

PRACTICA PERSONAL MÒDUL 3

EL PROJECTE: Soc desenvolupant amb la llibreria "pygame" un joc amb mòduls classes; bddbdd i carregat més o menys amb el seu entorn Spyder ver. 3.3.4, importar llibreries pygame, (moure per finestres) i split3 (entorn en bdd), creació fitxer .dat per guardar les puntuacions del jugador i fer el seu tractament, segon packed de l'apk, possibles llibreries d'un random, time date, datetime , pygame.locals -- Ver 1.9.6 - PY game - Imatges amb llicència dans d'un packed o carpeta , tipus d'auxiliares png , jpg , gif ...

REQUISIT:

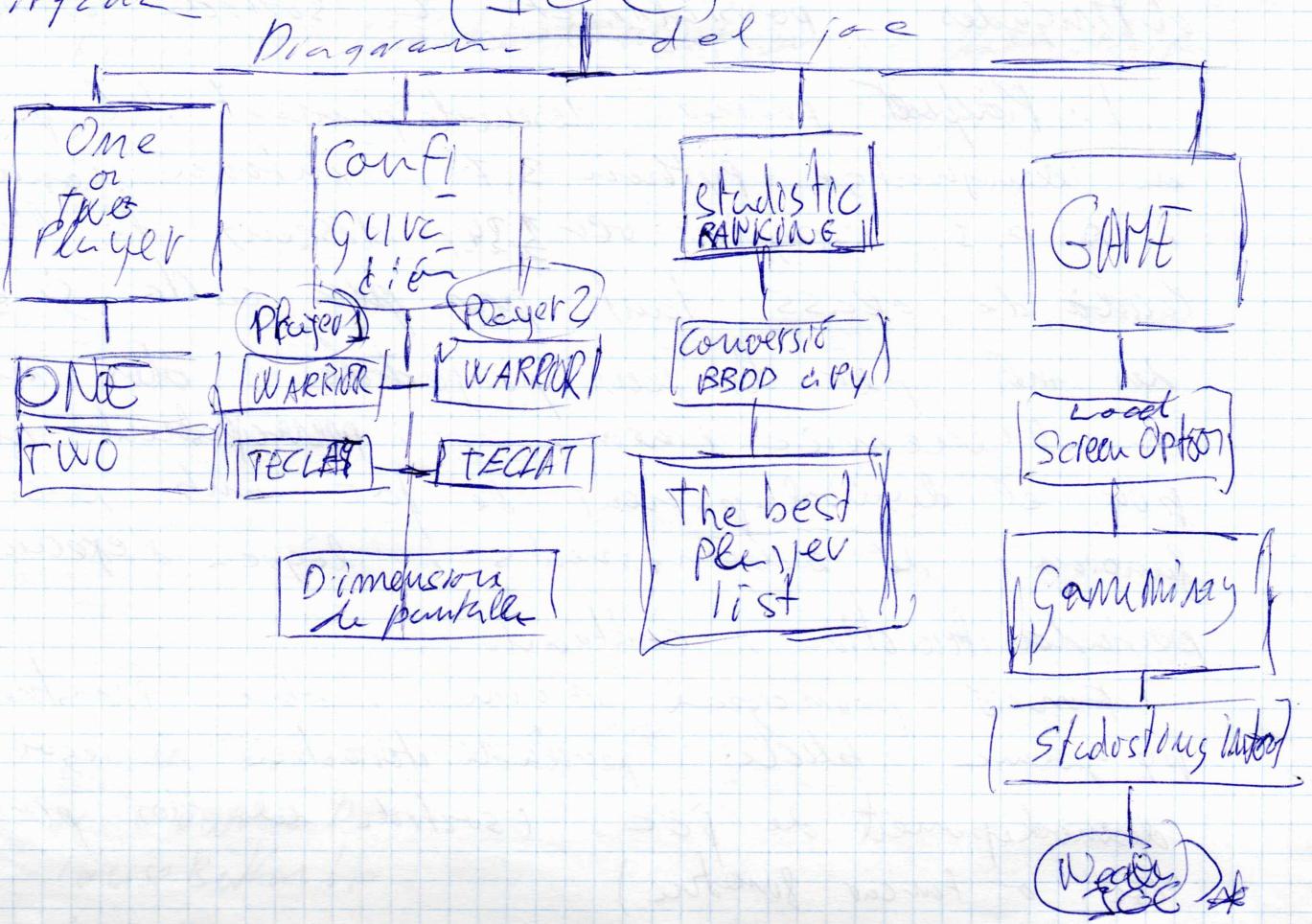
Implementar un joc d'arcade, temàtica Dragon Ball Z, apte per 1 o 2 jugadors amb l'el·lipse de lluitador, teclats de control, dimensions de pantalla i estadístiques de màximes puntuacions obtingudes . possiblitat de sortida de l'ajt amb ESC.

