

Programació VideoJoc sobre l'Origen de l'Agricultura a la Prehistòria

Alexis Domínguez Cachinero

Resum

El projecte proposat consta d'una aplicació que permeti, mitjançant la interacció amb l'usuari, simular els inicis i posterior evolució de l'agricultura a la prehistòria. Ho farem mitjançant una població d'individus que es desplaça per un mapa representatiu de la zona que connecta Orient amb Europa i evoluciona segons unes condicions, favorables o no, que l'usuari pot modificar al seu gust per estudiar i aprendre dels diferents casos proposats.

Aquesta aplicació es programa seguint els estudis arqueològics que demostren com la població es multiplica i avança cap a zones favorables al cultiu, fent que les decisions que pren la població del nostre programa, simuli de la forma més precisa possible el comportament i evolució humana en aquest àmbit.

Paraules Clau: Videojoc, programació, agricultura, prehistòria, enginyeria, simulador, educació, educatiu, història, evolució.

Abstract

The project proposed features an application that allow, through of the interaction with the user, simulate the starts and evolution of the agriculture at the prehistory. We will do it by means of a population of individuals that displaces for a map representing the zone that connects Orient with Europe and evolves as some conditions, favourable or not, that the user can modify by his decision to study and learn of the different cases proposed.

The application is programmed following the archaeologists studies that show how the population multiplies and advances to favourable zones to the crop, doing that the decisions taken by the population of our program, simulate of the most precise shape possible the behaviour and human evolution in this field.

Index Terms: Videogame, programming, agriculture, prehistory, engineering, simulator, education, educational, history, evolution.



1 INTRODUCCIÓ

Aquest treball es realitza sota la petició i supervisió del departament de Prehistòria de la facultat de filosofia i lletres de la UAB, qui fa de client sol·licitant de l'aplicació per l'estudi de l'evolució de l'agricultura a la prehistòria. L'objectiu del programa és aconseguir simular aquesta part de l'evolució humana d'una forma interactiva i visual per a l'usuari, fent possible modificar paràmetres i variables que ens demostraran que va passar i/o que hauria passat si les condicions haguessin estat diferents, manipulant-lo des d'una interfície d'usuari.

A continuació desenvoluparem els objectius a assolir, la metodologia a utilitzar per portar-los a terme i la planificació i desenvolupament del projecte. Al software li direm *Neolithic expansion and diffusion* (NED).

- E-mail de contacte: alexis.dc.92@gmail.com
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Yolanda Benítez (Enginyeria)
Juan Antonio Barceló (Dept. Prehistory.)
- Curs 2018/19

2 ESTAT DE L'ART

Actualment existeixen diferents aplicacions i estudis d'arqueologia relacionats amb aquesta matèria tant dedicades a l'entreteniment com a la investigació d'aquest entorn. Per una banda tenim les versions "gamificades" com per exemple *Evolving planet*, un joc mòbil de simulació de l'evolució desenvolupat amb la col·laboració del Centre de Supercomputació de Barcelona; per l'altra banda tenim estudis i projectes de simulació computacional com ara el treball de prehistòria publicat per Salvador Pardo-Gordó "Sistemas complejos adaptatius i simulació computacional en Arqueologia", aquests més enfocats a la investigació arqueològica.

Aquest projecte pretén dirigir-se més aviat cap a una versió jugable, tot i que es pot aprendre i seguir desenvolupant per tal d'enfocar el funcionament cap a l'àmbit de la investigació.

3 REQUISITS I OBJECTIUS A ASSOLIR

Mitjançant les reunions amb el client hem definit una sèrie de requisits mínims que tindrà el nostre projecte:

- **Software funcional:** Al final del projecte, hem d'obtenir un Software que funcioni correctament per tal d'estudiar el tema tractat.
- **Evolucionable:** El programari ha de permetre que altres persones puguin continuar treballant en ell, afegint noves funcionalitats i millorant la simulació.
- **Mapa realista:** El mapa generat ha de ser fidel al real en la zona on van iniciar l'agricultura (Orient Proper) per tal que la simulació sigui molt aproximada a la realitat, dissenyant un comportament realista segons la topografia.
- **Comportament realista:** Un cop finalitzat, voldrem que la simulació tingui un comportament raonable i que ens permeti aprendre.
- **Jaciments agricultura:** Hem de dissenyar els jaciments principals, els d'agricultura, de forma que puguem definir i estudiar com es reproduïxen i desplacen pel mapa.
- **Jaciments caça:** Els altres jaciments a dissenyar són els de caça/recol·lecció, aquests són els jaciments més primitius que interactuaran amb els nous de diferents formes escollides per l'usuari, per tal d'estudiar la influència del comportament.
- **Comportament modificable per l'usuari:** L'usuari ha de disposar de diferents opcions per modificar els comportaments i característiques de la simulació, fent així l'estudi i l'observació de forma interactiva.
- **Extracció de dades per l'estudi:** Un cop estigui funcionant tot, és convenient generar informes que mostrin l'evolució de la simulació.

4 METODOLOGIA A UTILITZAR

La metodologia que utilitzarem per a desenvolupar aquest projecte serà SCRUM, una metodologia de tipus agile, en la qual distribuïm les tasques principals en subtasques d'una duració determinada, permetent fer les modificacions oportunes a mesura que avanci el projecte, per organitzar i definir aquesta metodologia utilitzarem el software Trello. En aquest cas, no posarem màxims ni mínims de temps, ja que en haver-hi tasques molt diferents (Desenvolupament, recerca, planificació...) no podem considerar-les totes en les mateixes condicions definides, si no que ho anirem desenvolupant cíclicament en espiral. A continuació, en l'apartat de planificació, desenvoluparem els diferents grups de tasques, així com les subtasques definides a cadascun d'aquests grups.

