

AGRAIMENTS

M'agradaria agrair la creació d'aquest projecte a tota la gent del laboratori de gràfics, al tutor, al professor Charles Wüthrich de la Universitat de Weimar i a tota la gent del GILAB (Graphics & Imaging laboratory) de la Universitat de Girona que m'ha donat suport en algun que altre moment.

També vull agrair als meus amics i en especial a la meva parella per la paciència que han tingut que durant un temps els he tingut una mica abandonats.

Finalment m'agradaria donar moltes gràcies als meus pares i a tota la meva família que m'han donat un suport incondicional amb tot el que ha fet falta, sobretot amb els dinars de la meva mare i de la mare de la meva parella que han fet més agradables els dies sencers a la universitat.

1. Introducció	4
1.1. Presentació.....	4
1.2. Una mica d'història	5
1.3. Planificació	7
2. Llibreries físiques.....	8
2.1. Què és una llibreria física?	8
2.2. Elecció de llibreries físiques	9
2.3. Resum de les diferents llibreries físiques	10
2.3.1. Newton Game Dynamics.....	10
2.3.2. NVIDIA® PhysX™ Technology	13
2.3.3. Open Dynamics Engine.....	16
2.3.4. Bullet Physics Library.....	19
2.3.5. Tokamak Physics Engine.....	21
2.3.6. Havok™	23
2.3.7. Physics Abstract Layer	25
2.4. Quadre comparatiu de llibreries físiques	26
2.5. Elecció de la llibreria física.....	27
2.6. NVIDIA PhysX Technology	28
2.6.1. Introducció	28
2.6.2. PhysX by AGEIA	28
2.6.3. PhysX by NVIDIA.....	29
2.7. Integració de llibreries físiques en motors de joc	30
3. Generador de contingut físic.....	31
3.1. Introducció	31
3.2. Carregador des de fitxer SVG	32
3.2.1. Introducció	32
3.2.2. Format SVG	32
3.2.3. Analitzador sintàctic SVG	33

3.2.4. Extracció de dades d'un SVG	34
3.2.5. Creació d'un fitxer SVG	34
3.3. Generació de murs	35
3.3.1. Sistema utilitzat.....	35
3.3.2. Col·locació dels rajols	36
3.3.3. Els fonaments	38
3.3.4. Unions	39
3.4. Disseny	40
3.4.1. Fitxes de cas d'us.....	40
3.4.2. Diagrama de classes	41
3.4.3. Diagrames de seqüència	43
3.5. Integració en el motor de jocs Ogre 3D.....	45
3.5.1. Introducció a Ogre3D	45
3.5.2. Integració de l'aplicació a Ogre3D	45
3.6. Proves d'execució.....	46
3.6.1. Introducció	46
3.6.2. Primer Joc de proves	46
3.6.3. Segon joc de proves	47
3.6.4. Tercera Prova	48
3.7. Limitacions i possibles millores de l'aplicació	49
3.7.1. Limitacions	49
3.7.2. Possibles Millores.....	49
4. Manual d'ús de l'aplicació	50
5. Conclusions.....	51
6. Bibliografia.....	52
Pàgines Web.....	52
Llibreríes físiques.....	52
Diccionaris i enciclopèdies	52
Llibres	53

1. INTRODUCCIÓ

1.1. PRESENTACIÓ

En el món dels videojocs el realisme és un punt molt important a tenir en compte ja que dóna més sensació a l'usuari d'estar immers en el videojoc.

Això passa en part per aconseguir realisme en la dinàmica dels objectes i fer que aquests segueixin les lleis de la física de Newton. Per això s'han desenvolupat diverses llibreries que s'anomenen "motors de física" (physics engines), que empen variables com la massa, la velocitat, la fricció i la resistència del vent.

Els objectius d'aquest projecte seran l'estudi de diferents llibreries físiques existents, la seva comparació i com s'integren en els motors de jocs.

A més a més, la generació de contingut amb comportament que respongui a les funcions definides a aquestes llibreries no és trivial i per aquest motiu també es desenvoluparà una aplicació per generar murs de forma semiautomàtica que respongui a impactes.

Per assolir aquests objectius caldrà:

- Comparar els cossos rígids, unions i funcionament en general de diferents llibreries físiques: Newton Game Dynamics, NVIDIA PhysX Technology, Open Dynamics Engine, Bullet Physics Library, Tokamak Physics Engine i Havok.
- Implementar una aplicació que donant-li una imatge en planta d'una paret o conjunt de parets en format vectorial i les mides d'un maó, generi murs que puguin reaccionar de forma adequada quan rebin l'impacta d'una massa determinada.

L'aplicació s'implementarà en C++ i amb l'entorn de desenvolupament Microsoft Visual Studio 2005. La visualització serà amb OpenGL.

1.2. UNA MICA D'HISTÒRIA

Des de l'aparició dels ordinadors el món dels gràfics i videojocs ha estat en constant desenvolupament i expansió.

Un dels primers jocs que es va desenvolupar va ser l'OXO (Veure Fig. 1.1), cap a l'any 1952, era igual que el joc del tres en ratlla però per ordinador. Aquest joc disposava d'uns gràfics bastant escassos i l'únic que mostrava per pantalla era text. Tampoc va ser gens comercial ja que es va desenvolupar per l'ordinador EDSAC que era únic i estava situat a l'Universitat de Cambridge.

```
9 8 7      NOUGHTS AND CROSSES
6 5 4      BY
3 2 1      A S DOUGLAS, C.1952

LOADING PLEASE WAIT...

EDSAC/USER FIRST (DIAL 0/1):1
DIAL MOVE:
```

Fig. 1.1: OXO (Alexander S., 1952)

Avui en dia això no seria res però, en aquell moment, va ser un gran avanç ja que no es buscava tenir una bona qualitat gràfica sinó que es volia aconseguir la interacció entre la màquina i l'usuari. El fet de poder competir l'ordinador contra l'usuari va ser el pas important i a partir d'aquí ja es podia plantejar de crear uns gràfics que no fos només text sinó que mostrés ja alguna imatge encara que fos amb dos colors.

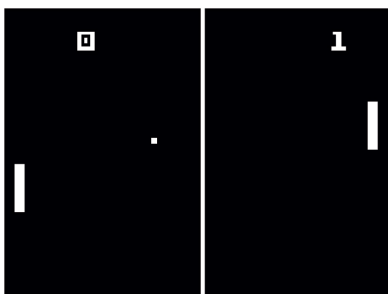


Fig. 1.2: PONG (Atari, 1972)

El següent joc que hi va haver, el qual ja va ser molt més comercial i amb uns gràfics millorats, va ser el PONG (Veure Fig. 1.2), desenvolupat per Atari l'any 1972. Aquest joc era una simulació del tenis taula en el qual es podia jugar contra la màquina o contra un altre jugador, només de començar es veien dos marcadors, dos pales, una pilota i una línia divisòria que separava el terreny en dos. Aquest joc fou molt estès gràcies a que va sortir per ordinadors, consoles de joc i estava en molts màquines recreatives.

En aquest cas no només es tractava de text sinó que ja es van incorporar imatges animades interactives tot i que només amb dos colors i en 2D degut a que el maquinari de l'època no permetia més.

Més endavant, a mesura que s'anava millorant el hardware, va ser possible afegir varietat de colors en els videojocs. Al principi n'hi havia pocs però cada vegada va ser possible incrementar més la quantitat juntament amb la resolució de la pantalla. En aquesta fase hi començava a haver una gran diversitat de jocs i eren més assequibles per al públic en general perquè el preu dels ordinadors ja no era tan elevat com al principi.

Un dels molts jocs destacables d'aquesta època és el Bomberman (Veure Fig. 1.3) desenvolupat per Hudsonsoft al 1983. Jocs com aquest, a part dels gràfics, incorporaven so, tot i que era a 8 bits i reproduïts normalment a través d'un altaveu intern.



Fig. 1.3: Bomberman (Hudsonsoft, 1983)

Arribats a aquest punt els jocs disposaven de so, un sistema d'intel·ligència artificial i uns gràfics que necessitaven ser millorats donat que la quantitat de colors no era gaire elevada i a més eren en 2D.



Fig 1.4: Day of the Tentacle (Lucas Arts, 1993)

En aquest moment doncs, la tendència va ser desenvolupar gràfics en 3D. Al principi era complicat per la potència de les màquines però es van començar a desenvolupar jocs en 2D simulant l'eix Z i així intentar semblar jocs en 3D. Un dels jocs representatius d'aquest estil és el Day of the Tentacle (Veure Fig. 1.4) desenvolupat per Lucas Arts l'any 1993.

Pocs anys després però, va sortir un dels primers jocs en 3D que va causar més sensació en el públic. Això va ser a l'any 1996 amb el Super Mario 64 (Veure Fig. 1.5) que sortia de les mans de Nintendo juntament amb la consola de joc a què anava dirigit, la Nintendo 64. Aquest joc començava a ser un 3D tal com l'entenem avui en dia i inclús renderitzava ombres en alguns objectes.

Malgrat haver arribat a aquest nivell de gràfics encara no hi havia prou realisme en aquest aspecte degut a que faltaven efectes tals com reflexions, refraccions, ombres, penombres i textures.

Aquest és un dels camps que s'ha decidit dedicar més esforços en els últims anys fins a aconseguir uns resultats sorprenents tant en el camp dels videojocs com en el del cinema, o la simulació precalculada, entre d'altres.



Fig. 1.5: Super Mario 64 (Nintendo, 1996)

Tot i aquest avanç s'ha vist que no és suficient i que de fet els usuaris el que volen a part de més realisme gràfic és més realisme en general. Per aconseguir això últimament s'intenta treballar fort en tota una sèries de camps molt importants com ara la intel·ligència artificial, els efectes sonors, la simulació física o la interactivitat de l'usuari amb el joc.

Avui en dia la tendència que està prenent una gran part dels videojocs és a no centrar tant tota l'atenció cap al jugador sinó a centrar-la més cap al món on està immers. La intenció sol ser aconseguir un món on aquest pugui funcionar sense ell i que tot el que va succeint es vagi acumulant en aquest món. Per exemple, si en una banda del món un nen llencés una pilota de beisbol i trenqués el vidre d'una casa, aleshores quan el personatge anés en aquesta casa es trobaria el vidre trencat. Això no només ha de passar amb aquest cas sinó que ha de ser així amb tota la multitud de fets que poden succeir en el nostre món de joc.

Un altre estil de joc que cada vegada pren més força és el "serious gaming". Aquests són jocs que van més enllà del pur entreteniment, poden ser per entrenar a bombers, militars o altra gent, també per fer simulacions de transaccions, demolicions o avions o poden servir en altres camps no relacionats amb l'entreteniment. Un dels exemples més coneguts avui en dia pot ser l'ús del Microsoft Flight Simulator X (Veure Fig. 1.6) en una assignatura de l'Enginyeria Tècnica Aeronàutica.



Fig 1.6: Microsoft Flight Simulator X (Microsoft, 2006)

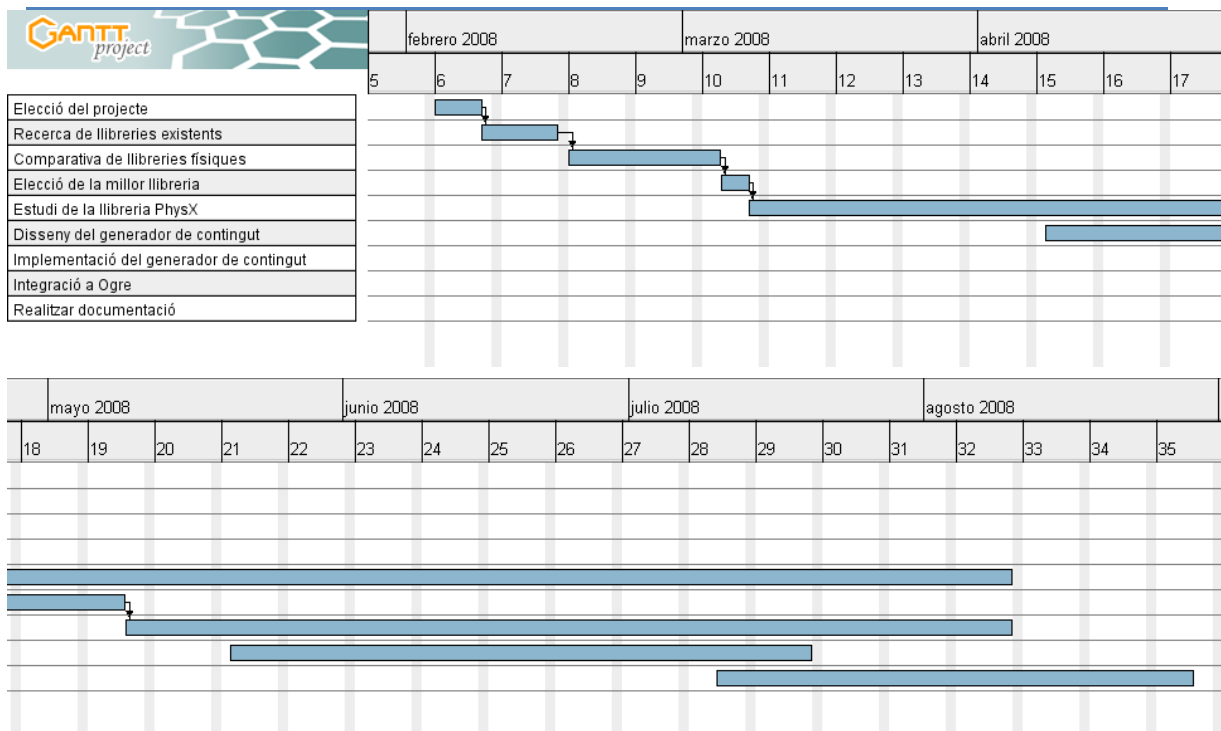
Per aquests últims estils de joc és a on juga una gran importància la simulació física per així poder aconseguir la màxima similitud amb la realitat.

