Ksenija Bidzāne

B\_Dig\_p-2

UI RPG

**Projektam tiek izpildīti šādi uzdevumi:** pēc spēlētāja nāves spēle apstājas, pēc pretinieka nāves parādās nākamais pretinieks. Ir iespēja ieslēgt un izslēgt vairogu, kas laika gaitā var salauzties.

**No papildu uzdevumiem tika veikti:** ir trīs dažādu veidu pretinieki, katrs ar unikālām spējām un ieročiem. Spēlētājs var izvēlēties vienu no trim ieročiem un mainīt tos cīņas laikā, katram ierocim ir savs efekts un bojājums.

**Kas ir redzams:** ekrānā mēs varam redzēt spēlētāja un pretinieku vārdu un veselību. Ir pogas uzbrukuma veikšanai, vairoga aktivizēšanai/deaktivizēšanai un teksts, kas parāda tā stāvokli, kā arī pogas ieroču izvēlei. Teksts, kas parāda spēlētāja un ienaidnieka nodarīto bojājumu apjomu.

**Mantošana:** tika izveidota bāzes klase ‘Character’ no kuras **manto** gan spēlētājs, gan pretinieki.

**Enkapsulācija:**’Berserker’ klasēi ir aggresion += aggresioGain; return aggresion / 10; **getter** struktūra. ‘Archer’ un ‘Goblin’ klasēm ir ‘ return totalDamage;’ **getter** struktūra. ‘Shield’ klasēi ir ‘return remainingDamage; ’ un ‘ return damage’ **getter** structūras. ‘Character’ klasēi ir ‘activWeapon = newWeapon;’ **setter** struktūra.

**Polimorfisms:** katrai no trim pretinieku klasēm ir ‘public **override** int Attack()’ funkcija, katrai no trim ieroču klasēm ir ‘public **override** void ApplyEffect(Character character)’ funkcija. ‘Character’ klasēi ir ‘ public void GetHit(int damage)’ un ‘public void GetHit(Weapon weapon)’ **overload** funkcija.

**Abstrakcija:** tika izveidota **abstrakta** ieroču klase ‘Weapon’ ar ‘ApplyEffect’ **abstracto** metodi un ‘GetDamage’ parasto metodi. No šīs abstraktās klases **manto** ‘Spear’, ‘Poisoned dagger’ un ‘Bow’ klases.