5 PLANIFICACIÓ

En aquest apartat s'explicarà la planificació de les tasques definides en relació al temps disponible, resumit i expressat gràficament al Diagrama de Gantt annex.

5.1 Definició

El primer dels quatre blocs principals anirà des de l'inici el 20-sept fins al 20-oct, tindrà una durada d'un mes i englobarà tant les entrevistes amb la tutora i el client com la part de recerca d'informació sobre el projecte i la planificació i requisits d'aquest, decidint el llenguatge a utilitzar (Netlogo) i estudiant el funcionament i possibilitats d'aquest.

5.2 Disseny Mapa

El segon bloc principal tindrà una durada de quasi 2 mesos, compresos entre el 22-oct i el 5-dec. Aquest bloc el dedicarem a desenvolupar el mapa, tant la recerca i disseny d'imatges que s'adaptin als nostres requisits com l'especificació i programació dels *patches* que compondran l'escenari de la simulació. També es començaran a dissenyar els agents per fer proves.

5.3 Disseny Agents

El tercer gran bloc, consta de 2 setmanes i mitja compresos entre el 6 i el 21 de desembre. En aquesta part haurem de continuar i finalitzar el disseny dels agents, tenint en compte totes les variables necessàries que aquests tindran i que després utilitzarem en els mecanismes o funcions.

5.4 Disseny Mecanismes

A l'últim bloc del projecte dedicarem un període d'un mes, del 24-dec al 24-gen. En aquesta última part, haurem de desenvolupar les funcions que s'encarreguen de la correcta simulació de l'entorn. Encara que ja haurem dissenyat algunes de bàsiques per tal de fer proves. Aquest mes el dedicarem a fer que els agents es reproduïxin adequadament i segueixin un algorisme realista en la simulació del comportament.

6 ANÀLISI

A continuació desenvoluparem les diferents tasques en subtasques més petites en les quals establim un període de temps, reflectit al Diagrama de Gantt.

6.1 Disseny Mapa

Aquestes tasques estan destinades a la creació i funcionament dels anomenats *patches*, les cel·les que utilitza Netlogo per a simular un entorn.

6.1.1 Buscar Imatges

Per tal de poder treballar cada tipus de terreny de forma diferent, necessitem una imatge realista de la zona a estudiar. Per això haurem de trobar i editar, diferents imatges de satèl·lit, en les quals segons el color podem definir com es comportaran els agents sobre el terreny.

6.1.2 Especificació

En aquesta part definirem les funcions de les diferents variables que afecten als *patches*, en principi són quatre, la topografia, la productivitat, els recursos i l'accessibilitat des de *patches* adjacents. Així com la relació entre aquestes variables permetent a l'usuari fer combinacions diferents.

6.1.3 Programació Patches

Posteriorment programarem les variables identificades, una a una, tenint en compte que pot ser necessari modificar altres variables ja definides. A més, definirem els rangs de color del mapa amb els que treballarem, aquests aniran relacionats amb la topologia.

6.2 Disseny Agents

Executant-se sobre els patches, tindrem els agents, objectes que representen les poblacions o jaciments, aquest bloc es dedica a la seva definició i programació.

6.2.1 Tipus agricultura

L'agent principal i sobre el que gira aquest projecte és el jaciment d'agricultura, aquí programem les variables que faran possible la modificació del seu comportament segons la interacció que trobi.

6.2.2 Tipus Caça

L'altre agent que trobarem serà el jaciment de caça/recol·lecció, aquest haurà de tenir diferents comportaments quan es trobi amb una població d'agricultors.

6.2.3 Definició moviment/any

Aquí definirem, mitjançant diferents proves i càlculs, la relació que tindran el moviment a través dels patches per cada any transcurrit.

6.2.4 Definició ticks/any

Aquí definirem, mitjançant diferents proves i càlculs, la relació que tindran els ticks de temps que executa el programa amb els anys transcorreguts.

6.3 Disseny Mecanismes

Els Mecanismes seran els algorismes de comportament que treballarem en aquest projecte, en aquest bloc els definim i programem.

6.3.1 Reproducció de jaciments

Aquest mecanisme defineix la forma en que els jaciments s'expandiran pel mapa, segons la interacció amb els patches en que es trobin.

6.3.2 Supervivència

En aquesta part definirem com un jaciment pot continuar existint, segons unes dependències i esdeveniments que poden definir si el jaciment serà temporal o permanent.

6.3.3 Abandonament

L'abandonament seria el fracàs de supervivència, si es donen les condicions adequades, un jaciment antic es deixarà morir i la població continuarà avançant cap a una altra zona.

6.3.4 Transmissió cultural

Aquí es definiran les diferents actuacions que tindrà un jaciment de caça/recol·lecció quan es troba amb un jaciment d'agricultura. Ho podrà escollir l'usuari i principalment seran tres reaccions, fugir de la zona, lluitar-la o aprendre agricultura, convertint així, el jaciment a l'altre tipus.

7 DESENVOLUPAMENT

7.1 Definició

La primera part del projecte consta de les reunions amb tutora i client en les que hem definit les parts a tractar i els requisits de funcionament.

