

PARCIAL NOVENBREAssignatura: **Disseny de Software**

Data: 8 de Novembre de 2016

Curs: **2016/2017****Problema (60 punts): Temps 2:00h.**

Es vol especificar el funcionament d'un joc d'aventura mono-jugador, on cada jugador té un nom, una contrasenya, monedes, punts de vida i una imatge (o avatar) associada. Del joc es té informació sobre el nom, la versió, la productora del joc i les plataformes on es pot jugar (PlayStation, PC, XBox, etc.). Com tot joc d'aventura, el joc té diferents nivells que el jugador anirà desbloquejant segons els seus punts d'experiència. El joc mostra un escenari diferent a cada nivell on el jugador pot trobar diferents objectes que pot intercanviar per monedes. Els objectes poden aparèixer en diferents escenaris, però el jugador només podrà agafar-ne un de cada tipus.

A cada nivell, el jugador podrà guanyar punts d'experiència o monedes i perdre vida segons els reptes que vagi passant en el nivell concret. Els nivells com a mínim tenen un repte. El joc ofereix diferents reptes, de tipus tutorial, d'entrenament o de combat. Només amb els de tipus combat el jugador podrà guanyar punts d'experiència i perdre punts de vida, mentre que en els reptes de tipus entrenament guanyarà monedes i punts d'experiència. El sistema en tot moment ha de saber si un jugador ha passat un repte, està jugant-lo o encara el té pendent de passar. Un cop es passa un repte, es marca com repte aconseguit actualitzant els punts de vida, les monedes i els punts d'experiència del jugador i es poden activar nivells més alts, si és el cas.

En el joc també hi han personatges controlats pel propi joc (Non-Player-Characters). Els reptes de tipus tutorial tenen un NPCs que és tutors del jugador, mentre que en els reptes d'entrenament apareixen personatges neutrals propis de cada repte que no poden fer mal al jugador. Finalment en els reptes de combat els NPC's poden ser tant personatges neutrals com enemics. El jugador guanya quan ha aconseguit tots els reptes de tots els nivells sense que el matí, és a dir mentre que els seus punts de vida no siguin zero.

Especifica el Model de Domini, que lliuraràs al final de l'examen junt amb les respostes a les següents preguntes:

1. (20 punts) Omple la següent taula amb els resultats del teu Model de Domini, justificant a cada fila, el mecanisme del model de domini que has fet servir, les classes que estan implicades (les pots dibuixar) i el raonament de la utilitat de la teva solució:

