CPP_Stacks_Interface
+ Push(ACPP_Card*) : void + Pop(): ACPP_Card*

CPP_DiscardPile

+ShuffledArray:TArray<ACPP_Card*>
+DiscardPileArray:TArray<ACPP_Card*>

+ AddCardsFromDrawPile(ACPP_Card*) : void
+EmptyDiscardPile():ACPP:Card*
+AddCardsToShuffledPile():void

QueueInterface

+Enqueue():APawn*
+Dequeue():void

| +1

BoardersGameMode

+PlayerArray: TArray<APawn*>
+Player1Class: TSubclassOf<APlayerPawn>
+Player2Class: TSubclassOf<APlayerPawn>
+CurrentPlayer: APawn*
+Player1: APlayerPawn*
+Player2: APlayerPawn*
+PlayerController: APlayerController*
+Player1SpawnPoint: TSubclassOf<AActor>
+Player2SpawnPoint: TSubclassOf<AActor>

#BeginPlay(): void
+Tick(float):void
+Enqueue_Implementation(APawn*):void
+Dequeue_Implementation():APawn*
+SpawnPlayers():void
+EndTurn():void
+SwitchPlayer():void
+ResetPlayer(APawn*):void
+PLayer1SpawnLocation():FVector
+PLayer2SpawnLocation():FVector
+EndGame():void

+1

+1

PlayerComponent

+SpawnedPieces: TArray<ACPP_Piece*>
+MaxPoints: int32
+Points: int32
+CurrentPoints: int32
+Hand: UCPP_Hand*

#BeginPlay(): void
+TickCompoment(float,ELevelTick,FActorComponentTickFunction*):void
+PieceBelongToPlayer(ACPP_Piece*): bool
+RemoveTotalPieces(ACPP_Piece*):void
+ActionCost(int32):void
+FillArray(ACPP_Piece):void
+RefillPoints():void

CPP_Stack

+CardsArray:TArray<ACPP_Card*>
+Role: FRole
+DiscardPile: UCPP_DiscardPile*

+ Push_Implementation(ACPP_Card*) : void
+ Pop_Implementation(): ACPP_Card*

FTeam

+ TeamName : FString
+ TeamColor : FColor

+FTeam(FString, FColor)

FRole

+RoleTeam: FTeam
+RoleName: FString
+RoleSprite: UPaperSprite*
+ClassIndex: int32

+FRole(FTeam, FString,UPaperSprite*)
+FRole(FTeam,FString)

CPP_Card

+ Mesh: UStaticMeshComponent*
+CardName: FString
+CardCost: int32
+CardRole: FRole
+CardActionGrid : TArray<F2DVectorInt>
+CardAttack:TArray<AActor*>

+SpawnAttack(---):void
#BeginPlay():void
+Tick(float):void

Action

- ActionName : string

+ Effect() : void

CPP_Hand

- DrawPiles: TArray<UCPP_Stack*>
+MaxCards: int
+Cards: TArray<ACPP_Card*>
+Sorter: UCPP_CountingSort*

- DrawCard(UCPP_Stack*) : void
+UseCard(int): ACPP_Card*

CPP_Piece

- MaxHealth : int
-CurrentBoard: ACPP_Board*
+TriggerBox: UBoxComponent*
+MeshComponent: USkeletalMeshComponent*
+Health: int
+BoardPosition: F2DVectorInt
+MovementOptions: TArray<F2DVectorInt>
+PlayerOwner: APlayerPawn*

#BeginPlay():void
+Tick(float):void
+Damage(int): int
+MoveTowards(F2DVectorInt): bool
+Onclicked(): void

PlayerPawn

+ Camera : class UCameraComponent*
+Imc: class UInputMappingContext*
+Wasd: class UInputAction*
+LookAction: class UInputAction*
+ClickLeftAction: class UInputAction*
+ClickRightAction: class UInputAction*
+PlayerComponent: class UPlayerComponent*
+bRightMouseButton: bool
-my_enum:: enum
+SavedPiece: ACPP_Piece*

#BeginPlay(): void
+Tick(float):void
+PlayerMovement(const FInputActionValue&):void
+CameraLook(const FInputActionValue&):void
+OnClick():void
+CheckActor(AActor*):bool
+RightMouseButtonIsclicked():void
+RightMouseButtonNotclicked():void
+Deselect():void
+SavePreviousPiece(ACPP_Piece):void

PlayerUI

- Classes : array<ClassUI>

+ DrawUI() : void

ClassUI

+ ClassInfo : ClassPiece

CPP_CountingSort

+ CardsForSorting: TArray<ACPP_Cards*>

+Sort():void
+PrintArray():void
+FillHand(ACPP_Card*):void

Board

-TileArray2D: TArray<UFTileArray*>
+Mesh: UStaticMeshComponent*
+InitBoardSize: F2DVectorInt
+TeamAffiliationRowCount: int
+TileSize: FVector
+InitBoardPadding: float
+InitBoardMargin: float
+InitBoardDepth: float
+TileActor: TSubclassOf<ACPP_Tile>

#BeginPlay():void
+Tick(float): void
+SetBoardSize(const F2DVectorInt): bool
+SetBoardTeamAffiliation(const int): bool
+GetTileAt(const int, const int): ACPP_Tile*
+GetTileAt(F2DVectorInt): ACPP_Tile*
+GetBoardSize(): F2DVectorInt
+DrawBoard(const FVector,const FVector,const float, const float, const float):void

0..*

FTileArray

+TileArray: TArray<ACPP_Tile*>

0..*

CPP_Tile

+bIsOccupied: bool
+TeamAffiliation: int
+OccupyingActor: AActor*
+Mesh: UStaticMeshComponent

#BeginPlay(): void
+Tick(float):void
+DrawTile(const FVector, const FVector):void

+1

CPP_Pathfinding

+RunPathfinding(F2DVectorInt, TArray<F2DVectorInt>, ACPP_Board*, int): TArray<UCPP_AlgorithmPath*>

+1..8

F2DVectorInt

+ Direction : Vector2D
+operator: bool
+X: int
+Y: int

+ F2DVectorInt(const int, const int)
+WithinRange(const F2DVectorInt&, const F2DVectorInt&): const bool
+WithinRange(const F2DVectorInt&): const bool
+operator+(const F2DVectorInt&): F2DVectorInt
+operator-(const F2DVectorInt&): F2DVectorInt

+1

+1

PieceSpawner

+SpawnPoints: TArray<AActor*>
+Pieces: TArray<TSubclassOf<ACPP_Piece>>
+Player: TSubclassOf<APlayerPawn>

#BeginPlay(): void
+Tick(float):void
+SpawnPieces():void
+AssignPlayer(ACPP_Piece):void

CPP_AlgorithmPath

+Position: F2DVectorInt
+PathCost: int
+Parent: UCPP_AlgorithmPath*

+UCPP_AlgorithmPath(F2DVectorInt)
+UCPP_AlgorithmPath(F2DVectorInt, int, UCPP_AlgorithmPath*)