Un cop fetes les reunions es dedica un temps a investigar les eines a utilitzar, el client és el departament de Prehistòria de la facultat de filosofia i lletres de la UAB i ens indica que acostumen a treballar amb un llenguatge de programació anomenat NetLogo, per tant vaig investigar el seu funcionament.

NetLogo és un llenguatge interpretat basat en *agent-based modeling*, utilitzat principalment per a la investigació i simulació d'agents en arqueologia, biologia, etc.

Molt resumidament, es basa en la interacció entre patches (cel·les del espai) i turtles (Agents que es mouen i/o interactuen per sobre d'aquests patches). Aquestes variables són les que permeten treballar els diferents comportaments mitjançant funcions, produint una interfície molt visual per a l'usuari.

Després d'investigar l'abast d'aquest llenguatge procedim a la planificació de les diferents tasques en el temps disponible.

Un cop feta aquesta primera part, es fan noves reunions periòdicament per corroborar el procediment establert amb el Client i la Tutora, contemplant i introduint els canvis que proposa el client per tal de tenir un projecte funcional en la duració disponible, es decideix utilitzar com a agents els mateixos jaciments d'agricultura o caça, en comptes de les diferents poblacions com proposàvem al principi, ja que aquestes tenen moltes més variables socials a tenir en compte i hauríem de dedicar la major part del projecte a estudiar-les i dissenyar-les.

7.2 Disseny mapa

La primera part per treballar amb els diferents patches del mapa, és buscar unes imatges adequades, ja que treballarem amb els diferents rangs de color i per això necessitem una imatge via satèl·lit que sigui suficientment realista de la zona a estudiar. Agafem diferents imatges suficientment clares i amb bona resolució i, després de modificar-les adequadament amb photoshop, fem proves al carregar la imatge per tal que el mapa sigui el més realista possible.

7.2.1 Especificació

En el nostre mapa a treballar, el qual tindrà una resolució de 1000*700 patches, cada patch representa un àrea real de $5*5 = 25\text{km}^2$.

Les variables que hem decidit definir sobre cada patch són les següents:

- Productivitat: Serà un valor numèric que es mourà en un rang entre 0-9 i que posteriorment utilitzarem per les probabilitats de cultiu, estarà definit pel tipus de terreny.
- Productivitat actual: la productivitat anirà variant depenent de l'activitat sobre el patch, per tant aquesta variable ens indica el valor de productivitat sobre el màxim del patch
- Accessibilitat: També serà numèric i anirà des de "1 - accessible fàcilment" a "10+ -inaccessible", això en un futur podrà canviar segons les opcions de què disposem, per exemple, amb barques, les illes serien accessibles.

7.2.2 Programació Patches

Per a tenir uns patches funcionals del mapa, es dissenyen dues versions del mapa que l'usuari pot escollir a l'interfície, una simplificada amb només dues zones (colors) que representen la terra i l'aigua, i una altra més realista amb 7 zones diferents, aquestes zones representades per diferents colors tindran propietats particulars, definides per les següents variables:

- access-cost: Dificultat d'accés a un color, des d'un de diferent.
- productivity: Multiplicador màxim que indica la fertilitat de la zona.
- Productivity-now: Multiplicador actual que indica la fertilitat real de la zona, anirà variant.

Es podran afegir nous mapes amb diferents zones i/o ubicacions per l'usuari, sempre que compleixin l'estàndard de colors utilitzat al codi el qual contempla 10 rangs especificats a la següent taula. Si s'utilitzen altres colors, el programa els convertirà al color més pròxim.

Colors	Netlogo	RGB	A-C	Prod
1.Blau	- 95 -	[45 141 190]	50	0
2.Verd1	- 67 -	[128 227 137]	1	2
3.Verd2	- 56 -	[122 192 99]	3	1.5
4.Verd3	- 53 -	[54 109 36]	5	1
5.Groc	- 48 -	[247 247 173]	1	1.5
6.Taronja	- 28 -	[249 195 162]	3	0.8
7.Marró1	- 35 -	[157 110 72]	6	0.6
8.Marró2	- 33 -	[97 68 44]	8	0.5
9.Blanc	- 9.9 -	[255 255 255]	9	0.1
10.Lila	- 115 -	[124 80 164]	10	0

7.3 Disseny Agents

En aquest punt del desenvolupament, parlarem dels agents dissenyats amb els quals podrem utilitzar les funcions programades posteriorment.

7.3.1 Disseny Agents jaciment d'agricultors

Els agents dissenyats corresponen als jaciments d'agricultura que mitjançant les funcions que després veurem, ens permetran observar com es multipliquen i desplacen pel mapa, segons el nostre algoritme.

Les dades que pertanyen a cada jaciment serien les següents:

Localització definida al mapa per coordenades X,Y, group-age indicant l'edat des que es va crear el grup, settle-age indicant l'edat del jaciment en aquesta posició, num-adults per comptar el nombre d'individus reproductors, num-kids per indicar el nombre de nens menors (aquests són menys productius i no es poden reproduir), consumption indicant l'alimentació necessària pel grup i production indicant el que genera aquest grup.

Al començament només crearem un (o els que l'usuari vulgui) petit grup que anirà desplaçant-se o dividint-se, en unes coordenades escollides del mapa.

7.3.2 Definició moviment/any Definició ticks/any

Aquesta part ha estat definida recomanable i temporalment com a 10 patches/tick pel correcte desenvolupament de la resta de funcions, encara que es podrà modificar segons hi veiem la necessitat, també l'usuari pot definir-la a la simulació.

