



Beer Game

Jordi Fernández Rodríguez
Grau Multimèdia
Videojocs

Jordi Duch Gavalrà/Helio Tejedor Navarro
Joan Arnedo Moreno

14 de Gener del 2018



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FITXA DEL TREBALL FINAL	A
PRESENTACIÓ	1
FINALITAT	4
CONTEXT D'APLICACIÓ	4
METODOLOGIA	5
CONCEPTUALITZACIÓ.....	5
ELEMENTS DEL JOC.....	7
PROGRAMARI.....	7
ESCENARI	11
NAVEGACIÓ.....	11
EINES	12
COMPTADORS.....	12
PROCESSAT	13
SCRIPTS	14
ELEMENTS GRÀFICS.....	20
PROVA AMB USUARIS	21
RESULTATS	22
VERSIÓ 1.....	22
COMENTARIS DELS USUARIS.....	24
VERSIÓ 2: INCORPORANT MILLORES.....	26
ASPECTES A MILLORAR PER FUTURES EDICIONS	27
CONCLUSIONS	31
BIBLIOGRAFIA	32

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Beer Game</i>
Nom de l'autor:	<i>Jordi Fernández Rodríguez</i>
Nom del consultor/a:	<i>Jordi Duch Gavaldà Helio Tejedor Navarro</i>
Nom del PRA:	<i>Joan Arnedo Moreno</i>
Data de lliurament (mm/aaaa):	<i>01/2018</i>
Titulació o programa:	<i>Grau Multimèdia</i>
Àrea del Treball Final:	<i>Videojocs</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Videojoc, cervesa, gestió</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>Finalitat: Desenvolupar la primera versió jugable d'un videojoc de gestió de recursos per a produir cervesa: <i>Beer Game</i>. Context: Aquest joc està dirigit a jugadors casuals que fan servir el mòbil o la tablet per jugar i estan interessats en l'elaboració de cervesa artesana. Aquest joc permet aprendre les bases del procés. També es pot oferir el joc a empreses productores de cervesa que el podrien oferir com a obsequi. Metodologia: Es va fer servir el motor Unity i l'eina Ultimate Isometric Toolkit. Primer, es van centrar els esforços en el funcionament del joc i la seva programació respecte al comportament de les eines i dels menús. Després es van implementar totes les millores gràfiques i es van fer proves amb usuaris. Els comentaris d'aquests es van introduir en la segona versió. Resultats: La primera versió va ser provada per 3 persones que van comentar alguns aspectes a millorar com ara la necessitat d'incloure un tutorial, més informació als menús i visibilitat del final de procés, aspectes inclosos en la segona versió. El resultat final inclou el procés complet d'elaboració de cervesa, des de el conreu i producció de les matèries primeres fins l'obtenció del producte final. Conclusió: Ha estat possible dissenyar una primera versió jugable d'un joc de gestió de producció de cervesa artesana. Malgrat això, hi ha alguns aspectes que s'han de millorar: incloure una fase de bar així com molts més productes (per exemple diferents tipus de cereals) per incrementar la variabilitat.</p>	
Abstract (in English, 250 words or less):	
<p>Purpose: To develop the first version of a farming-based game to produce beer: <i>The Beer Game</i>. Context: This game target casual players who mainly use the mobile or the tablet to play and are interested in the brewing process.</p>	

This game allow them to learn the bases of that process while having a first contact with the different aspects of the production. Brewing companies could also offer this game as a gift to their customers. **Methodology:** I used Unity and Ultimate Isometric Toolkit. First, all my efforts were centred in designing and scripting. Then, I implemented all the graphic improvements and the first version was tested by users. After their comments, a second version was done. **Results:** The beta-testers commented the following aspects to improve the game: the need to include a tutorial; more information related to the menus; and visibility at the end of process. These improvements were implemented in the second version. The final result includes the complete process of brewing, from harvest to bottling. **Conclusion:** it has been possible to design the first playable version of a brewing game. In spite of this, there are some aspects that need to be improved, such as the inclusion of a pub and more products (i.e. different types of grains) in order to increase variability.

Links de descarrega de materials:

Assets:

https://www.dropbox.com/s/0bdj0agbkknvmcm/assets_BeerGame.zip?dl=0

Build:

https://www.dropbox.com/s/2psldik86so5xcb/build_BeerGame.zip?dl=0

Presentació:

https://www.dropbox.com/s/71vhue8k6yrwigg/Presentaci%C3%B3_TFG-BeerGame_JordiFernandez.mp4?dl=0

PRESENTACIÓ

Beer Game és un videojoc on ens encarreguem de l'elaboració de cervesa artesanal. No només ens centrem en el procés de fermentació si no que ens haurem d'encarregar de cultivar els cereals, els llúpols i fruits necessaris per una banda, i, per l'altra, crear els diferents elements químics que intervenen.

Tenint en compte els diferents processos, combinacions d'ingredients i temps de fermentació, obtindrem diferents tipus de cervesa. Depenent de tot això, es pot donar el cas de que el procés falli i s'hagi de llençar tot el producte, perdent material i diners ja que, si aconseguim elaborar bones cerveses, obtindrem premis per a poder continuar. Finalment, vendrem les cerveses a la botiga als diferents personatges que vinguin a la granja.



No es pretén que sigui un simulador real de com és el procés de creació de cervesa ja que intervenen masses factors i elements com ara la temperatura, els temps de cocció, macerat, etc. i la quantitat d'ingredients. Això no vol dir que Beer Game no vagi evolucionant en un futur per aconseguir que, amb cada nova versió, s'acosti més a com es fa realment els diferents tipus de cervesa.

Tipus de joc

Ens trobem davant d'un joc de gestió de recursos. Aquest tipus de jocs es caracteritzen perquè no tenen una història lineal i l'usuari ha de complir els objectius proposats per avançar en el joc on normalment són del tipus crear una ciutat, un parc temàtic, una granja, etc. A més a més, durant aquest procés, el jugador ha de controlar els recursos que intervenen en cada tipus de joc, depenent del nivell de realisme aquests seran de menys a més quantitat.

SIMCITY

Un dels primer jocs de gestió de recursos el trobem a la saga SimCity de Will Wright, on es reptava al jugador a crear una ciutat des de zero. Per tal d'anar fent créixer la seva ciutat havia de construir fabricues, centrals elèctriques, cases per a la gent que es mou a la ciutat per tal de fer-la créixer, etc. I, a partir d'aquí, s'havia de gestionar tots els aspectes que intervenen en una gran ciutat per a que la gent estigués feliç, com ara el transport, la sanitat, seguretat, carreteres i els impostos.



SimCity 2000 Special Edition, ©Origin

The SIMS

Seguint en aquesta mateix línia de joc, Will Wright va sorprendre amb un videojoc on passàvem a controlar i gestionar les vides d'aquells ciutadans al més pur estil Big Brother, es tracta del títol Sims que encara avui en dia ven milions de còpies de la seva quarta entrega.



The Sims, ©Maxis



Però per posar un exemple clar del tipus de videojoc que és Beer Game, el podríem trobar a “Hay Day” de SuperCell, creadors també dels famosos “Clash of Clans” i “Clash Royale”, on hem de gestionar els aspectes principals d’una granja com poden ser cuidar els animals per aconseguir llet, llana, ous, etc. o conrear el camp fins produir diferents elements per després vendre’ls a personatges AI o a altres jugadors de manera on-line i aconseguir, d’aquesta manera, diners per millorar els nostres recursos o edificis de la granja per a que siguin més eficients.



HayDay, ©SuperCell

Tornant a Beer Game, al tractar-se d’un gestor de recursos, la majoria de processos els haurem de fer a través de menús i sub-menús, tenint sempre en compte la quantitat de material que necessitem i del que disposem. Si no existeixen aquests elements no podrem passar al següent procés. S’ha de tenir en compte que és una primera versió del joc i, per tant, no es pot comparar directament als exemples plantejats anteriorment però si que és un punt de partida per a acabar fent un joc d’elaboració de cervesa que pot servir de guia per a la gent interessada en aquest procés ja que s’intentarà que les taules d’ingredients i temps de fermentació siguin proporcionals als reals per a que la durada del joc no sigui massa llarg i pugui acabar avorrint. D’aquesta manera,

l'usuari haurà de conrear els cereals, llúpols i la resta de materials que es necessiten per crear una cervesa específica.

FINALITAT

La finalitat d'aquest projecte és desenvolupar la primera versió d'un videojoc de gestió de recursos per a produir cervesa. No d'una manera realista si no com a presentació de com és el procés i que la gent pugui veure'l d'una manera senzilla.