1.3. PLANIFICACIÓ

La planificació del projecte s'ha fet amb el programa Gantt Project.

Les feines s'han estructurat de la següent manera:

Nom	Inici	Fi
Elecció del projecte	4/02/08	9/02/08
Recerca de llibreries existents	9/02/08	17/02/08
Comparativa de llibreries físiques	18/02/08	5/03/08
Elecció de la millor llibreria	5/03/08	8/03/08
Estudi de la llibreria PhysX	8/03/08	10/08/08
Disseny del generador de contingut	8/04/08	9/05/08
Implementació del generador de contingut	9/05/08	10/08/08
Integració a Ogre	20/05/08	20/07/08
Realitzar documentació	10/07/08	29/08/08



2. LLIBRERIES FÍSQUES

2.1. QUÈ ÉS UNA LLIBRERIA FÍSICA?

Una llibreria física és un conjunt d'eines que permeten simular la física de Newton en una aplicació. D'aquesta manera, durant la simulació, en tot moment es compleixen sempre les 3 lleis de Newton:

- 1a llei:** Tot sistema físic aïllat, és a dir, en què no actua damunt d'ell cap força exterior, roman en el seu estat de repòs o de moviment rectilini uniforme.
- 2a llei:** La variació de l'estat de repòs o de moviment rectilini i uniforme d'un sistema físic és proporcional a la força exterior que se li aplica, i es manifesta en la mateixa direcció que la força ($F = m \cdot a$).
- 3a llei:** Tota acció (força) aplicada a un sistema físic s'oposa una reacció igual i de sentit contrari. També conegut com a principi d'acció i reacció.

Per aconseguir millorar el realisme a més a més de complir-se aquestes tres lleis es poden definir tota una sèrie de paràmetres per cada cos com són la densitat, el coeficient de fregament estàtic i dinàmic o el coeficient de restitució.

A més, una llibreria física permet crear unions de tot tipus entre els cossos per així restringir-los el moviment que poden realitzar. Per crear algunes unions com són les d'un vehicle o les d'un ragdoll¹ solen disposar d'eines per crear aquests cossos enters amb les unions ja creades.

Algunes llibreries poden simular els efectes de la roba, camps de força o el comportament de cossos tous com podria ser un matalàs o un coixí. També poden simular partícules ja siguin sòlids o fluids.

Una de les limitacions que presenten avui en dia les llibreries físiques són que no poden simular la física einsteiniana ni altres branques de la física com són la física quàntica, la física atòmica o la física nuclear.

¹ Unió d'esferes, càpsules i altres volums geomètrics per a simular una forma humana amb les seves articulacions

2.2. ELECCIÓ DE LLIBRERIES FÍSiques

Per escollir quina llibreria s'utilitzarà per l'aplicació s'han analitzat i s'han comparat classificant-les segons les següents característiques:

- Cossos rígids i simulació física
 - Tipus de cossos que es poden crear juntament amb les propietats que poden tenir, diferents tipus d'unions i comportaments i els mètodes de simulació que es poden utilitzar.
- Control de personatge
 - Eines disponibles per controlar el personatge ja sigui per pujar escales, caminar o altres aspectes relacionats.
- Vehicles
 - Eines i funcions que faciliten la creació de vehicles.
- Optimitzacions
 - Optimitzacions disponibles de la llibreria ja sigui segons la plataforma, els fils d'execució o altres possibles aspectes.
- Simulació de partícules
 - Possibilitat per simular efectes de les partícules ja siguin de sòlids o de líquids.
- Roba
 - Possibilitat per simular efectes de la roba ja siguin trencaments o doblecs i simulació d'efectes de cordes.
- Cossos tous
 - Possibilitat per simular cossos tous (flams, coixins, neumàtics, ...).
- Camps de força
 - Possibilitat per simular camps de força (vent, corrent marítim, ...).

La informació utilitzada per a realitzar el resum de cada llibreria física s'ha extret de la pàgina web, dels manuals, dels tutorials i dels registres de versió de cadascuna. És per aquest motiu que en algunes la informació és molt més extensa que en d'altres depenent de l'interès que té cada una per promocionar-se i d'informar de les seves possibilitats.

2.3. RESUM DE LES DIFERENTS LLIBRERIES FÍSQUES

2.3.1. NEWTON GAME DYNAMICS V1.53 (25/05/2006)



2.3.1.1. PRESTACIONS DE LES QUALS DISPOSA

2.3.1.1.1. COSSOS RÍGIDS I SIMULACIÓ FÍSICA

- Cossos rígids (estàtics i dinàmics)
- Diferents tipus de primitives de col·lisió:
 - Esfera
 - Caixa
 - Col·leccions d'esferes i caixes
 - Elevació de terreny
 - Malles poligonals
 - Malles definides per l'usuari
- Diferents tipus d'unions:
 - Esfèriques
 - Motoritzades
 - Definides per l'usuari
- Diferents comportaments per les unions:
 - De molla
 - D'amortidor
 - Normal
- Diferents tipus de restriccions per les unions:
 - De tirabuixó
 - De frontissa
 - De colissa
- Suport per el control i la creació de ragdolls
- Suport per a dos models de fricció diferents
- Detecció de col·lisions contínua
- Funcions de llançament de rajos
- Permet augmentar precisió perdent velocitat i viceversa
- Gestió de l'escena

2.3.1.1.2. CONTROL DE PERSONATGE

- Suport per el control de personatge

2.3.1.1.3. VEHICLES

- Funcions per la fàcil implementació de vehicles

2.3.1.1.4. OPTIMITZACIONS

- No especifica optimitzacions específiques per a diferents plataformes ni per a múltiples fils d'execució

2.3.1.1.5. SIMULACIÓ DE PARTÍCULES

- No té suport per a la simulació de sistemes de partícules

2.3.1.1.6. ROBA

- No té suport per a la simulació d'objectes de roba

2.3.1.1.7. COSSOS TOUS

- No té suport per a la simulació de cossos tous

2.3.2.1.8. CAMPS DE FORÇA

- No té suport per a la creació de camps de força

2.3.1.2. ALTRA INFORMACIÓ D'INTERÈS:

- Conté documentació de cada funció en detall
- Conté manuals d'aprenentatge
- Està disponible per Windows, Mac i Linux
- És una llibreria de poc pes, ràpida i fàcil d'utilitzar
- És una llibreria fàcilment integrable amb el projecte el qual es treballa
- Disposa de fòrums actualitzats per poder compartir dubtes i problemes amb altres desenvolupadors
- La llibreria és gratuïta amb llicència pròpia però el codi no és obert

2.3.1.3. RESUM:

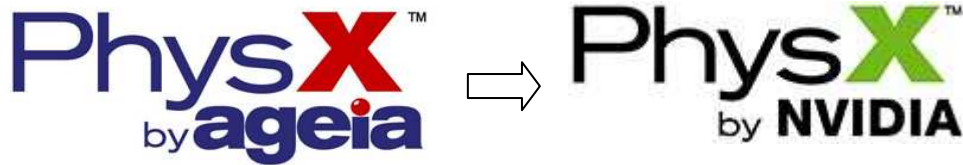
Pel que fa a les prestacions d'aquesta llibreria física estan una mica limitades, de totes maneres per el nostre cas està bé ja que tampoc es necessiten moltes prestacions i això ajuda a que siguin lleugeres, senzilles i fàcils d'utilitzar.

El fet que no disposi de simulació de partícules, efectes de roba, creació de cossos tous i camps de força no és cap impediment ja que no està previst utilitzar cap d'aquestes prestacions.

Cal destacar que aquesta llibreria està disponible per Windows, Mac i Linux, fent més fàcil el desenvolupament multi plataforma tot i que només per a PC. A més disposa de documentació en detall de cada funció la qual facilita molt més el desenvolupament de les aplicacions realitzades juntament amb els manuals d'aprenentatge i els fòrums de consulta de què disposa.

El lloc principal on s'ha de parar més atenció és en la data de l'última versió de la llibreria ja que es pot veure que va sortir fa més de dos anys. Aquest fet podria ser molt problemàtic en cas de trobar algun error en alguna funció, no es podria saber quan seria la pròxima versió si és que arribés a sortir mai. Si es vol trobar positivisme en aquest fet cal tenir en compte que, si no es tragués cap actualització més, aleshores sempre es tindria l'aplicació adaptada a l'última versió de la llibreria.

Per contrarestar el fet de que l'última versió és bastant antiga es pot destacar els fòrums de consulta que es posen a disposició dels desenvolupadors que sí estan al dia.



2.3.2.1. PRESTACIONS DE LES QUALS DISPOSA

2.3.2.1.1. COSSOS RÍGIDS I SIMULACIÓ FÍSICA

- Cossos rígids (dinàmics, estàtics i cinemàtics)
- Diferents tipus de primitives de col·lisió:
 - Esfera
 - Caixa
 - Càpsula
 - Plans
 - Elevació de terreny
 - Formes convexes
 - Malles triangulars
- Diferents tipus d'unions:
 - Esfèriques
 - De revolució
 - Prismàtiques
 - Cilíndriques
 - Fixes
 - A distància
 - De politja
- Suport per a la creació i edició de ragdolls amb unions
- Models de fricció i materials com a suport al comportament de les col·lisions a la superfície dels objectes
- Detecció de col·lisions contínua
- Llançament de rajos, recorreguts, disparadors i tests de solapament en la detecció de col·lisions
- Permet activar i desactivar la detecció de col·lisions segons la definició d'uns grups i uns filtres de col·lisió
- Ofereix la possibilitat d'un sol sentit d'interacció (El cos 'X' pot afectar al cos 'Y' però no viceversa)
- Notificació de contactes el qual permet als desenvolupadors activar altres accions quan un objecte específic interactua o modifica el comportament de varis contactes

2.3.2.1.2. CONTROL DE PERSONATGE

- Permet pujar automàticament petits escalons i escales
- Disposa de definició de parts on es pot caminar i a on no. Aleshores el moviment es para automàticament quan s'entra a una part del nivell on no es pot caminar
- El control del personatge està inclòs al codi font i per tant permet al desenvolupador fer a mida el comportament per adaptar-se a les diferents necessitats

2.3.2.1.3. VEHICLES

- Simulació de vehicles per llançament de rajos com si fossin un únic cos rígid
- Disposa de rodes de forma esfèrica amb un sofisticat model de fricció per els neumàtics
- La suspensió dels cotxes està basada en unions, disposa d'un sistema de geometries complexes de direcció i suspensió
- La simulació és lleugera

2.3.2.1.4. OPTIMITZACIONS

- Suport de hardware per PhysX-Ready GeForce Processors (Sèrie 8, Sèrie 9 i Sèrie 200)
- Suport de hardware per AGEIA PhysX Accelerator
- Posseeix un control molt acurat del control de múltiples fils d'execució
- Suport per passos asíncrons
- Suport per l'anàlisi de rendiment (AgPerfMon)
- Optimitzacions per a PS3 i Xbox 360

2.3.2.1.5. SIMULACIÓ DE PARTÍCULES

- Conté dos models de simulació de partícules diferents:
 - Smoothed Particle Hydrodynamics
 - Simple
- Ofereix a l'usuari de controlar cada partícula de fluid individualment
- Ofereix un i dos sentits d'interacció amb cossos rígids, roba i cossos tous.
- Permet la simulació d'explosions i efectes de restes de l'explosió.