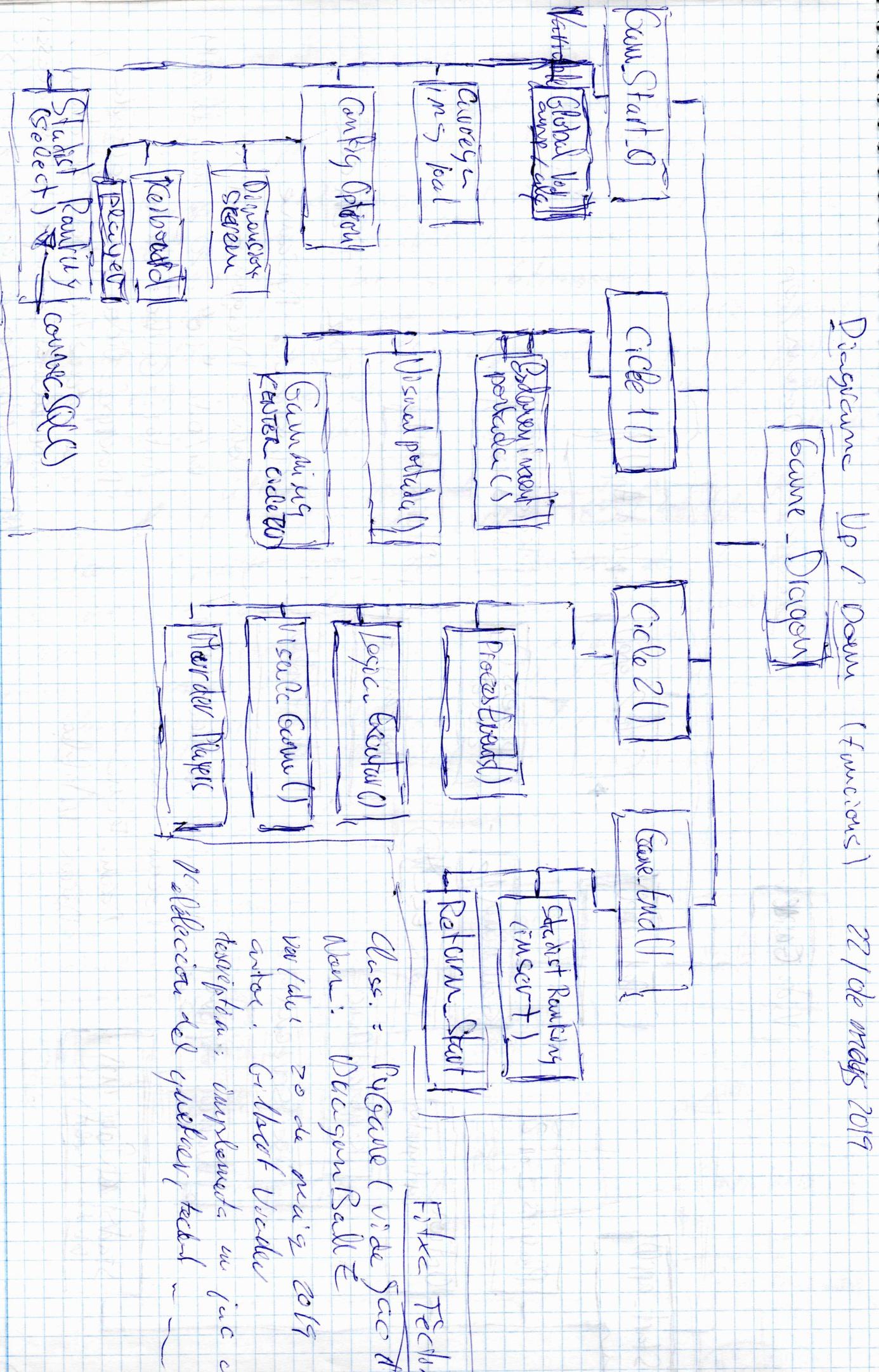
1.- Plaçant primer desenvolupament del programari en lenguatge python 3.7 , entorn Conda 3 o A.P.I. Spyder ver.3.3.4, descent de l'accesibilitat de classes tant per fer clica sencera, des més per els seus projectis , obres més en l'enemic i arena , ; pygame permet una llarga per el desenvolupament de joc amb possibles funcions de esdeveniments , lògica d'execució , portada d'acció , construcció ...