CONTEXT D'APLICACIÓ

Beer Game està pensat per a que pugui jugar tothom ja que la navegació és simple i no requereix que el jugador sigui experimentant en el món dels videojocs. Per tant, un primer àmbit d'aplicació el podríem trobar en el jugadors casuals que majoritàriament fan servir el mòbil o la tablet per jugar.



Per una altra banda, cada cop hi ha més gent interessada en l'elaboració de cervesa artesana i Beer Game podria ser una manera d'aprendre les bases del procés i tenir un primer contacte amb els diferents aspectes de la producció. Seria una manera de **gamificació** dins del procés de creació de cervesa.



<http://www.educaciontrespuntocero.com/noticias/gamificacion-y-game-based-era-digital/69585.html>

La **gamificació o ludificació** s'entén com lús de tècniques, elements i dinàmiques més pròpies dels jocs i l'oci aplicades a activitats que no són de lleure per tal de potenciar la motivació, millorar la productivitat o, simplement, activar l'aprenentatge.

Per últim, es pot oferir el joc a empreses productores de cervesa ja sigui una gran multinacional o un negoci petit de cervesa artesana. Aquestes podrien oferir el videojoc als seus clients com a obsequi introduint la seva marca en ell. Moltes companyies ofereixen mini jocs a les seves web com per exemple ColaCao o Disney. En aquest cas es podrien oferir codis de descarrega a través de les principals plataformes d'aplicacions (App Store, Steam o Google Play).



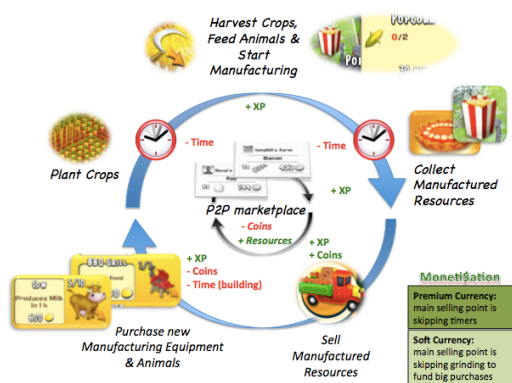
METODOLOGIA

CONCEPTUALITZACIÓ

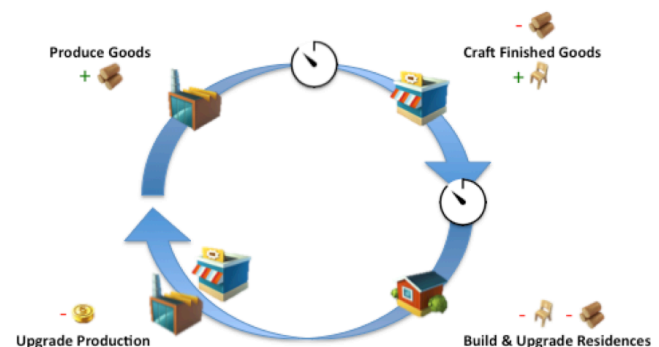
Com hem vist a l'apartat anterior el joc no té una història com podria tenir un videojoc tradicional, en aquest cas ens trobem en un espai fixe ambientat en una granja amb els diferents departaments i elements amb els que el jugador pot interaccionar per tal d'elaborar la cervesa que vulgui i vendre-la als personatges que vinguin. Aquests tindran prioritats per un tipus de cervesa o una altra i el jugador haurà de produir la que toqui per a cada personatge.

Per fer la versió inicial del joc es planteja un nivell de tutorial per ensenyar les mecàniques del joc i com poder elaborar la primera producció de cervesa. El joc no farà servir els temps reals d'elaboració de cervesa si no que ho farà en qüestió de segons per fer-ho més dinàmic. S'ha de tenir en compte que pels temps de macerat i cocció poden ser hores i pels temps de fermentació poden arribar a ser setmanes.

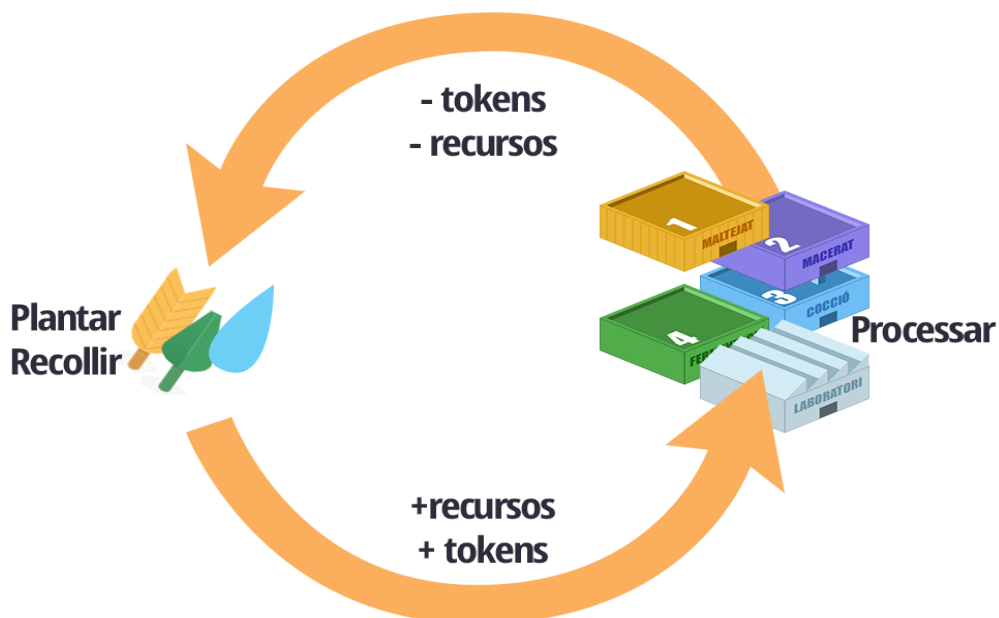
En els comentaris de la PEC1 se'm va recomanar llegir sobre el "core loop" dels videojocs tipus "Farmville". Aquest concepte ve donat per esquematitzar el procés principal del joc, quins elements necessitem per portar a terme una altra tasca, què ens dona punts o recompenses, quins processos depenen de l'usuari i quins no, etc. A continuació es pot veure uns exemples de "core loop" i el que es va realitzar per a Beer Game:



Core loop Hay Day, ©SuperCell



Core loop SimCity BuildIt, @EA



Core loop Beer Game

ELEMENTS DEL JOC

- Magatzem (on trobem tots el materials que necessitem per al joc)
 - Camps de conreu: el jugador pot posar les àrees deis camps allà on vulgui
 - Cereals: inicialment de civada però en versions futures poden haver-hi de blat, arròs, etc.
 - Llúpol: es pot cultivar als camps de conreu també
 - Fruites: tipus de fruites que es poden cultivar per afegir al procés.
- Laboratori (el jugador crea els llevats que necessiti per afegir-los al fermentat)
 - Llevats Ale: per a fermentació entre 18°C i 24°C
 - Llevats Lager: per a fermentació entre 7°C i 12°C
 - Saques de llevat: angleses, belgues, de blat alemanyes, americanes.
- Locals de processat (cada procés té un local diferent per a controlar els elements que intervenen)
 - Macerat
 - Cocció
 - Fermentat
 - Embotellat
 - Botiga (aquí es podran vendre les cerveses un cop embotellades)

PROGRAMARI

Per portar a terme el videojoc es va considerar fer servir un d'aquests tres engines: Unreal, GameMaker Studio o Unity.

A continuació es descriuen els diferents engines fent èmfasi en els punts forts de cadascun d'ells:

- Unreal va ser creat per la companyia Epic Games i fet servir en el seu shooter del mateix nom publicat a l'any 1998. Va ser la base de jocs posteriors com "Deus Ex", "Gears of War", "Bioshock" i molts més. Actualment es pot trobar la seva quarta generació a la web per descarregar de manera gratuïta. Aquesta versió es totalment completa incloent fins i tot les actualitzacions periòdiques i qualsevol persona pot crear el seu propi videojoc. Si es donés el cas que un videojoc creat amb Unreal es comercialitzés de manera oficial, Epic Games tindria el 5% dels beneficis cada trimestre sempre i quan el producte superi els seus primers 3000 dòlars.



**UNREAL
ENGINE**



Deus Ex Human Revolution, ©Eidos

Unreal pot ser molt útil pels videojocs d'acció tipus "Final Fantasy", "Batman Arkham City" o la resta comentats abans, és un motor molt potent i totalment operatiu ofert de manera gratuïta. Va ser una molt bona iniciativa per part d'Epic Games però per les necessitats que intervenen a la creació de Beer Game no era oportú fer servir aquest tipus de motor. Beer Game és un projecte molt més senzill a nivell gràfic i, per tant, era necessari trobar un motor amb les característiques adequades per portar-lo a terme.

