

# AIC2018: Descripció del joc

## 1 Lore

Mentres unes formigues, escarbats, aranyes i abelles s'estan barallant de forma desapercibuda en una parcel·la insignificant de terreny, hi ha alhora dues grans nacions competint pel territori que l'engloba. Cadascuna d'aquestes nacions inverteix tot el que pot en aconseguir el millor exèrcit possible, però mantenir aquests tots soldats resulta molt car. Per això, totes dues nacions demanen el favor del Déu de la guerra, que promet ajudar l'exèrcit que li faci més ofrendes. Qui guanyarà en aquest escenari?

## 2 Resum del joc i objectius

El joc consisteix en destruir l'exèrcit enemic. Això es pot fer matant totes les seves tropes o bé aconseguint reunir 2000 punts de victòria per oferir al Déu de la guerra. Els punts de victòria es poden aconseguir recolectant-los pel mapa o comprant-los. El cost de comprar punts de victòria és  $50 + r/20$  unitats de fusta, on  $r$  és el número de ronda. L'equip que aconsegueixi aquest objectiu o l'equip que destrueixi totes les unitats de l'equip enemic guanya la partida. En cas que al cap de 2000 torns cap equip hagi aconseguit algun d'aquests objectius, l'equip guanyador es determina amb el següent criteri, per ordre:

1. Equip que té més punts de victòria .
2. Equip tal que la suma de recursos disponibles més el valor de cadascuna de les seves unitats és més gran.
3. Aleatòriament.

## 3 Economia

L'únic recurs del joc és la fusta, que es consumeix al crear noves unitats. Aquest recurs és compartit per tot l'equip i s'aconsegueix automàticament al començament de cada ronda a un ritme de 10 unitats de menjar per ronda. Els *treballadors* poden aconseguir fusta extra plantant i talant arbres, però cada unitat, a excepció de les casernes, consumeix dos unitats de fusta per torn i es gasten al començament del torn d'aquesta unitat.

## 4 Mapa

El mapa consisteix en una quadrícula rectangular de dimensions entre  $20 \times 20$  i  $50 \times 50$ . Totes les unitats ocupen exactament una casella i totes les caselles del mapa són accessibles per unitats excepte si hi ha un roure o aigua. Els roures, tot i ser inaccessibles, es poden destruir si són atacats o talats prou cops.

Cada unitat té un rang de visió finit, és a dir, només pot observar les caselles que són prou a prop de la seva posició.

Al començament de cada ronda (inclosa la inicial) poden aparèixer punts de victòria en caselles accessibles del mapa, aquests punts de victòria es poden recollectar amb qualsevol unitat adjacent<sup>1</sup>. Tanmateix, aquests punts de victòria son limitats i desapareixen al cap de 50 torns.

La probabilitat que apareixin punts de victòria en una casella (a la qual no ha aparegut cap punt de victòria en els últims 50 torns) ve donada per

$$p = \frac{1}{50\sqrt{T}}$$

on  $T$  és el total de caselles sense aigua al mapa.

Cada equip comença amb entre un i tres *treballadors* sobre el mapa, a més es garanteix la simetria dels mapes i dels punts de victòria que vagin apareixent per evitar afavorir un equip sobre l'altre. Aquesta simetria pot ser horitzontal, vertical o rotacional.

## 5 Arbres

Hi ha dos tipus d'arbres, els *roures* i els *arbres petits*. Les unitats poden passar per sobre dels arbres petits però els roures són impassables. Tots els arbres del mapa al començament de la partida són roures i la seva vida pot variar. Per altra banda els arbres petits han de ser plantats pels treballadors i tenen sempre 40 de vida màxima. Aquests arbres resistents comencen amb 1 punt de vida i recuperen 4 punts de vida durant primers 10 torns, on es diu que estan *en creixement*. Si un arbre és talat o atacat durant aquest període, l'arbre es destrueix. Passat el període de creixement els arbres recuperen dos punts de vida per ronda.

