

BalloonBlitz igra

Tim: Andrea Koraca, Karla Grižić, Latica Grediček

1. OPIS IGRE

BalloonBlitz je 2D igra u kojoj igrač kontrolira balon koji se kontinuirano podiže sve dok ima goriva ili nastane kolizija sa preprekama. Igrač može pomicati balon ulijevo koristeći tipku A i udesno koristeći tipku D. Vertikalno pomicanje je automatsko i igrač njime ne može upravljati. Prepreke – ptice i oblaci se stvaraju nasumično na lijevoj ili desnoj strani ekrana i pomiču se horizontalno, a kada dotaknu kraj ekrana odbijaju se od istog i mijenjaju smjer kretanja. Kako bi se osiguralo da se balon prividno kontinuirano kreće vertikalno, a da prepreke nemaju samostalno vertikalno kretanje postavljena je maksimalna visina, odnosno vertikalni prag do kojeg se balon može podići, a da i dalje ostane vidljiv na ekranu, a kada balon dosegne taj prag ostatak svijeta se pomiče prema dolje kako bi se dobio dojam daljnjeg vertikalnog podizanja balona. Balon može pokupiti power-upove koji mu privremeno daju neku dodatnu mogućnost. Može pokupiti gorivo koje može biti manje ili veće pa tako manje daje 30%, a veće 70% dodatnog goriva, a može pokupiti i dulji ili kraći štit koji mu daje 10 ili 20 sekundi mogućnosti kolizije sa preprekama bez gubitka života, odnosno imunost na prepreke. Ukoliko balon nema aktivan štit svaka kolizija sa preprekama mu oduzima po jedan život, a ukoliko se iskoriste sva tri moguća života igra završava. Isto tako se kroz igru konstantno smanjuje količina dostupnog goriva koja kada dođe na 0% oduzima život balonu. Bodovi se prikupljaju kontinuirano prolaskom vremena i postizanjem visine. Cilj igre je postići što veću visinu uz izbjegavanje prepreka.

2. INSTALACIJA I POKRETANJE

Preduvjeti:

- Pygame 2.6.1
- Pygame (instalacija korištenjem pip install pygame)

Instalacija:

- Klonirati repozitorij sa GitHub-a
- Osigurati da je assets folder na ispravnoj lokaciji kako bi se prikazale slike balona, prepreka i power-upova
- Pokrenuti igru izvršavanjem BalloonBlitz.py datoteke

Struktura datoteka

- Balloon Blitz/
 - assets/
 - balloon.png
 - cloud.png
 - fuel.png
 - pigeon.png
 - shield.png
 - core/
 - factory.py
 - gameloop.py
 - gamestats.py
 - settings.py
 - objects/
 - balloon.py
 - button.py
 - decorator.py
 - obstacle.py
 - powerup.py
 - scoreboard.py
 - BalloonBlitz.py

3. ZAHTJEVI

Statički funkcionalni zahtjevi:

1. Pritiskom na tipke a i d balon se mora pomicati horizontalno
2. Balon se mora konstantno podizati dok god ima goriva
3. U trenutku kolizije balona sa oblakom ili pticom igra mora završiti
4. U trenutku nestanka goriva igra mora završiti
5. Oblaci i ptice se cijelo vrijeme pomiču horizontalno, a u trenutku sudara s rubovima prozora mijenjaju horizontalan smjer

6. Oblaci i ptice se ne kreću vertikalno, a izlaskom iz prozora se brišu iz memorije
7. Cijelo vrijeme se mora prikazivati i osvježavati trenutačno postignuta visina kao i najviša postignuta visina od svih pokušaja
8. Balon može pokupiti barem dva različita *power-upa*
9. Štit *power-up* određeno vrijeme omogućava koliziju balona s preprekama bez kraja igre
10. Gorivo *power-up* povećava količinu goriva u balonu

Dinamički funkcionalni zahtjevi:

1. Promjena postavki igre (brzina balona, brzina prepreka ...) na jednom mjestu
2. Dodavanje novih vrsta prepreka i mogućnost upravljanja preprekama kao flotom i jednostavno računanje kolizija
3. Mogućnost dodavanja novih vrsta *power-upova*
4. Jednostavna mogućnost promjene postojećih prepreka
5. Jednostavna promjena načina stvaranja objekata u igri (više/manje prepreka, više/manje *power-upova*)