Cada tick del programa correspondrà a un any i el moviment dels jaciments per any el definirem segons el valor assignat a la variable global movility, que correspondrà al radi de distància a explorar cada any.

7.4 Disseny Mecanismes

Com a mecanisme principal hem dissenyat cicle-of-activity, que és una funció per decidir si el patch sobreviu (producció > consum) i quin tipus de reproducció faria el jaciment llavors.

Cridant després les funcions necessàries segons la decisió que es prengui, a continuació les explicarem totes.

7.4.1 Reproducció jaciments

La reproducció de jaciments en l'algorisme cicle-of-activity es fa de la següent forma, si el patch on es troba un grup, produeix més del que consum, neixen nous individus; en cas contrari es decideix si desplaçar-se (grup inferior a 4 adults) o dividir-se (grup superior a 4 adults).

En qualsevol d'aquestes dues decisions, tindrem en compte la variable movility per estudiar el radi de moviment disponible, escollint la zona més productiva per a començar un nou jaciment.

En cas que al nou patch tampoc puguin sobreviure, com a última opció es faria un infanticidi, per després tornar a calcular les opcions que tenen de sobreviure els adults, si tot i això fos negatiu, s'eliminarien tants adults com fes falta per sobreviure (poden ser tots, moririen d'inanició).

7.4.2 funcions dissenyades

Aquí explicarem simplificadament les funcions que fan possible el comportament descrit anteriorment.

- **SET-UP:** es carrega el mapa seleccionat i s'estableixen les variables introduïdes a la interfície d'usuari
- **POPULATE-WORLD:** un cop carregat el mapa, si fem click al botó populate, activem aquesta funció, que ens permet crear tants agents com clicks fem al mapa. Cada cop que es clica a una ubicació, es genera un nou agent, això es pot activar i desactivar encara que la simulació hagi començat
- **GO:** bucle central de l'aplicació, el qual genera un fitxer amb tots els agents introduïts per l'usuari (ho veurem a la secció "reports") amb la indicació de coordenades i any. Inicialitza i suma el comptador d'anys utilitzant el paràmetre global "any inicial" i finalment crida a l'algoritme de decisió, cycle-of-activity.
- **CICLE-OF-ACTIVITY:** Cada tick (Any) cada grup local produeix energia i per a això consumeix el que produeix. (Procediment SURVIVE (produeix - consumeix)). Si sobreviu (produeix el que necessita consumir), la població creix (crida a BORN). Si el que un grup produeix no és suficient per a mantenir a tot el grup aquella part del grup que supera la capacitat de producció es desplaça seguint els criteris següents:
 - Si la part que ha de desplaçar-se té més de 4 adults, aquests adults funden un nou agent en un altre patch (DIVIDE). El nou agent no té nens en el seu moment inicial, per la qual cosa tots els nens existents en el primer grup es queden.
 - Si la part que ha de desplaçar-se té menys de 4 adults, i no es poden fer 2 grups de 4, s'haurà de moure l'assentament complet (conducta més lògica, per a evitar la mort d'individus abans d'hora, en MOVE veurem si ocorre).

Després d'això crida a les funcions DIE-RATE, GROW, ACTUALIZE y PRODUCTIVITY-REGEN

- **PRODUCE:** calcula la Producció d'aliment
En forma funcional: $Y=(L \times \alpha)^\beta$ on:
 - Y (valor de producció durant 1 any) (unitats consum)
 - L = quantitat de treball (adults+(kids*0.5))
 - α = productivitat actual del patch
 - β =rendiment del treball (global parameter: input work_performance). Inicialment $\beta \geq 1$

En aquesta funció, també es redueix la productivitat del patch un 10% per cada any en què es treballa sobre aquest.

- **CONSUME:** Calcula el total d'aliment necessari per mantenir un jaciment: Consum = adults + nens.
- **DIE-RATE:** Calcula la probabilitat que un nen mori cada any, en cada assentament, segons la variable utilitzada per l'usuari. També calcula la mort per edat de l'assentament.
- **BORN:** Segons la probabilitat introduïda per l'usuari, calcula si neixen nens aquest any (1 per parella).
- **GROW:** Calcula si els nens passen a adults aquest any, hi ha un 30% que ocorre, si ocorre, la meitat dels nens passen a ser adults.
- **DIVIDE:** Calcula quanta gent sobra en l'assentament i crida MOVE per aquest sobrant, si és menor a 4 assumeix que sobren 4, si després del càlcul, queden menys de 3 persones al jaciment, crida a MOVE per a tot el grup, si es pot dividir:
 - Neix un nou grup i la seva EDAT S'INICIA A ZERO
 - S'inicia el seu comptador de nens (0) i reproductors (adults sobrants).
 - Es desplaça al patch de major productivitat dins de determinada àrea de veïnatge (MOVE)
- **MOVE:** crida a EXPLORE i rep el millor patch, al qual es desplaça.
 - Si No existeix cap patch, moren (STARVE)
 - Si es mouen i així i tot no poden sobreviure, llavors es redueix el nombre de nens del grup i es recalcula el consum (INFANTICIDE). Si malgrat la mortalitat de nens la producció < consum, llavors es moren els adults, si així i tot no sobreviuen, mor el jaciment.
- **EXPLORE:** Cerca i retorna el millor patch dins del àrea Movility:
 - En el rang movility/2 (50% de la distància màxima), investiga tots els patches
 - En el rang movility*0.8 (80% del max), investiga els patches de menor cost d'accés o aquells que només superen el Access-cost actual en 4.
 - En el rang movility (100%) investiga els patches de menor cost o els que superen el Access-cost en 1
 - Una vegada ha investigat tots, tria el millor patch segons la productivitat actual.

