Tenebris

Grupp 24

Designdokument

V. 1.2

2018-05-05

# Dokumenthistorik

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Beskrivning | Författare |
| 180411 | <1.0> | Skapandet av designdokument | Dionist Peci, Anton Persson |
| 180505 | <1.2> | Uppdatering av designdokument. | Anton Persson |
| 180507 | <1.3> | Uppdatering av klassdiagram beskrivningar. | Dionist Peci |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Innehåll**

[Dokumenthistorik 2](#_gjdgxs)

[Designdokument](#_30j0zll) 3

[Syfte](#_1fob9te) 3

[Ordlista](#_3znysh7)3

[Systemdiagram](#_tyjcwt) 4

[Användningsfallsdiagram](#_3dy6vkm) 5

[Scenarion/Användningsfallsbeskrivningar](#_1t3h5sf) 5

[Användargränssnitt](#_2s8eyo1) 6

# Designdokument

# Syfte

Syftet med det här dokumentet är att förklara designen på vårt spel. Det innehåller ett klassdiagram som förklarar vad varenda klass gör och hur de relaterar till varandra. En ordlista som förklarar alla ord som vi har använt, ett användningsfallsdiagram som förklarar vad användaren kan göra med spelet. Slutligen har vi ett fåtal bilder på hur vi har tänkt att vårt gränssnitt ska se ut, med kompletterande text.

# Ordlista

Tangent inputs : Helt enkelt handlingen av att trycka på en tangent på tangentbordet

Mana: En resurs spelaren använder för att använda spells.

Spells: Olika saker spelaren kan göra i spelet, till exempel att kasta en eldboll på en fiende.

Inventory: Helt enkelt spelarens förråd, till exempel en väska där spelaren har alla sina saker i.

Key bindings: Key bindings är knappar som har en funktion, alltså knappen är bunden till någon funktion.

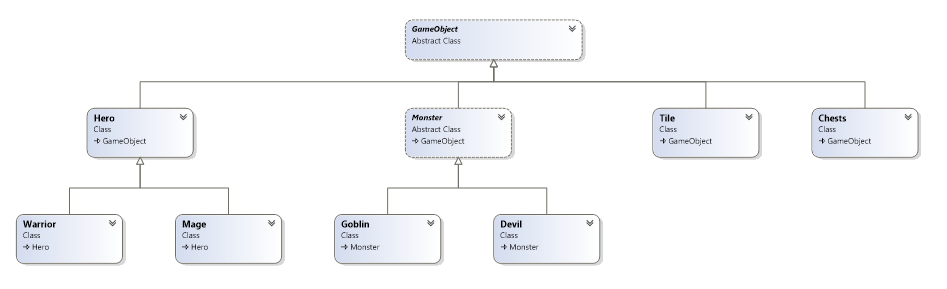
HP( Health Points): HP är spelarens liv resurs, om den blir noll så dör spelaren.

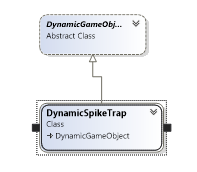
Platform: En struktur eller area som spelaren och fiender går på, till exempel flygande trottoar.

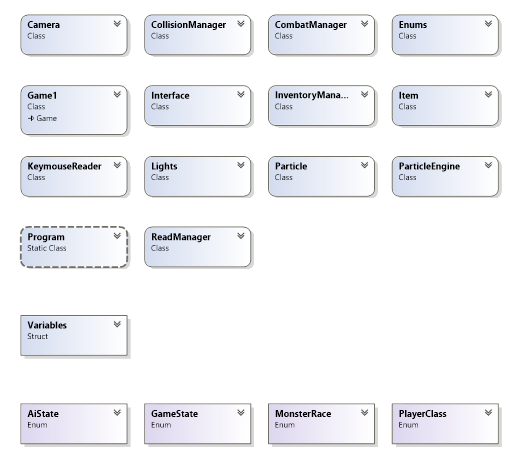
Gamestate: Vilken status spelet har för nuvarande. Denna bestämmer “vilken av spelen” användaren spelar.