4. KAKO IGRATI

Kontrole:

- Pritiskom na tipku A balon se pomiče ulijevo
- Pritiskom na tipku D balon se pomiče udesno

Mehanika igre:

- Balon se konstantno podiže sve dok ima goriva
- Gorivo se smanjuje tijekom vremena i kada dođe na 0% gubi se život
- Kolizijom sa preprekama se također gubi život ukoliko balon nema aktivan štit

- Kolizijom sa power-upovima balonu se dodaje određena količina dostupnog goriva ili aktivira štit

Kraj igre:

- Igra završava kada su izgubljeni svi životi
- Prikazuje se Play gumb za ponovno pokretanje i resetiranje igre (high score – najveća postignuta visina ostaje sačuvan.

5. ARHITEKTURA KODA

5.1. Core

factory.py

Sadrži factory klase koje kreiraju objekte igre.

Klase:

- BalloonFactory – tvornica za kreiranje balon objekta
 - create_balloon(): kreira instancu balona
- ObstacleFactory – tvornica za kreiranje prepreka
 - create_obstacle(): kreira prepreke – ptice i oblaci
- PowerUpFactory – tvornica za kreiranje power-upova
 - create_powerup(): kreira power-upove – gorivo i štit

Koristi se obrazac tvornice kako bi se kreiranje objekata igre razdvojilo od logike igre što olakšava izmjenu i proširenje stvaranja objekata ukoliko bude potrebe za tim.

gameloop.py

Rukovanje glavnom petljom igre. Procesiranje evenata (pritisci na tipke ili pritisak miša), ažuriranje objekata, pomicanje svijeta, detekcija kolizija, bodovanje, prikazivanje objekata na ekranu.

Klase:

- GameLoop:

- `check_events()`: procesira ulazne evente (tipkovnica i miš)
- `start_new_game()`: resetira stanje igre za pokretanje nove
- `display_message()`: postavlja statusnu poruku ili obavijest o prikupljenom power-upu
- `reset_position()`: resetira poziciju balona na startnu
- `update_game()`: ažurira objekte u igri, pomicanje, kolizije i bodovanje
- `spawn_obstacles()`: nasumično stvaranje prepreka korištenjem `ObstacleFactory`-a
- `spawn_powerups()`: nasumično stvaranje power-upova korištenjem `PowerUpFactory`-a
- `update_screen()`: renderira sve objekte i poruke na ekranu
- `run_game()`: glavna petlja u kojoj se koristi `clock` kako bi se osigurala neovisnost o frame rate-u

gamestats.py

Prati statistiku igre - životi, bodovi (visina), high score (najveća postignuta visina), gorivo, status igre (aktivna/neaktivna)

Klase:

- Singleton: Metaklasa koja osigurava postojanje samo jedne `GameStats` instance.
- `GameStats`: Klasa za praćenje statistike igre
 - `__init__()`: inicijalizira statistiku
 - `reset_stats()`: Resetira statistiku koja se mijenja - gorivo, životi i bodovi, ali ne i high score (najveću postignutu visinu)

Koristi se obrazac jedinstveni objekt kako bi se osiguralo postojanje samo jedne `GameStats` instance u igri.

settings.py

Definira sve postavke igre Balloon Blitz – dimenzije ekrana, brzine kretanja balona, prepreka i power-upova, stope stvaranja prepreka i power-upova, bodovanje, dostupno gorivo i slično.

Klase:

- Settings
 - `__init__()`: inicijalizira statičke postavke
 - `initialize_dynamic_settings()`: Inicijalizira postavke koje se mijenjaju tijekom igre.

5.2. Objects

balloon.py

Definiranje klase Balloon koja predstavlja objekt kojim upravlja igrač.