Funció principal main() , obre finestra pygame blida porta d'interacció , segon bucle desenvolupament de joc . (sortida dels dos per rebre "ESC" o ferca finestra)

Segon paquet resultat de les imatges més visibles, que serà accessible des de els formats predefinitos de l'app per mitjans de pàgina "Load" i una redimensionat dinàmicament i proporcionalment de les imatges d'altres i ampliades en la nostra finestra. Imatges adaptades amb transparències del google chrome navegador "google chrome". Formats dels fitxers d'imatge són .png, .jpg, .jpng, .pdf - reservats als drets de distribució sense usos comercials, llicència comunitària - homologació.

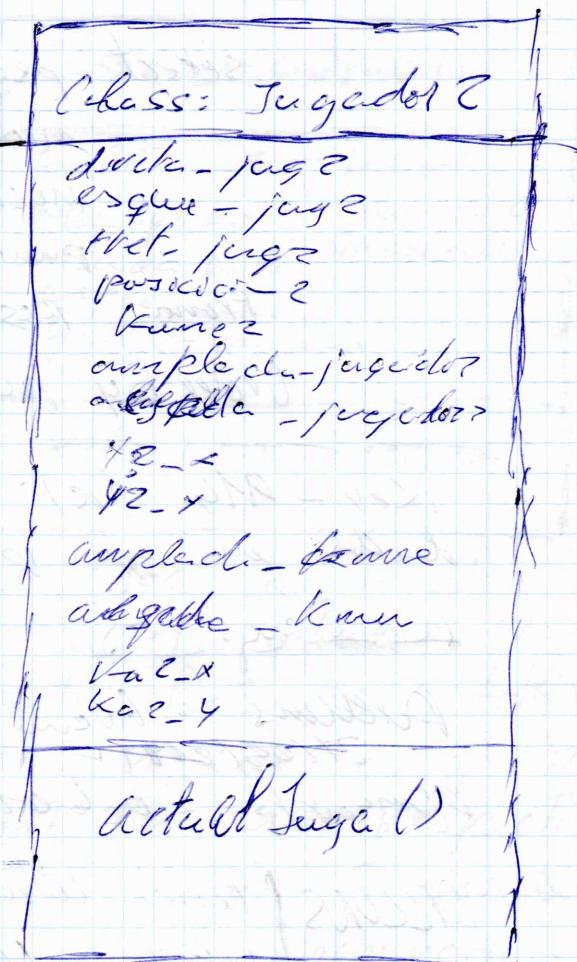
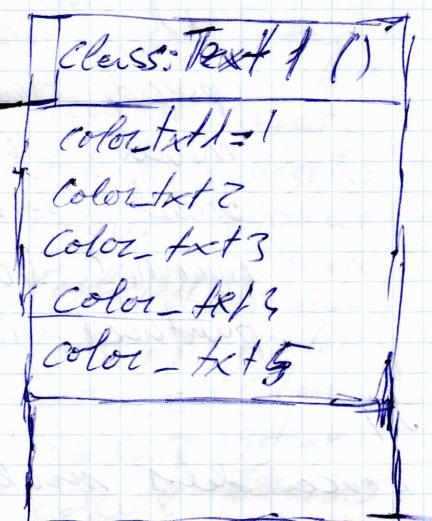
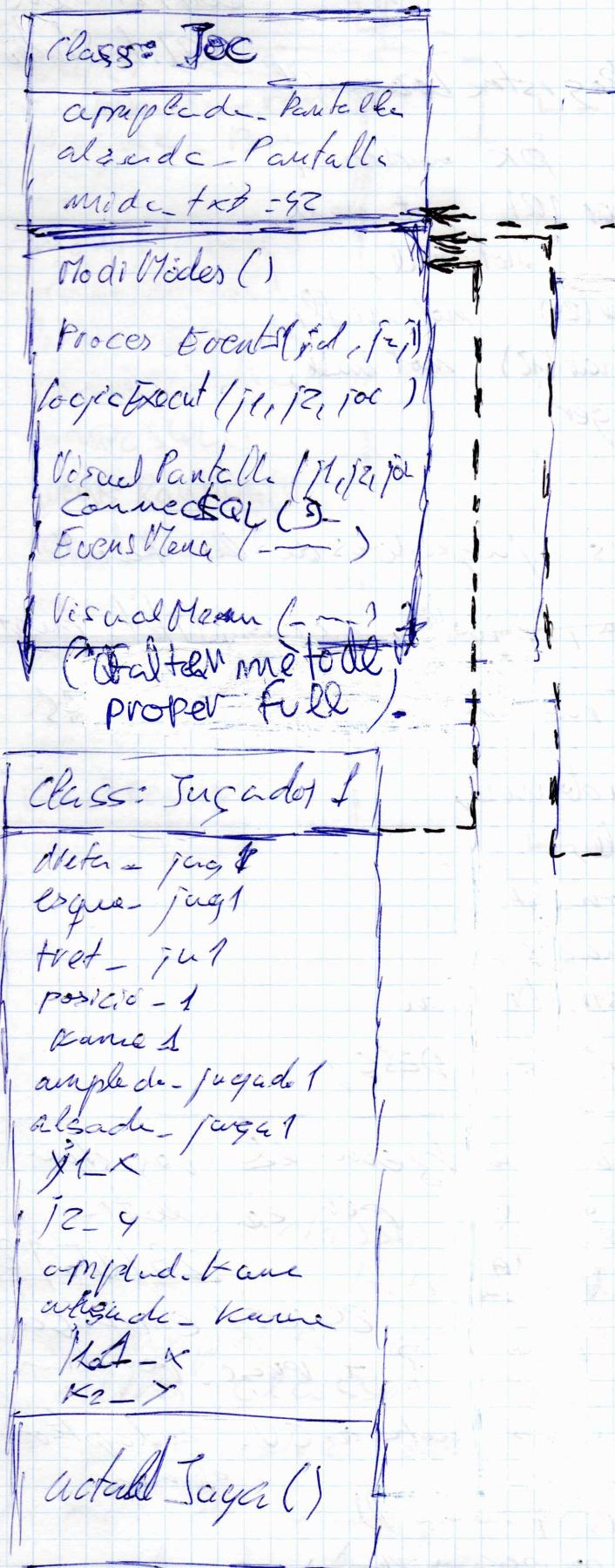
Tercer: últim paquet resultat de la BBDD possible formats .dat & .db editat en Squilte Studio en presentem Squilte Studio com creatiu de la funció que recullirà els punts adquirits amb diverses persones o possessors, dates d'inscripció en la base, codi electrònic, mecanisme, etc... editat en back-end; sortides d'visualització a pàgines amb tractament de les dades i tipografica personalitzada.