D'aquesta forma es contempla que no pugui explorar els patches de més difícil accés si aquests no estan prop de la ubicació actual de l'assentament.

- **INFANTICIDE:** Resta suficients nens perquè el procediment SURVIVE quedi en positiu, si així i tot no queda positiu, resta suficients adults per a aquest propòsit, en cas que l'assentament quedi amb menys de 3 adults, l'elimina (STARVATION).
- **ACTUALIZE:** Actualitza les variables consumption i production de cada jaciment i elimina els jaciments amb 1 sol adult.
- **PRODUCTIVITY-REGEN:**
Rendiment decreixent de la productivitat del patch. A l'inici del tick el valor de productivitat de partida és el mateix que al final del tick anterior.
Si no hi ha ocupació humana, a cada tick, el patch recupera 30% de la seva productivitat original, de manera que al cap de tres ticks (tres anys), el patch ha tornat pràcticament a la seva productivitat original.
Es contempla que la productivitat actual no pugui sobrepassar la màxima.
- **NATURAL-DISASTER:** Botó que s'utilitza en temps d'execució del programa per a simular una catàstrofe mundial (malaltia, plaga, fenomen meteorològic, etc) reduint 1/3 de la població total. Això ho fa mitjançant el ID de l'assentament, per tant només funciona una vegada, les següents eliminaran 1/3 dels nous assentaments creats des de l'última pulsació, però no dels totals.
- **REPORTS:** Fitxers amb resultats de la simulació:
 - Opció per a crear un Snapshot de l'execució en qualsevol moment (locations.txt). És un fitxer que conté: ID Agent, Coordenades (X, Y) num_adults num_nens, production.
 - Crea un Fitxer la primera vegada que es clica sobre GO!, suposant que s'ha poblat el mapa prèviament, es genera un arxiu amb els assentaments inicials seleccionats (init-agents.txt). És un fitxer que conté: ID, CoordX, CoordY, Year.

8 PROVES

Les primeres proves realitzades les trobem escollint un mapa base per tal de desenvolupar la simulació, després de provar diferents versions, em vaig decantar per fer-ne un des de zero en photoshop, degut a que les zones entre un color i un altre creaven patches de colors semblants però no contemplats en el codi, i encara que finalment es contemplen o converteixen tots els colors, podien crear zones discordants. Per a tenir uns patches funcionals del mapa, es dissenya una primera versió on el reduïm a 4 zones diferents per tal d'ajustar les variables dels patches i fer que tinguin un comportament raonable, posteriorment quedarà com a mapa de 7 zones, contemplat fins a 10 diferents a nivell de codi.

A diferència de la funció "produce", la majoria no venien donades per una fórmula o un algorisme conegut, això es tradueix en decidir com actuarà cada funció mitjançant prova i error, ajustant els percentatges i els valors de les variables utilitzades per tal que el comportament sigui el més natural possible.

A l'algorisme de decisió principal "cicle-of-activity" s'han provat diferents versions fins a arribar a l'actual, tenint en compte diferents mides de grups al separar-se, tenint en compte sacrificis parcials o totals dels grups, divisions amb o sense nens, etc. Tot això comporta modificacions a les funcions move i divide, principalment.

Dintre de la funció "die-rate" està implementada la mort accidental de nens als jaciments mitjançant una probabilitat escollida per l'usuari, junt aquesta mort, tenim la mort per inanició, contemplada en la separació d'un jaciment per falta de recursos, a més d'aquestes dues morts, s'han fet proves de mort per edat del grup, però finalment en aquesta versió es descarta, degut a que el jaciment no pot contemplar l'edat dels individus sinó només la del propi jaciment, fent que quan un jaciment porti X anys viu es mori, encara que es tracti d'un jaciment jove i productiu.

La primera versió de la funció "explore" era molt més simple, agafava la variable de moviment introduïda per l'usuari com a radi i explorava tot el que existia en aquest radi, això en la simulació s'interpreta com si fos igual de fàcil explorar 1km de muntanya i 1km d'esplanada. Per tant, mitjançant proves es va arribar a la versió final la qual dependent de la distància té en compte diferents dificultats d'accés.

En un principi, la introducció del jaciment inicial es feia mitjançant coordenades que podia introduir l'usuari manualment, després de diferents proves es va crear l'opció de fer-ho amb clicks sobre el mapa de forma molt més còmoda per l'usuari, implementant també que pogués introduir més d'una coordenada inicial.

Una funció fruit de diferents proves és "natural-disaster", la qual simula un fenomen meteorològic o una malaltia que redueix dràsticament la població total, en futures versions s'hauria de decidir i polir el funcionament exacte, actualment redueix 1/3 de la població i es pot fer 1 cop per simulació.

9 EXPOSICIÓ DELS RESULTATS

Els resultats obtinguts en les diferents proves finals del projecte són satisfactoris, s'aconsegueix simular l'evolució i distribució dels jaciments agrícolers d'una forma suficientment realista i visual com per analitzar i jugar amb els diferents comportaments establerts.