Klase:

- Balloon – klasa koja predstavlja igračev balon
 - `__init__()`: inicijalizacija balon objekta. Učitava sliku balona, skalira ju i postavlja inicijalnu poziciju balona na ekranu te postavlja status štita.
 - `update()`: ažurira poziciju balona ovisno o kretanju i količinu goriva. Ako se balon kreće lijevo ili desno ažurira se x koordinata i osigurava se da balon ne izađe izvan granica ekrana. Ako ima dostupnog goriva balon se kreće vertikalno, a gorivo se smanjuje tijekom vremena.
 - `update_shield()`: ažurira timer štita, kada vrijeme istekne štit se deaktivira
 - `blitme()`: crta balon na ekranu

button.py

Definira se klasa Button za prikazivanje Play gumba za pokretanje igre.

Klase:

- Button – Play gumb se pokazuje kada igra završi ili je neaktivna.
 - `Init()`: postavljanje gumba
 - `prep_msg()`: renderira tekst gumba
 - `draw_button()`: crta gumb na ekranu

decorator.py

Definiranje dekoratora za dodavanje power-upova balonu.

Klase:

- PowerUpDecorator – bazni (apstraktni) dekorator za power-upove, dodaju se efekti balonu.
 - apply(): apstraktna metoda. Implementira se u podklasama.
- FuelPowerUpDecorator – povećavanje dostupne količine goriva za neki postotak.
 - apply(): primjenjuje efekt povećanja goriva.
- ShieldPowerUpDecorator – aktiviranje štita na balonu za određeno vrijeme (ovisno koji je štit pokupljen).
 - apply(): aktivira štit na balonu

Koristi se obrazac dekorator kojim se dodaje ponašanje power-upa balonu bez da se mijenja sama Balloon klasa.

obstacle.py

Definiraju se bazna klasa za prepreke i specifične prepreke - ptice i oblaci, kao i kompozit za grupiranje istih.

Klase:

- Obstacle – bazna klasa za prepreke
 - init(): inicijalizacija objekta prepreke, učitava se slika prepreke, postavlja pozicija stvaranja i brzina kretanja
 - update(): Pomicanje prepreka horizontalno, kad dođu do kraja ekrana odbijaju se i kreću u suprotnu stranu. Kada izađu iz ekrana brišu se iz igre.
- Bird – konkretna prepreka (ptica)
 - update(): proširuje update() iz Obstacle klase i zrcali sliku ptice ovisno o smjeru kretanja
- Cloud – konkretna prepreka (oblak)
- ObstacleComposite – grupiranje prepreka i upravljanje istima.

- add()
- remove()
- update()
- draw()

Koristi se obrazac kompozit za upravljanje skupom prepreka kao jednim entitetom.

powerup.py

Definiraju se power-up bazna klasa i specifične klase - gorivo i štit, kao i kompozit za grupiranje istih.

Klase:

- PowerUp – bazna klasa za power-upove
 - update(): ažurira vertikalnu poziciju power-upa.
- FuelPowerUp – nasljeđuje PowerUp i definira efekte (amount)
- ShieldPowerUp – nasljeđuje PowerUp i definira efekte (duration)
- PowerUpComposite – grupiranje power-upova i upravljanje istima.
 - add()
 - remove()
 - update()
 - draw()

Koristi se obrazac kompozit za upravljanje skupom power-upova kao jednim entitetom.

scoreboard.py

Prikazuje statistiku igre – bodove (visinu), najveće osvojene bodove (visinu), postotak dostupnog goriva i živote, power-up poruke (koji power-up je aktiviran) i status poruke iz GameLoop-a.

Klase:

- Scoreboard – upravlja prikazom statistike i poruka
 - prep_score(): renderira tekst trenutnih bodova (visine)
 - prep_high_score(): renderira tekst najviše osvojenih bodova (visine)