2.3.2.1.6. ROBA

- Permet la simulació d'objectes de roba
- Suporta les col·lisions que pot tenir un objecte de roba amb ell mateix
- Simulació dels trencaments i la pressió a la roba
- Opció 'Metal Cloth', la qual permet simular deformacions com poden ser a un barril, a la carrosseria d'un cotxe o a portes metàl·liques

2.3.2.1.7. COSSOS TOUS

- Permet la creació de cossos tous
- Disposa de NVIDIA PhysX Viewer el qual permet la fàcil creació de cossos tous.
- Suporta les col·lisions que pot tenir un cos tou amb ell mateix

2.3.2.1.8. CAMPS DE FORÇA

- Disposa de camps de força amb diferents formes:
 - Esfera
 - Càpsula
 - Caixa
 - Malles convexes
- Disposa de funcions per el control de la quantitat i la direcció de la força

2.3.2.2. ALTRA INFORMACIÓ D'INTERÈS:

- Llibreria física molt potent i ràpida que s'ajuda amb el hardware a través d'un processador de la targeta dedicat a fer els càlculs físics
- Pel que fa a la llicència hi ha la versió gratuïta i la versió de pagament.
 - Versió gratuïta: No tens el codi obert i només et permet desenvolupar per PS3 i per a PC però sense l'obligatorietat del hardware.
 - Versió de pagament: Proporciona el codi del kit de desenvolupament i permet desenvolupar per Xbox 360.

2.3.2.3. RESUM:

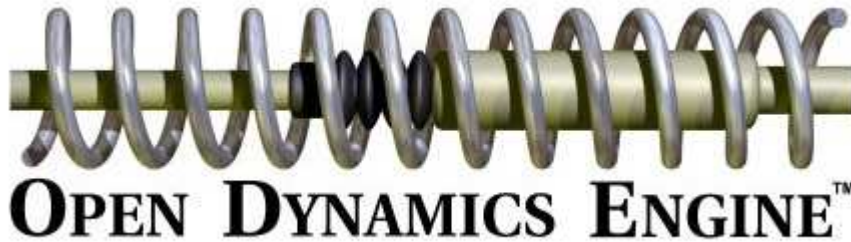
La llibreria PhysX és molt completa ja que permet simular aspectes molt interessants tals com efectes de roba, cossos tous, camps de força i sistemes de partícules i fluids. Tot i l'ampli ventall de simulació que ofereix cal destacar que l'especialització en cadascun dels camps no hi manca gens.

Un dels aspectes més destacables d'aquesta llibreria és el fet de que la simulació física es pot realitzar a través d'un hardware específic per així incrementar la velocitat de simulació. El hardware pot ser un AGEIA PhysX Accelerator o una targeta NVIDIA GeForce de la sèrie 8, 9 o 200. El segon tipus de hardware sol ser bastant estès als ordinadors actuals i és per això que cal remarcar la simulació per hardware.

A més a més, el fet que disposi de versió gratuïta, tenint en compte tot el que té, està molt bé encara que aquesta versió no disposi del codi font.

Finalment cal dir que està bastant ben documentat i disposa d'exemples i fòrums de consulta per Internet.

Alguns jocs que s'han fet amb aquesta llibreria són: Unreal Tournament 3, Age of Empires III, Tom Clancy's Rainbow Six Vegas o Rail Simulator entre d'altres.



2.3.3.1. PRESTACIONS DE LES QUALS DISPOSA

2.3.3.1.1. COSSOS RÍGIDS I SIMULACIÓ FÍSICA

- Cossos rígids amb distribució arbitrària de la massa
- Diferents tipus de primitives de col·lisió:
 - Esfera
 - Caixa
 - Cilindre tapat
 - Pla
 - Raig
 - Malla triangular
 - Elevació de terreny
- Diferents espais de col·lisió:
 - Simple
 - Taula de hash multi resolució
 - Arbre quadràtic
 - 'Sweep and Prune'
- Diferents tipus d'unions:
 - Esfèrica/còncava
 - Amb restricció de frontissa i frontissa-2
 - Fixes
 - De motor angular
 - Angulars
 - Prismàtiques
 - Amb restricció de pistó
 - De pla 2D
- Un model de contacte i fricció el qual està basat en el Dantzig LCP encara que ODE implementa una aproximació més ràpida del model de fricció de Coloumb
- Un mètode de simulació el qual les equacions del moviment són derivades d'un model basat en la velocitat dels multiplicadors de Lagrange

2.3.3.1.2. CONTROL DE PERSONATGE

- No té control de personatge integrat

2.3.3.1.3. VEHICLES

- No disposa de fàcil implementació de vehicles

2.3.3.1.4. OPTIMITZACIONS

- Optimitzacions específiques de cada plataforma

2.3.3.1.5. SIMULACIÓ DE PARTÍCULES

- No té suport per la simulació de sistemes de partícules

2.3.3.1.6. ROBA

- No té suport per la simulació d'efectes de roba

2.3.3.1.7. COSSOS TOUS

- No té suport per a la simulació de cossos tous

2.3.3.1.8. CAMPS DE FORÇA

- No té suport per a la simulació de camps de força

2.3.3.2. ALTRA INFORMACIÓ D'INTERÈS:

- És una llibreria física bona per la simulació de cossos rígids articulats
- Està pensada per ser usada interactivament o en temps real
- La llibreria dóna més importància a la rapidesa i a l'estabilitat que a una física acurada
- ODE utilitza contactes durs a l'hora de detectar la col·lisions de dos cossos enlloc d'utilitzar altres mètodes com els marges que poden fer que no siguin tant precisos i més propensos a errors
- Utilitza un sistema de detecció de col·lisions integrat però amb l'opció d'utilitzar-ne un de propi si es vol
- És una llibreria de codi implementada en C++ oberta sota les llicències GNU Lesser General Public License i BSD-style license

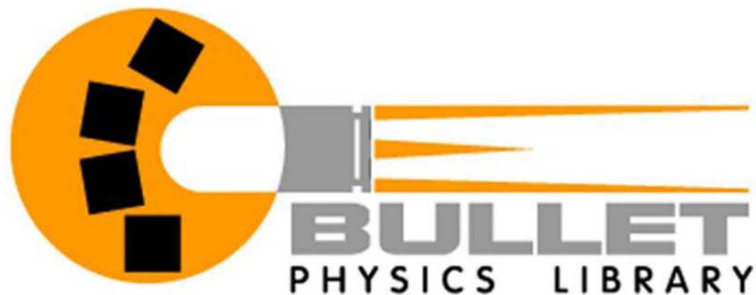
2.3.3.3. RESUM:

Aquesta llibreria està molt especialitzada en les unions entre cossos i les col·lisions que poden tenir entre ells. Per altra banda però, no permet la simulació de roba, cossos tous, camps de força ni simulació de partícules. A més a més no disposa de control de personatge ni tampoc una implementació senzilla de vehicles.

Tot això en principi no hauria de ser cap obstacle ja que el que s'utilitzarà més en aquest projecte és el tema de les unions i amb aquest aspecte està molt especialitzat. A més l'altre necessitat que es té

és la bona detecció de col·lisions i aquesta llibreria està molt bé ja que utilitza un sistema de detecció de col·lisions molt més precís que la majoria de llibreries.

Pel que fa a documentació, n'hi ha però no és molt completa. A més no té fòrum per dubtes tot i que el que sí té és un tutorial i una àrea Wiki.



2.3.4.1. PRESTACIONS DE LES QUALS DISPOSA

2.3.4.1.1. COSSOS RÍGIDS I SIMULACIÓ FÍSICA

- Cossos rígids (estàtics, dinàmics i cinemàtics)
- Diferents tipus de primitives de col·lisió:
 - Caixa
 - Esfera
 - Càpsula
 - Cilindre tapat
 - Con
 - Col·leccions d'esferes
 - Malla convexa
 - Malla còncava
 - Elevació de terreny
- Diferents tipus d'unions:
 - Esfèrica/còncava
 - De frontissa
 - De con giratori
 - De colissa
 - Motoritzades
 - Genèriques amb 6 graus de llibertat
- Detecció de col·lisions de forma discreta i contínua
- Llançament de rajos amb filtrat de col·lisió
- Suport per la creació de ragdolls

2.3.4.1.2. CONTROL DE PERSONATGE

- Control de personatge bàsic

2.3.4.1.3. VEHICLES

- Simulació de vehicles amb definició de paràmetres

2.3.4.1.4. OPTIMITZACIONS

- Versió de múltiples fils d'execució per a màquines multi nucli disponible
- Versió per a Cell SPU optimitzada disponible a través de Sony PS3 Devnet
- Suport de C# executable amb l'XNA per Windows i Xbox 360

2.3.4.1.5. SIMULACIÓ DE PARTÍCULES

- No té suport per a la simulació de partícules

2.3.4.1.6. ROBA

- Simulació de roba i cordes

2.3.4.1.7. COSSOS TOUS

- Simulació de cossos tous i volums deformables

2.3.4.1.8. CAMPS DE FORÇA

- No té suport per a la simulació de camps de força

2.3.4.2. ALTRA INFORMACIÓ D'INTERÈS:

- Possibilitat d'importació i exportació a COLLADA
- Compilació multi plataforma i amb suport de Collada
- Llibreria de codi obert en C++ sota llicència Zlib i gratuïta per us comercial
- Disponible per a Playstation 3, Xbox 360, Nintendo Wii i PC

2.3.4.3. RESUM:

Aquesta llibreria està molt bé ja que disposa d'una gran varietat de primitives de col·lisió i tipus d'unions tot i que per l'aplicació no es necessiten totes. A més a més disposa de detecció de col·lisions contínua que és ideal per detectar col·lisions de cossos que van a gran velocitat. Altres aspectes a destacar són el suport per a la creació de ragdolls, la simulació de vehicles, roba, cordes i cossos tous encara que per al moment no són necessaris. Tampoc és necessària ni la simulació de partícules ni la de camps de força que aquesta llibreria no té.

Un fet remarcable d'aquesta llibreria és la quantitat de plataformes diferents que la suporta juntament amb el fet de que és gratuïta per a us comercial. A més a més el suport a COLLADA pot ser molt útil per si es vol guardar els resultats i exportar-ho a una altre aplicació.

També disposa d'exemples i de fòrums de consulta per Internet, malgrat això però, l'últim manual de Bullet que hi ha no està actualitzat.



2.3.5.1. PRESTACIONS DE LES QUALS DISPOSA

2.3.5.1.1. COSSOS RÍGIDS I SIMULACIÓ FÍSICA

- Diferents tipus de primitives de col·lisió:
 - Caixa
 - Esfera
 - Càpsula
 - Malla convexa
 - Malla triangular
 - Elevació de terreny
- Diferents tipus d'unions:
 - De frontissa
 - De colissa
 - Amb motor
- Ofereix un únic mètode iteratiu per resoldre les unions el qual permet escollir l'equilibri entre la rapidesa i la precisió.
- Posseeix un model de fricció molt realista i amb un cost computacional molt baix
- Permet construir models que puguin ser trencats quan es produeixi una col·lisió

2.3.5.1.2. CONTROL DE PERSONATGE

- No té suport per el control de personatge

2.3.5.1.3. VEHICLES

- No té suport per la fàcil implementació de vehicles

2.3.5.1.4. OPTIMITZACIONS

- És especialista en l'optimització de grans apilonaments d'objectes

2.3.5.1.5. SIMULACIÓ DE PARTÍCULES

- Conté suport per a la simulació d'efectes de partícules

2.3.5.1.6. ROBA

- No té suport per a la simulació de roba

2.3.5.1.7. COSSOS TOUS

- No té suport per a la simulació de cossos tous

2.3.5.1.8. CAMPS DE FORÇA

- No té suport per a la simulació de camps de força

2.3.5.2. ALTRA INFORMACIÓ D'INTERÈS:

- Per la resolució d'unions no utilitza grans matrius el qual fa que eviti la limitació de l'ample de banda de la memòria d'algunes consoles de joc

2.3.5.3. RESUM:

Aquesta llibreria està bastant limitada ja que no suporta molta part de simulació com són els apartats de roba, cossos tous i camps de força. A més no hi ha gaire varietat en els diferents tipus d'unions i no té suport per implementar vehicles ni control de personatge.

Malgrat tot això, seria una possible llibreria a utilitzar ja que està especialitzada en l'apilament d'objectes i totes les funcionalitats que es poden necessitar per l'aplicació hi són. A més el fet de tenir la potestat de decidir entre rapidesa i precisió fa que li doni més flexibilitat segons les possibilitats del hardware.

Cal destacar que la documentació de la què disposa és bastant pèssima, no té cap manual concret i si es vol informació s'ha de mirar les especificacions que hi ha de cada funció. De totes maneres hi ha alguns exemples que poden ajudar a l'hora de desenvolupar amb la llibreria juntament amb un fòrum de consulta per Internet tot i que una mica incomplet.

