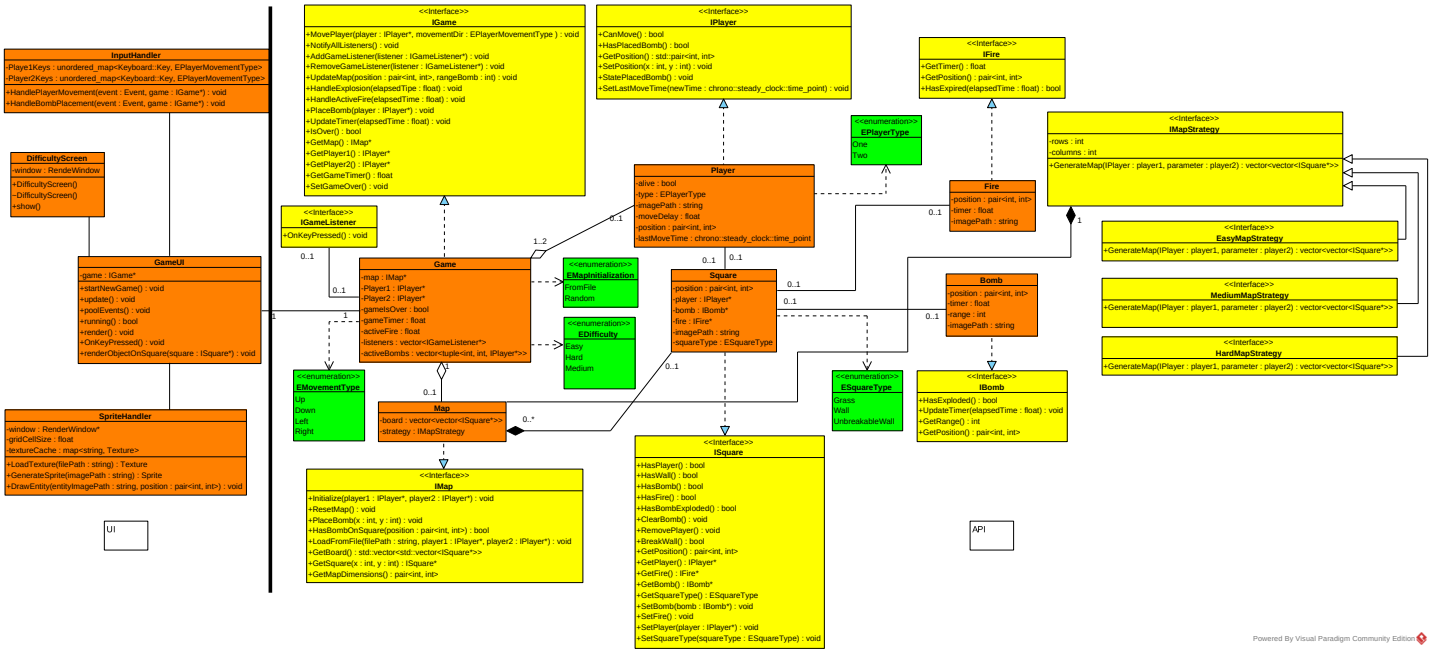
InputHandler
-Playe1Keys : unordered_map<Keyboard::Key, EPlayerMovementType>
-Player2Keys : unordered_map<Keyboard::Key, EPlayerMovementType>
+HandlePlayerMovement(event : Event, game : IGame*) : void
+HandleBombPlacement(event : Event, game : IGame*) : void
DifficultyScreen
-window : RenderWindow
+DifficultyScreen()
~DifficultyScreen()
+show()
GameUI
-game : IGame*
+startNewGame() : void
+update() : void
+pollEvents() : void
+running() : bool
+render() : void
+OnKeyPressed() : void
+renderObjectOnSquare(square : ISquare*) : void
SpriteHandler
-window : RenderWindow*
-gridCellSize : float
-textureCache : map<string, Texture>
+LoadTexture(filePath : string) : Texture
+GenerateSprite(imagePath : string) : Sprite
+DrawEntity(entityImagePath : string, position : pair<int, int>) : void
UI
<<Interface>>
IGame
+MovePlayer(player : IPlayer*, movementDir : EPlayerMovementType) : void
+NotifyAllListeners() : void
+AddGameListener(listener : IGameListener*) : void
+RemoveGameListener(listener : IGameListener*) : void
+UpdateMap(position : pair<int, int>, rangeBomb : int) : void
+HandleExplosion(elapsedTipe : float) : void
+HandleActiveFire(elapsedTime : float) : void
+PlaceBomb(player : IPlayer*) : void
+UpdateTimer(elapsedTime : float) : void
+IsOver() : bool
+GetMap() : IMap*
+GetPlayer1() : IPlayer*
+GetPlayer2() : IPlayer*
+GetGameTimer() : float
+SetGameOver() : void
<<Interface>>
IPlayer
+CanMove() : bool
+HasPlacedBomb() : bool
+GetPosition() : std::pair<int, int>
+SetPosition(x : int, y : int) : void
+StatePlacedBomb() : void
+SetLastMoveTime(newTime : chrono::steady_clock::time_point) : void
<<Interface>>
IFire
+GetTimer() : float
+GetPosition() : pair<int, int>
+HasExpired(elapsedTime : float) : bool
<<enumeration>>
EPlayerType
One
Two
<<interface>>
IMapStrategy
+rows : int
-columns : int
+GenerateMap(IPlayer : player1, parameter : player2) : vector<vector<ISquare*>>
Player
-alive : bool
-type : EPlayerType
-imagePath : string
-moveDelay : float
-position : pair<int, int>
-lastMoveTime : chrono::steady_clock::time_point
Fire
-position : pair<int, int>