- GameMaker Studio és una eina basada en el llenguatge de programació interpretat i un kit de desenvolupament de software (SDK). Va aparèixer al mercat a l'any 1999 amb el nom d'Animo i va ser creat pel professor Mark Overmars, no va ser fins a l'any 2011 que se'l va començar a conèixer com a GameMaker. Alguns dels seus jocs més coneguts són "Spelunky" o "Hotline Miami".



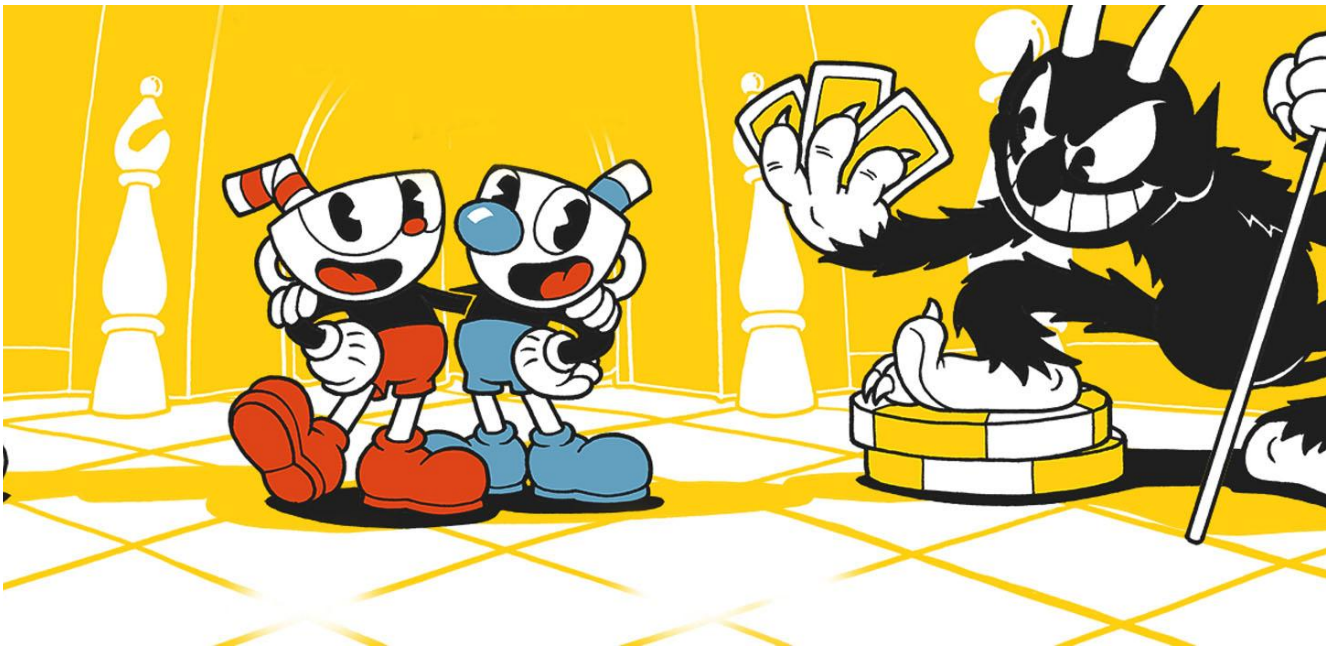
Spelunky, ©Derek Yu

El seu llenguatge de programació està basat en Delphi, això fa que a l'hora d'escriure sigui més flexible ja que, per exemple, no és necessari incloure el punt i coma al final de cada línia. La llicència que ofereix és gratuïta però força limitada ja que no ofereix prou recursos i no permet la publicació en cap plataforma, si que existeix una versió comercial amb moltes més característiques addicionals.

- Unity va ser creat per la companyia Unity Technologies i publicat al 2005, actualment es troba en la versió 5 i ofereixen dues opcions: Unity Professional i Unity Personal. S'han creat molts jocs amb aquest motor però els més coneguts són: "HeartStone", "Firewatch" o el recent "Cuphead".



Unes de les característiques més destacables és que a partir de la versió 3.0 va incorporar una eina per a depurar scripts anomenada MonoDevelop i, a més, Unity pot utilitzar-se juntament amb altre programari com Blender, Cinema4D, Maya, Adobe Photoshop, etc. D'aquesta manera, totes les modificacions que es fan a nivell gràfic, s'actualitzen en les instàncies o sprites del projecte. També incorpora una modalitat de crear jocs en 2D o, el que s'anomena 2.5, un tipus de 2D que simula certa tridimensionalitat i que, pel que sembla, és el que es fa servir en molts jocs de vista isomètrica. També incorpora molts recursos i paquets d'elements gràfics per descarregar de manera gratuïta des del seu "Unity Asset Store".



Cuphead, ©Studio MDHR

Finalment es va decidir utilitzar Unity ja que, tal i com s'ha dit, la seva web i Internet en general estan plens de recursos i tutorials per començar a fer servir el programari. Per una altra banda, es va comprovar que era el més adient per al tipus de joc que es volia crear, oferia eines per treballar en vistes isomètriques d'una manera senzilla i, d'aquesta manera, poder dedicar més esforços a la part del disseny del joc i els seus elements.

ESCENARI

Tal i com es va plantejar inicialment, el joc es desenvolupa en un espai isomètric. Per tant, vaig investigar sobre el tema i vaig trobar una eina per Unity anomenada "Ultimate Isometric Toolkit" que em va semblar molt útil per a aquest projecte.



El que ens permet aquesta eina es transformar les coordenades tridimensionals dels objectes de tal manera que podem treballar fàcilment en un espai 2D isomètric amb sprites. Es pot trobar en aquest enllaç:

<https://www.assetstore.unity3d.com/en/#!/content/42058>

El que s'ha de tenir en compte és que els sprites abans esmentats han de respectar les proporcions de la vista isomètrica amb la que s'està treballant. De no ser així els elements no es complementarien correctament i no obtindríem el resultat desitjat.

NAVEGACIÓ

Un cop generat l'escenari he creat dos scripts per acostar i allunyar la càmera i per moure-la respectivament. La idea és que l'usuari faci servir el botó esquerre del ratolí i desplaçí l'escenari en la direcció que vulgui. Per una altra banda, amb la roda del ratolí pot augmentar o disminuir el zoom de tot l'escenari.

Tot i que en un principi es va pensar per fer servir un ratolí, això també pot funcionar per mòbils i tablets. La navegació en aquest cas es realitzaria amb els dits.

EINES

Per l'apartat d'eines es va crear un "canvas" amb Unity on es van incorporar totes les eines i menús que necessita el joc, ja sigui per plantar com per cultivar o per crear la cervesa en els diferents processos.



Cereal: amb aquesta eina plantem la civada que és el material principal per fer la cervesa. Més endavant es pot implementar una sèrie de sub-menús per escollir quin tipus de cereal volem plantar i cadascú tindrà el seu sprite per a poder identificar-lo.



Llúpul: igual que amb el cereal, aquesta eina ens servirà per plantar el llúpul i més endavant s'afegiran diferents tipus d'aquesta planta a través de sub-menús.



Aigua: servirà per recollir l'aigua que quedarà emmagatzemada per al procés de la cervesa.



Recol·lectar: aquesta eina servirà per recollir tot el cereal i llúpul que estigui llest i quedaran emmagatzemats per al seu posterior procés.

COMPTADORS

Fent servir les eines de l'aigua i de recol·lectar, es va implementar una sèrie de comptadors que ajuden a saber la quantitat d'ingredients que hi ha al magatzem. Aquests es troben a la part superior de la pantalla sempre visibles ja que són els elements principals del joc.

Al tractar-se d'un joc de conreu es necessita una sèrie d'informació de tots els materials que l'usuari disposa, és per això que es va decidir deixar com a comptadors principals els d'aigua, conreu i tokens i per a cada procés es va incloure els comptadors en el seu menú corresponent.

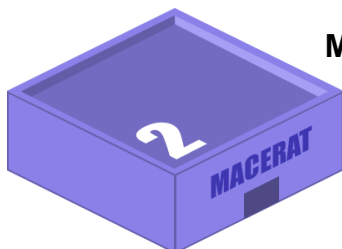
Cada edifici ens mostra la quantitat d'elements que s'han creat i els elements que es poden crear això vol dir que, si no tenim res, no podem passar al següent procés.