## 6 Distàncies

Degut a que es més fàcil treballar amb nombres enters, les distàncies sempre es donaran en unitats quadrades tant en aquest document com a la documentació del joc. És a dir, la distància entre un punt  $P_1 = (x_1, y_1)$  i un punt  $P_2 = (x_2, y_2)$  serà  $(x_1 - x_2)^2 + (y_1 - y_2)^2$ . Per exemple, si una unitat situada al  $(0,0)$  pot atacar fins a distància 12 vol dir que pot atacar una unitat al  $(2,1)$  ja que  $2^2 + 1^2 \leq 12$ , però no una al  $(2,3)$ .

També direm que dues caselles són *adjacents* si es troben a distància menor o igual a 2.

## 7 Unitats

Cada unitat rep un identificador (ID) únic escollit aleatòriament entre 1 i 10000. Quan una unitat nova es crea, hi ha un període de 5 torns en el qual la unitat està en construcció no es pot controlar.

---

<sup>1</sup>La definició d'adjacent es troba a la següent secció.

Els torns restants de construcció d'una unitat donada es poden consultar mitjançant les funcions del controlador.

Les unitats de combat adquireixen experiència (EXP) a mesura que van lluitant. Quan una unitat mor, totes les unitats de l'altre equip situades a distància menor o igual a 48 al voltant de la unitat destruïda guanyen 5 unitats d'experiència. A més, si aquesta unitat ha sigut destruïda per un atac o habilitat d'una unitat de l'equip enemic, totes les unitats creades posteriorment que siguin del mateix tipus que l'unitat destructora comencen amb 1 punt d'experiència adicional. Aquesta experiència adicional es denota com *experiència passiva*, és acumulativa i es pot consultar mitjançant les funcions del controlador.

Totes les unitats comencen a nivell 0, quan una unitat acumula prou experiència, aquesta puja de nivell i rep bonificacions (això pot passar immediatament al crear-se si l'experiència passiva d'aquest tipus d'unitat és prou gran). Les bonificacions són del tipus següent:

- Nivell 1: Es desbloqueja l'habilitat passiva. Aquesta habilitat està sempre present.
- Nivell 2: Es desbloqueja l'habilitat activa. Aquesta habilitat es pot utilitzar mitjançant les funcions del controlador, però té un temps de cooldown fins que es pugui tornar a fer servir.
- Nivell 3: La vida i l'atac de la unitat són augmentats.

Els treballadors no guanyen experiència, però disposen permanentment d'una habilitat activa i una passiva.

Hi ha sis tipus d'unitats: treballadors, casernes, soldats, arquers, cavallers i ballestes. Els treballadors tenen un poder de combat molt limitat, però són les úniques unitats del joc que poden plantar i talar arbres. Els soldats, arquers, cavallers i ballestes són *unitats de combat* amb diferents característiques, les quals juntament amb les dels treballadors es detallen al proper requadre. Totes les distàncies es donen al quadrat i els valors al nivell 3 surten entre parèntesi.

	Treballador	Soldat	Arquer	Cavaller	Ballesta
Cost (fusta)	100	100	140	160	200
Vida Màxima	40	60 (100)	20 (30)	120 (170)	20 (30)
Atac	1	6 (8)	10 (14)	6 (8)	32 (48)
Rang d'atac	2	8	20	4	35
Mínim rang d'atac	-	-	9	-	20
Rang de visió	16	24	20	48	10
Cooldown de moviment	2	2	3	1	4
Cooldown d'atac	1	1	2	2	4
Habilitats passives	Chop	Close Combat	Binoculars	Shield	Heavy Machinery
Habilitats actives	Plant	Counter	Poison	Dash	Root
Cooldown activa	1	8	15	10	7
Rang activa	2	8	20	8	35
Nivell 1 (EXP)	-	25	15	20	0
Nivell 2 (EXP)	-	75	150	50	90
Nivell 3 (EXP)	-	175	250	150	225

Els treballadors poden crear altres treballadors, però no poden crear altres unitats de combat di-

rectament. Per això han de crear una **caserna** amb 500 de fusta, que és una estructura immòbil amb 300 de vida màxima i 48 de rang de visió que no pot atacar ni té habilitats. Les casernes, a diferència de les altres unitats, triguen 25 torns en construir-se. Totes les habilitats passives i actives de les unitats seran detallades a la secció 9.

