

# Less Fun Version of Warcraft 3

## Bevezető

A Warcraft 3 egy olyan stratégiai játék, amelyben az ellentétes csapatok, miután létrehozták saját bázisukat, csatával döntenek el, hogy ki legyen a végső nyertes. A feladat egy hasonló játék szimulálása, leegyszerűsítve. A feladatban egymással háborúzni nem kell, csupán létrehozni saját bázisunkat.

A **Building**, **Resources**, **Simulation**, **Unit**, **UnitType** osztályokat ne módosítsuk a feladatok teljesítése során.

## Feladat

A **Simulation** osztály a program belépési pontja, itt hozunk létre egy **Base** objektumot. A **Base** osztály konstruktorának egy paramétere van; a bázis neve.

A bázis feladata, hogy létrehozzon bizonyos épületeket, illetve munkásokat. Mind a munkásoknak, mind az épületeknek vannak közös tulajdonságai; mennyi aranyba kerül létrehozni (**goldCost**), mennyi fába kerül létrehozni (**woodCost**), mennyi ételbe kerül fenntartani (**foodCost**, ez az épületeknél 0), melyik bázishoz tartozik (**owner**), mi a típusa (**unitType**), és mennyi időbe telik elkészíteni (**buildTime**). Ezen információk tárolására hoztuk létre a **Unit** absztrakt osztályt, ami számon tartja melyik bázishoz tartozik az adott egység (**owner**), illetve hogy milyen típusú az adott egység (**unitType**). Az egységek típusát egy Enummal határozzuk meg, amelyben tároljuk a **goldCost**, **woodCost**, **foodCost**, **buildTime** attribútumokat is, ennek a neve **UnitType**.

A **UnitType** lehetséges értékei:

Name	Gold Cost	Wood Cost	Food Cost	Build Time (ms)
FARM	80	20	0	2000
LUMBERMILL	120	0	0	4000

BLACKSMITH	140	60	0	5000
PEASANT	75	0	1	1000

A **Resources** osztály segít számon tartani, hogy egy bázis milyen nyersanyagokkal rendelkezik (**gold**, **wood**), illetve mennyi embert tud ellátni (**capacity**, **capacityLimit**). A **capacity** mondja meg, hogy a jelenleg létrehozott egységek ellátására mennyi élelmiszer szükséges, a **capacityLimit** pedig azt, hogy maximum mennyit tudunk ellátni. **A capacity nem haladhatja meg a capacityLimit-et! A capacityLimit kezdő értéke legyen 5.**

A **Peasant** osztály a dolgozókat fogja szimbolizálni. Ez a **Unit** osztályból származik le, és a következő funkciókkal rendelkezik, melyeket implementálni kell a feladat teljesítéséhez:

- **startMining:** Elkezd aranyat termelni. Minden 100ms elteltével termel 10 aranyat.
- **startCuttingWood:** Elkezd fát vágni. Minden 100ms elteltével termel 10 fát. **A dolgozó nem tud egyszerre fát is vágni is aranyat is termelni.**
- **tryBuilding:** Megpróbál felépíteni egy épületet, amennyiben van elegendő erőforrása annak felépítésére, és éppen nem aranyat vagy fát termel. Ez visszafele is igaz – amíg építkezik, addig nem tud aranyat vagy fát termelni.

A **Building** osztály szimbolizálja az épületeket. A *Lumbermill* és *Blacksmith* nem befolyásolja a játék menetét. Minden *Farm* 10-zel növeli az élelmiszer ellátási kapacitást a **Resources** osztályban.

A **Base** osztály konstruktorában hozzunk létre 5 **Peasant**-et, amiből 3 elkezd aranyat termelni, egy fát vágni, egy pedig nem csinál semmit kezdetben. Az osztály rendelkezik egy **startPreparation** módszerrel. Ennek módszernek a feladata, hogy elérjük az alábbi állapotot

- Legyen 10 **Peasant**, amiből 5-en aranyat termelnek, 2-en fát vágnak. Figyeljünk, hogy ne legyen több **Peasant**, mint amennyi ellátására elegendő élelmiszerünk van. Egyszerre csak egy **Peasant**-ot lehessen képezni.
- Rendelkezzünk 3 *Farm*-mal.
- Rendelkezzünk egy *Blacksmith*-tel.
- Rendelkezzünk egy *Lumbermill*-lel.

Miután ezt elértük, a dolgozók hagyják abba a nyersanyagtermelést. Amint van nyersanyag és kapacitás megépíteni valamit, vagy létrehozni egy új **Peasant**-ot, egyből kezdjük is azt el.