PROCESSAT

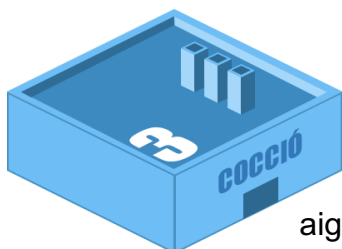
El processat de la cervesa es fa a través dels edificis creats prèviament. Clicant a sobre de cadascú, apareix un menú on podem seleccionar els elements que volem crear i a la part dreta de la pantalla apareix el magatzem on podem veure quina quantitat de cada material tenim creat fins al moment.



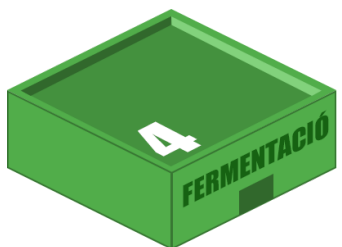
Molta o maltejat: es clica a cada tipus de malta que es vol crear a partir del cereal que disposem. Quan s'està processant cada element no es pot interactuar amb aquell mateix botó o tancar el menú.



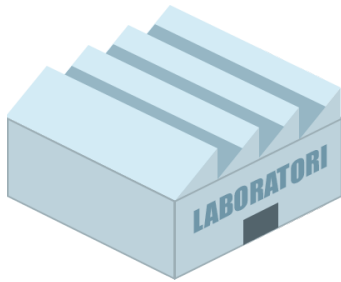
Macerat: aquí es comença a processar el tipus de cervesa que volem, cada una descomptarà automàticament la quantitat de malta i aigua necessaris.



Cocció: igual que amb el macerat, per a cada tipus de cervesa es descomptarà la quantitat de most que s'ha creat a la maceració així com la quantitat de llúpul i aigua.



Fermentació: aquí es descomptarà el material que s'ha generat a la cocció i la quantitat de llevat.



Laboratori: sense llevat no es pot fer la fermentació, per tant, abans s'haurà de crear el llevat corresponent per a aquest procés.

SCRIPTS

- **Control de la càmera**

- *CameraDrag.cs*

Aquest script controla el moviment de la càmera al prémer el botó esquerre del ratolí i l'arroseguem en la direcció que volem de l'escenari.

```
5 public class CameraDrag : MonoBehaviour {
6     //private Vector3 ResetCamera;
7     private Vector3 Origin;
8     private Vector3 Diference;
9     private bool Drag=false;
10    void Start () {
11        //ResetCamera = Camera.main.transform.position;
12    }
13    void LateUpdate () {
14        if (Input.GetMouseButton (0)) {
15            Diference=(Camera.main.ScreenToWorldPoint (Input.mousePosition))- Camera.main.transform.position;
16            if (Drag==false){
17                Drag=true;
18                Origin=Camera.main.ScreenToWorldPoint (Input.mousePosition);
19            }
20        } else {
21            Drag=false;
22        }
23        if ((Drag==true) && (MainIsoScript.currentTool == "none")){
24            Camera.main.transform.position = Origin-Diference;
25        }
26    }
27 }
```

- *CameraZoom.cs*

Script amb el que a través de la roda del ratolí ens acostem o en allunyem de l'escenari. En ell s'especifiquen la velocitat, suavitat de moviment i el valor màxim i mínim de la càmera.

```
5 public class CameraZoom : MonoBehaviour {
6
7     public float zoomSpeed = 5;
8     public float targetOrtho;
9     public float smoothSpeed = 30f;
10    public float minOrtho = 3.0f;
11    public float maxOrtho = 10.0f;
12
13    void Start() {
14        targetOrtho = Camera.main.orthographicSize;
15    }
16
17    void Update () {
18
19        float scroll = Input.GetAxis ("Mouse ScrollWheel");
20        if (scroll != 0.0f) {
21            targetOrtho -= scroll * zoomSpeed;
22            targetOrtho = Mathf.Clamp (targetOrtho, minOrtho, maxOrtho);
23        }
24
25        Camera.main.orthographicSize = Mathf.MoveTowards (Camera.main.orthographicSize, targetOrtho, smoothSpeed * Time.deltaTime);
26    }
27 }
```

- **Control dels elements del joc**

- *GameState.cs*

Després del feedback rebut a la PAC2 vaig crear un GameObject amb el seu script corresponent on es recollia la informació del elements del joc, com una mena de magatzem central. Aquest arxiu es diu GameState i és la base per a que el joc

pugui sumar i restar els elements per a cada procés del joc.

```
284 public void sumaImperialIPA(int value)
285 {
286     nImperialIPA += value;
287     countImperialIPA.text = "Imperial IPA: " + nImperialIPA.ToString ();
288 }
289
290 public void sumaAmericanStout(int value)
291 {
292     nAmericanStout += value;
293     countAmericanStout.text = "American Stout: " + nAmericanStout.ToString ();
294 }
295
296 public void sumaLlevatAle(int value)
297 {
298     nLlevatAle += value;
299     countLlevatAle.text = "Ale: " + nLlevatAle.ToString ();
300 }
```

```
7 public class GameState : MonoBehaviour {
8
9     public Text countToken;
10    public int nToken;
11
12    public Text countCivada;
13    public int nCivada;
14
15    public Text countLupol;
16    public int nLupol;
17
18    public Text countAigua;
19    public int nAigua;
20
21    public Text countPale;
22    public int nPale;
23 }
```

- **Eines**

- *MainIsoScript.cs*

Aquest petit script permet identificar si tenim o no cap eina seleccionada.

```
5 public class MainIsoScript : MonoBehaviour {
6
7     public static string currentTool = "none";
```

- *tool.cs*

S'inclouen les funcions necessàries per a la selecció dels botons de les eines, aquestes modifiquen el *MainIsoScript.cs* en el cas de que es seleccioni el cereal, el llúpol, el cubell o la falç.

```
5 public class tool : MonoBehaviour {
6
7     public void Tool_Cereal()
8     {
9         MainIsoScript.currentTool = "cereal";
10        Debug.Log ("tool cereal");
11    }
12
13    public void Tool_Hop()
14    {
15        MainIsoScript.currentTool = "hop";
16        Debug.Log ("tool hop");
17    }
18
19    public void Tool_Water()
20    {
21        MainIsoScript.currentTool = "water";
22        Debug.Log ("tool water");
23    }
24
25    public void Tool_Scythe()
26    {
27        MainIsoScript.currentTool = "scythe";
28        Debug.Log ("tool scythe");
29    }
30
31 }
```

- *plant.cs*

S'encarrega de les funcions de plantat de la civada i del llúpol així com de la seva recollida.

Es va solucionar el problema per a que, quan es clicava a sobre del canvas, no s'interactués amb l'escenari. Això era clau per a quan teníem els menús oberts de cada edifici de processat ja que podíem obrir diferents menús alhora i la idea era tenir només un.

```
70 void OnMouseDown()
71 {
72     if (!EventSystem.current.IsPointerOverGameObject ()) {
73         if ((MainIsoScript.currentTool == "cereal") && (GetComponent<SpriteRenderer> ().sprite == NoPlant)) {
74             GetComponent<SpriteRenderer> ().sprite = TerraCereal;
75             timer = delay;
76             return;
77         }
78     }
79 }
```

Es va fer servir un temporitzador per a que els sprites corresponents canviessin fins arribar a l'estat final per poder ser recollit.

```
30 void Update ()
31 {
32     timer -= Time.deltaTime;
33     if (timer <= 0)
34     {
35         if (GetComponent<SpriteRenderer> ().sprite == TerraCereal)
36         {
37             GetComponent<SpriteRenderer> ().sprite = TallCereal;
38             timer = delay;
39             return;
40         }
41     }
42 }
```

Tenint en compte que la falç estigui seleccionada per l'usuari, al fer clic a sobre de la civada o el llúpol deixarà l'sprite original i sumarà a l'script *GameState.cs* la quantitat de material i tokens definits.

```
85     if ((MainIsoScript.currentTool == "scythe") && (GetComponent<SpriteRenderer> ().sprite == PlantaCereal)) {
86         GetComponent<SpriteRenderer> ().sprite = NoPlant;
87         GameObject.FindWithTag ("state").GetComponent<GameState> ().sumaCivada (10);
88         GameObject.FindWithTag ("state").GetComponent<GameState> ().sumaToken (5);
89         return;
90     }
91 }
```

- *pickWater.cs*

Al ser un àrea diferent de l'escenari, es va crear un script específic per recollir l'aigua amb l'eina del cubell. De la mateixa manera que l'anterior, aquest suma la quantitat especificada a *GameState.cs*

```
24 void OnMouseDown()
25 {
26     if (!EventSystem.current.IsPointerOverGameObject ()) {
27         //Debug.Log ("clicked on water");
28         if (MainIsoScript.currentTool == "water") {
29             GameObject.FindWithTag ("state").GetComponent<GameState> ().sumaAigua (20);
30             return;
31         }
32     }
33 }
```

- **Elaboració**

- *menu_open.cs*

Es va aplicar aquest script als diferents edificis de l'escenari per a que, al clicar a sobre de cada un, s'obrís els seu menú de processat corresponent.

```
30     void OnMouseDown()
31     {
32
33         if (!EventSystem.current.IsPointerOverGameObject ()) {
34             menuObj.SetActive (true);
35             storageObj.SetActive (true);
36             storageObjPos = new Vector3 (335, 10, 0);
37             storageObj.transform.localPosition = storageObjPos;
38             prevStorageA.SetActive (true);
39             prevStorageAPos = new Vector3 (-335, -10, 0);
40             prevStorageA.transform.localPosition = prevStorageAPos;
41             return;
42     }
```

Es va fer servir una variable publica de tipus GameObject per a cada element: el menú principal, el magatzem propi i el magatzem del procés anterior. D'aquesta manera l'script permet posicionar el magatzem del procés anterior a la part esquerra del menú principal i el magatzem propi a la dreta.