# Klassdiagram

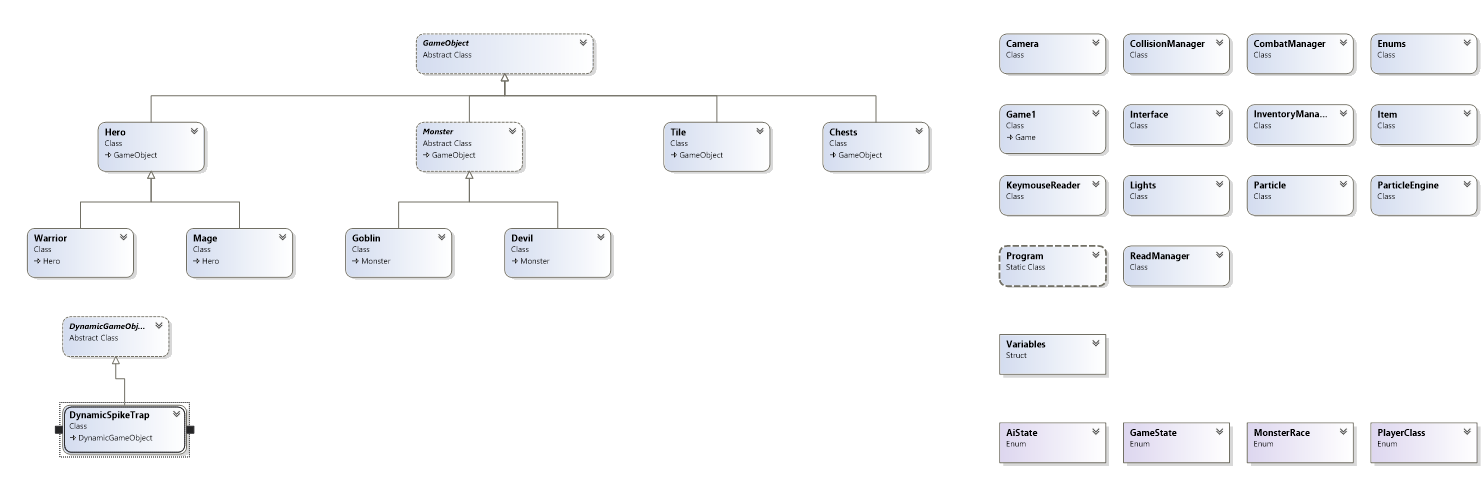
**Uppdelat klassdiagram:**







**Fullständig klassdiagram:**



# 

GameObject är en abstract klass som agerar som en mall för alla dens subklasser (spelobjekt) (superklass) till exempel: alla objekt i spelet kommer ha en draw metod och en update metod som de ärver från GameOjbect klassen.

Hero klassen är en mall för alla huvudkaraktärer i spelet, just nu har vi warrior och mage som instanser.

Monster klassen är en mall för alla fiender i spelet, just nu har vi goblin och devil som instanser.

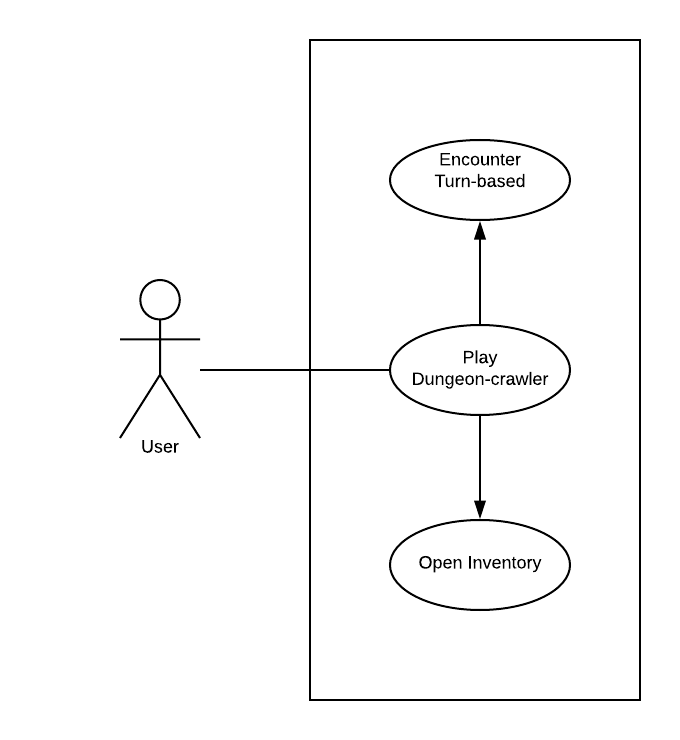
Tile klassen är för platformarna i spelet och chest klassen är kistor i spelet.