2.3.6. HAVOK™ V5.5.0 (19/02/2008)



2.3.6.1. PRESTACIONS DE LES QUALS DISPOSA

2.3.6.1.1. COSSOS RÍGIDS I SIMULACIÓ FÍSICA

- Diferents tipus d'unions:
 - Motoritzades
 - De cadena
 - Trencables
 - Configurables
- Detecció de col·lisions contínua
- Suport per a la creació de ragdolls
- Tecnologia MOPP™ per a compactar representacions de grans malles de col·lisió
- Sistema per a la destrucció d'objectes en temps real

2.3.6.1.2. CONTROL DE PERSONATGE

- Eines per configurar el comportament del personatge
- Integració completa del personatge amb la simulació

2.3.6.1.3. VEHICLES

- Suport per a la fàcil creació de vehicles

2.3.6.1.4. OPTIMITZACIONS

- Suport per a Microsoft Xbox 360, Sony PLAYSTATION 3, Nintendo Wii, Microsoft Xbox, Sony PlayStation 2, PSP i PC. Optimitzada per treure el màxim rendiment del hardware en totes elles.
- Optimització per múltiples fils d'execució

2.3.6.1.5. SIMULACIÓ DE PARTÍCULES

- No disposa de simulació de partícules

2.3.6.1.6. ROBA

- Suport per a la simulació de cossos de roba com pantalons, capes, samarretes, cortines, ...
- Control de la massa, la resistència amb l'aire i la capacitat de doblegar-se i desdoblegar-se
- Possibilitat de limitar els moviments de la roba i de tenir col·lisions amb altres objectes

- Simulació de trossos de roba que s'hagin trencat de la roba principal

2.3.6.1.7. COSSOS TOUS

- Suport per a la creació de cossos tous

2.3.6.1.8. CAMPS DE FORÇA

- No disposa de simulació de camps de força volumètrics

2.3.6.2. ALTRA INFORMACIÓ D'INTERÈS:

- Conté plug-ins per Autodesk 3Ds Max, Maya i Avid SOFTIMAGE|XSI
- Disposa de depuració d'errors visual
- És una llibreria física amb una gran robustesa i fàcilment integrable amb l'aplicació
- Disposa de documentació de suport, ajudes detallades i fòrums de consulta per Internet
- La llibreria física està estructurada en blocs per donar una major modularitat: Havok Behavior™, Havok Physics™, Havok Animation™, Havok Cloth™ i Havok Destruction™
- És una llibreria gratuïta només per a us no comercial

2.3.6.3. RESUM:

Aquesta és una llibreria física molt potent i robusta i disposa de moltes prestacions. El suport optimitzat per una multitud de plataformes fa que no tingui limitacions a l'hora de portar un joc d'una plataforma a una altra. A més està modularitzat la qual cosa facilita el desenvolupament amb ella juntament amb els plug-ins de què disposa. També cal dir que el nivell d'aprofundiment en cada camp és bastant considerable i és per això que la fa destacable.

De totes maneres un dels inconvenients que té és que només és gratuït per a ús no comercial tot i que a nosaltres no ens afecta.

Per últim cal destacar la documentació i fòrums de consulta per Internet.

Alguns jocs que s'han fet amb aquesta llibreria són: Assassin's Creed, Battlefield: Bad Company, Stuntman: Ignition, Top Spin 3 o The Godfather entre d'altres.

Physics Abstraction Layer

2.3.7. PHYSICS ABSTRACT LAYER

La Physics Abstract Layer no és cap llibreria física però cal comentar-la ja que com el seu nom indica és una capa d'abstracció de diferents motors físics. El que permet aquesta eina és englobar diferents llibreries físiques en una mateixa aplicació. A més la utilització és senzilla i fàcilment adaptable en el projecte i no restringeix cap motor de física ja que en el cas que l'eina no disposi del motor que es vol utilitzar disposa de les eines i tutorials necessaris per afegir-lo.

Un dels principals avantatges que té aquesta eina és que et permet comparar el mateix comportament d'una sèrie d'objectes amb diferents motors físics i d'aquesta manera poder veure quin d'ells s'assembla més al resultat que es vol obtenir (Veure Fig. 2.1).

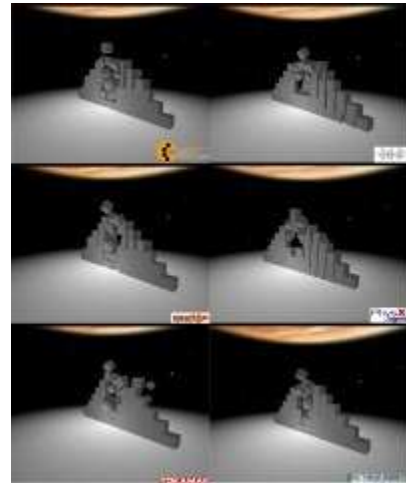


Fig. 2.1: Execució amb Physics Abstract Layer

2.4. QUADRE COMPARATIU DE LLIBRERIES FÍSIQUES

	<i>Control de personatge</i>	<i>Vehicles</i>	<i>Optimitzacions</i>	<i>Simulació de partícules</i>	<i>Roba</i>	<i>Cossos tous</i>	<i>Camps de força</i>
Newton Game Dynamics	Sí ¹	Sí ¹	No ²	No ²	No ²	No ²	No ²
NVIDIA PhysX Technology	Sí ¹	Sí ¹	Sí ¹	Sí ¹	Sí ¹	Sí ¹	Sí ¹
Open Dynamics Engine	No ²	No ²	Sí ¹	No ²	No ²	No ²	No ²
Bullet Physics Library	Sí ¹	Sí ¹	Sí ¹	No ²	Sí ¹	Sí ¹	No ²
Tokamak Physics Engine	No ²	No ²	Sí ¹	Sí ¹	No ²	No ²	No ²
Havok	Sí ¹	Sí ¹	Sí ¹	No ²	Sí ¹	Sí ¹	No ²

¹ Per saber la informació detallada de cada prestació cal mirar les pàgines anteriors amb la informació de cada llibreria.

² Això és tenint en compte la documentació i la recerca feta, es podria donar la possibilitat que tingués alguna prestació però que no s'hagi trobat.

2.5. ELECCIÓ DE LA LLIBRERIA FÍSICA

Un cop s'han estudiat i analitzat les diferents llibreries s'han pogut saber les capacitats de cadascuna i quines s'adaptaven millor a les nostres necessitats.

En principi com que l'aplicació es desenvoluparà per a PC i sobre Windows no hi ha cap limitació de les llibreries anteriorment analitzades. Les prestacions de les que disposen tampoc són cap impediment ja que només es necessita crear esferes, caixes, volums convexos i unions que es puguin trencar entre les caixes.

El sistema de detecció de col·lisions no importa quin s'utilitzi sempre i quan sigui ràpid, robust i precís. De totes maneres aniria bé que tingués la possibilitat de detecció de col·lisions contínua ja que aquest sistema permet detectar la col·lisió entre dos cossos que vagin a grans velocitats.

La llicència que tingui la llibreria és important que no restringueixi massa la seva utilització i que no s'hagi de pagar per desenvolupar amb ella. El fet de si s'ha de pagar per la versió comercial de moment no interessa.

La documentació és un aspecte molt important juntament amb els fòrums de consulta per si sorgeix algun problema tenir un lloc on anar a buscar resposta o si més no a preguntar-ho.

Amb aquestes limitacions només es descarta la llibreria Tokamak i també es podria descartar la Newton pel fet de que fa dos anys de l'última versió. De totes maneres l'aspecte més interessat és el fet de l'acceleració per hardware de la llibreria PhysX. Per aquest motiu doncs aquesta serà la llibreria utilitzada per l'aplicació.

2.6. NVIDIA PHYSX TECHNOLOGY

2.6.1. INTRODUCCIÓ

NVIDIA PhysX Technology és un motor de física desenvolupat per AGEIA i posteriorment adquirit per NVIDIA aquest mateix any 2008. La peculiaritat bàsica que ha tingut sempre aquesta tecnologia és que dona la possibilitat d'accelerar la simulació física a través d'un hardware especialitzat.

2.6.2. PHYSX BY AGEIA

Abans que NVIDIA adquirís AGEIA la simulació física es podia fer a través de l'AGEIA PhysX accelerator (Veure Fig. 2.2). Aquest hardware consistia en una targeta PCI la qual tenia una unitat de processament dedicada exclusivament a fer els càlculs necessaris per dur a terme la simulació física. D'aquesta manera s'aconseguia alliberar una gran càrrega de càlcul de la CPU i augmentar molt més la rapidesa de la simulació.



Fig. 2.2: AGEIA PhysX accelerator

Així doncs la idea general que volia aconseguir AGEIA Technologies era que en una aplicació el processament de càlculs estigues dividit en tres unitats de processament, cadascuna especialitzada en la seva tasca (Veure Fig. 2.3).

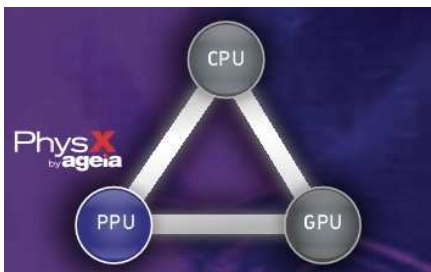


Fig. 2.3: Unitats de processament

Aquestes tres unitats eren la PPU, especialitzada en els càlculs de la simulació física, la GPU, especialitzada en els càlculs per a una òptima visualització gràfica i la CPU, encarregada de realitzar les tasques de propòsit general que no són capaces de realitzar ni la PPU ni la GPU.

2.6.3. PHYSX BY NVIDIA

Amb l'adquisició d'AGEIA per NVIDIA el plantejament inicial ha canviat una mica. Aprofitant que PhysX està dissenyat per ser accelerat amb un hardware amb processadors potents i centenars de nuclis, NVIDIA ho ha combinat amb la capacitat de càlcul paral·lel de les seves targetes gràfiques per així incrementar exponencialment les possibilitats d'aquest motor físic i aconseguir espectaculars efectes.

A primera vista pot semblar una mala idea perquè tant la simulació gràfica com la física han de compartir la mateixa targeta, però això no és cap inconvenient ja que l'ús de la GPU es reparteix segons les necessitats i així, per exemple, quan no necessitem un gran processat gràfic ho podem aprofitar amb el processat físic i viceversa. A més a més sempre hi ha la possibilitat d'afegir una segona targeta gràfica.

Les targetes de NVIDIA que ara mateix suporten la simulació física per hardware són totes les GeForce sèrie 8, sèrie 9 i la nova sèrie 200 (Veure Fig. 2.4). A més continuen mantenint el suport per el AGEIA PhysX accelerator.



Fig. 2.4: NVIDIA GeForce GTX 280

2.7. INTEGRACIÓ DE LLIBRERIES FÍSQUES EN MOTORS DE JOC

La integració de les llibreries físiques en un motor de joc a vegades sol ser un tema una mica delicat.

Hi ha alguns motors com poden ser Unity o Unreal engine que ja porten el PhysX Technology integrat i a l'hora de voler utilitzar una altra llibreria física a vegades no és possible. Aquest fet és un gran avantatge perquè la integració del motor de física amb el motor de jocs és molt més completa i el desenvolupador no s'ha de preocupar per els conflictes que puguin tenir els dos motors. Per altra banda, el petit inconvenient que té, és que t'obliga a utilitzar el sistema de física que ell ha escollit.

D'altres en canvi porten el seu propi sistema de simulació física amb la possibilitat d'integrar-ne d'altres. Alguns exemples de motors de joc d'aquest estil serien Torque o Vicious engine. Això va bé sempre i quan les funcionalitats físiques que proporciona el motor de joc s'adaptin a les necessitats del desenvolupador, en cas contrari es pot integrar el motor de física desitjat.

Malgrat tot, no tots els motors de jocs estan preparats per simular física, i si es vol aquesta capacitat, s'ha d'escollir la llibreria que més s'adapti a les nostres necessitats i intentar integrar-la en el motor.

3. GENERADOR DE CONTINGUT FÍSIC

3.1. INTRODUCCIÓ

Un cop escollida la llibreria física a utilitzar a l'aplicació llavors el segon aspecte més important a tenir en compte és com realitzar correctament un generador de contingut físic.

Un generador de contingut físic és una aplicació que donant-li una sèrie de paràmetres i característiques et pot generar tota una sèrie de cossos, unions i altres elements de la llibreria de forma automàtica i amb tots els paràmetres físics ja definits. D'aquesta manera amb pocs minuts es pot tenir tot un món preparat a mostrar un comportament basat amb la física de Newton.

El que cal destacar com a requisits que ha de tenir és que sigui senzill, fiable, flexible i com més automatitzat millor. El problema que tenen a vegades els generadors de contingut és que són massa limitats, és a dir, a vegades no permeten decidir tot el que l'usuari desitja. També pot passar que no funcioni correctament o que a l'hora d'utilitzar-lo sigui massa complicat i és per això que s'intentarà evitar tots aquests problemes que sabem que poden tenir.

El primer pas que s'ha decidit és que la base de la qual partirà el generador serà un fitxer que emmagatzema gràfics vectorials. Aquest fitxer contindrà un esquema en planta d'alguna construcció amb la qual el generador s'encarregarà a partir d'aquesta imatge, juntament amb una alçada i unes mides de rajol, de generar unes parets o murs de rajols que siguin fidels a la imatge en planta. A més de l'alçada i les mides de cada rajol també es podrà escollir el material amb què estarà feta la paret definint la seva densitat i la força d'unió entre els maons.