Quan executem l'aplicació ens trobem amb un menú guiat amb els punts 1-4 on podem seleccionar els valors de les diferents variables (o deixar-les per defecte) i un dels dos mapes disponibles, després podem seleccionar els punts on comença la simulació i donar inici a aquesta. Un cop iniciada, trobarem diferents monitors on veure dades de l'execució tals com l'any o els jaciments totals i alguns botons per interactuar o extreure dades de la simulació.

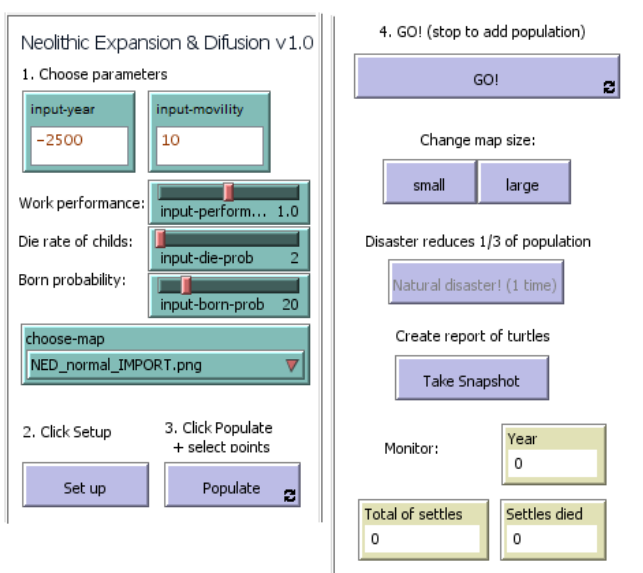


Fig. 2 - Menú de la interfície d'usuari.

Observem que els jaciments es van reproduint per les zones de nivells similars (depenent del cost d'accés) i prioritant el desplaçament per zones de costa i amb alta productivitat, evitant pujar a les muntanyes i zones menys productives sempre que no es trobin en un cas extrem, degut a que tenen més dificultat d'accés i menys productivitat, factors que es tenen en compte a l'hora d'escollir un nou patch en la zona de mobilitat.

Obtenim també els resultats esperats quan iniciem un jaciment en zona muntanyosa, baixen als millors patch i continuen el moviment per la costa.

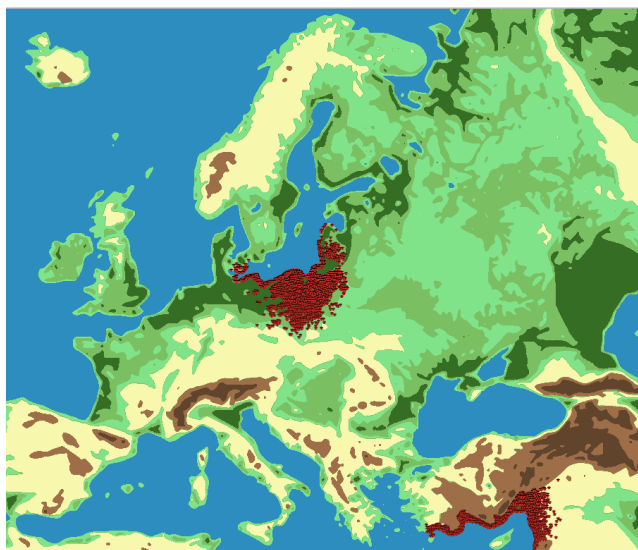


Fig. 1 - Any 220 des de inici amb 3 jaciments inicials distribuïts pel mapa.



Fig. 3 - Any 683 des de inici amb 3 jaciments inicials distribuïts pel mapa.

El comportament a l'hora de buscar nous patches per crear un nou jaciment és el correcte i esperat, abans d'escollir el millor patch com hem comentat, es comprova si es pot subsistir i si no és així, es desplaça o divideix com hem comentat anteriorment, tenint en compte que en aquest escenari dos jaciments diferents no poden col·laborar entre ells i per tant es busquen patches lliures.

Es pot observar puntualment que alguns jaciments apareixen a les illes més properes a la costa encara que la navegació no està implementada, això és degut a que la variable moviment assignada és suficient per arribar a aquest patch, no és una falla a tenir en compte pel fet que es pot interpretar com si aquestes poblacions poguessin arribar-hi nedant o amb bales rudimentàries tal com si travessessin un riu, tenint en compte que el seu moviment és superior a la distància a nedar.

Quan el programa porta un temps d'execució, es comença a ralentitzar en la seva execució, un comportament comprensible degut a que el creixement dels jaciments serà exponencial si les condicions d'evolució son adequades.

10 CONCLUSIONS I LÍNIES OBERTES

Com a conclusions, actualment podem dir que tenim un programa funcional, una primera versió utilitzable però amb possibilitats d'ampliació i millora.

Tenint en compte que és un software desenvolupat des de 0 en el temps disponible per dur a terme el treball, s'han assolit els objectius proposats en gran mida, afegint i desenvolupant nous objectius i millores que es demanaven per al correcte funcionament del software.

El menú de la UI permet escollir algunes variables seleccionades per tal de modificar el comportament de la simulació i permet escollir mitjançant clicks la posició dels jaciments inicials, creant una experiència còmode i jugable per a l'usuari, encara que en l'apartat monitors només tenim els més bàsics per interpretar la simulació, en cas que el software continués com a eina d'investigació caldria afegir monitors i gràfiques amb més dades útils.

Encara que el funcionament del software és acceptable per fer simulacions i proves, sempre es pot millorar, sobretot en els algorismes de decisió que, com hem comentat en apartats anteriors, les variables i fórmules utilitzades han estat desenvolupades mitjançant proves, per tal que executin una simulació el més realista possible, es podrien millorar amb algorismes més precisos.