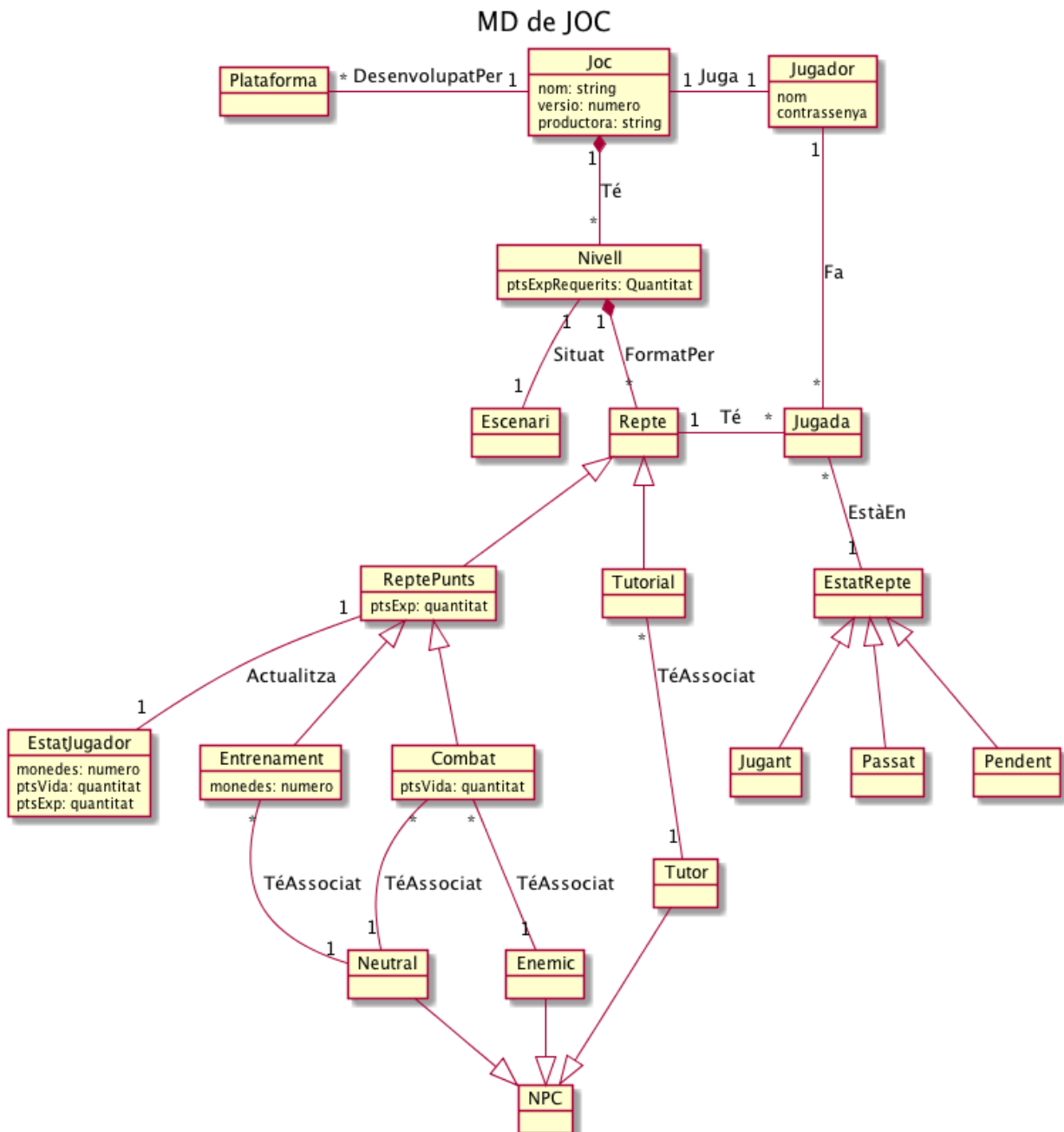
Has utilitzat...? Quantes?	Quines classes estan implicades?	Per a què ho fas servir?
<p>Especificació de classes</p> <p><input checked="" type="checkbox"/> Si</p> <p><input type="checkbox"/> No</p> <p>En cas afirmatiu, quantes vegades?</p> <p>_____1_____</p>	<p>La classe Especificació Objecte i Objecte</p>	<p>Per a poder controlar que un el jugador només agafi un sol tipus d'objecte i en canvi en el joc puguin haver tantes repeticions d'objectes com es vulgui. Es a dir, en el joc podrà haver tantes pilotes com es vulgui (instància d'objecte), però només hi haurà una especificació de pilota (amb la que es farà una associació des del jugador)</p>

Has utilitzat...? Quantes?	Quines classes estan implicades?	Per a què ho fas servir?
<p>Agregació</p> <p><input type="checkbox"/> Si</p> <p><input checked="" type="checkbox"/> No</p> <p>En cas afirmatiu, quantes vegades? _____</p>		
<p>Composició</p> <p><input checked="" type="checkbox"/> Si</p> <p><input type="checkbox"/> No</p> <p>En cas afirmatiu, quantes vegades? _____2_____</p>	<p>Joc està compostat per nivells i Nivell està compostat per reptes</p>	<p>S'ha considerat Joc com un TOT format per nivells. Si només es considera això, és un agregació. Considero que si s'esborressin tots els nivells, el joc s'esborraria</p>
<p>Herència</p> <p><input checked="" type="checkbox"/> Si</p> <p><input type="checkbox"/> No</p> <p>En cas afirmatiu, quantes vegades? _____4_____</p>	<p>S'ha usat herència per a:</p> <ol style="list-style-type: none"> 1. especialitzar Repte en els diferents tipus de reptes (Repte sensePunts (tutorial) i RepteAmbPunts). 2. El RepteAmbPunts l'he especialitzat a la seva vegada en entrenament i combat), 3. Els NPCs s'ha especialitzat en els tipus de NPCs (tutors, neutrals i enemics) 4. L'estat d'una jugada s'ha especialitzat en Pendent, Jugant i Passat 	<p>S'ha usat herència per tres raons:</p> <ul style="list-style-type: none"> - Posar en comú paràmetres dels fills en el pare, com pot ser el cas de RepteAmbPunts que agrupa els reptes que tenen punts - Per a poder detallar associacions concretes dels fills, com per exemple entre la classe Enemic i Combat o l'associació entre tutor i tutorial. Aquestes associacions més detallades no serien possibles si no hagués fet herència del Repte ni dels NPCs - Per a poder detallar concretament en el Model de Domini informació de diferents tipus de classes o situacions, com l'estat que pot ser Pendent, Jugant i Passat

Altres:

2. (20 punts) Dibuixa i justifica les classes conceptuais, les associacions i els atributs del teu Model de Domini que permeten definir els nivells que té el joc i que permeten modelar un moment de la sessió del joc, on el jugador té reptes ja superats, reptes que està jugant i reptes pendents de passar.