Alla manager klasserna hanterar deras respektive roll för programmet. Till exempel tar CombatManager hand om combat i spelet och InventoryManager inventory osv.

Klasserna som ligger under utility mappen är verktyg för spelet som får det att fungera. Camera används för att spelarkaraktären alltid ska vara i mitten av skärmen. Interface är gränssnittet, particle och particle engine är partiklar, lights är för ljuset, enums är för alla states i spelet till exempel gamestate eller playerclass.

**Scenario/Användningsfallsbeskrivningar**

# Användningsfallsdiagram



Varje use case beskrivs i ett eget use case dokument som utförligt går igenom varje nuvarande funktion i enlighet till hur användaren upplever spelet.

**<Play Dungeon-crawler>**

Beskriver alla funktioner som första “Gamestate”:et innehar i ett kontinuerligt flöde. Det innehåller input från användaren och hur systemet reagerar på eventuell input. Första “Gamestate”:et är själva dungeon-crawler delen av spelet. Allt från karaktärens rörelse till spelets pussel mekaniker. Referera till “Use case: Dungeon-crawler” för vidare information.

**<Encounter Turn-based combat>**

Beskriver alla funktioner som andra “Gamestate”:et innehar i ett kontinuerligt flöde. Det innehåller input från användaren och hur systemet reagerar på eventuell input. Andra “Gamestate”:et är ett turn-based combat system. Allt från attack animationer till spell effekter.

Referera till “Use case: Encounter turn based combat” för vidare information.