El raster del mapa té un funcionament correcte tenint en compte costos d'accés i productivitat, però és un comportament pla a tot el mapa, si volem tenir en compte reduccions de productivitat per la latitud o altres modificacions parcials s'haurà de crear un raster nou que inclogui aquestes noves dades.

Com podem observar al diagrama de Gantt annex, el desenvolupament ha anat variant respecte al plantejat en un principi segons avançava, afegint nous objectius i deixant de banda uns altres inicialment proposats per tal d'obtenir un producte funcional dintre dels termes establerts, però deixant obertes possibilitats a seguir desenvolupant.

Els objectius que no s'han pogut assolir degut a això són la creació de l'agent caçador, amb la conseqüent programació de recursos animals als patches i la interacció amb l'agent agricultor i la funcionalitat de navegació, que tindria en compte si els agents poden arribar a les illes més llunyanes, principalment.

Altres idees de millora que van sorgir durant l'etapa de desenvolupament van ser, a més de les esmentades anteriorment, modificar la taxa de regeneració segons el tipus de sol, podent escollir-ho amb un raster de latituds i desenvolupar correctament la funció de mort per edat del jaciment, en cas que només siguin ancians.

En canvi, ja que aquests objectius no s'han assolit a temps, s'ha desenvolupat un mapa molt més complex, passant dels 4 nivells (diferents zones) als 10 nivells, tenint en compte que els mapes que es poden introduir seran diferents i es convertiran seguint el codi de colors indicat en apartats anteriors, a més, l'algorisme cycle-of-activity ha estat modificat i millorat diferents cops, fent el comportament bàsic dels agents a l'hora de desplaçar-se i dividir-se el més realista possible i tenint en compte noves opcions, com per exemple fer sacrificis en cas de no poder subsistir, també s'ha afegit l'opció de crear snapshots amb la informació d'un instant concret de la simulació, per tal d'estudiar i comparar diferents execucions, entre altres objectius.

AGRAÏMENTS

Aprofito per agrair l'ajuda i col·laboració de Yolanda Benítez com a tutora d'enginyeria, Juan Antonio Barceló com a tutor del departament de Prehistòria, Xavier Vilà, Francesc Josep Miguel Quesada i Salvador Pardo-Gordó, com a col·laboradors i equip interessat en el projecte i finalment a la meva família i amics que m'han animat i donat suport al llarg del grau.

BIBLIOGRAFIA

- [1] Netlogo official website. (1999-2018). Uri Wilensky. <https://ccl.northwestern.edu/netlogo/>
- [2] Netlogo teacher Channel. (2009). Gabriel Wurzer. <https://www.youtube.com/user/gabrielwurzer/>
- [3] Isern Neus, "Front sreading in population dynamics models. Theory and application to the Neolithic transition" 2011 Thesis
- [4] Sean M. Bergin, "Mechanisms and Models of Agropastoral Spread During the Neolithic in the West Mediterranean: The Cardial Spread Model", 2016
- [5] Pardo-Gordó Salvador, "Sistemas complejos adaptativos y simulación computacional en Arqueología", TRABAJOS DE PREHISTORIA 74, N.º 1, enero-junio 2017, pp. 9-25, ISSN: 0082-5638
- [6] Steele James, "Human Dispersals: Mathematical Models and the Archaeological Record", Human Biology, 81(3):121-140, Wayne State University Press 2009
- [7] Ryan E. Hughes, Erika Weiberg, Anton Bonnier, Martin Finné and Jed O. Kaplan, "Quantifying Land Use in Past Societies from Cultural Practice and Archaeological Data" Institute of Earth Surface Dynamics, University of Lausanne, 1015 Lausanne, 2018 Switzerland
- [8] Fort Joaquim, "Demic and cultural diffusion propagated the Neolithic transition across different regions of Europe", 2015, J. R. Soc. Interface 12:20150166.

ANNEX

Aquest apartat el reservem per afegir dades d'ajuda a la comprensió sobre l'evolució del software.

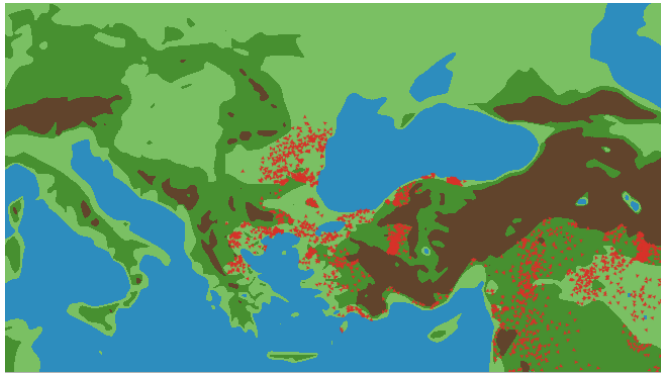


Fig. 5 - Simulació en etapa mitjana del desenvolupament

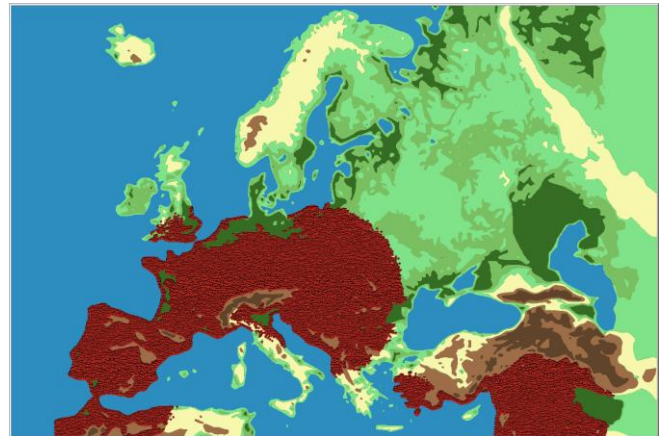


Fig. 4 - Simulació en etapa final del desenvolupament.