Diagrammes UML

Models de classes / Classes



Relacionem les classes Jugador1, Jugador2 i Textos, amb la dependència de la classe principal "Joc", ja que necessitaran del seu constructor els atributs d'ampleda i alsada

Class : Joc()

Super Class
ambulada_Pan
alçada_Pan

modiMides()
procesEvents()
logica_Executor()
visual_Pantalla()
conneccSQLC()
events_Ranking()
visual_Rank()
eventsMenu()
visualMenu()
guine_over()
events_Register()
visual_Register()

Class : CarregaImag()

Name	ball0
Trunks	ball4
Trunks	ball3
Vegete	Vegete
goku	Vegete
goku	goku
goku	goku
Frees	goku
Fons_M8	picolo
goku	goku
goku	goku
goku	goku
a16	
a16v	
sangohan	
sangohan	

Class : Balonrac()

ball | ball0
holes_speed | (carregat)
alsada_ball | select(marge)
anplada_ball
ball_x = random
ball_y = 0

actual_hole (joc, lug)

Tenim la classe principal "Joc" amb els principals atributs de la seva alçada i la seva ampletat de la finestra PyGame , tots els altres classes depenen d'ella i aquestes altres paràmetres per trobar els seus respectius atributs de formes d'objectes , són 5 classes dependent de la classe principal anomenades , Jugador 1 , Jugador 2 , Text1 , Balonrac , CarregaImg.