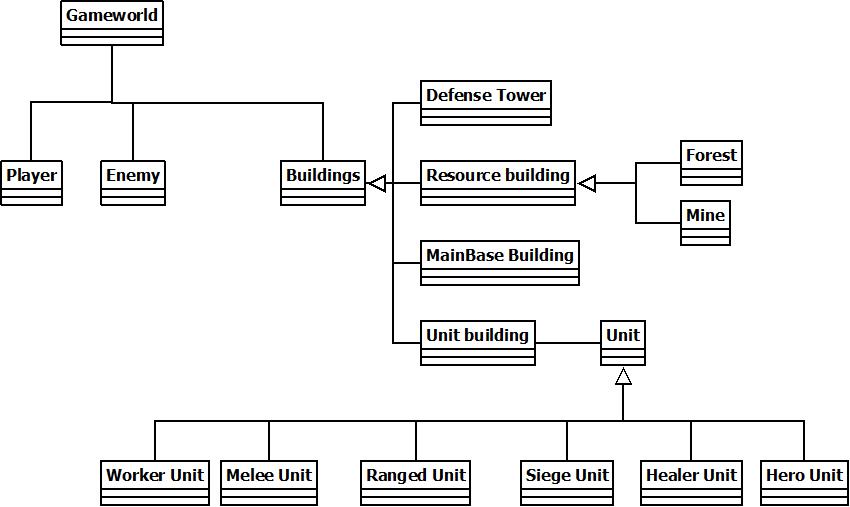
Basebuilder

* Spil Ide
  + Formålet med spillet er at bygge en base der automatisk producerer resourcer og træner soldater der vil angribe fjendens base, man vinder ved at ødelægge fjendens hovedbase og man taber hvis ens egen base bliver ødelagt. Spillet styres med W A S og D for at bevæge kameraet og musen for at vælge bygninger og bestemme hvor de skal bygges. Der vil også være en AI kontrolleret fjende der vil bygge bygninger for at angribe spilleren og forsvare sig selv.