## 8 Moviment, atac i cooldowns

Cada unitat té un cooldown de moviment, un cooldown d'atac i un cooldown d'habilitat activable assignats que augmenten o decreixen de forma independent l'un de l'altre. La unitat només podrà realitzar l'acció desitjada en cas que el cooldown assignat a aquella acció sigui inferior a 1. En particular, si més d'un cooldown és inferior a 1, la unitat pot fer més d'una acció al mateix torn (com per exemple, atacar i moure's).

Cada cop que la unitat es mou, ataca o fa servir la seva habilitat, se li suma certa quantitat al cooldown de l'acció efectuada. La quantitat que se suma està indicada a la taula de la secció 5. Tanmateix, si la unitat es vol moure en una direcció diagonal, el cooldown de moviment sumat es multiplica per 1.4142, que és aproximadament  $\sqrt{2}$ . Al principi de cada torn tots els cooldowns baixen en una unitat.

Les unitats no poden atacar a través de roures, és a dir, una unitat no pot atacar ni utilitzar una habilitat activable sobre una casella si el segment que uneix el centre d'una unitat al centre de la casella objectiu passa per l'interior d'una casella amb un roure (es permet, però, que el segment només passi per la cantonada d'una casella amb un roure). A més, les unitats poden atacar roures no obstruïts per altres roures. Si una unitat ataca una posició (o un objecte en aquesta posició) on hi ha més de dos objectes (per exemple, una unitat i un arbre petit), tots els objectes de la casella reben el mal.

## 9 Mecàniques especials

### Treballadors

- Poden construir casernes per 500 de fusta o altres treballadors.
- **Chop:** Al atacar un arbre, si aquest arbre no era petit li redueix la vida en 24 punts i l'equip del treballador guanya 6 unitats de fusta, d'altra manera només li redueix la vida en 12 punts i l'equip del treballador guanya 12 unitats de fusta.
- **Plant:** Poden plantar un *arbre petit* per 180 de fusta en una casella adjacent.

### Casernes

- Poden crear soldats, arquers, cavallers i ballestes. Mentres una caserna està creant una unitat no pot fer-ne més.
- No poden recollir punts de victòria

## Soldats

- **Close Combat:** Els seus atacs peguen el doble si la unitat que rep l'atac està a una casella adjacent.
- **Counter:** A la unitat objectiu se li suma el doble de cooldown el proper cop que ataqüi.

## Arquers

- **Binoculars:** El rang de visió augmenta a 30.
- **Poison:** La unitat perd 1 punt de vida durant 10 torns.

## Cavallers

- **Shields:** Els atacs rebuts a distància estrictament més gran que 5 es redueixen a la meitat.
- **Dash:** La unitat es mou instantàniament a la casella objectiu.

## Ballesta

- **Heavy Machinery:** El cooldown de moviment i d'atac és compartit. És a dir, atacar suma cooldown de moviment i viceversa.
- **Root:** A la unitat objectiu se li suma el doble de cooldown de moviment el proper cop que es mogui.

# 10 Comunicació i visió

Cada unitat pot veure només una secció del mapa, que es correspon a les caselles que estiguin dins el seu radi de visió (en unitats quadrades). Si una casella és dins el seu radi de visió, aleshores la unitat pot demanar informació sobre aquesta casella mitjançant el controlador (més informació a la documentació i a la secció 12).

Les unitats s'executen independentment i no comparteixen memòria. Per exemple, els objectes visibles per una unitat no són visibles per la resta d'unitats. Tot i així, existeix una única manera de comunicar-se entre unitats: per a cada equip hi ha un array compartit d'enters de mida 100000 on tots els seus components estan inicialitzats a zero. Al seu torn, cada unitat pot llegir una o diverses posicions d'aquest array i/o sobre escriure una o diverses posicions mitjançant les funcions *read()* i *write()* del controlador.