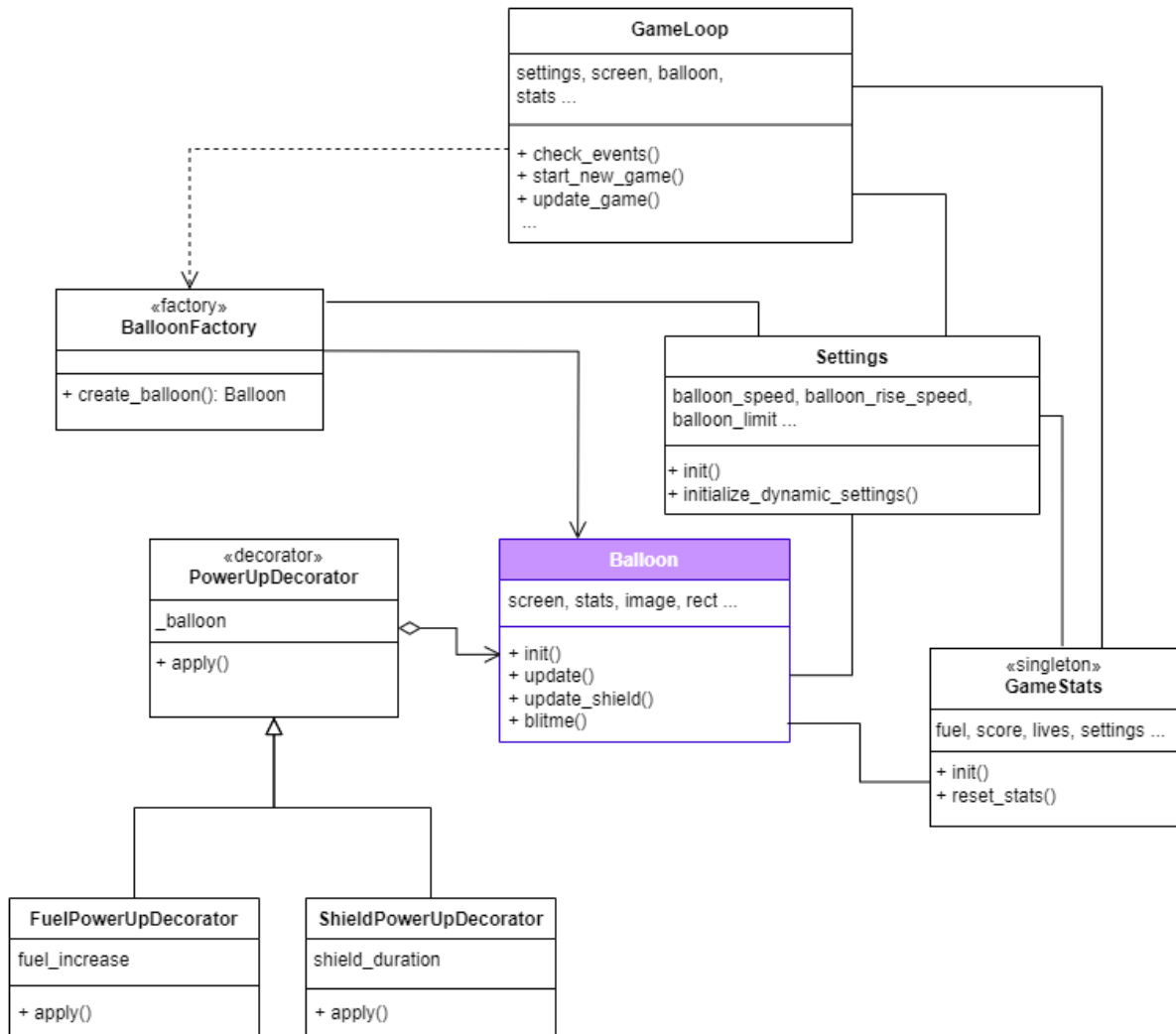
- `prep_fuel()`: renderira tekst trenutne količine dostupnog goriva
- `prep_lives()`: renderira tekst dostupnog broja života
- `set_active_powerup()`: postavlja poruku za trenutno aktivirani power-up
- `set_status_message()`: postavlja statusnu poruku
- `show_score()`: crta svu statistiku i poruke na ekranu