- *molta.cs*

Amb aquest script es transforma la civada que l'usuari a recollit a malta. Està aplicat al primer edifici (Maltejat) i, a través del menú emergent i els botons corresponents, es resta la quantitat de civada necessària per a cada tipus de malta que es vol produir. Passats uns segons es passa la quantitat definida a l'script GameState.cs on servirà per continuar amb els processos posteriors.

```
87     public void moltaPale()
88     {
89         if (civadaPale <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nCivada)) {
90             activePale = true;
91             GameObject.Find ("Button_MaltaPale").GetComponent<Button> ().interactable = false;
92             GameObject.Find ("Close").GetComponent<Button> ().interactable = false;
93             GameObject.FindWithTag ("state").GetComponent<GameState> ().restaCivada (civadaPale);
94             timer = delay;
95             return;
96         } else {
97             warning.SetActive (true);
98         }
99     }
```

```

9     float timer = 5f;
10    float delay = 5f;
11
12    bool activePale;
13    bool activeCristal;
14    bool activeChocolate;
15    bool activeMunich;
16    bool activeNegra;
17
18    int civadaPale = 5;
19    int civadaCristal = 10;
20    int civadaChocolate = 15;
21    int civadaMunich = 20;
22    int civadaNegra = 30;

```

Es van definir una sèrie de variables: controlar el temps que trigaria en fer-se els elements, quantitat de civada que es necessita per a cada tipus de malta i unes booleans per quin procés està actiu.

A partir de condicionals es van crear unes funcions per a cada botó del menú, aquí es deixen inactius els botons necessaris i comença el compte enrere.

Quan el temporitzador arriba a zero, els elements generats s'actualitzen amb els condicionals de l'apartat Update.

```

36    void Update () {
37        timer -= Time.deltaTime;
38        if (timer <= 0)
39        {
40            if (activePale == true)
41            {
42                GameObject.FindWithTag ("state").GetComponent<GameState> ().sumaPale (50);
43                GameObject.Find ("Button_MaltaPale").GetComponent<Button> ().interactable = true;
44                GameObject.Find ("Close").GetComponent<Button> ().interactable = true;
45                activePale = false;
46                return;
47            }

```

- *macerat.cs*

Seguint la mateixa estructura que l'script anterior, aquí es restarà les quantitats de malta i d'aigua per al tipus de most que es vol fer.

```

63    public void mostImperialIPA()
64    {
65        if ((ipaAigua <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nAigua))
66            && (ipaPale <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nPale))
67            && (ipaCristal <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nCristal))
68            && (ipaChocolate <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nChocolate))) {
69            activeImperialIPA = true;
70            GameObject.Find ("Button_ImperialIpa_Macerat").GetComponent<Button> ().interactable = false;
71            GameObject.Find ("Close").GetComponent<Button> ().interactable = false;
72            GameObject.FindWithTag ("state").GetComponent<GameState> ().restaAigua (ipaAigua);
73            GameObject.FindWithTag ("state").GetComponent<GameState> ().restaPale (ipaPale);
74            GameObject.FindWithTag ("state").GetComponent<GameState> ().restaCristal (ipaCristal);
75            GameObject.FindWithTag ("state").GetComponent<GameState> ().restaChocolate (ipaChocolate);
76            timer = delay;
77            return;
78        } else {
79            warning.SetActive (true);
80        }

```

Un cop passat el temps necessari, es suma el most a GameState.cs.

```

40    void Update () {
41        timer -= Time.deltaTime;
42        if (timer <= 0)
43        {
44            if (activeImperialIPA == true) {
45                GameObject.FindWithTag ("state").GetComponent<GameState> ().sumaImperialIPAMost (mostIPA);
46                GameObject.Find ("Button_ImperialIpa_Macerat").GetComponent<Button> ().interactable = true;
47                GameObject.Find ("Close").GetComponent<Button> ().interactable = true;
48                activeImperialIPA = false;
49                return;
50            }

```

- *coccio.cs*

Per a la cocció es continua el procés a partir del macerat, és a dir, s'agafa la quantitat de most creat anteriorment i s'aplica a la cocció

```
58 public void coccioImperialIPA()
59 {
60     if ((ipaAigua <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nAigua))
61         && (ipaMost <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nImperialIPAMost))
62         && (ipaLupol <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nLupol))) {
63         activeImperialIPA = true;
64         GameObject.Find ("Button_ImperialIpa_Coccio").GetComponent<Button> ().interactable = false;
65         GameObject.Find ("Close").GetComponent<Button> ().interactable = false;
66         GameObject.FindWithTag ("state").GetComponent<GameState> ().restaAigua (ipaAigua);
67         GameObject.FindWithTag ("state").GetComponent<GameState> ().restaImperialIPAMost (ipaMost);
68         GameObject.FindWithTag ("state").GetComponent<GameState> ().restaLupol (ipaLupol);
69         timer = delay;
70         return;
71     } else {
72         warning.SetActive (true);
73     }
74 }
75
36 void Update () {
37     timer -= Time.deltaTime;
38     if (timer <= 0)
39     {
40         if (activeImperialIPA == true) {
41             GameObject.FindWithTag ("state").GetComponent<GameState> ().sumaImperialIPAcoccio (coccioIPA);
42             GameObject.Find ("Button_ImperialIpa_Coccio").GetComponent<Button> ().interactable = true;
43             GameObject.Find ("Close").GetComponent<Button> ().interactable = true;
44             activeImperialIPA = false;
45             return;
46         }
47     }
48 }
```

- *fermentacio.cs*

En el cas de la fermentació es va fer servir la mateix estructura amb l'afegit de que quan acaba el procés, apareix una il·lustració dient la quantitat d'ampolles que s'han generat.

```
61 public void fermentacioImperialIPA()
62 {
63     if ((ipaCoccio <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nImperialIPACoccio))
64         && (ipaLlevat <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nLlevatAle))) {
65         activeImperialIPA = true;
66         GameObject.Find ("Button_ImperialIpa_Ferm").GetComponent<Button> ().interactable = false;
67         GameObject.Find ("Close").GetComponent<Button> ().interactable = false;
68         GameObject.FindWithTag ("state").GetComponent<GameState> ().restaImperialIPAcoccio (ipaCoccio);
69         GameObject.FindWithTag ("state").GetComponent<GameState> ().restaLlevatAle (ipaLlevat);
70         timer = delay;
71         return;
72     } else {
73         warning.SetActive (true);
74     }
75 }
76
37 void Update () {
38     timer -= Time.deltaTime;
39     if (timer <= 0)
40     {
41         if (activeImperialIPA == true) {
42             GameObject.FindWithTag ("state").GetComponent<GameState> ().sumaImperialIPA (imperialIPA);
43             GameObject.Find ("Button_ImperialIpa_Ferm").GetComponent<Button> ().interactable = true;
44             GameObject.Find ("Close").GetComponent<Button> ().interactable = true;
45             finalIPA.SetActive (true);
46             activeImperialIPA = false;
47             return;
48         }
49     }
50 }
```

- *labs.cs*

Igual que en les processos anteriors pel laboratori es van crear les funcions corresponents per a cada botó dels diferents tipus de llevat. En aquest cas, es fan servir els tokens obtinguts quan es recullen la civada i el llúpul.

