elindul egy új játék
2	GIVEN	egy játékban vagyunk
	WHEN	menüt előhozzuk, új játék indítása
	THEN	új játék indul

AS A		Játékos
I WANT TO		kilépni a játékból
1	GIVEN	főmenüben vagyunk
	WHEN	kilépés gomba rákattintunk
	THEN	bezárul a játék

<b>2</b>	<i>GIVEN</i>	egy játékban vagyunk
	<i>WHEN</i>	menüt előhozzuk, kilépés a játékból gombra kattintunk és ezt az előugró ablakban megerősítjük
	<i>THEN</i>	bezárul a játék
<b>AS A</b>		Játékos
<b>I WANT TO</b>		megállítani a játékot
<b>1</b>	<i>GIVEN</i>	egy játékban vagyunk
	<i>WHEN</i>	előhozzuk a menüt
	<i>THEN</i>	szünetel a játék

<b>AS A</b>		<b>Játékos</b>
<b>I WANT TO</b>		lerakni egy tornyot
<b>1</b>	<i>GIVEN</i>	van rá pénzünk
	<i>WHEN</i>	rákattintottunk egy torony ikonjára, rákattintunk egy üres mezőre
	<i>THEN</i>	lerakódik egy torony, kifizetjük az árát
<b>2</b>	<i>GIVEN</i>	nincs elég pénz a toronyra
	<i>WHEN</i>	rákattintunk a torony ikonjára
	<i>THEN</i>	nem enged rákattintani, szürke a gomb
<b>AS A</b>		<b>Játékos</b>
<b>I WANT TO</b>		fejleszteni egy tornyot
<b>1</b>	<i>GIVEN</i>	van rá elég pénzünk
	<i>WHEN</i>	meglévő toronyra kattintunk
	<i>THEN</i>	a torony a következő szintre fejlődik, kifizetjük az árát
<b>2</b>	<i>GIVEN</i>	nincs elég pénz a fejlesztésre
	<i>WHEN</i>	meglévő toronyra kattintunk
	<i>THEN</i>	nem enged rákattintani, szürke a gomb
<b>AS A</b>		<b>Játékos</b>
<b>I WANT TO</b>		fejleszteni a bázist/várat
<b>1</b>	<i>GIVEN</i>	van rá elég pénzünk
	<i>WHEN</i>	a bázisra kattintunk
	<i>THEN</i>	a bázis a következő szintre fejlődik, kifizetjük az árát
<b>2</b>	<i>GIVEN</i>	nincs elég pénz a fejlesztésre
	<i>WHEN</i>	a bázisra kattintunk
	<i>THEN</i>	nem enged rákattintani, szürke a gomb
<b>AS A</b>		<b>Játékos</b>
<b>I WANT TO</b>		lerombolni egy tornyot
<b>1</b>	<i>GIVEN</i>	van felépített tornyunk
	<i>WHEN</i>	meglévő toronyra kattintunk
	<i>THEN</i>	a torony lerombolódik és a mező üres marad, megkapjuk a kifizetett ára egy részét