***BalloonBlitz.py* (main)**

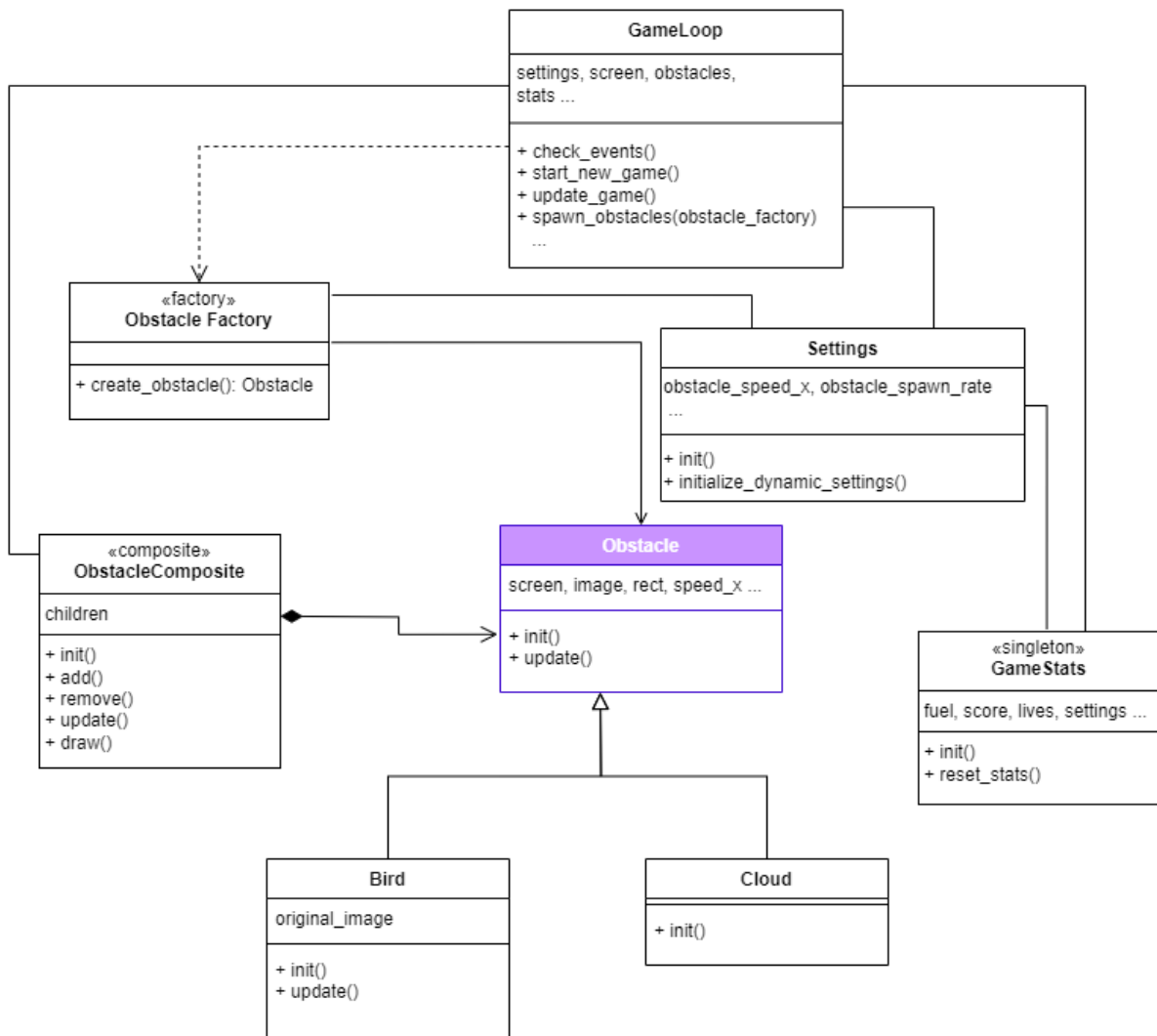
Ulazna točka igre, služi za pokretanje iste. Inicijalizira pygame, kreira instancu postavki (Settings) koja sadrži konfiguraciju igre, stvara glavni prozor prema dimenzijama iz postavki, inicijalizira GameStats (objekt statistike) koji prati bodove, high score, živote i gorivo, kreira također Scoreboard za prikazivanje statistike na ekranu. Kreira i Play gumb koji omogućava ponovno pokretanje igre te balon objektkoristeći BalloonFactory. Kreira kompozite za prepreke i power-upove te inicijalizira glavnu petlju igre (GameLoop).

