```
//public MainMenu
        //load all the ascii art

        //create variable int uiSelected to save current selection
        //create varbiable bool uiChosen to check if user wants to select current selected

        //print "Please fullscreen the application (f11) to continue, otherwise application might crash"
        //print Press any key to continue
        //wait for button press
        //clear console

        //while ui chosen is false

                //clear console
                //print the main logo ascii art

                //switch ui selected
                        //case 0
                                //print continue as selected
                                //print start
                                //print credits
                        //break

                        //case 1
                                 //print continue
                                //print start as selected
                                //print credits
                         //break

                        //case 2
                                //print continue
                                //print start
                                //print credits as selected
                        //break
                //end switch

                //switch user key input
                        //case "w" "uparrow"
                                //decrement selected if value is above 0 else do nothing
                        //break

                        //case "s" "downarrow"
                                //increment selected if value is below 2 else do nothing
                         //break

                        //default
                                //if any other button is pressed set selected to true
                        //break
                //end switch

                //clear console

        //end while loop
//end MainMenu
```

Entity

- baseDamageStats: int[]
- health: int
- armor: int
- strength: float
- criticalChance: int
- criticalMultiplier: float
- speed: float

+ GetHealth():int
+ TakeDamage(pDamage:Int):void
+ DealDamage():int
+ GetSpeed():float
+ SetBaseDamage(minDamage:int, maxDamage:int):void

Player

- writingStyles:WritingStyles
- name:string
- position:int[]

+CharacterCreator():void
+Move(pDungeon:DungeonTemplate):void
+CheckMove(pDungeon:DungeonTemplate pVertical:int, pHorizontal:int, pDirection:int):bool
+SetPlayerPosition(pPosition:int[]):void

Program

- mainMenu:MainMenu
- vischachaGate:VischachaGate

+main():void

VischachaGate

- writingStyles:WritingStyles
- player:Player

+Game():void

WritingStyles

+WriteLineByCharactersSlowly(input:string, delay:int): void
+WriteByCharactersSlowly(input:string, delay:int): void
+WriteLineByLineSlowly(input:string, delay:int): void
+OverWriteCurrentLine(): void

WritingStyles

+ WriteLineByCharactersSlowly(input:string, delay:int): void
+ WriteByCharactersSlowly(input:string, delay:int): void
+ WriteLineByLineSlowly(input:string, delay:int): void
+ OverWriteCurrentLine(): void

DugeonTemplate

+ Dungeon:RoomTemplate[,]
- asciiArtRooms:string[]
- startingRoom:int[]
- newRoomsCoordinates:list<int>

+ DungeonTemplate(pDungeonDimensions:int):void
- GenerateDungeon(pDungeonDimensions:int):void
- GenerateRooms():void
- GetNewRoomCoordinates(pCurrentRoomCoordinates:int[], pHallwayDirection:int):int[]
- RemoveStrayRooms():void
- CheckDungeonBounds(coords:int[]):bool
- LoadAsciiArt():void
+ GetStartingRoomCoordinates():int[]
+ GetRoomObject(pPosition:int[]):RoomTemplate

1 to ???

RoomTemplate

- possibleHallwayLayouts:list<bool>
- roomIndex:int
- hallwaysStatus:bool[]
- roomArt:string

+ RoomTemplate(pAsciiArt:string[]):void
- RandomHallways():void
- AssignRoomArt(pAsciiArt:string[]):void
+ GetHallways():bool[]
+ GetRoomArt():string
+ OverwriteHallways(pAsciiArt:string[], pHallways:bool[]):void

DungeonTemplate object

DungeonTemplate Constructor | LoadAsciiArt Method | GenerateDungeon Method

GenerateRooms Method

dimensions given?

yes | no

create a 2d array of object RoomTemplate based on the dimensions given by the constructor

create a 2d array of object RoomTemplate based on the base dimensions set by the constructor

load text file (containing ascii art) into string array

Create a new RoomTemplate object in middle of the 2d array

overwrite the new objects hallway property

Save the newly created rooms location in list

Grab first room from list

Create new rooms around first room from list based on criteria

new rooms needed?

no | yes

add newly created rooms to list

remove first room from list

rooms in list of new rooms?

yes | no

RoomTemplate object(s)

RandomHallways Method	AssingHallways Method	OverWriteHallways Method
generate a random number between 0 and 14 (including 14), store index in int	assign this object a string based on the stored index int	
		uses the given values to edit the current rooms hallways
	assign this object a string based a new stored index int	

RoomTemplate object(s)

RandomHallways Method	AssingHallways Method
generate a random number between 0 and 14 (including 14), store index in int	assign this object a string based on the stored index int

criterias met?

no | yes