

1 Implementation

1.1 SquareFormation

I vores arbejde med enemy formationerne har vi fundet det nødvendigt at fjerne de enemies som spawner tættest på kanterne. Dette er for at der ikke er nogen enemies som bevger sig ud af vores play area når vi giver dem vores ZigZagDown movement strategy. I praksis fungerer vores algoritme ved at vi først tjekker om vi skal begynde at spawn enemies på næste 'lag' af skærmen, eller om der fortsat skal stilles enemies på det nuværende lag. Når vi har fundet ud af dette tjekker vi at vores enemy ikke skal til at spawnes på lagets nulte eller syvende plads. Vi tjekker dette ved at udregne 'i modulus 8' — da der kan være otte enemies per lag. Hvis vi er på disse pladser, så fortsætter vi med at spawn den næste enemy uden at gøre yderligere. På denne måde får vi kun enemies i midten af vores play area.

1.2 StateMachine

Vores StateMachine bliver kaldt af vores Game objekt når spillet startes op. I vores StateMachine constructor stilles den aktive state til MainMenu således at spillet starter i vores menu. Vi har valgt at StateMachine selv er ansvarlig for at subscribe til de events den skal bruge da det til dels bliver mere overskueligt at det er StateMachine som subscriber til disse events, og til dels fordi vi mener at hver klasse så vidt muligt bør håndtere sig selv. Det kommer desværre til at gå ud over overskueligheden af hvilke events vores program benytter, men vi mener at det ovenstående opvejer dette.

Kommunikationen i vores program fungerer ved, at hver gang en af vores states i vores program ønsker at skifte til en anden state, så får vores StateMachine dette at vide, og ændrer dernæst sin ActiveState til den nye state. I vores ProcessEvent metode har vi skrevet noget logik som skelner mellem om vores StateMachine får en InputEvent eller en GameStateEvent. Hvis vores event er en InputEvent, så sender StateMachine denne videre til den aktive state og eventen bliver dernæst overladt til HandleKeyEvent i den aktive state. Hvis StateMachine får en GameStateEvent så starter vi en switch case hvor vi tjekker for hvilken state ActiveState skal ændres til. I tilfældet hvor der ønskes at skiftes til GameRunning staten, gør vi et ekstra tjek for at finde ud af om det er MainMenu som skifter staten, eller om det er GamePaused som skifter staten. I tilfældet hvor det er MainMenu, da vil vi gerne starte spillet som helt nyt, og vi kalder derfor InitializeGameState for at starte helt forfra. Men hvis det er GamePaused som ønsker at kalde GameRunning, så vil vi gerne fortsætte fra det tidligere spil, og derfor kaldes InitializeState ikke. Vi gør dette tjek ved at kigge på om den nuværende ActiveState er MainMenu eller GamePaused inden vi skifter ActiveState.