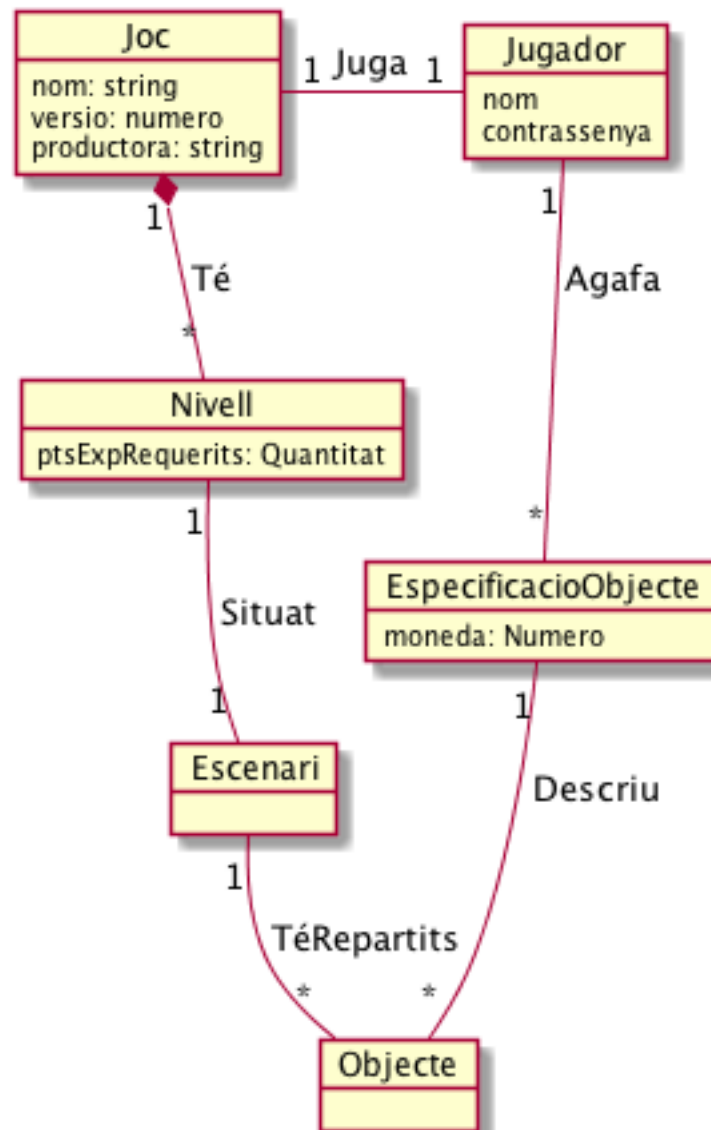
La part del model de domini que correspon a aquesta pregunta és la següent:



S'ha detallat la classe Jugada per a considerar diferent l'estructura del Joc (format per Nivells i Reptes) de la part del procés del Joc (Jugador-Jugada-Estat i Jugada-Repte). L'Estat de la Jugada d'un Repte s'ha especialitzat en tipus "Pendent", "Jugant" i "Passat".

3. (20 punts) Dibuixa i justifica les classes conceptuais, associacions i atributs del teu Model de Domini que permeten restringir, si és possible, que el jugador pugui agafar els objectes només un cop però considerant que els objectes poden estar repetits en diferents escenaris del joc. Raona les teves decisions.

Aquí cal fer una classe `ObjecteEspecificacio` que es relaciona només amb `Jugador` i la classe `Objecte` (que són les diferents instàncies de l'`Objecte` en el Joc). La classe `Objecte` es relaciona amb la classe `Escenari`. Així s'evita que un jugador tingui més d'un objecte del mateix tipus i es mostra directament en el Model de Domini.



MD complet:

MD de JOC

