* programmet ska kodas i C++ och använda grafikbiblioteket SDL (närmare bestämt SDL2)
* objektorienterad programmering ska användas: programmet ska vara uppdelad i klasser med användning av oo-tekniker som inkapsling, arv och polymorfism
* tillämpningsprogrammeraren ska skyddas mot att göra svårupptäckta fel som att använda värdesemantik för objekt av polymorfa klasser
* det ska finnas en gemensam basklass (säg Sprite) för alla figurer, denna basklass ska vara förberedd för att vara en rotklass i en klasshierarki (om tillämpningsprogrammet önskar göra subklasser till den)
* inkapsling: datamedlemmar ska vara privata om inte väldigt speciella skäl föreligger
* det ska inte finnas något minnesläckage, du ska se till att dynamiskt allokerat minne städas bort
* spelmotorn ska kunna ta emot input (tangentbordshändelser, mushändelser) och reagera på dem enligt tillämpningsprogrammets önskemål, eller vidarebefordra dem till tillämpningens objekt
* enkel kollisionsdetektering: man ska kunna kolla om den omgivande rektangeln för en Sprite har kolliderat med den omgivande rektangeln för en annan Sprite
* programmet ska vara kompilerbart och körbart på en dator under både Linux och MS Windows (alltså inga plattformspecifika konstruktioner) med SDL 2 och SDL\_ttf, SDL\_image och SDL\_mixer
* man ska kunna ange en högsta "frame rate" (antalet frames per sekund)
* det ska finnas olika typer av Sprites i en klasshierarki i den generella delen, det kan t.ex. vara rörliga Sprites (för figurer som flyttar sig själva och där man kan ange hur mycket de ska flytta sig i x- resp. y-led per tick) och orörliga Sprites. Objekt av dessa klasser ska endast kunna skapas dynamiskt och värdesemantik ska vara förbjuden för dem
* en av subklasser till Sprite ska kunna vara en animerad figur, representerad internt antingen med flera bilder som växlar efter ett angivet antal tick eller med en sprite sheet
* spelmotorn ska vara förberedd för att tillämpningen vill installera kortkommandon (enkla tangenttryckningar). Tillämpningen ska kunna ange en tangent och en call back-funktion som ska anropas då denna tangent har tryckts ner
* spelmotorn ska vara förberedd för hantering av spelnivåer (levels). Tillämpningar ska kunna skapa flera olika "scener" bestående av olika Sprite-objekt och bakgrundsbilder och lätt växla mellan dem
* kortkommandon (tangenttryckningar) ska kunna knytas även till medlemsfunktioner i angivna objekt (förutom till fria funktioner). Det ska vara lätt för tillämpningen att ange antingen en funktionspekare till en fri funktion eller kombinationen av en objektpekare och en funktionsmedlemspekare tillsammans med tangenten
* det ska finnas en grafisk komponent (Sprite-subklass?) som ska kunna användas som ett inmatningsfält för text, som tillämpningen kan använda för textinmatning (t.ex. namn på spelaren för en high score-lista). Det behöver inte (men får gärna) finnas editeringsmöjligheter för inmatningsfältet.
* kollisionsdetektering ska göras på pixelnivå, med hänsyn tagen till genomskinliga pixlar
* spelmotorn ska implementera viss funktionalitet hos en fysikmotor, nämligen att rörliga Sprites kan påverkas av en gravitation samt ska kunna studsa mot varandra vid kollision osv. Sprites ska ha en viss elasticitet som påverkar deras studs-beteende.