Se bilag 1 og 2 for OOA og OOD diagram

* Player (BaseBuilderPlayerCharacter, BaseBuilderPlayerController)
  + BaseBuilderPlayerCharacter
    - Fields
      * TopDownCameraComponent, kameraet brugt af spilleren
      * CameraBoom, et UspringArmComponent brugt til at holde TopDownCameraComponent en bestemt distance fra spillerens pawn position
      * CursorToWorld, et decal brugt til at vise spillerens mus’ position i spilverdenen
    - Methods
      * GetTopDownCameraComponent, returnerer TopDownCameraComponent
      * GetCameraBoom, returnerer CameraBoom
  + BaseBuilderPlayerController
    - Fields
      * Money, brugt til at holde styr på hvor mange resourcer spilleren har
      * OwnBase, en AActor der holder styr på spillerens egen base, bruges af HUD til at skrive basens liv til skærmen og skulle bruges til at lade fjendes units finde og angribe spilleren
      * EnemyBase, en AActor, brugt som OwnBase til at holde styr på fjendens base og lade spillerens units angribe fjenden
      * SelectedBuilding, en AActor klasse brugt til at bestemme hvilken bygning spilleren vil bygge
    - Methods
      * SetupInputComponent, bruges til at sætte spillerens museklik til at kalde OnClick
      * OnClick, bruges til at spawne en bygning spilleren har valgt, dette gøres først ved at checke om spillerens har valgt en bygning, hvortil der checkes om spilleren har nok resourcer til at bygge den, efter det laves der et raycast der checker om spilleren har klikket på et område de kan bygge i og efter det bliver SpawnActor kaldt, bygningens SetFaction bliver kaldt og bygningens pris bliver trukket fra spillerens resourcer
  + Player klassen kontrollerer kameraet og valg og placering af valgte bygninger samt håndtering af spillerens tilgængelige resourcer
  + Placering a bygninger bestemmes gennem et raycast der checker for om spilleren har klikket på et område der er defineret som spillerens base område hvortil der bliver checket for om spilleren har nok resourcer til at bygge den valgte bygning og hvis de har bliver den valgte bygning spawnet på det punkt. Der er et problem med den måde det virker på ved en bygning kan laves der kolliderer med en anden bygning, en måde at løse dette kunne være ved at have en hitbox på størrelse med den bygning spilleren prøver at bygge og checke om den kolliderer med en anden bygning.
* Building(BaseBuilding)
  + Fields
    - TeamID, en int der bruges til at bestemme hvilket hold bygningen er på og hvilke units er fjender
    - Health, en float der bruges for at holde styr på bygningens liv. Hvis den rammer nul bliver OnDeath kaldt.
    - Description, en Fstring der bruges til at skrive bygningens beskrivelse i HUD
    - Price, en int der bestemmer hvor mange resourcer bygningen koster
  + Methods
    - SetFaction, bruges til at sætte TeamID når bygningen bliver spawnet, bliver kaldt fra PlayerController
    - OnDeath, bruges til at slette bygningen når dens Health når 0
    - TakeDamage bruges til at få bygningen til at tage skade når den bliver angrebet
  + En Bygningsskabelon som alle andre bygninger er baseret på.
  + Bygningen har et faktions ID, en beskrivelse, en byggepris og livskraft. Derudover indeholder den TakeDamage funktionen der bruges til at lade andre ting i spillet angribe den og gøre skade.
* Defence Tower(StandardTower)
  + Fields
    - Target, en APawn reference der bruges til at angribe fjender.
    - AttackSpeed, en float der bruges til at holde styr på et tårns angrebs fart.
    - Damage, en float der holder styr på tårnets kampkraft.
    - AttackTimer, en float der tæller ned fra AttackSpeed til nul. Når den rammer nul kan tårnet angribe, derefter sættes den til AttackSpeed.
    - Firing, en bool der holder styr på om tårnet kan angribe.
  + Methods
    - Attack, bruges til at tildele skade til Target.
  + Formålet med Tårnene er at virke som en forsvarsbygning, der angriber fjendtlige soldater.
  + Defence Tower bruger AIPerception til at se soldater. AI Perception fungerer sådan at alle enheder der indeholder en AIPerceptionStimuliSource for den specifikke Perception Source kan ”ses” Af enheden. Tårnet kører altså alle enheder den kan se igennem et loop, der tjekker om de er Soldater, og hvis de er, om soldatens faktions ID svarer til den faktions ID tårnet arver fra BaseBuilding. Hvis de er ens bliver Soldaten Ignoreret, ellers begynder tårnet at skyde. Hvis tårnets AttackTimer løber ud kan den køre sin Attack funktion, hvor modstanderen bliver tildelt skade efter tårnets styrke, og AttackTimer’en sættes tilbage. Hvis tårnet bliver for skadet dør det.
  + Det var meningen at spilleren skulle bygge tårne for at forsvare sin base fra AI fjendens angreb men siden AI elementet ikke blev færdiggjort er der ingen grund til spilleren bygger sine egne tårne.
* Resource Building (BPFactory, FactoryComponent)
  + BPFactory er et blueprint lavet fra BaseBuilding der gør brug af et FactoryComponent for at give det en speciel funktion
