

1 Implementation

Vores spil er skrevet i C#, hvor at der skabes et projekt til hver class. Spillet fungerer ved, at der i vores main-funktion instantieres et objekt af klassen "Game", hvorefter der kaldes en metode til denne kaldet "gameLoop". Denne method gameLoop kører spillet i et loop, ved konstant at kalde eventBus, hvilket er en instans af Galagabus, fremt spillet fortsat skal køre. Det er i denne eventbus, at de forskellige classes og method i spillet kaldes således, at spillet fungerer. Idet eventBus initialiseres, da bliver en liste af de forskellige GameEvents liges initialiseret og tilføjet til metoden InitializeEventBus. Disse events fungerer som den måde, hvormed spillet udfører sine handlinger. Disse events kunne eksempelvis være forskellige states, som spillet foregår i (Spillet er kørende, spillet er pauset, spillet er i startmenuen), eller det kunne være et input fra spilleren, som spillet skal udføre. Eksempelvis kunne spilleren trykke på en piletast for at bevæge sig, hvorledes der bliver kaldt et InputEvent, der registrerer, hvad inputtet var og sørger for, at inputtet bliver udført korrekt. I dette tilfælde vil det inkludere, at bevægelsen foregår fremt man holder piletasten nede og skal stoppe så snart man slipper knappen igen. Dette gøres ved at, at ProcessEvent, som nemlig tager den pågældende EventType som input og kalder de korrekte methods alt efter, om man trykker på tasten eller slipper den. Disse methods, der bliver kaldt, som f.eks. MoveRight, vil så selv kalde en method til Shape, en instans af hvilken er spilleren selv, hvis Direction ændres således, at spilleren bevæger sig. Som fremgår i ovenstående eksempel fungerer spillet altså ved, at mange classes kalder hinandens methods og udfører handlinger således, at spillet virker som forventet. Spillet er i praktisk forstand blevet implementeret ved, at det givne directory DIKUArcade indeholder mange forskellige interfaces (eksempelvis IGameState), der giver et krav til, hvilke forskellige methods, alle de objekter som instantieres udfra pågældende interface, skal agere. Disse interfaces er i vores kode blevet implementeret på passende måder.