-timer : float
-imagePath : string
<<Interface>>
IGameListener
+OnKeyPressed() : void
Game
-map : IMap*
-Player1 : IPlayer*
-Player2 : IPlayer*
-gameIsOver : bool
-gameTimer : float
-activeFire : float
-listeners : vector<IGameListener*>
-activeBombs : vector<tuple<int, int, IPlayer*>>
<<enumeration>>
EMapInitialization
FromFile
Random
<<enumeration>>
EDifficulty
Easy
Hard
Medium
Square
-position : pair<int, int>
-player : IPlayer*
-bomb : IBomb*
-fire : IFire*
-imagePath : string
-squareType : ESquareType
Bomb
-position : pair<int, int>
-timer : float
-range : int
-imagePath : string
<<Interface>>
EasyMapStrategy
+GenerateMap(IPlayer : player1, parameter : player2) : vector<vector<ISquare*>>
<<Interface>>
MediumMapStrategy
+GenerateMap(IPlayer : player1, parameter : player2) : vector<vector<ISquare*>>
<<Interface>>
HardMapStrategy
+GenerateMap(IPlayer : player1, parameter : player2) : vector<vector<ISquare*>>
<<enumeration>>
EMovementType
Up
Down
Left
Right
Map
-board : vector<vector<ISquare*>>
-strategy : IMapStrategy
<<enumeration>>
ESquareType
Grass
Wall
UnbreakableWall
<<Interface>>
IBomb
+HasExploded() : bool
+UpdateTimer(elapsedTime : float) : void
+GetRange() : int
+GetPosition() : pair<int, int>
<<Interface>>
IMap
+Initialize(player1 : IPlayer*, player2 : IPlayer*) : void
+ResetMap() : void
+PlaceBomb(x : int, y : int) : void
+HasBombOnSquare(position : pair<int, int>) : bool
+LoadFromFile(filePath : string, player1 : IPlayer*, player2 : IPlayer*) : void
+GetBoard() : std::vector<std::vector<ISquare*>>
+GetSquare(x : int, y : int) : ISquare*
+GetMapDimensions() : pair<int, int>
<<Interface>>
ISquare
+HasPlayer() : bool
+HasWall() : bool
+HasBomb() : bool
+HasFire() : bool
+HasBombExploded() : bool
+ClearBomb() : void
+RemovePlayer() : void
+BreakWall() : bool
+GetPosition() : pair<int, int>
+GetPlayer() : IPlayer*
+GetFire() : IFire*
+GetBomb() : IBomb*
+GetSquareType() : ESquareType
+SetBomb(bomb : IBomb*) : void
+SetFire() : void
+SetPlayer(player : IPlayer*) : void
+SetSquareType(squareType : ESquareType) : void
API
1..2
0..1
1
0..*
Powered By Visual Paradigm Community Edition