

TDP005 Projekt: Objektorienterat system

Designspecifikation

Författare

Martin Kuiper, marku849@student.liu.se

Jim Teräväinen, jimte145@student.liu.se

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Första utkast för designspecifikationen.	2020-11-27

2 Klassbeskrivningar

2.1 Player

Player ska kunna uppdatera sin position, beroende av vilka knappar som är intryckta samt vilket state Player har. Player har en överlagrad render()-funktion, då den behöver kunna uppdatera hur den renderas beroende på vad som händer Player-objektet. Funktionen tick() kallar alla funktioner som ska köra när spelet uppdateras.

2.1.1 Datamedlemmar

- state - påverkar hur objektet beter sig när tick() kallas.
- weapon - ett objekt av klassen Weapon ägs av Player.
- immune - hur länge Player ska vara osårbar.
- health - ärvd från Character, minskar när Player tar skada.
- damage - ärvd från Character, påverkar Players vapens skada.
- speed - ärvd från Character, påverkar hastigheten av objektets förflyttning.
- shape - ärvd från Textured Object, påverkar hur objektets textur ritas ut.
- center - ärvd från Game_Object, bestämmer objektets position på skärmen.
- hitbox - ärvd från Game_Object, bestämmer storleken på objektets kollisionsområde.

2.1.2 tick(Time)

För att flytta Player kallas move_player().

För att flytta och skjuta vapnet kallas move_weapon().

Player frågar World vilka objekt den kolliderar med. Kolliderar Player med ett Enemy-objekt kallas take_damage().

Datamedlemmen 'immune' minskas med Time om den inte är 0.

2.1.3 move_player()

Om någon av tangenterna som ska flytta spelarkarakteren till höger eller vänster är intryckta ska Players position uppdateras i den riktningen, så länge Player inte befinner sig vid banans kant.

Normalt har spelaren state 'standing'. Har spelaren state 'falling' flyttas spelaren mot botten av skärmen. Har spelaren state 'jumping' flyttas spelaren mot toppen av skärmen.

När spelaren trycker på hopp-knappen ändras state till 'jumping' om det inte redan är 'jumping' eller 'falling'. När spelaren har haft state 'jumping' en viss tid ändras state till 'falling'. Om den understa delen av Player-objektet kolliderar med den översta delen av en plattform sätts 'falling' till 'standing'.

2.1.4 move_weapon()

Players position bestämmer Weapon-objektets position. Input från tangentbord eller muspekare bestämmer vapnets rotation. Vapnet har en datamedlem 'cooldown' som påverkar hur ofta det kan avfyras. Är 'Avfyravapen'-knappen intryckt och vapnets 'cooldown' 0 skapas en projektil. Projektillen rör sig i riktningen av vapnets rotation och har en datamedlem 'damage' som sätts till Players 'damage'.

2.1.5 take_damage(int)

Players 'health' subtraheras med skadan som skickas som parameter till funktionen.

Datamedlemmen 'immune' sätts enligt hur länge Player ska vara odödlig efter att ha tagit skada.

2.2 Hud

HUD står för Heads-Up Display och har som syfte att visa relevant information för spelaren. Hud-klassen ska visa Player-objektets nuvarande hälsa (health) samt spelarens insamlade pengar.

2.2.1 Datamedlemmar

- player_ptr - pekare till player för att kunna hämta dess hälsa.
- shape - ärvd från Textured Object, påverkar hur objektets textur ritas ut.
- center - ärvd från Game_Object, bestämmer objektets position på skärmen.
- hitbox - ärvd från Game_Object, bestämmer storleken på objektets kollisionsområde.

2.2.2 tick(Time)

Player-objektets hälsa hämtas och en grafik med den informationen ritas ut i kanten av skärmen ovanpå alla andra objekt.

Mängden insamlade pengar hämtas från Game_States datamedlem player_info och renderar detta på liknande sätt.