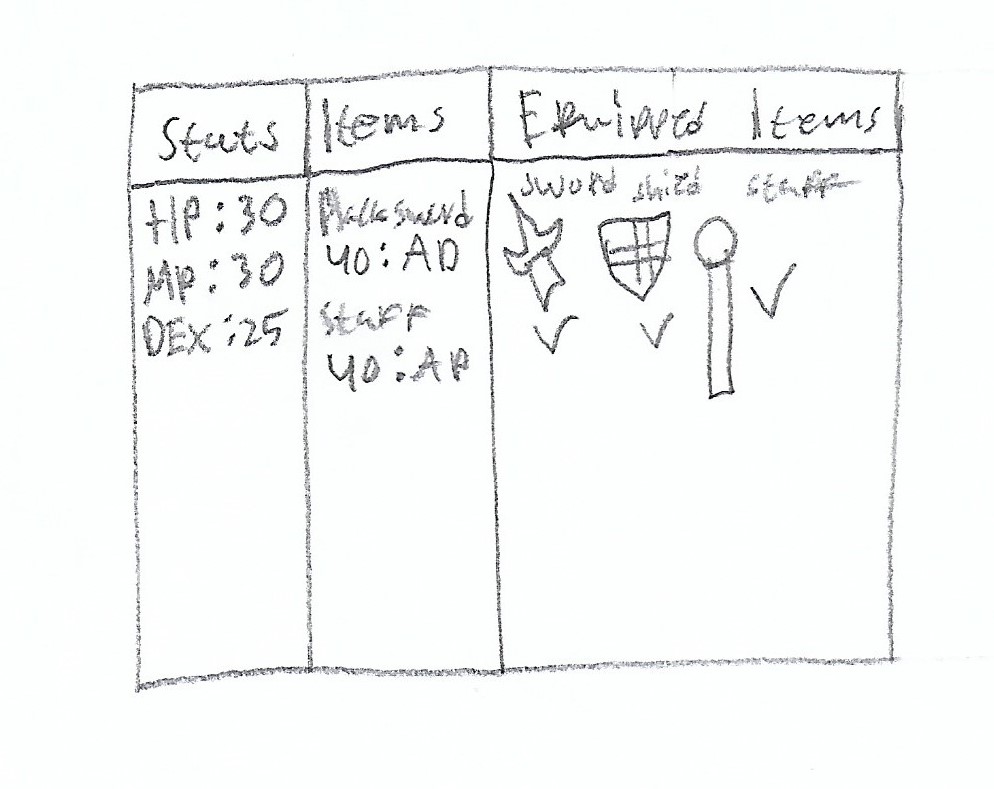
**<Open Inventory>**

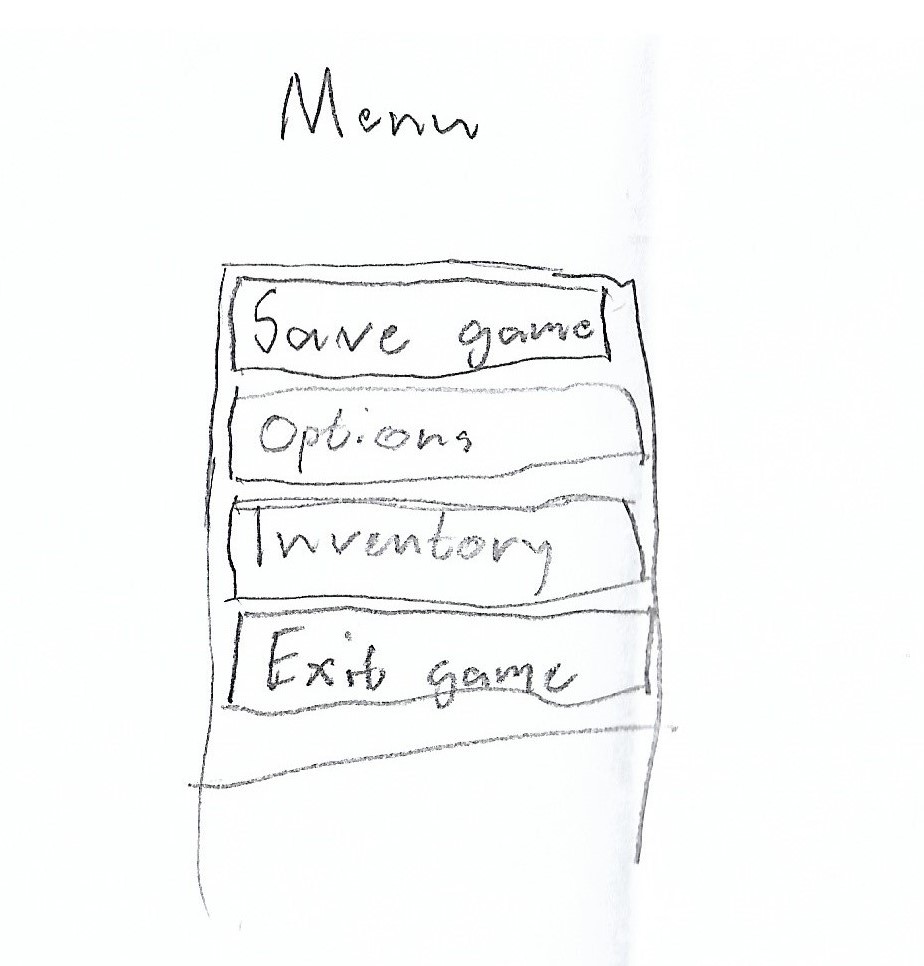
Beskriver alla funktioner som inventoryn innehar i ett kontinuerligt flöde. Det innehåller input från användaren och hur systemet reagerar på eventuell input. Allt från attribute points system till inventory management. Referera till “Use case: Open inventory” för vidare information.