3.2. CARREGADOR DES DE FITXER SVG

3.2.1. INTRODUCCIÓ

A l'hora d'escollir el format del fitxer interessa que sigui un format gràfic de tipus imatge ja que així és fàcil de visualitzar i d'intuir què carregarà el generador només mirant el fitxer d'entrada. Posant només aquest requisit ens queden tota una multitud de formats d'imatge com poden ser JPEG, GIF, BMP, PICT, DIB, PNG, SVG, TIF, TARGA, etc.

El segon requisit que ha de tenir és que ha de ser un format d'imatge vectorial ja que conté informació dels punts i polígons que forma la imatge i així és molt més fàcil de tractar i analitzar. Afegint aquest requisit amb l'anterior queden només formats d'imatge vectorial com ara AI, CDR, CGM, DXF, DWG, EPS, FLA, PDF, SVG, etc.

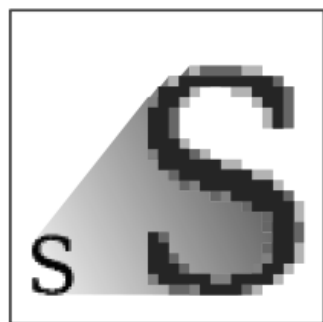
Finalment, l'últim requisit que cal parar especial atenció, és que ha de ser un format fàcilment interpretable perquè per el que s'ha de fer servir serà la manera més senzilla de tractar-lo i és el que pot evitar més problemes. Així doncs el format de fitxer d'imatge vectorial amb una estructura més simple i entenedora és el format SVG.

3.2.2. FORMAT SVG

El format SVG (Scalable Vector Graphics) permet representar gràfics vectorials bidimensionals ja siguin animats o estàtics. Aquest format és una especificació de l'XML el qual és un metallenguatge etiquetat desenvolupat per el World Wide Web Consortium àmpliament usat a la web i en eines d'importació i exportació.

Els tres tipus d'objectes gràfics que permet el SVG són:

- Gràfics vectorials (Veure Fig. 3.2)
- Gràfics rasteritzats (Mapes de bits) (Veure Fig. 3.1)
- Text



BITMAP
.jpeg .gif .png

Fig. 3.1: Gràfic rasteritzat



OUTLINE
.svg

Fig. 3.2: Gràfic vectorial

Tot seguit es pot veure el codi d'un fitxer SVG juntament amb la imatge que representa (Veure Fig. 3.3):