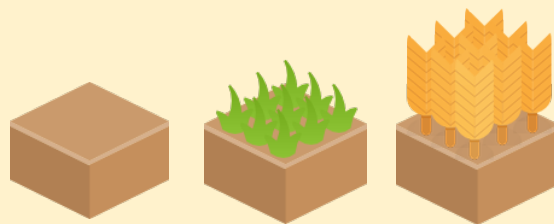
Si l'usuari no té els recursos necessaris, apareix una finestra d'advertència.

```
92     public void llevatAle()
93     {
94         if ((nTokenSimple <= (GameObject.FindWithTag ("state").GetComponent<GameState> ().nToken)) {
95             activeAle = true;
96             GameObject.Find ("Button_LlevatAle").GetComponent<Button> ().interactable = false;
97             GameObject.Find ("Close").GetComponent<Button> ().interactable = false;
98             GameObject.FindWithTag ("state").GetComponent<GameState> ().restaToken (nTokenSimple);
99             timer = delay;
100            return;
101        } else {
102            warning.SetActive (true);
103        }
104    }
```

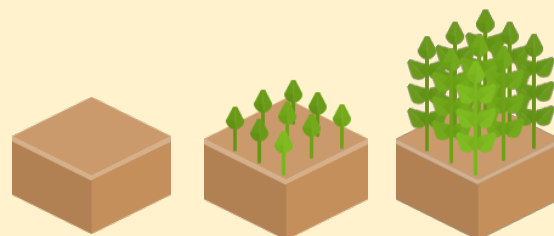
Un cop generats aquests scripts, s'han aplicat a l'element del Canvas i dins de cada botó s'ha seleccionat la funció necessària en cada cas.

ELEMENTS GRÀFICS

Durant el procés de creació de Beer Game es van anar generant els diferents elements gràfics de l'escenari com els sprites dels materials que es plantaven i anaven creixent amb el pas del temps. També es van dissenyar els edificis que identifiquen a cada procés.



Es va anar amb molta cura per a que la perspectiva fos sempre isomètrica, en cas contrari, el resultat visual no hauria estat el desitjat i s'hauria d'haver tornat a repetir el disseny.



Pel que fa als menús, en les primeres versions es van fer servir botons molt bàsics del propi canvas de Unity. En la versió final es poden veure els botons amb cada icona corresponent per a poder identificar els elements que intervenen



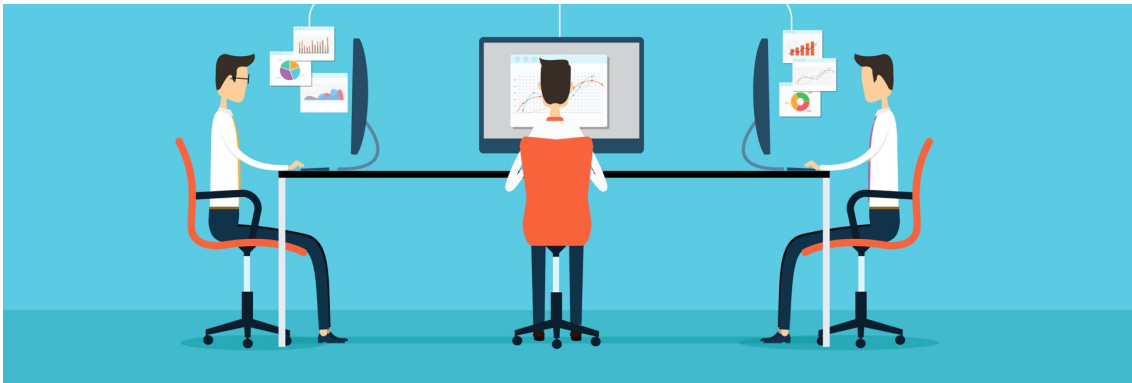
en el procés. També s'ha fet d'aquesta manera per a que l'usuari es familiaritzi amb els diferents tipus de color que pot tenir una malta o una cervesa ja fermentada.

També es van dissenyar uns panells per als comptadors principals d'aigua, civada, llúpul i tokens.



PROVA AMB USUARIS

Un cop es va fer l'entrega de la PEC 3 el 28 de desembre es va passar a una petita fase de prova amb tres usuaris. Aquest grup estava format per dues dones i un home, entre 35-40 anys amb experiència com usuaris habituals de videojocs i aficionats al món de les cerveses i els seus diferents tipus que hi ha al mercat.



Es va realitzar aquesta prova principalment per veure que en pensaven de Beer Game i poder comprovar que el joc funcionava bé i no provocava errors (veure apartat de resultats).

Les proves es van fer amb un build del joc tant per Mac OS com per Windows i amb ratolí i el trackpad del portàtil.

RESULTATS

En aquest apartat es veuran els resultats de la primera versió de Beer Game, és a dir, la que es va entregar a la PEC3 i els resultats de la segona versió amb els canvis que es van realitzar un cop passada la prova amb els usuaris i implementat el nou grafisme.

VERSIÓ 1

A l'obrir el joc ens trobem davant de l'escenari amb el canvas principal que inclou les eines i els comptadors d'aigua, civada, llúpol i tokens. Aquí l'usuari pot desplaçar-se pel terreny i acostar-se o allunyar-se amb el ratolí tal i com s'ha explicat a apartats anteriors.



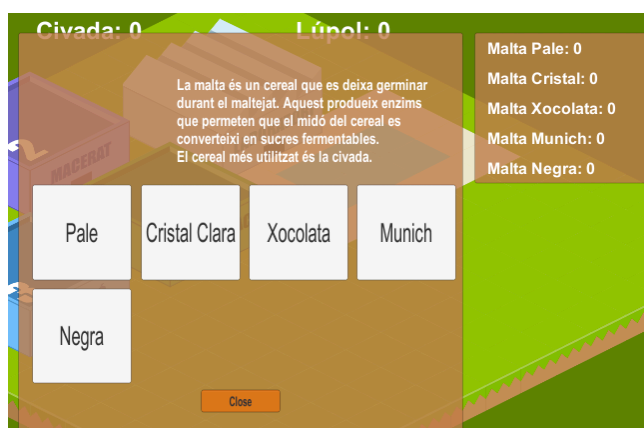
Al seleccionar l'eina de civada o de llúpol, l'usuari pot començar a plantar a sobre del terreny. Al fer clic, l'sprite del terreny canvia al color terra, hi ha dos fases de creixement per a la civada i el llúpol: una de tall i la final que és quan ja es poden recollir. En aquest punt, es pot seleccionar l'eina de la falç i fer clic a sobre de l'element del terreny que es vol recollir. Cada cop que es recull una part de cereal o de llúpol l'usuari obté tokens per gastar al laboratori.

Amb l'eina del cubell es pot clicar a sobre de l'àrea de l'aigua per recollir-la.

Durant tot moment podem observar com es sumen els elements recollits en els comptadors de la part superior de la pantalla

Maltejat o Molta

Al fer clic a sobre de l'edifici de maltejat s'obre el menú on podem crear els diferents tipus de malta per a fer la cervesa. Cada malta es fa a partir d'una



quantitat específica de civada que s'anirà restant del seu comptador corresponent. Durant el procés d'un element no es pot tancar el menú.

A la part dreta es pot veure el magatzem de maltes creades.

Macerat

En el menú de macerat ja es pot escollir quin tipus de cervesa es vol elaborar. L'usuari té dos botons: Imperial IPA i American Stout. En aquest procés es fa servir els tipus de maltes necessàries per a cada cervesa i la quantitat d'aigua. Al fer clic en un d'ells, els ingredients necessaris es resten de l'script GameState.cs. Passats uns segons es suma la quantitat de most al magatzem.



Cocció

Aquí s'agafa el most resultant del procés previ i s'afegeix el llúpul necessari depenent del tipus de cervesa que s'estigui elaborant. Igual que en el menú de macerat, l'usuari té dos botons per escollir la cervesa que vol continuar fent. Un cop acabada la cocció tindrem el material necessari per passar al procés de fermentació.

Fermentació

Fent clic a sobre d'aquest edifici l'usuari obre el menú de fermentació, el pas final per aconseguir la cervesa que es vol elaborar. Igual que en els processos anteriors, es pot escollir entre els dos tipus de cervesa disponibles i es continuarà amb el seu procés restant el producte obtingut de la cocció i afegint el llevat que s'hagi creat prèviament en el laboratori. Un cop acabat el temps necessari es sumen, en aquest cas, les ampolles resultants per la quantitat de most que s'ha generat a l'inici.

Laboratori

Com ja s'ha comentat abans, al laboratori es genera el llevat per a dur a terme la fermentació. Quan l'usuari obre aquest menú pot escollir entre els diferents tipus de llevat disponibles tot i que en aquestes primeres versions només es fa servir el llevat de tipus Ale.