# 11 Energia

L'energia és una mesura aproximada de la quantitat d'instruccions que executa una unitat al llarg d'un torn. Més concretament, cada instrucció de bytecode que executa una unitat costa una unitat

d'energia, a excepció de les instruccions de les classes pròpies del joc, dels quals el cost és constant i està disponible a la documentació. Per als usuaris no familiaritzats amb Java, no és necessari saber exactament el que són les instruccions de bytecode, tan sols cal tenir en compte que l'energia consumida creix a mesura que la unitat executa més instruccions, i que es pot comprovar experimentalment quina quantitat queda i quina quantitat es porta gastada amb les funcions del controlador.

Quan una unitat sobrepassa el límit d'energia permesa (actualment posat en 15000 unitats d'energia), aquesta unitat es pausa i continua el seu torn a la següent ronda. És recomanable procurar mai passar-se de l'energia máxima permesa, ja que perdre un torn pot ser crucial en moltes situacions.

## 12 Instruccions per l'usuari

Els jugadors han d'omplir la funció *run()* de la classe *UnitPlayer*. D'entrada es passa un controlador (*UnitController*) per la unitat, que permet a l'usuari donar ordres a la unitat i relacionar-se amb l'entorn, per exemple, la classe *UnitController* té funcions per detectar el que hi ha a certa casella, per donar ordres d'atacar/moure's/utilitzar habilitats, etc., o per llegir/escriure a l'array compartit. Per més detalls, a la documentació hi ha disponible tota la informació necessària sobre les classes pròpies del joc.

La funció *run()* funciona de la següent manera: quan es genera una unitat nova i passen els torns de construcció s'executa la funció *run()* d'aquesta unitat. La funció *run()* s'executa fins cridar la funció *yield()* del controlador o fins que se supera el límit d'energia permesa (més informació a la secció 11). Per aquest motiu, un cop una unitat acaba de fer totes les tasques desitjades (atacar, moure's, comunicar-se, etc.) s'ha de cridar la funció *yield()*. Així s'indica que es vol acabar el torn d'aquesta unitat. Si no s'indica, la funció *run()* continuarà executant-se fins superar l'energia permesa i acabarà el torn automàticament aleshores. S'ha de tenir en compte que no acabar el torn cridant *yield()* pot portar a comportaments de la unitat no desitjats en el futur, ja que és difícil predir des d'on tornarà a executar-se al següent torn.

Si en algun moment la funció *run()* retorna (acaba), la unitat mor. Per aquest motiu és recomanable assegurar que la funció *run()* mai retorni, per exemple amb un bucle *while(true)*. En general, és recomanable utilitzar l'esquema disponible als exemples *nullplayer* i *demoplayer*.

## 13 Informació sobre la implementació

Aquesta secció es pot ignorar si l'usuari no està prou familiaritzat amb Java.

Cada unitat s'executa en un thread independent que es pausa al finalitzar el seu torn. Aquest thread es reactiva a cada ronda de la partida mantenint l'ordre d'execució relatiu entre unitats. Per seguretat, es prohibeix l'accés a totes les classes de Java fora de *lang*, *math*, i *util*, i fins i tot a algunes sub-classes i funcions d'aquestes classes. De totes maneres, aquestes classes són totalment prescindibles pel joc actual.

També, per motius de seguretat, es prohibeix l'ús de variables estàtiques<sup>2</sup>. Més endavant conside-

---

<sup>2</sup>Som conscients que hi ha maneres sofisticades de compartir memòria entre threads sense fer servir variables estàtiques. Els codis pujats seran revisats manualment per evitar aquests casos, i en cas que un equip intenti violar aquesta restricció, aquest equip serà desqualificat.

rarem executar cada unitat en un procés independent amb la seva propia JVM.