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG
1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11
.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
version="1.1"
width="467" height="462">
<!-- This is the red square: -->
<rect x="80" y="60" width="250" height="250" rx="20" fill="red"
stroke="black" stroke-width="2px" />
<!-- This is the blue square: -->
<rect x="140" y="120" width="250" height="250" rx="40" fill="blue"
stroke="black" stroke-width="2px" fill-opacity="0.7"/>
</svg>
```

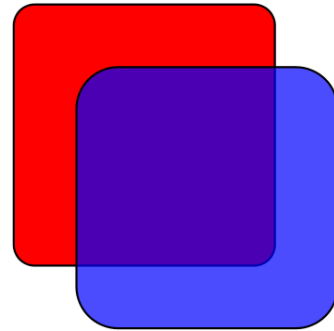


Fig. 3.3: Imatge SVG

3.2.3. ANALITZADOR SINTÀCTIC SVG

Pel que fa a l'eina utilitzada per analitzar sintàcticament el fitxer SVG ha estat la llibreria Expat. Aquesta llibreria és fruit d'un projecte de codi obert el qual permet analitzar fitxers XML i com que el SVG no és més que una especificació de XML doncs ens és ideal per extreure tota la informació que puguem necessitar.

Primerament es va intentar utilitzar una llibreria anomenada AGG (Anti-Grain Geometry) que està especialitzada en els fitxers SVG i el seu tractament. De totes maneres no es va veure viable continuar endavant amb aquesta llibreria degut el nivell de complexitat és molt alt respecte les poques funcionalitats que es necessiten.

És per això doncs, que s'ha decidit utilitzar la llibreria Expat encara que s'ha hagut d'analitzar cada etiqueta del fitxer XML quin element és, però, malgrat tot, és molt més senzilla d'utilitzar.

3.2.4. EXTRACCIÓ DE DADES D'UN SVG

Ara que tenim el format de fitxer i l'analitzador sintàctic que farem servir, l'únic que ens falta és extreure les dades necessàries del fitxer SVG.

Dels fitxers SVG no s'utilitzaran tots els tipus d'objectes gràfics que permet, els únics que es faran servir seran els gràfics vectorials i dintre d'aquests les línies poligonals. A més tampoc es tindrà en compte ni el color ni la gruixària de les línies, l'única cosa que interessa és les posició dels punts de les línies i les unions entre ells. En cas de que passin més informació o més objectes dels que es demanen aleshores no es tindran en compte. Un model d'imatge que es pot carregar és el de la Figura 3.4.:

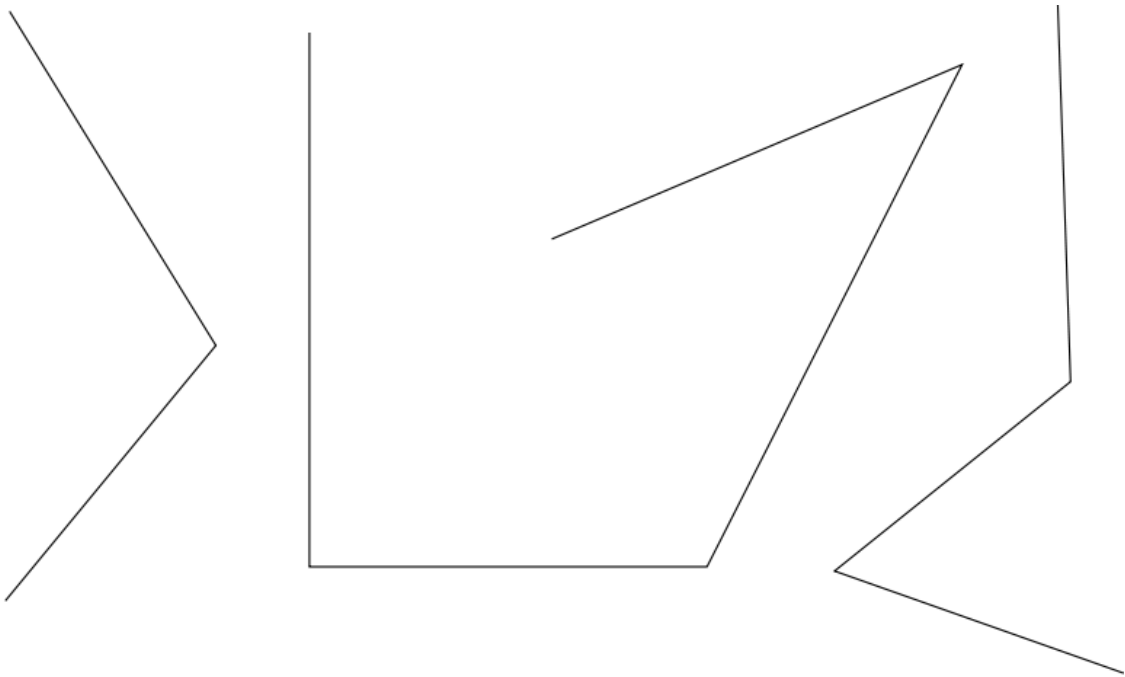


Fig. 3.4: Possible imatge SVG com a fitxer d'entrada

Com es pot veure, a través del fitxer no se li passa informació de l'alçada de cada mur, la força amb la qual estan units els rajols ni tota la informació referent a les característiques del rajol (alçada, amplada, profunditat i densitat). Tota aquesta informació l'entrarà interactivament l'usuari a través del teclat per així, amb un mateix fitxer SVG, pugui entrar diferents informacions d'alçada i característiques de maó per diferents execucions.

3.2.5. CREACIÓ D'UN FITXER SVG

Per crear una imatge en format SVG es pot utilitzar programes d'edició vectorial tals com el Corel Draw o l'Adobe Illustrator entre d'altres. També es poden buscar imatges en format SVG de la xarxa.

En tots dos casos cal tenir en compte que les línies poligonals que s'hi representin s'han de guardar com a tal i no com a gràfic rasteritzat.

3.3. GENERACIÓ DE MURS

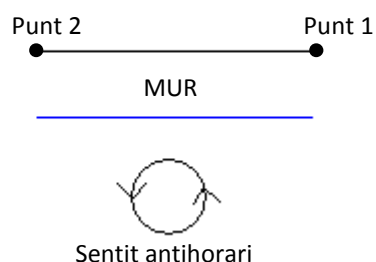
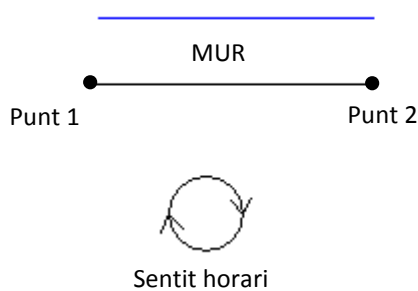
3.3.1. SISTEMA UTILITZAT

Un cop s'han obtingut la quantitat de línies poligonals que té la imatge, juntament amb la informació dels punts, el que toca fer és crear per cada línia poligonal un mur.

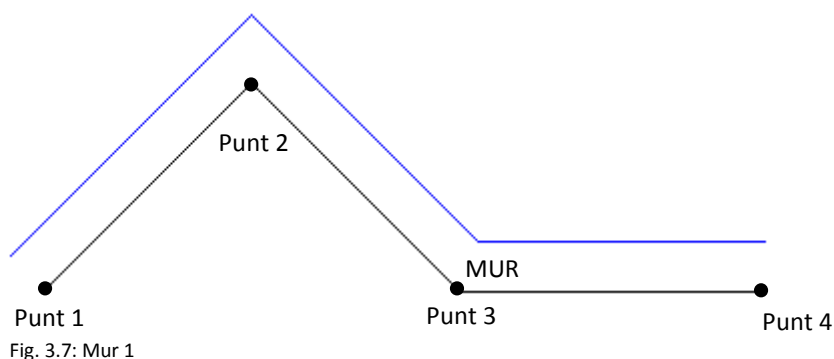
El que cal tenir en compte a l'hora de crear un mur és que la línia no té gruixària a diferència del mur. És per això que s'ha de decidir a quina banda de la línia es construirà el mur o si es construirà la meitat per cada banda.

En el nostre cas s'ha decidit que a l'hora de construir el mur es podrà triar la banda per la qual es construeix però haurà de ser per una banda o per l'altra i no per el mig. L'avantatge de fer-ho així és que et dona llibertat a l'hora de decidir i així per exemple si es vol construir un mur al costat d'un carrer es pot evitar que el mur n'envaeixi un tros.

El sistema utilitzat va segons l'ordre dels punts al moment de la creació de la recta poligonal. Si a l'hora de crear els punts es fa en el sentit de les agulles del rellotge aleshores el mur es crea cap a l'exterior (Veure Fig. 3.5), si per al contrari es crea en el sentit contrari a les agulles del rellotge aleshores el mur es crea cap a l'interior (Veure Fig. 3.6).



Tot seguit es pot veure un exemple de la creació de dos murs que parteixen d'una línia poligonal amb la mateixa forma però amb els punts donats en ordre invers:



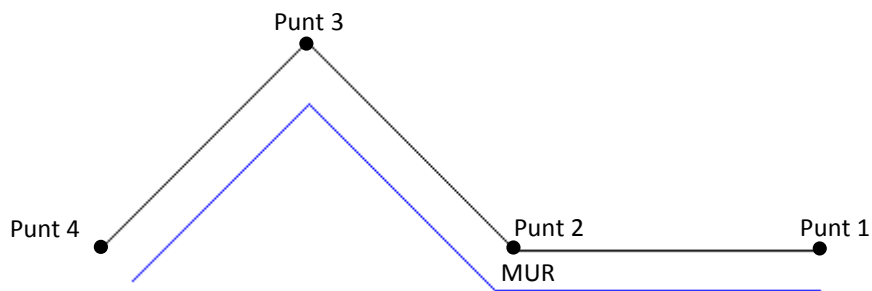


Fig. 3.8: Mur 2

3.3.2. COL·LOCACIÓ DELS RAJOLS

Per col·locar els rajols en el mur es farà amb aparell a trencajunt de pla (Veure Fig. 3.9), una tècnica en què la junta vertical d'una filera de rajols es troba exactament al mig del rajol de la filera anterior. Aquesta tècnica és molt usada en la construcció perquè així quan es construeix un mur l'estabilitat d'aquest no depèn només del formigó que s'hi ha posat sinó que amb aquesta tècnica, té estabilitat per ell mateix. De totes maneres per construir el mur es tindran en compte tres zones diferents on s'haurà de tenir especial cura a l'hora de posar els rajols. Aquestes zones són l'inici i el final del mur, les cantonades i els trams rectes:

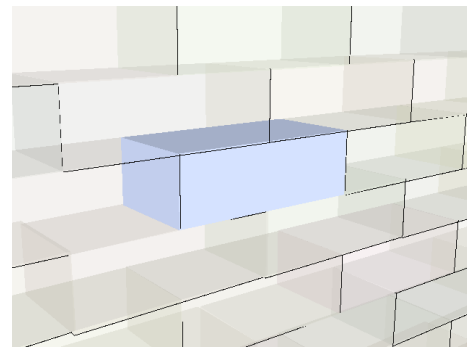


Fig. 3.9: Paret amb aparell a trencajunt de pla

3.3.2.1. INICI I FINAL

Al principi i al final del mur caldrà tenir en compte que segons la fila caldrà acabar-la amb mig rajol enlloc d'un. Aquest cas el tindrem sempre ja que com que el que tractem sempre són línies poligonals sempre tindran inici i final.

En el cas de l'inici si la fila és un número imparell començarà amb un rajol enter (Veure Fig. 3.10), si per al contrari és un número parell començarà amb mig rajol (Veure Fig. 3.11).



Fig. 3.10: Inici mur: Pis imparell

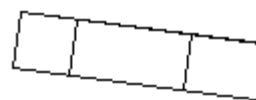


Fig. 3.11: Inici mur: Pis parell

En el cas del final la decisió de si es posa un rajol enter a la fila amb número parell o a la fila amb número imparell dependrà de la manera que s'ajusti més a la distància definida per l'usuari.

3.3.2.2. CANTONADA

A les cantonades dels murs es col·locaran els rajols tallats per així evitar que sobresurtin trossos de rajol per algun dels costats del mur. A més s'haurà de tenir en compte l'angle de cada cantonada segons aquests casos: major de 0° i menor de 90° (Veure Fig. 3.12 i Fig. 3.13), igual a 90° (Veure Fig. 3.14 i Fig. 3.15), major de 90° i menor de 180° (Veure Fig. 3.16 i Fig. 3.17), major de 180° i menor de 270° (Veure Fig. 3.18 i Fig. 3.19), igual a 270° (Veure Fig. 3.20 i Fig. 3.21) i major de 270° i menor de 360° (Veure Fig. 3.22 i Fig. 3.23).

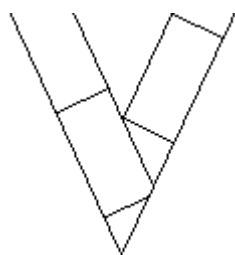


Fig. 3.12: Pis parell: $0^\circ < \text{angle} < 90^\circ$

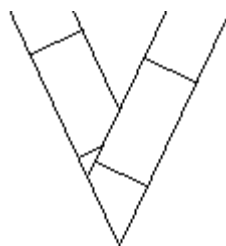


Fig. 3.13: Pis imparell: $0^\circ < \text{angle} < 90^\circ$

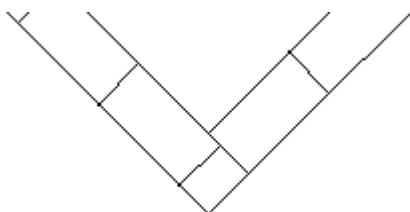


Fig. 3.14: Pis parell: angle = 90°

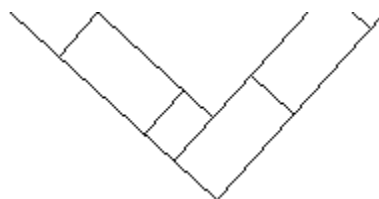


Fig. 3.15: Pis imparell: angle = 90°

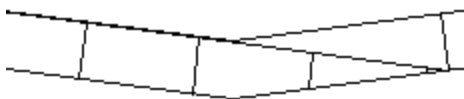


Fig. 3.16: Pis parell: $90^\circ < \text{angle} < 180^\circ$



Fig. 3.17: Pis imparell: $90^\circ < \text{angle} < 180^\circ$

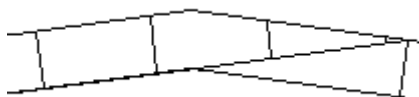


Fig. 3.18: Pis parell: $180^\circ < \text{angle} < 270^\circ$



Fig. 3.19: Pis imparell: $180^\circ < \text{angle} < 270^\circ$

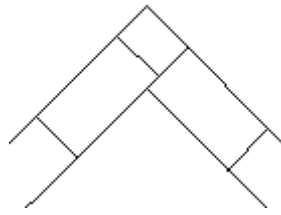


Fig. 3.20: Pis parell: angle = 270°

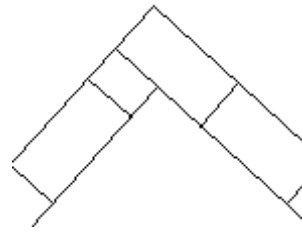


Fig. 3.21: Pis imparell: angle = 270°

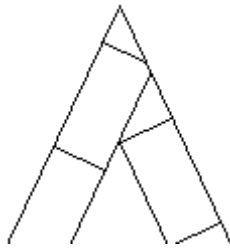


Fig. 3.22: Pis parell: 270° < angle < 360°

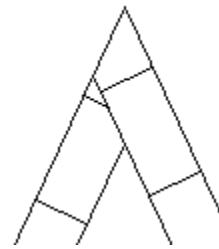


Fig. 3.22: Pis imparell: 270° < angle < 360°

Cal tenir en compte que en els casos que una línia poligonal estigui formada només per dos punts aleshores aquest cas no existirà.

3.3.2.3. TRAM RECTE

En els trams rectes la construcció serà normal i només caldrà tenir en compte que les juntes verticals de cada fila caiguin just al mig del rajol de la fila anterior.

3.3.3. ELS FONAMENTS

Per aconseguir una millor estabilitat de la paret s'han de construir uns fonaments a on estiguin agafats els rajols, per fer-ho s'utilitzaran els cossos cinemàtics. Aquest tipus d'objecte té la peculiaritat que les interaccions físiques no els hi afecten però si que afecten a altres cossos, si un cos xoca amb un objecte cinemàtic aleshores l'objecte rebotirà però el cinemàtic es quedarà quiet.

Així doncs, la primera fila inferior de la paret, enlloc de construir-la amb objectes dinàmics es farà amb objectes cinemàtics. A més s'enterrarà la meitat del rajol sota terra per donar més sensació de fonament (Veure Fig. 3.23).

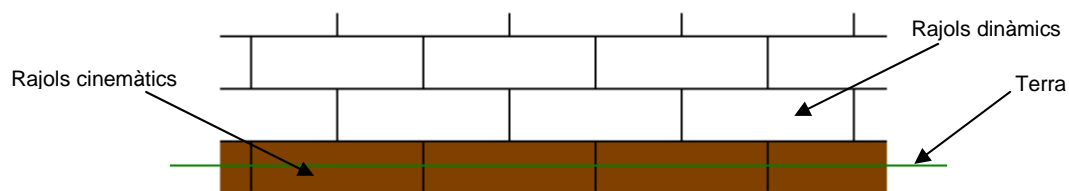


Fig. 3.23: Fonaments (Vista lateral)

3.3.4. UNIONS

