

TDP005 Projekt: Objektorienterat system

Designspecifikation

Författare

Elliot Johansson, elljo130@student.liu.se

Lukas Freyland, lukfr510@student.liu.se

Nadim Lakrouz, nadla777@student.liu.se

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Första utkast	2022-11-25

2 Player

Syftet med playerklassen är att representera den karaktär spelaren styr. Spelaren kan gå runt med karaktären och samtidigt använda magiska formler i form av cirklar runt spelaren. Formlerna kommer att följa spelarens rörelser. Spelaren kan röra sig runt hela spelplanen men inte utanför.

Funktionen `regenerate_mana` kommer sakta återställa manapoäng med `sf::Clock` och `sf::Time`. `Hitpoint` och `mana` visar spelaren hur mycket liv hen har kvar samt om hen kan använda en magisk formel. `Hitpoint` och `mana` visas i form av "bars" som ändras i storlek procentuellt med hur många `hitpoints` samt `mana` finns kvar.

2.1

- `double hitpoint{}` - Håller antalet "hitpoints" spelaren har vilket är spelarens liv.
- `int hitpointMax{}` - Håller ett fast värde på det maximala antal "hitpoints". Används för att få fram ett procentuellt värde av hur många `hitpoints` som finns.
- `double HpPercent{}` - Ett procent värde som används för att veta hur stor HP-baren ska vara relaterbart med hur många "hitpoints" spelaren har kvar.
- `double mana{}` - Håller antalet mana spelaren har. Mana bestämmer om spelaren kan använda en trollformell.
- `double manaMax{}` - Ett fast värde på det maximala antalet mana spelaren har. Används för att få fram ett procentuellt värde av hur mycket mana som finns.
- `double ManaPercent{}` - Antal procent av mana jämfört med `manaMax`. Används för att veta hur stor Mana-baren ska vara relaterbart till antalet mana.
- `sf::Clock m_clock` - sfml variabel för att hantera tiden.
- `sf::Time m_elapsed` - sfml variabel för att hantera tiden.
- `void RestartClock()` - Startar om tiden. Mana ska uppdateras en gång varje intervall och `RestartClock()` startar om intervallet.
- `void updatePlayerGUI()` - Uppdaterad "Graphical User Interface" för spelarens Hp- och Mana-bar.

- `int set_hp()` - Ändrar privata variabeln "hitpoint" när spelaren ska förlora eller få tillbaka hitpoints.
- `int get_hp()` - Hämtar privata variabeln hitpoint för att bl.a. stoppa spelaren från att få mer hitpoints än hitpointMax.
- `int get_mana()` - Hämtar privata variabeln mana för att bl.a. se till att trollformler bara kan användas när spelaren har mana.
- `void regenerate_mana()` - Använder sig av `m_clock` och `m_elapsed` för att öka privata variabeln mana ett bestämt antal varje intervall.
- `void update()` - uppdaterar mana, position, återställer klockan och anropar `updatePlayerGUI()`.
- `void renderPlayerGUI(sf::RenderTarget & target)` - Renderar spelarens Hp- och Mana-bars med bestämd längd och position.
- `void render(sf::RenderTarget & target)` - Renderar spelarens karaktär.
- `void resetPlayer()` - Återställer spelaren om spelaren skulle dö.

3 Enemies

”Enemies” klassens syfte är att representera fienderna som i intervaller kommer förflytta sig mot spelaren på spelplanen. Vid kollision med magiska formler så kommer det påverka fienderna på olika sätt. Exempelvis kommer de antingen dö eller bli stoppade beroende på vilken magisk formel som spelaren använder.

Vid kollision med spelaren tar spelaren skada fram tills spelaren inte längre är i kontakt med fienden.

3.1

- float size - Storlek på fiender. Bestämmer radien på circle.
- double xMovement - Fienders rörelse i X-led.
- double yMovement - Fienders rörelse i Y-led.
- sf::CircleShape circlesize - Formen på en fiende. Används som hitbox för att rita ut fienden.
- sf::Vector2f location - Förvarar fienders position.
- sf::Texture texture - Innehåller Fienders textur.
- std::vector <sf::Texture*> textures - En container som förvarar alla fienders olika texturer.
- Enemy(float size, sf::Texture &texture) - Skapar ett fiende objekt med bestämd storlek och textur och sätter texturen till circle.
- void update(sf::Vector2i pos) - Uppdaterar fienders position enligt xMovment och yMovment.
- void render(sf::RenderTarget & target) - Renderar circle.
- sf::CircleShape getCircle() - Hämtar circle.
- void setLocation(float x, float y) - Bestämmer positionen av fienderna.
- sf::Vector2f getLocation() - Hämtar position av fiender
- int getDistanceCircles(Enemy& otherEnemy) - Hämtar avståndet mellan två fiender.
- bool checkCollision(Enemy& otherEnemy) - Kontrollerar om två fiender överlappar varandra med hjälp av getDistanceCircles(Enemy& otherEnemy) funktionen.
- double getXMovement() - Hämtar rörelser i X-led.
- double getYMovement() - Hämtar rörelser i Y-led.

- `void setMovement(sf::Vector2i pos)` - Bestämmer vilket riktning fiender ska gå mot då de ska alltid röra sig mot spelaren.
- `sf::Vector2f spawnPoint()` - Bestämmer var fienders skapas på spelplanen.

4 Diskussion och filhantering

Vår design är uppbyggd av en Engine som har kontroll över flera "states". De här "statsen" är `Menu_state`, `Game_state` och `Gameover_state`, alla ärver av `Base_state`. De olika "statsen" gör det lätt att hantera olika spelfaser genom att skapa olika dedicerade objekt i de olika "statsen". Det går att dela upp hanteringen av objekten så att allt inte ligger på ett och samma ställe. Till exempel det är onödigt att ha en Player-objekt i `Menu_state` då Player-objekten endast behöver vara i `Game_state`.

Vårt program använder sig av en "resource_manager" som kan hantera olika typer av filer. På så sätt behöver vi bara använda en funktion för att hantera alla externa filer, istället för att ha flera olika funktioner vilket skulle leda till programmet körslångsammare. Flera funktioner skulle göra minneshantering svårare då det skulle skapas fler minnesläckor.

Hanteringen av "Spells" kunde ha gjorts bättre. Nu skapas olika "spell"-objekts för varje magiska formel. Det här kunde ha gjorts bättre genom att använda en vector av baspekare till "spells". Den vectorn kan iterera igenom alla "spells" och anropa deras "update" och "render" funktioner. På grund av tidsbrist kunde vi inte implementera vector-lösningen utan fick nöja oss med det vi hade.

I projektet används en config-fil där det går att ändra olika konstanter inom spelet. Bland annat går det att ändra mana och hp som spelar karaktären har, kan ändra hur mycket mana som ska regenereras samt hur snabbt fiender rör sig.

5 Diagram