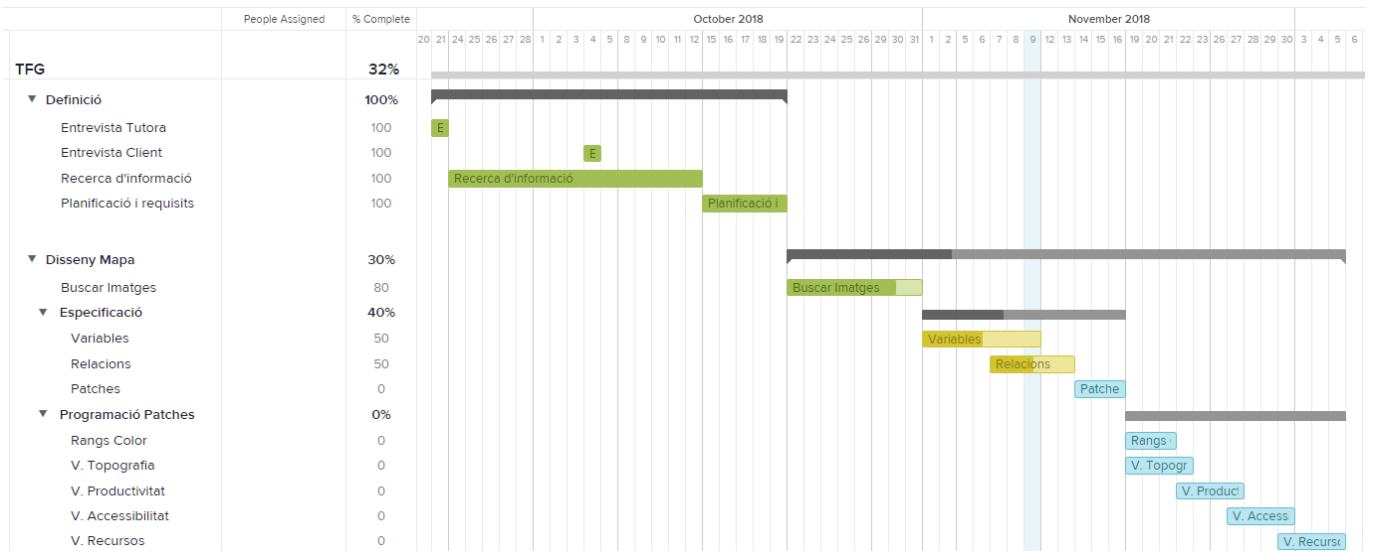


Fig. 7 - Diagrama de Gantt inicial, part 1

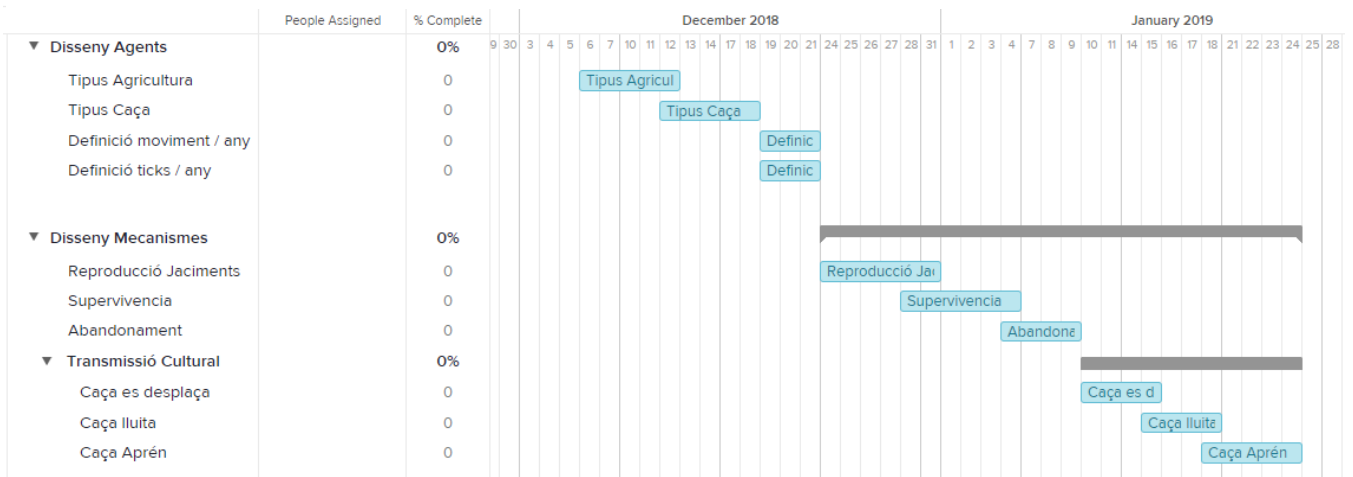


Fig. 6 - Diagrama de Gantt inicial, part 2