

Objektorienterad analys

Strike: local defensive

Projekt i kursen TDDC76, Linköpings Universitet

Projektmedlemmar:

- Jesper Otterholm
- Lage Ragnarsson
- Erik Sköld
- Rasmus Vilhelmsson
- Isak Wiberg
- Filip Östman

2014-11-11

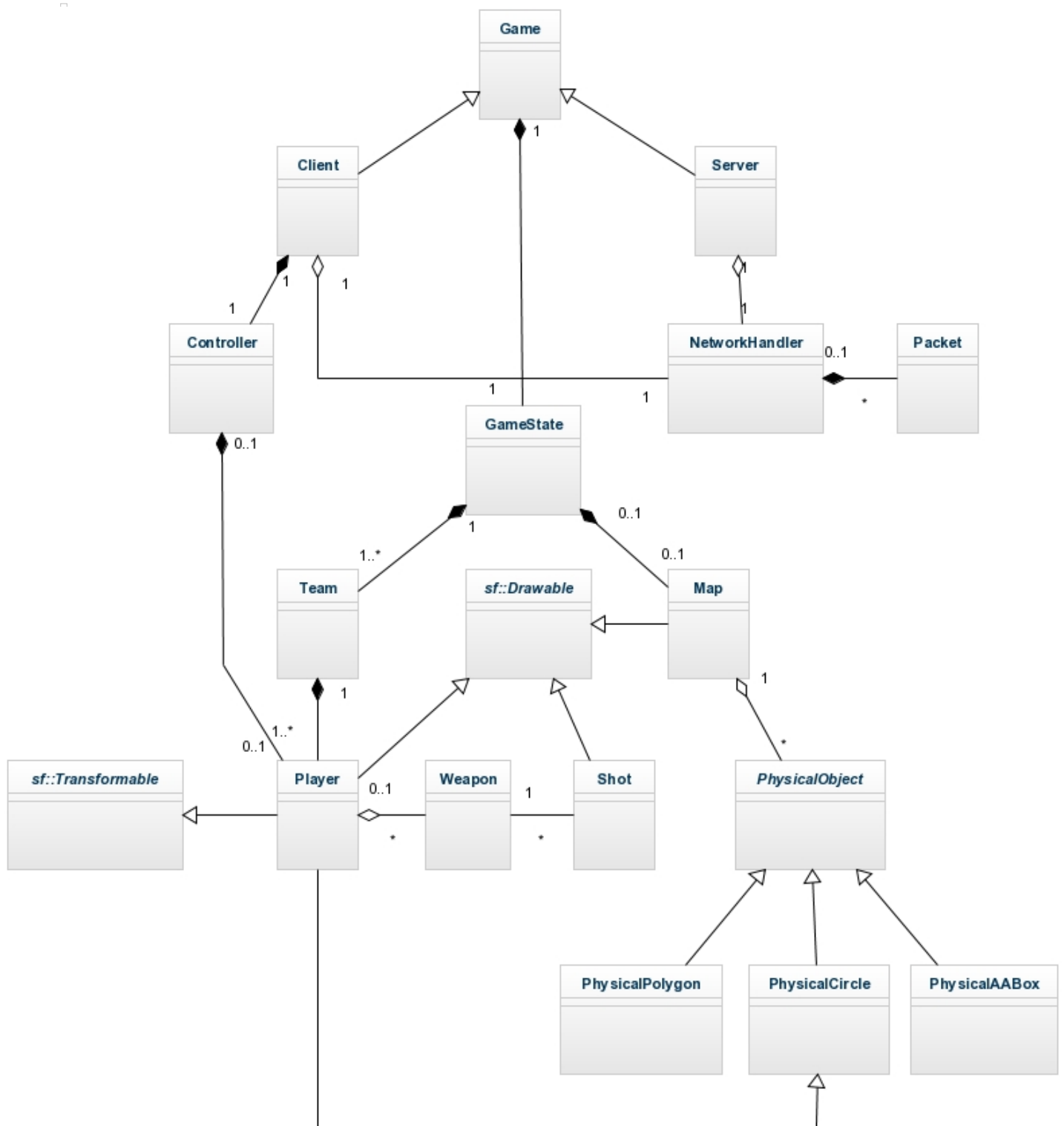
Sammanfattning

Detta dokument ger en översiktlig klasstruktur för projektet. Klasserna beskrivs dels i ett diagram samt senare kort i text. I diagrammet kan man också se hur klasserna relaterar till varandra.