S'ha de recordar que, quan s'està generant un producte en qualsevol dels edificis, els botons del producte en qüestió i el de tancar queden bloquejats.



COMENTARIS DELS USUARIS

Jugabilitat

Durant les proves el joc funcionava correctament, no es va penjar ni va donar cap tipus d'error d'scripts o de qualsevol altre tipus. Durant les sessions de prova es van recollir els comentaris dels usuaris i, en general, com a punt positiu és que el joc i la idea va agradar molt. El fet d'anar plantant la civada o el llúpul i veure com creix els va engrescar a plantar més sobre el terreny.



També va agradar veure quines són les fases d'elaboració de la cervesa i llegir les petites explicacions que es mostren.



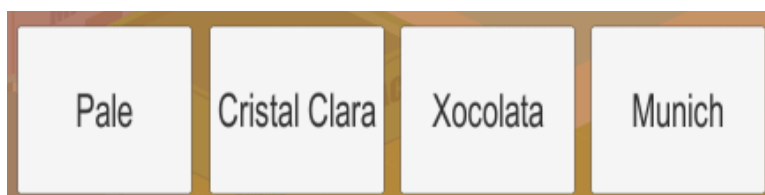
Com a punts negatius hi va haver una opinió unànime de que es necessitava algun tipus de tutorial per saber què s'havia de fer per començar a jugar. Es trobaven davant de la pantalla sense saber per on començar.

Finalment es va demanar que s'identifiqués més el final del procés de fermentació per poder saber la quantitat d'ampolles de cervesa disponibles.

Disseny

Pel que fa al disseny, va agradar l'escenari isomètric ja que els usuaris estaven acostumats a aquest tipus de joc fent servir aquesta vista. Tot i que el joc no estava acabat a nivell gràfic el disseny general els hi va semblar correcte.

Per una altra banda, van trobar a faltar més informació dels elements disponibles dins dels menús de processat. Per exemple, saber quines quantitats i tipus de malta estaven disponibles quan s'obria el menú de macerat i poder relacionar les icones amb la malta que es demanava per a cada most.

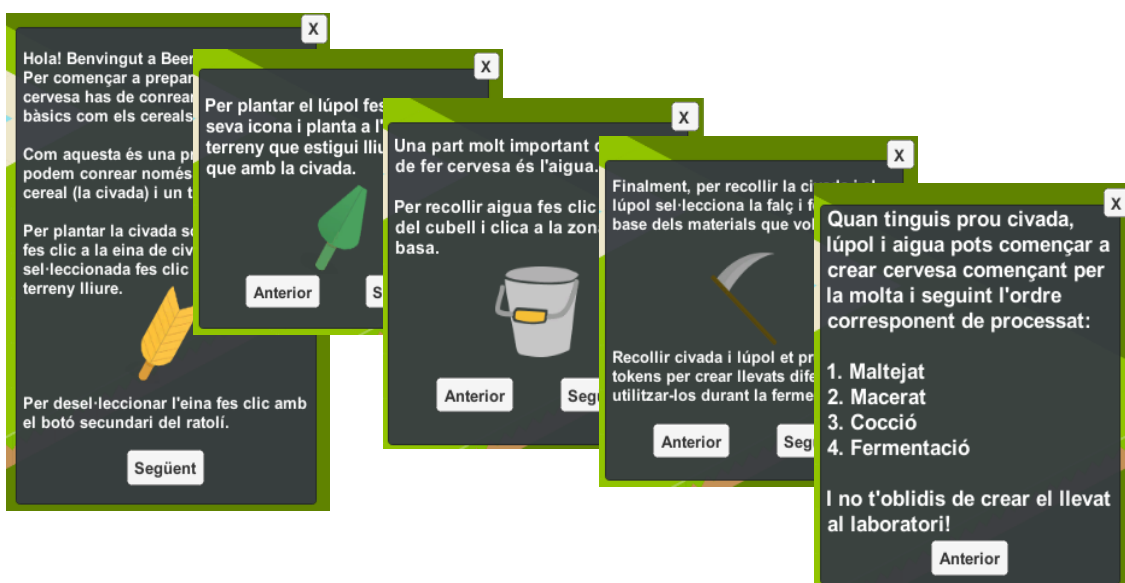


Satisfacció general

Un cop acabada la prova els usuaris van trobar el joc força entretingut pel simple fet de generar la cervesa, conrear els materials i recollir-los. Si que és cert que no hi ha cap objectiu final i que simplement es tracta d'elaborar cervesa però, tot i així, es una bona manera d'introduir-se en el procés i aprendre els aspectes principals.

VERSIÓ 2: INCORPORANT MILLORES

Seguint els comentaris de les proves amb usuaris, s'ha introduït un tutorial a través de finestres i botons on s'explica per on començar. La primera finestra apareix només iniciar-se el joc a la part dreta de la pantalla i l'usuari pot avançar o retrocedir al seu gust. També pot tancar-la amb el botó inclòs a la part superior dreta de la pròpia finestra, d'aquesta manera queda un botó d'ajuda de color taronja just a sota del comptador de tokens que l'usuari pot clicar per tornar a obrir el tutorial.



S'ha inclòs en el procés de fermentació una imatge final a mode de medalla per a que, quan el temps acaba, aparegui dient la quantitat d'ampolles que s'han produït a partir del most generat en el macerat i la coccíó. Per fer desaparèixer aquesta imatge simplement es pot fer clic a sobre d'ella.

També s'ha afegit el magatzem del procés previ per a que l'usuari pugui veure fàcilment la quantitat de material que té i què pot o no pot fer a partir d'ell. D'aquesta manera el magatzem del procés previ queda a l'esquerra de la pantalla i el magatzem del procés actual es troba a la dreta. També s'ha modificat completament el grafisme dels botons per a que resulti més fàcil identificar els tipus de malta i els tipus de cervesa.



A part de les millores aplicades seguint els comentaris dels usuaris, també s'ha canviat el disseny dels comptadors per a que sigui més adient a un joc i no simplement un text a la part superior de la pantalla. D'altra banda s'han incorporat les millores suggerides pels consultors durant el procés d'avaluació però les que no, s'enumeren a continuació.



ASPECTES A MILLORAR PER FUTURES EDICIONS

Hi ha molts elements que bé per falta de temps o bé per manca d'experiència no s'ha pogut implementar en el joc, per tant, en aquest apartat es plantegen primer les millores immediates a tenir en compte per al joc i, segon, les millores a llarg termini com idees d'evolució del joc.

Millores a curt termini

Des del primer moment es va plantejar que l'usuari hagués d'elaborar la cervesa per tal de vendre-la als personatges que apareixerien damunt del terreny. Aquesta fase del joc no s'ha pogut portar a terme per falta d'experiència amb Unity i C# i no hi ha hagut temps per implementar-lo.

Per tant, com a primera millora s'ha d'incloure la fase del bar o botiga i que els personatges apareguin al voltant demanant el tipus de cervesa que volen a través de globus de còmic amb icones característiques de cada una. Aquests personatges arriben a l'escenari, es mantenen allà una estona i finalment marxen, és a dir, que si l'usuari no



s'afanya a donar el tipus de cervesa que vol, no aconseguirà diners. A conseqüència d'això s'ha d'incloure l'element de diners al joc així com nivell d'experiència del jugador. Això es convertirà en diferents millores en el joc.

La següent millora a aplicar és la d'incloure un tipus de comptador o compte enrere mentre s'està processant un element. Aquest pot ser de tipus digital, un cercle que es va omplint o una barra decreixent. Es va intentar incloure aquest element per a la versió final però no es va aconseguir que funcionés correctament.



Una tercera millora per aplicar immediatament és la d'incloure més tipus de cervesa. En aquest moments només s'ofereix l'Imperial IPA i American Stout però n'hi ha de molts més tipus diferents. Combinant aquesta millora amb la de la botiga, es tindria molta més varietat entre els clients i faria el joc menys repetitiu.

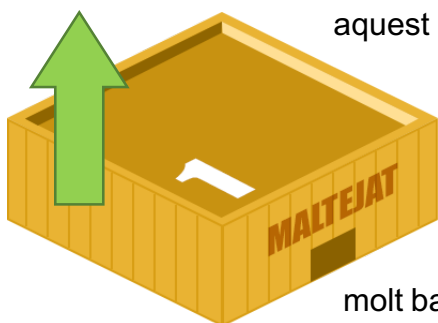
També s'ha de tenir en compte la de crear una pantalla principal d'entrada al joc amb un botó de començar a jugar o de continuar en el cas que ja s'hagi començat una partida. En conseqüència, també s'ha de crear una funció per guardar l'estat

en el que es troba el joc i que l'usuari el pugui recuperar al fer clic en el botó continuar.