  + FactoryComponent
    - Fields
      * Timer, en float der bruges til at holde styr på hvor lang tid der er gået siden bygningen sidst gav spilleren resourcer
      * ResourceAmount, mængden af resourcer der bliver givet til spilleren hvert sekund
    - Methods
      * TickComponent, bliver kaldt hver frame og bruges til at tælle Timer ned til 0 hvor spiller bliver givet en mængde resourcer og Timer bliver sat til 1 sekund igen
  + Resource Building skulle have været en bygning som worker units ville samel resourcer fra og aflevere til MainBase building men vi nåede ikke at få lavet Worker Units så i stedet giver den passivt resourcer hvert sekund
  + Bygningen er lavet ud fra BaseBuilding og er givet et component for at give det funktionaliteten til at give resourcer
* Unit Building(SpawnBuilding, SpawnComponent)
  + SpawnBuilding er et blueprint laver fra BaseBuilding og gør brug af SpawnComponent til at lave nye units for spilleren i et interval
  + SpawnComponent
    - Fields
      * SpawnTime, hvor lang tid der går i sekunder før bygningen spawner en ny unit
      * SpawnPawn, en pawn variable der bruges til at bestemme hvad komponentet skal spawne
      * Timer, en variabel der bruges til at tælle ned og holde styr på hvor lang tid der er gået siden der sidst blev spawnet noget
    - Methods
      * BeginPlay, Sætter timer til SpawnTime
      * TickComponent, tæller Timer ned indtil den rammer 0 hvor Spawn bliver kaldt og Timer bliver sat til Spawntime
      * Spawn, Forsøger at spawne en ny BasePawn og kalde den Initialize for at give den et mål og en faction
  + Det var meningen at der var flere slags Unit Buildings der lavede deres egen version af BasePawn men grundet tidsmangel og fokus på andre vigtigere ting nåede vi ikke at lave det.
* Melee Unit(BasePawn, BPBasePawn)
* BPBasePawn er et blueprint lavet fra BasePawn, dette var lavet for at give det mulighed for at gøre brug af nogle components der ikke var tilgængelig gennem C++
  + BasePawn
    - Fields
      * Faction, en int der bruges til at bestemme hvilket hold enheden er på, og hvad der er fjendtligt.
      * Health, en float der bruges for at holde styr på enhedens liv. Hvis den rammer nul bliver OnDeath kaldt.
      * AttackSpeed, en float der bruges til at holde styr på enhedens angrebs fart.
      * Damage, en float der holder styr på tårnets kampkraft.
      * AttackTimer, en float der tæller ned fra AttackSpeed til nul. Når den rammer nul kan tårnet angribe, derefter sættes den til AttackSpeed.
      * Firing, en bool der holder styr på om enheden kan angribe.
      * Target, en AActor reference der bruges til at sætte et mål på banen og angribe fjender.
      * BaseTarget, en AActor reference der bruges til at holde styr på hvor fjendens hovedbase står.
    - Methods
      * SetNewMoveDestination, Bruges til at sætte enhedens destination efter Target.
      * TakeDamage, bruges til at skade enhedens Health, hvis Health går under 0 køres OnDeath.
      * OnDeath, bruges til at slette enheden når den er død.
      * Attack, bruges til at tildele skade til Target.
  + Formålet med disse Units er at ødelægge fjendens base ved at angribe fjendens forsvar og soldater indtil de når fjendens hovedbase, som de skal ødelægge.
  + Melee Units bliver spawnet fra en Unit Building, og arver bygningens faktions ID og Fjendens MainBase i Initialize funktionen. Med SetNewMoveDestination funktionen bliver den sat til at bevæge sig til dets ”Target” der per default er fjendens MainBase. Den navigerer mod denne base indtil den via AIPerception ser en fjende, om end det er Fjendens base, et Tårn eller en fjendtlig Unit. Denne fjende sættes til ”Target”, og SetNewMoveDestination funktionen sætter Unit’en imod den nye fjende. Hvis Unit’en kommer tæt nok på, og AttackTimer’en er udløbet, køres Attack funktionen, hvor fjenden tager skade efter Unit’ens styrke, og AttackTimer’en sættes tilbage. Hvis Unit’en tager skade køres TakeDamage funktionen, hvor Unit’ens ”Health” sættes ned efter fjendens skade. Efter al skade tjekkes der om ”Health” er under 0. Hvis den er køres OnDeath funktionen, Der sletter Unit’en.
  + Det var meningen at der var mange forskellige slags units, der havde deres egne styrker og svagheder.
* MainBaseBuilding(BPMainBase)
  + BPMainBase er et blueprint lavet fra BaseBuilding og bruger samme angrebs system som StandardTower for at angribe fjendtlige units, den har også en Health variabel der holder styr på dens liv
* LoseSplashScreen/WinSplashScreen
  + Splashscreens brugt når spilleren vinder/taber for at lade spilleren gå tilbage til hovedmenuen
* GameHUD og SelectBuildingWidget
  + GameHUD og SelectBuildingWidget bruges samlet til at give spilleren information om deres resourcer, deres bases resterende liv, en beskrivelse af hvilken bygning de har valgt og knapper for at lade spilleren vælge en bygning
* MainMenu
  + Et blueprint brugt til at lave hovedmenuen af spillet

Bilag 1.



Bilag 2.