6. UML DIJAGRAMI

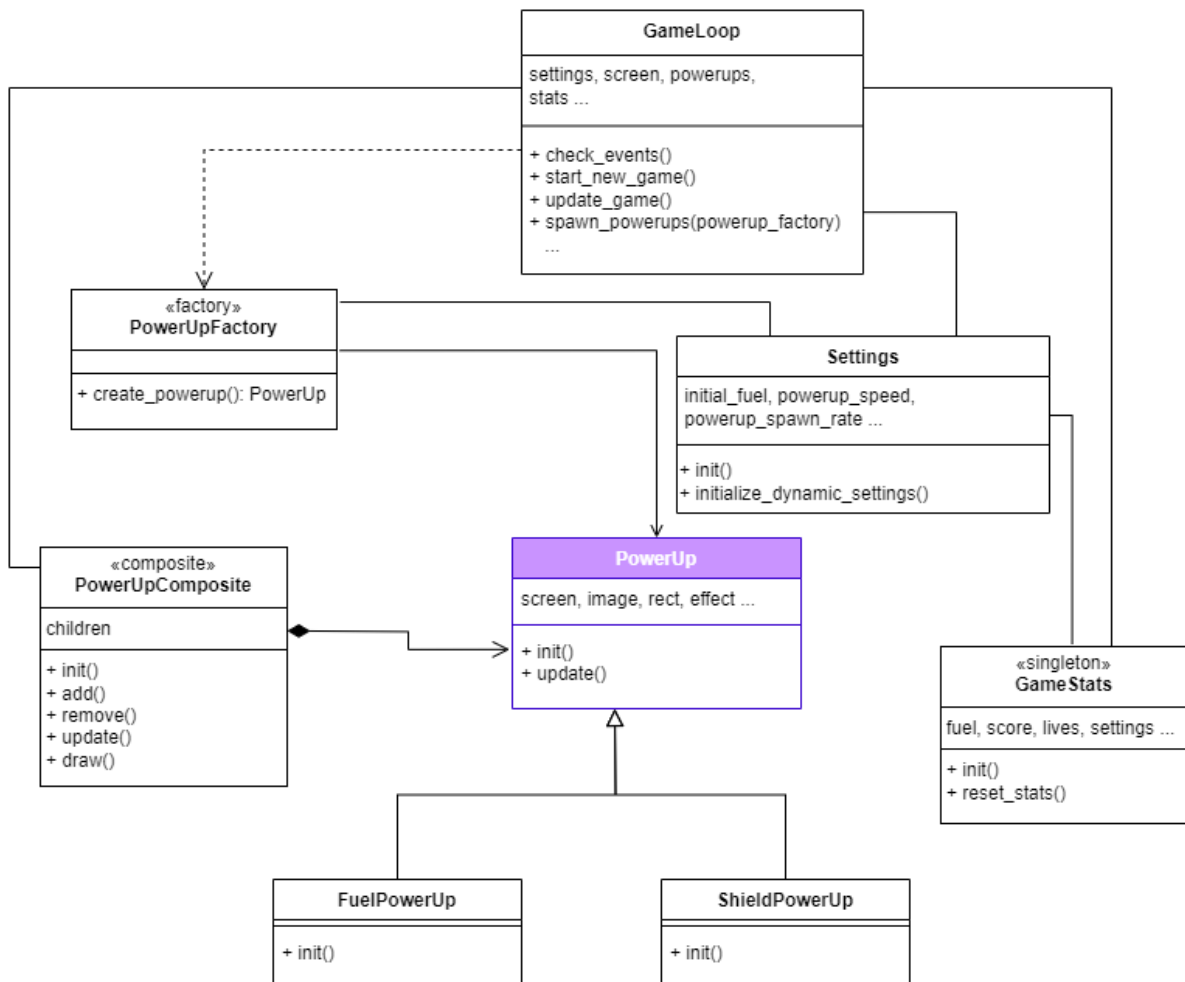
Klasa Balloon i ostale povezane klase



Klasa Obstacle i ostale povezane klase



Klasa PowerUp i ostale povezane klase



BallonBlitz – sve klase unutar igre