Per poder simular la força del ciment entre els rajols es farà a través d'unions punt a punt. L'ideal seria utilitzar unions de superfície amb superfície, però com que actualment la llibreria no en disposa, s'agafaran les de punt a punt ja que són les que s'ajusten més a les nostres necessitats i a més tenen la possibilitat de ser trencables. Això significa que se li pot definir una força límit i així quan se li aplica una força superior a aquesta aleshores la unió es trenca.

El sistema utilitzat per unir els rajols entre ells serà en forma de xarxa, les unions aniran des de cada rajol als dos que tingui a sobre i als dos que tingui a sota (Veure Fig. 3.24). També tenen contacte amb els rajols de cada costat però com que la major part de la superfície generalment es troba a sobre i a sota no s'uniran amb els dels costats per així reduir el nombre d'unions i poder exigir a la simulació millor precisió sense que vagi massa lent.

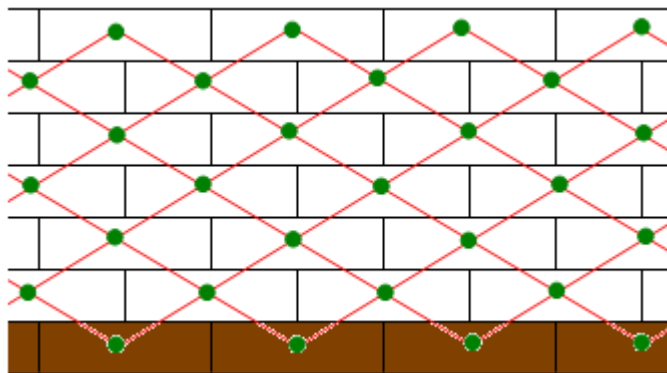


Fig. 3.24: Xarxa d'unions (Vista lateral)

Els punts d'ancoratge de cada unió aniran d'un extrem del rajol a l'extrem oposat de l'altre rajol (Veure Fig. 3.25). D'aquesta manera aconseguirem que la xarxa d'unions en lloc d'estar creada en un pla 2D estigui en un espai 3D. Aquesta tècnica sol ser bastant utilitzada quan es creen unions, ja que així amb menys càlculs el motor pot simular amb més precisió el comportament d'aquestes.

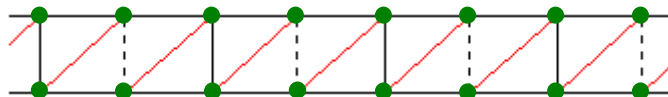


Fig. 3.25: Xarxa d'unions (Vista superior)

3.4. DISSENY

3.4.1. FITXES DE CAS D'US

Fitxes de cas d'us de les funcionalitats més importants:

CAS D'US: Engegar aplicació	
Actors:	Usuari
Precondició:	Cert
Flux principal:	<ol style="list-style-type: none">1. Inicialitzar el món físic amb els paràmetres corresponents2. Llegir fitxer SVG3. Per cada línia poligonal del fitxer<ol style="list-style-type: none">3.1. Llegir dades del mur des de teclat3.2. Crear mur
Subfluxos:	En el cas de no disposar d'una targeta d'acceleració física es mostrarà un missatge per pantalla advertint de que es simularà per software.
Postcondició:	L'aplicació ha arrencat correctament i ha generat una o varies parets de rajols segons la informació del fitxer "paret.svg" i segons les dades introduïdes per teclat de l'usuari.

CAS D'US: Llençar projectil	
Actors:	Usuari
Precondició:	L'aplicació esta engegada correctament
Flux principal:	<ol style="list-style-type: none">1. Interacció de l'usuari des de teclat (Premer Espai)2. Crear projectil amb una velocitat inicial en la mateixa direcció que la càmera i la mateixa posició.
Subfluxos:	-
Postcondició:	S'ha creat un projectil amb una certa massa, densitat, velocitat i tots els paràmetres físics definits. El projectil surt llençat des de la càmera.

3.4.2. DIAGRAMA DE CLASSES

L'aplicació s'ha estructurat en quatre classes principals i quatre més de suport. Aquestes classes utilitzen funcions de la llibreria de PhysX, la llibreria d'OpenGL i la llibreria de C++.

Les classes principals són:

Interficie: Aquesta classe s'encarrega de gestionar tot el tema relacionat amb la interfície i la interacció de l'aplicació amb l'usuari com l'entrada de teclat o el mouse.

Física: Aquesta classe s'encarrega de fer de pont amb la llibreria física i fer més amigable la crida de funcions físiques. A més és la responsable de crear el carregador per així llavors poder crear els objectes fidels al fitxer carregat.

Carregador: Aquesta classe s'encarrega d'obtenir tota la informació del fitxer SVG i emmagatzemar-la en una estructura de dades pròpia per així llavors poder subministrar la informació a la classe Física.

DrawObjects: Aquesta classe s'encarrega de dibuixar els objectes per pantalla juntament amb les ombres. Qui s'encarrega de fer les crides a aquesta classe és la classe Física ja que per dibuixar cada objecte es necessita tota la informació física d'on està en cada moment. Aquesta classe s'ha aprofitat amb algunes modificacions dels exemples de PhysX que la incorpora per fer el renderitzat d'objectes.

Les classes auxiliars són:

Forma: En aquesta classe s'hi emmagatzema tota la informació de cada objecte del fitxer SVG. És una classe general per les figures que permet el fitxer SVG.

Polyline: Aquesta classe és una especialització de la classe Forma i conté la informació de les línies poligonals del fitxer SVG.

Punt: És una classe auxiliar per emmagatzemar al informació de cada punt de les figures del fitxer SVG.

Mao: Aquesta classe emmagatzema la informació de com són els maons: alçada, amplada, profunditat i densitat.

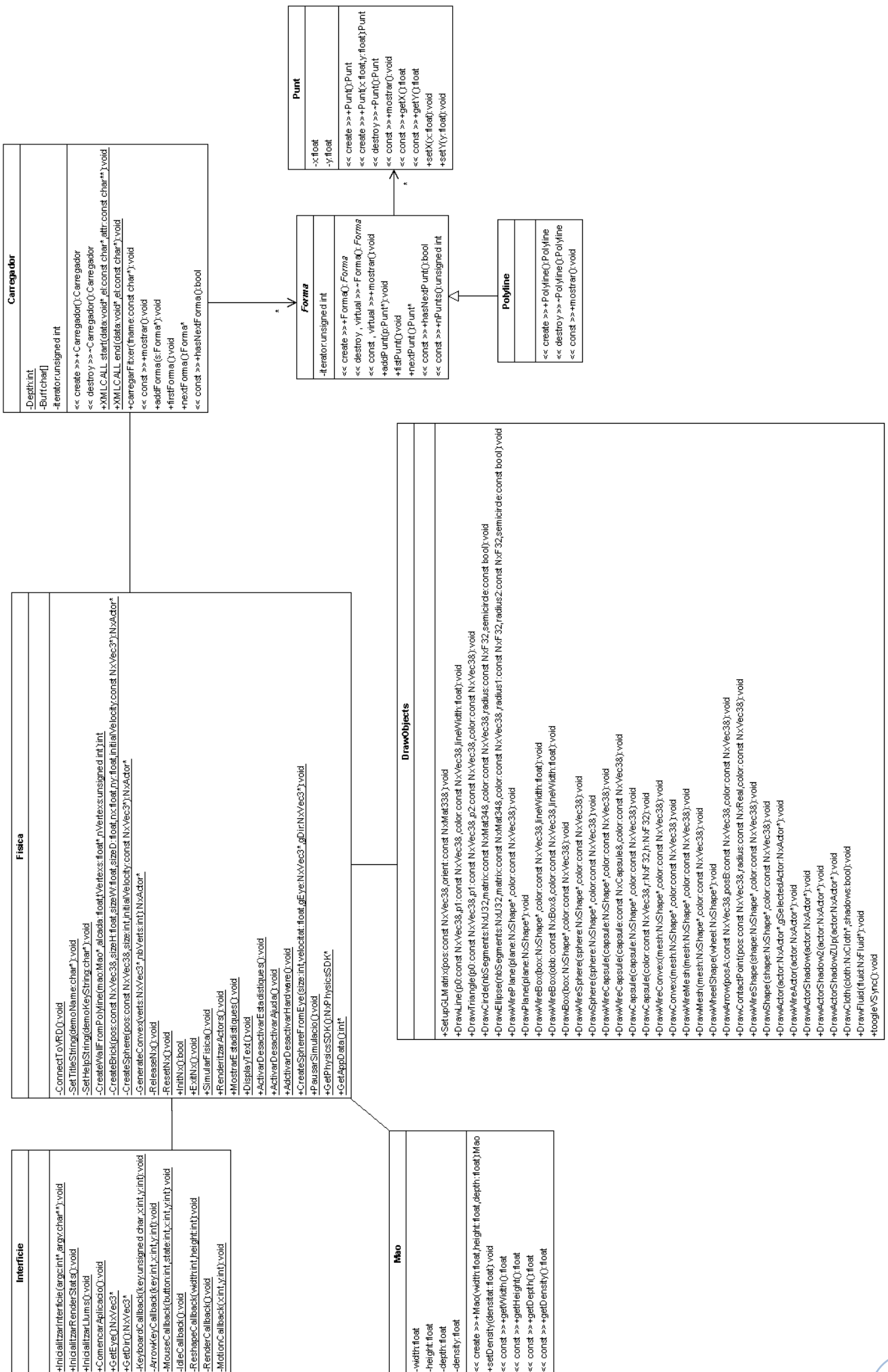
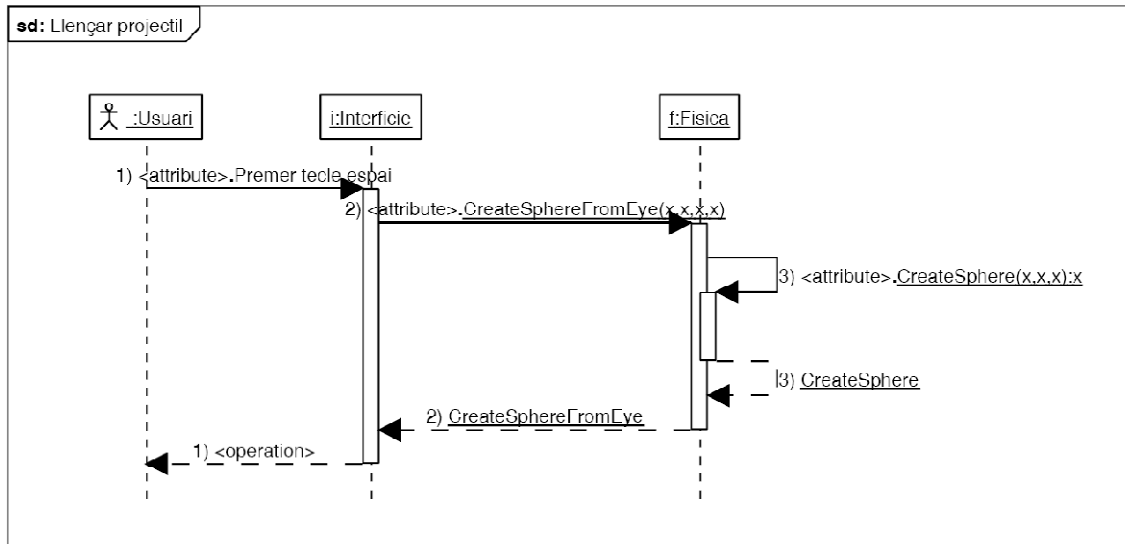


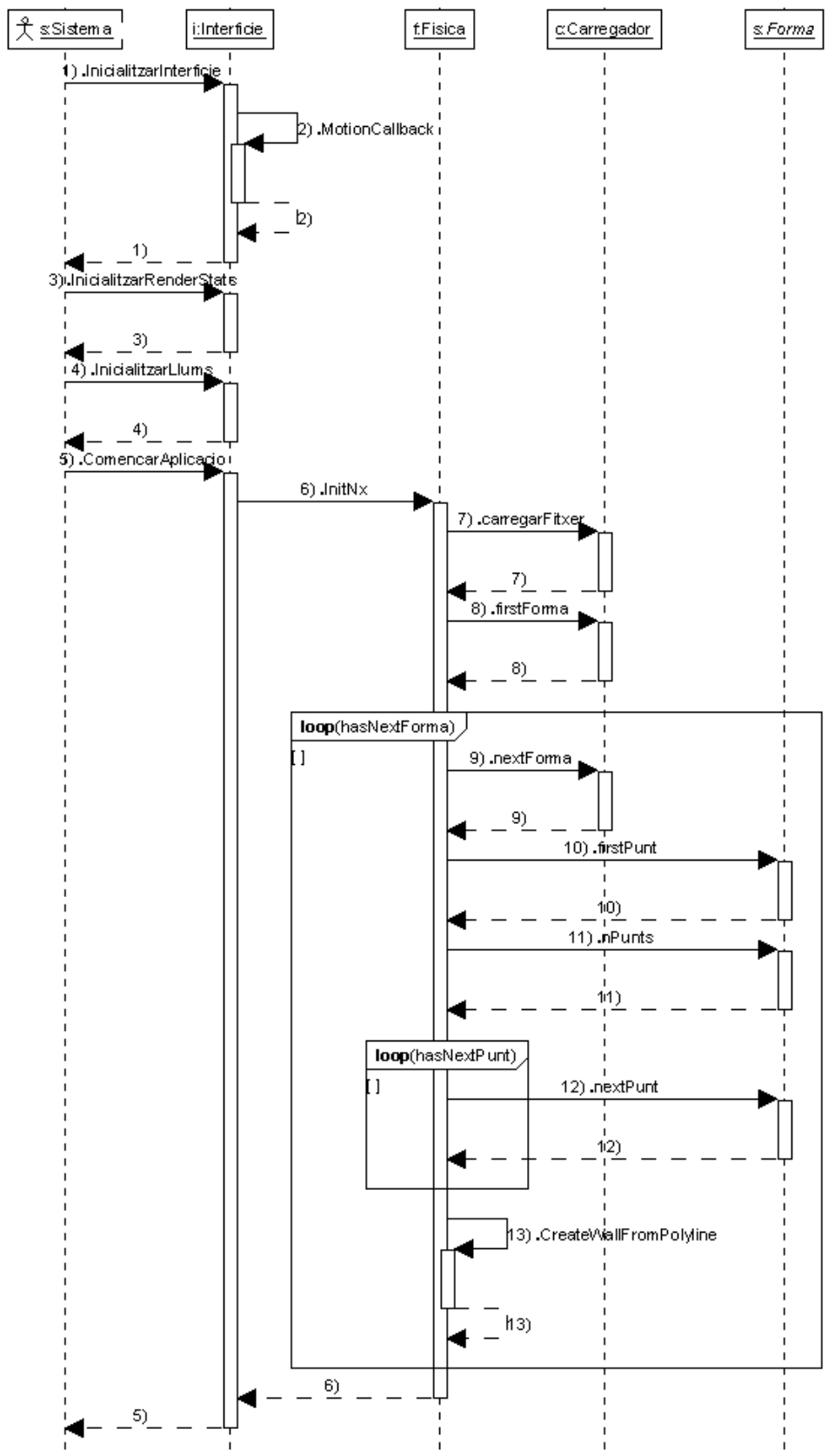
Fig 3.26: Diagrama de classes

3.4.3. DIAGRAMES DE SEQÜÈNCIA

Principals diagrames de seqüència de l'aplicació:



sd: Engegar Aplicació



3.5. INTEGRACIÓ EN EL MOTOR DE JOCS OGRE 3D

3.5.1. INTRODUCCIÓ A OGRE3D

Ogre 3D és un motor 3D de codi lliure orientat a objectes i implementat en C++ per facilitar així la feina als desenvolupadors i fer més intuïtiu el desenvolupament d'aplicacions gràfiques 3D accelerades per hardware.

Cal tenir en compte que no és un motor de jocs sinó que és un motor 3D i per tant tot el que sigui referent a intel·ligència artificial, xarxa, so o física s'hi ha d'integrar. Està disponible sota la llicència GNU Lesser General Public License (LGPL).



Fig. 3.27: Aplicació en Ogre 3D

3.5.2. INTEGRACIÓ DE L'APLICACIÓ A OGRE3D

Aprofitant que el departament d'Informàtica i Matemàtica Aplicada està adscrit en el projecte europeu Game Tools en el qual s'ha treballat amb Ogre 3D, s'ha intentat doncs integrar l'aplicació en aquest motor.

Per integrar a l'Ogre 3D la simulació física hi ha desenvolupats un seguit de conjunts de classes que embolcallen les llibreries físiques per permetre una més fàcil integració. Aquests conjunts de classes són OgreNewt, NxOgre, OgreODE i OgreBullet que permeten integrar els motors de física Newton Game Dynamics, NVIDIA PhysX Technology, Open Dynamics Engine i Bullet Physics Library respectivament.

El conjunt de classes que interessa és l'NxOgre. Aquest embolcall permet integrar PhysX a l'Ogre per així poder aprofitar la potència de simulació del PhysX i la potència de renderitzat de l'Ogre. Tot això seria perfecte però en aquesta aplicació s'ha intentat dur a terme la integració però no ha estat possible.

Aquesta llibreria embolcall està en constant desenvolupament i ara mateix està en un pas en que hi ha dues versions. Una versió és més o menys estable però li falten funcionalitats, algunes de les quals necessàries per aquesta aplicació. L'altre versió és molt més avançada i la que properament serà la definitiva. S'ha intentat integrar l'aplicació a Ogre amb aquesta segona versió de l'NxOgre però curiosament resulta que per la creació d'objectes convexos recomanen usar la primera versió mencionada. Així doncs per integrar l'aplicació a l'Ogre es necessitarien coses de les dos versions la qual cosa és impossible.

En qüestió de poc temps han començat a sortir tutorials del NxOgre i la seva web cada vegada pren una major magnitud. Potser d'aquí un temps serà possible la integració d'aquesta aplicació amb Ogre 3D per així obtenir un molt bon resultat gràfic.

3.6. PROVES D'EXECUCIÓ

3.6.1. INTRODUCCIÓ

Per testejar i comprovar les possibilitats de l'aplicació desenvolupada s'han fet tota una sèrie d'execucions amb diferent nombre de cossos a l'escena, simulat per hardware i per software i amb diferents graus de precisió. Les proves d'execució s'han dut a terme en un ordinador Pentium IV amb un processador a 3GHz, 2GB de memòria RAM i una targeta gràfica GeForce 8800 GTX de l'empresa NVIDIA.

Cal tenir en compte que les proves no són les mateixes si els actors² estan en moviment o estan en repòs. Això és degut a un sistema d'optimització que fa servir PhysX en el qual quan un cos es mou a menys d'una certa velocitat es posa en situació de "sleep". En aquest estat no es calcula la posició següent d'aquell objecte fins que se li aplica alguna força que el pugui tornar a posar en moviment.

L'altre factor a tenir en compte és que la precisió es mesura amb "solver iteration count"³. Aquest factor es té en compte quan els cossos estan units amb un o més cossos. És un sistema el qual segons el número que se li dona és el nombre de vegades que reitera sobre les restriccions de moviment de les unions. Així doncs, si hi ha un gran nombre de cossos units entre ells amb unions i interessa que la precisió sigui alta aleshores aquest nombre també va bé que sigui elevat.

3.6.2. PRIMER JOC DE PROVES

El primer joc de proves que s'ha realitzat ha estat per mirar la diferència que hi ha entre simular la física de l'aplicació per software o per hardware amb la mateixa quantitat de cossos i amb la mateixa precisió les dues execucions.

La prova s'ha fet amb 876 actors i un "solver iteration count" de 30 i s'han comparat el nombre de fotogrames per segon (fps). Hem analitzat just després de crear la paret i després de que els rajols s'anessin posant en repòs.

	<i>Hardware</i>	<i>Software</i>
Només d'engegar l'aplicació	9,4 fps	9,3 fps
Al cap de 1 minut	15,8 fps	15,5 fps