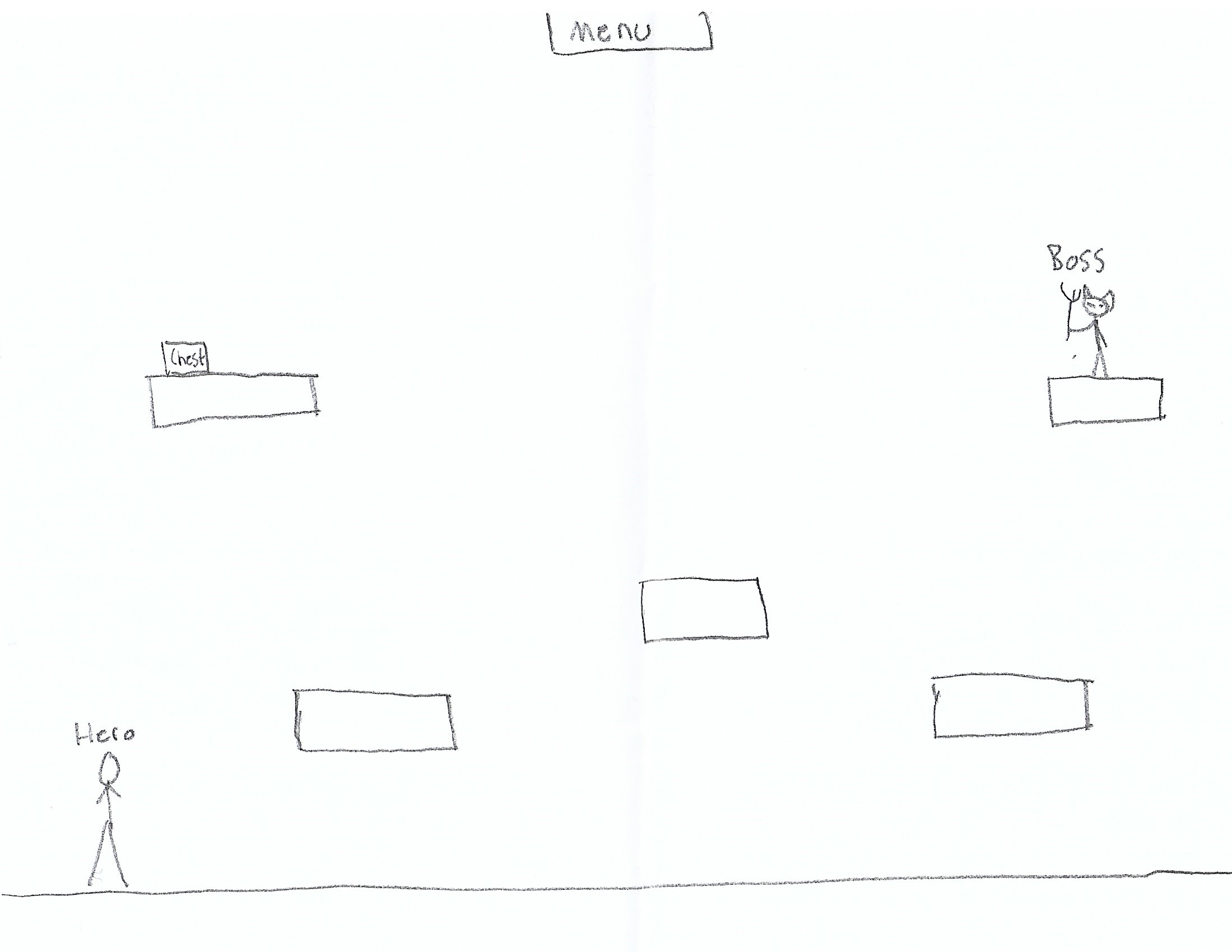
# Användargränssnitt

# 

Bilden visar hur det ska se ut när man möter bossar. Man kan se sitt egna liv, mana och motståndarens liv. Om man trycker på knapparna kommer man se alternativen i function window.

Bilden visar hur inventoryn ska se ut. Man kan alltså se informationen om items, vilka items man har och vilka items som man använder. Man kan även se sina stats. 

Detta är menyn som man kan få upp. Tanken är att inventoryn ska vara key bound men eftersom man kanske inte vet detta har vi options där man kan se key bindings och även att man kan nå inventory därifrån. Spelet kommer att pausas när denna meny är uppe.

Bilden visar hur spelet kommer att se ut när man hoppas på plattformar. Detta är en grov skiss och flera saker kommer att implementeras. Det kan förekomma fler mindre monster men också flera fällor och dekorationer.