Innehållsförteckning

1. Sammanfattning
2. Klassdiagram
3. Klassbeskrivningar
4. Användningsfall

Klassdiagram



Klassbeskrivningar

Game

Game är en abstrakt klass som har en gameloop och ett GameState-objekt.

Client

Client är en subklass till Game som skapas hos användaren när denne startar spelet. Den har en controller som hanterar användarens input. Dessutom har den samma networkhandler-objekt som server har. Detta objekt hanterar nätverkskommunikationen.

Server

Ärver från Game. I denna klass finns logiken för att köra en servers gameloop. Denna skiljer sig från Client då den till exempel inte behöver rita någon spelgrafik. Kommunikationen över nätverket kommer också att vara lite annolunda.

Packet

En klass för att kapsla in meddelanden för att skicka över nätverket. Kan skilja på packet av olika typ, t.ex positionsuppdateringar, skott, mm.

GameState

Detta objekt samordnar de olika delarna som beskriver spelsessionen. Den innehåller alla objekt som ska finnas i sessionen och alla objekt som skapas under sessionens gång.

Team

Klassen Team ska ha hand om spelare, deras lag och eventuellt ett poängsystem. Den är hårt knuten till GameState och Player är hårt bunden till Team.

Player

Player är klassen som hanterar spelaren, alltså liv, position och rörelse. Player är hårt bunden till Team och Controller. Player känner till weapon. Player ärver från sf:Transformable, sf:Drawable och PhysicalCircle.

Weapon

Weapon är en klass som hanterar spelarnas vapen. Det innefattar antal skott och hur mycket skada vapnet gör. Weapon är mjukt bunden till Player och känner till Shot.

Shot

Shotklassen ska hålla reda på position och rörelsevektor för avlossad projektil. Shot ärver från sf:Drawable och kommunicerar med Weapon.

sf::Drawable

`sf::Drawable` är en abstrakt klass i SFML-biblioteket som allt utritbart skall ärvas.

`sf::Transformable`

`sf::Transformable` är en abstrakt klass i SFML-biblioteket som ärvs av de klasser som ska kunna transformeras. Dvs förflyttas, roteras eller skalas.

`Map`

`Map` är en drawable för att den ska kunna rita ut sin bakgrundsbild. Den har en lista över physical objects som utgör banans geometri.

`PhysicalObject`

Abstrakt basklass för geometriska former som diverse linjer och strålar kan kollidera med.

`PhysicalPolygon`

Håller den geometriska beskrivningen av en godtycklig månghörning samt implementerar kollisionsalgoritmer för en sådan.

`PhysicalCircle`

Håller den geometriska beskrivningen av en cirkel samt implementerar kollisionsalgoritmer för en sådan.

`PhysicalAABBox`

Håller den geometriska beskrivningen av en axeljusterad rektangel samt implementerar kollisionsalgoritmer för en sådan.

`NetworkHandler`

En klass för att sköta kommunikation över nätverk. Innehåller mest metoder för att packetera och skicka data.

Användningsfall

Användaren startar en klient

Användaren startar den exekverbara spelfilen och kan därefter genom ett terminalfönster ange detaljer för att ansluta till en server.

En spelare skjuter ett skott

Gameloopen i Client registrerar alla knapptryckningar och skickar det vidare till Controller som hanterar fallet då skjutknappen är nedtryckt. Knappen var nedtryckt och Controller säger åt Player att han sköt. Player säger åt Weapon att avfyras som minskar sin ammunition och skapar ett Shot som får en startposition, riktningsvektor och möjligheten att rita ut sig själv eftersom det är en `sf::Drawable`.

Användaren startar en server

När man startar en server ska man kunna ange parametrar för hur servern ska bete sig, t.ex vilken bana som ska spelas, maximalt antal spelare. Servern kommer vara ett terminalprogram som användaren interagerar med när spelsessionen ska startas.