Amb aquesta prova es pot veure que la diferència entre la simulació per hardware i la simulació per software és mínima. Això és degut a que actualment la simulació física per hardware està pensada per ser utilitzada per roba, cossos tous i simulació de partícules. D'aquí poc es pretén aconseguir que la simulació de cossos rígids també es pugui dur a terme a través del hardware.

² Un actor és cada objecte de l'escena que està creat de tal manera que se li simula la física

³ Nombre d'iteracions que es realitzen per calcular les restriccions de moviment de les unions

3.6.3. SEGON JOC DE PROVES

El segon joc de proves que s'ha realitzat ha estat per comprovar com afectava la precisió en els fotogrames per segon.

Aquesta prova s'ha fet mirant amb una certa precisió quin era el mínim de fotogrames per segon que assolía l'aplicació després de llençar un projectil contra un mur. El nombre d'actors implicats ha estat de 876 més un del projectil. En tots els casos el projectil s'ha llençat des de la mateixa posició, amb la mateixa velocitat i amb la mateixa direcció. Aquesta execució s'ha fet simulant la física a través del hardware.

	<i>Hardware</i>
Precisió 30	12,9 fps
Precisió 25	13,6 fps
Precisió 20	15,3 fps
Precisió 15	16,2 fps
Precisió 10	20,3 fps
Precisió 5	24,3 fps

Com es pot observar a la taula, a mesura que s'incrementa la precisió disminueix el nombre de fotogrames per segon de l'aplicació. Si disminuïm molt la precisió també podem comprovar que el comportament del mur en l'impacta cada vegada s'assembla menys a la realitat. Així doncs la millor precisió podríem concloure que és 25 ja que a 20 o menys el comportament ja és massa inestable.

Hem provat valors més grans de 30 però surten resultats incoherents. Cercant a la documentació hem vist que el màxim en precisió és aquest valor i a partir d'aquí ens podríem trobar que ens sortissin comportaments inesperats.

3.6.4. TERCERA PROVA

La tercera i última prova s'ha realitzat per comprovar com afectava la quantitat d'actors de l'escena en els fotogrames per segon.

Per fer aquesta prova s'ha utilitzat una precisió de 25 i simulant la física per hardware. La prova s'ha fet mirant el mínim de fotogrames per segon que ha assolit l'aplicació després d'impactar un projectil en el mur.

	<i>Hardware</i>
221 actors	53,4 fps
386 actors	31,7 fps
491 actors	23,6 fps
646 actors	16,4 fps
711 actors	16,5 fps
876 actors	15,6 fps
906 actors	12,3 fps

Es pot comprovar com el nombre d'actors a l'escena influeixen en el nombre de fotogrames per segon. Cal tenir en compte que el cine emet a 24 fotogrames per segon però per exemple els videojocs d'acció solen anar a 60 fotogrames per segon. En el nostre cas dons aniria bé no tenir més de 500 actors a l'escena per així almenys aconseguir la mateixa qualitat que ofereixen en el cinema.

3.7. LIMITACIONS I POSSIBLES MILLORES DE L'APLICACIÓ

3.7.1. LIMITACIONS

Algunes de les limitacions que té actualment l'aplicació que s'ha desenvolupat són:

- L'aplicació només carrega línies poligonals on a més no poden tenir angles de 0 ni de 180 graus. Aquestes línies tampoc es poden creuar ni estar molt i molt juntes.
- Només carrega fitxers en format SVG amb un nom determinat i en una localització concreta.
- Quan hi ha una gran quantitat d'objectes el nombre de fotogrames per segon es redueix considerablement.

3.7.2. POSSIBLES MILLORES

Algunes de les millores que es podrien fer a l'aplicació que s'ha desenvolupat són:

- Poder carregar línies poligonals amb angles de 0 i 180 graus i que puguin creuar-se o estar juntes.
- Donar la possibilitat de carregar rectangles, quadrats i polígons en general.
- Poder carregar cercles o altres figures sense costats rectes.
- Poder carregar altres formats de fitxers vectorials o inclús poder dibuixar al moment per pantalla el que es vol carregar.
- Poder generar edificis sencers.
- Poder definir la ruta i nom del fitxer que es vol carregar.
- Implementar algun sistema multi resolució el qual permeti tractar inicialment cada paret com a un sol objecte i que a mesura que interressi una zona concreta es pugui anar dividint per així aconseguir parts més petites. D'aquesta manera s'aconseguiria reduir el problema de la quantitat d'objectes i es podria tenir molts més objectes a l'escena amb una major rapidesa.
- Integrar l'aplicació en un motor de jocs per aconseguir un millor resultat gràfic o sinó, donar la possibilitat de que els objectes generats puguin ser exportats a XML per així poder-los importar des d'un motor de jocs.

4. MANUAL D'US DE L'APLICACIÓ

Per instal·lar i executar correctament l'aplicació s'han de seguir els següents passos:

- Instal·lar l'aplicació en el directori desitjat
- En cas de no tenir cap fitxer SVG crear-lo a través de l'Adobe Illustrator o algun altre programa de creació de fitxers SVG
- Col·locar el fitxer SVG en el directori d'instal·lació amb el nom "paret.svg"
- Executar l'aplicació i gaudir d'ella

Nota: Si encara no s'ha fet, en cas de disposar d'una targeta gràfica GeForce Sèrie 8, Sèrie 9 o Sèrie 200 o la targeta AGEIA PhysX accelerator cal instal·lar el driver corresponent de la web de NVIDIA que permet simular la física a través d'aquest hardware.

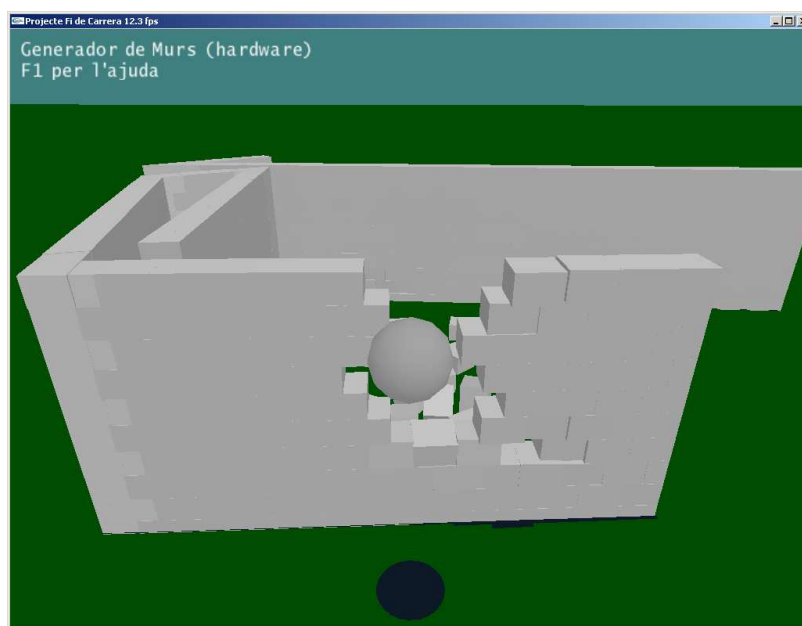


Fig. 4.1: Imatge de l'aplicació desenvolupada

5. CONCLUSIONS

L'objectiu principal d'aquest projecte era investigar les diferents llibreries de simulació física que hi ha actualment al mercat i comparar-les per saber quines prestacions donava cadascuna.

A més una de les dificultats que tenen aquestes llibreries és que la creació de contingut no és trivial ja que moltes vegades per crear una simple caixa s'ha de cridar una funció amb una multitud de paràmetres. Per aquest motiu doncs, es va decidir triar una d'aquestes llibreries analitzades i realitzar una aplicació, amb la qual, a partir d'una imatge en planta d'un o més murs o parets en format vectorial, es pogués generar l'equivalent d'aquesta imatge en 3D preparat per comportar-se segons la física de Newton.

Després de realitzar aquest projecte podem concloure que s'han assolit els objectius de la següent manera:

- S'han cercat i analitzat diferents llibreries físiques del mercat aconseguint així saber les prestacions de cada una i podent realitzar un quadre comparatiu.
- S'ha escollit la llibreria PhysX per ser la més prometedora per l'aplicació que es volia realitzar.
- S'ha desenvolupat una aplicació la qual permet:
 - Generar murs en funció de la imatge d'un fitxer SVG.
 - Visualitzar els murs generats.
 - Poder interaccionar amb els murs generats llençant projectils.

Aquest projecte ha estat una aproximació en el món de la simulació física en els videojocs i amb ell he descobert un nou camp de desenvolupament que permet aconseguir més realisme en les aplicacions que volen simular la realitat. A més a part descobrir l'apassionant món de la simulació física també m'ha servit per veure altres camps com són els motors de jocs, els diferents formats d'imatge, la visualització 3D, i molts més.

També m'ha servit per reflexionar què passaria si aquest projecte enlloc de fer-lo una persona sola el fes un equip de programadors com sol passar a la majoria de videojocs. Potser seria aleshores quan et pots trobar més diversitat d'opinions i valorar quina d'elles és la més adequada. També llavors es podria especialitzar varies persones per cada tasca i per exemple així uns es podrien dedicar a fons a carregar un fitxer svg, uns altres a dissenyar la construcció del mur, uns altres a implementar-lo, uns per renderitzar, etc.

Finalment he vist que a cada subtema del projecte es pot aprofundir tant com es vulgui i que arriba a un punt que s'ha de trobar mesura i decidir si ja s'ha aprofundit prou o s'ha de continuar treballant amb ell.

Tot plegat ha estat el fruit d'anys d'estudi que són el que m'han permès realitzar aquest projecte i al mateix temps incrementar la meva formació com a informàtic i com a persona.

6. BIBLIOGRAFIA

PÀGINES WEB

LLIBRERÍES FÍSIQUES

Newton Game Dynamics

Web: <http://www.newtondynamics.com/>

Fòrum: <http://www.newtondynamics.com/forum/>

NVIDIA PhysX Technology

Web producte: http://www.nvidia.com/object/nvidia_physx.html

Web desenvolupadors: <http://developer.nvidia.com/object/physx.html>

Fòrum: <http://developer.nvidia.com/forums/index.php?showforum=16>

Open Dynamics Engine

Web: <http://www.ode.org/>

ODE Wiki: http://opende.sourceforge.net/mediawiki-1.6.10/index.php/Main_Page

Bullet Physics Library

Web: <http://www.bulletphysics.com/Bullet/wordpress/>

Fòrum: <http://www.bulletphysics.com/Bullet/phpBB3/>

Tokamak Physics Engine

Web: <http://www.tokamakphysics.com/>

Havok

Web: <http://www.havok.com/>

DICCIONARIS I ENCICLOPÈDIES

Gran Diccionari de la Llengua Catalana

Web: <http://ec.grec.net/lexicx.jsp?GECART=0>

Enciclopèdia Catalana

Web: http://www.enciclopedia.cat/fitxa_v2.jsp?NDCHEC=0

WordReference

Web: <http://www.wordreference.com>

Traductor de la Generalitat de Catalunya

Web: <http://traductor.gencat.cat>

LLIBRES

Fernando, R., & Kilgard, M.J. (2003) *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*, Addison-Wesley

Galán, M. (2007) *Blender: Curso de Iniciación*, Inforbook's

Junker, G. (2006) *Pro OGRE 3D Programming*, Apress