Finalment, no hi ha hagut temps d'afegir una música i els diferents efectes de so per a cada interacció. Aquest aspecte s'ha de millorar de manera immediata.

Millories a llarg termini

Una primera millora per aplicar en el futur és la de que l'usuari pugui construir els edificis ell mateix a partir de tokens o diners. De la mateixa manera podria millorar aquest edificis per a que treballessin més ràpid, produïssin més tipus de recursos, etc.



Els magatzems també es podrien millorar de tal manera que en un nivell molt bàsic la capacitat fos molt baixa i aquesta anés augmentant conforme s'augmenta el nivell de l'edifici. Amb aquesta millora s'hauria de pensar d'introduir un magatzem conjunt o individual per guardar la civada, el llúpol, l'aigua i els tokens.

Incloure altres tipus de cereals com ara blat i altres tipus de llúpol, depenent de la cervesa i del moment que s'afegeix aquest en el procés de cocció, pot ser d'un tipus o d'un altre.

Afegir la opció de plantar arbres fruiters o cafè ja que hi ha molts tipus de cervesa a les que es poden afegir aquests tipus d'ingredients.

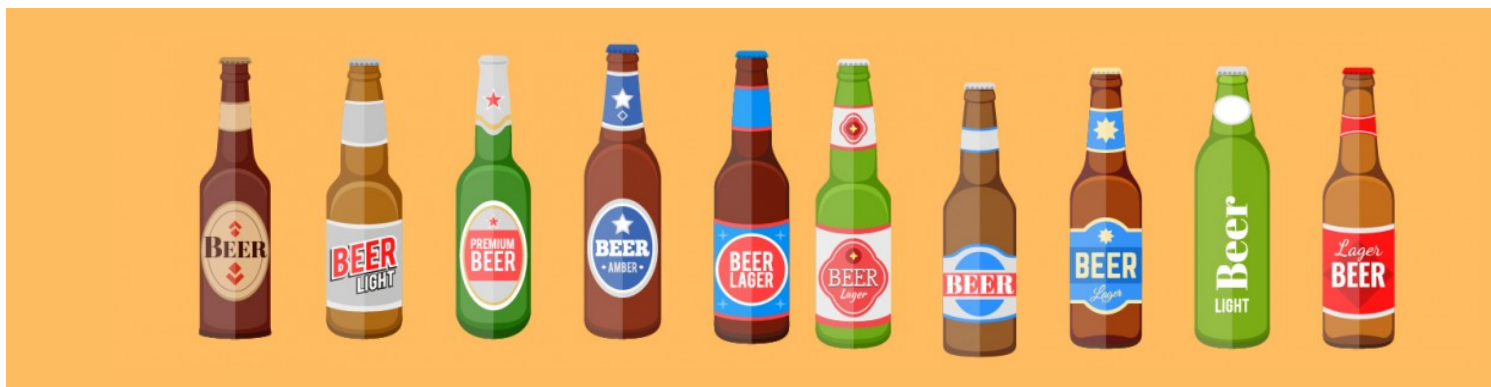
Com a millora a llarg termini també està previst generar animacions per als edificis mentre estan treballant i que el menú no hagi d'estar sempre obert, si no que treballi en segon pla. De la mateixa manera també es farien servir animacions per a les interaccions de l'usuari amb l'escenari com per exemple recollir la civada o el llúpol i que apareguin les icones d'aquests objectes i dels tokens flotant a sobre del lloc on es clica, o bé animacions de monedes sortint del personatge en el moment de donar-li la cervesa que volia.



Com ja s'ha dit, aquesta primera versió és molt bàsica i el procés d'elaboració s'ha simplificat per poder complir el termini d'entrega. Aquest procés es podria complicar una mica més si, per a cada edifici, s'introduïssin diferents mini jocs. Un exemple el tindriem duran la cocció, en realitat el llúpul no es llença tot de cop si no que es llença una part al principi, una altra passats uns minuts i una darrera part quan falten uns minuts per a que acabi el procés (no sempre és així, tot depèn del tipus de cervesa). Per això, es podria fer que el joc avisés a l'usuari que té un marge per afegir el llúpul i, en el cas que no ho faci en aquell temps, el procés falla perdent d'aquesta manera els materials utilitzats.



Finalment, es podria incloure la opció de xarxa social per a que els usuaris es connectessin entre ells per intercanviar o vendre cerveses ja fetes, maltes, llúpols, etc.



CONCLUSIONS

Aquest projecte m'ha ensenyat que és possible dissenyar una primera versió jugable d'un joc a partir d'una idea que feia temps que tenia en ment: la gestió de la producció de cervesa artesana. Malgrat això, i com s'ha comentat, hi ha alguns aspectes que s'han de millorar com ara incloure la fase del bar, que li donaria un valor afegit al joc per a que l'usuari tingués un objectiu clar de perquè està fent la cervesa, i, per una altra banda, incloure molts més productes per incrementar la variabilitat.

Personalment, la realització d'aquest videojoc ha estat una experiència enriquidora. En el moment d'escollir l'àrea del treball final dubtava entre la branca més audiovisual com fer un curtmetratge d'animació o la branca més tècnica com la del videojoc. La primera opció no suposava cap mena de repte per mi ja que porto molts anys dedicant-me al món de l'animació i, per tant, no m'aportaria gaires més coneixements. En canvi, quan vaig veure l'opció de videojocs li vaig donar moltes voltes però al final em vaig engrescar a realitzar aquest tipus de projecte ja que, encara que no aconseguís fer un gran videojoc, sí que seria una manera d'adquirir més coneixements i, el que és més important, fer-ho d'una manera totalment pràctica. Des de molt petit he jugat a videojocs i sempre m'ha interessat aquest sector, el fet d'haver tingut la oportunitat de crear un des de zero, tot i no ser un producte totalment acabat, és una experiència molt gratificant. De totes maneres, aquest treball final de grau no s'acaba aquí, ara que s'ha començat Beer Game hi ha molta feina per fer i aquest anirà evolucionant fins aconseguir un producte complet.



BIBLIOGRAFIA

Aspis, Matan. *6 Top Game Engines in 2017* [en línia]. (2017) <<http://www.discover sdk.com/blog/6-top-game-engines-in-2017>> [Consulta: 2 d'octubre del 2017]

Game Design HQX. (26 de març del 2016). *Unity5 – Planting, Gardening, Harvesting Gameplay* [Arxiu de vídeo].
< <https://www.youtube.com/watch?v=9wuoSx3BjvM>>

Hampson, Tim (2008). *Cerveza, el libro* (2a edició, traducció de: The Beer Book, 352pàgs). Madrid. Ediciones Akal, SA.

Hughes, Greg (2013). *Cómo elaborar cerveza casera* (traducció de: Home Brew Beer, 224pàgs). Barcelona. Ediciones Omega.

Katkoff, Michail. *Mid-Core succès part1: Core Loops* [en línia]. (2013) <https://www.gamasutra.com/blogs/MichailKatkoff/20131024/203142/MidCore_Success_Part_1_Core_Loops.php> [Consulta: 20 d'octubre del 2017]

Lovato, Nathan. *How to perfect your game's core loop* [en línia]. <<https://gameanalytics.com/blog/how-to-perfect-your-games-core-loop.html>> [Consulta: 20 d'octubre del 2017]

VV. AA. (2006). *Diseño de personajes para consolas portátiles. Videojuegos para móviles, sprites y gráficos con píxeles* (traducció de: Character design for Mobile devices. Mobile games, sprites and pixel art, 192pàgs). Editorial Gustavo Gili SL.

GameMaker: Studio [en línia]. (2017) < https://es.wikipedia.org/wiki/GameMaker:_Studio> [Consulta: 2 d'octubre del 2017]

Scripting API [en línia] (8 de d'octubre del 2017)

< <https://docs.unity3d.com/ScriptReference/>>

Unity. (20 d'agost del 2014). *UI Canvas – Unity Official Tutorials* [Arxiu de vídeo].

< <https://www.youtube.com/watch?v=OD-p1eMsyrU>>

Unity (motor de juego) [en línia]. (2017)

< [https://es.wikipedia.org/wiki/Unity_\(motor_de_juego\)](https://es.wikipedia.org/wiki/Unity_(motor_de_juego))> [Consulta: 2 d'octubre del 2017]

Unity User Manual (2017.3) [en línia] (8 de d'octubre del 2017)

< <https://docs.unity3d.com/Manual/index.html>>

Unreal Engine [en línia]. (2017)

<https://es.wikipedia.org/wiki/Unreal_Engine> [Consulta: 2 d'octubre del 2017]