BomberMan

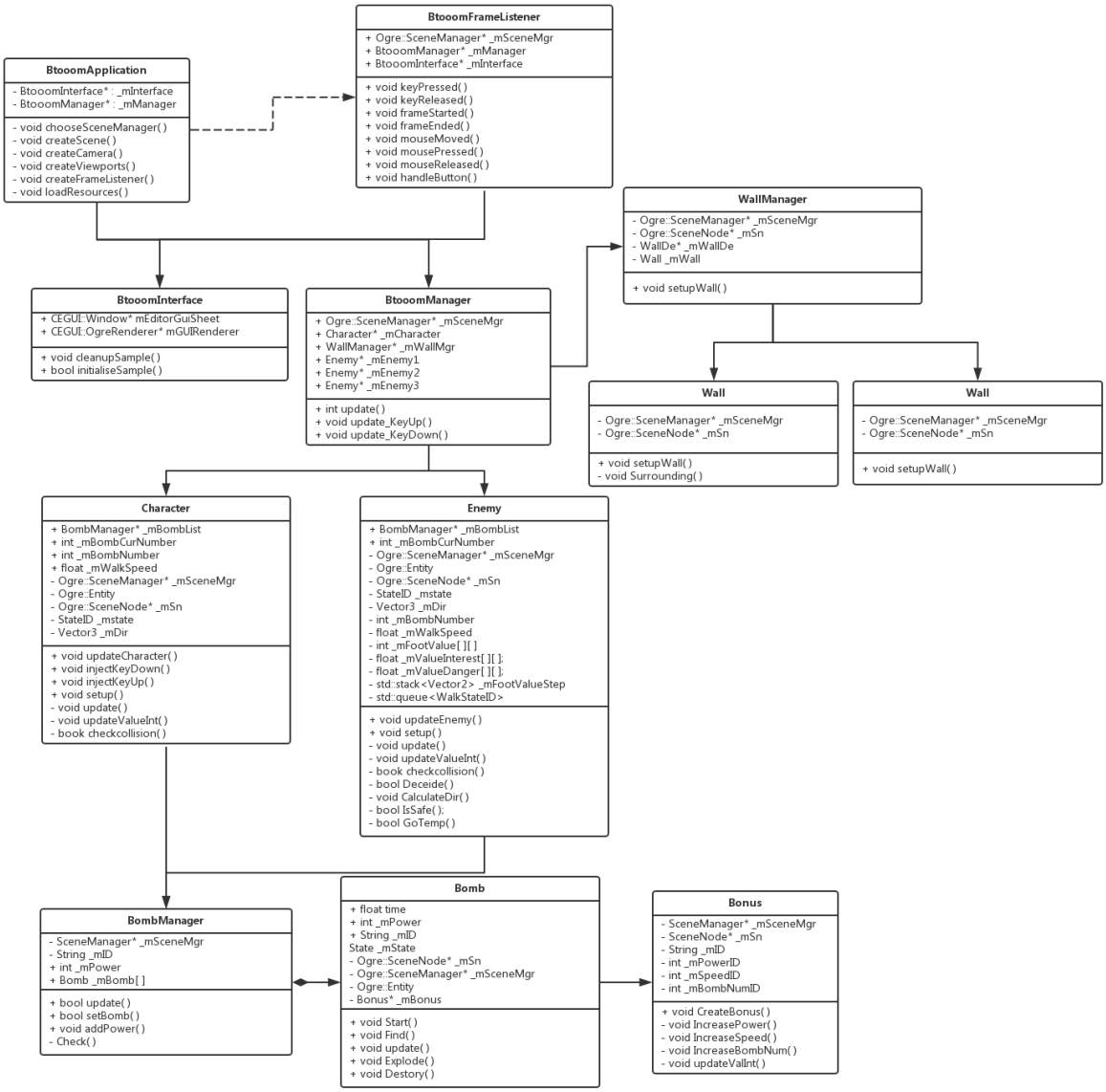
Introduction:

This is a simple 3d BomberMan game based on Ogre and CeGui. People can plays with almost three enemies. User can control his character by using arrow keys and set bomb by space key. There are two types of obstruction: the destructible brick and wood, the indestructible wall. There are three types of bonus: speed-up, bomb-power-up and bomb-number-up, which will be generated randomly by destroying the obstruction.

Implementation:

The BtooomApplication.h, which is a subclass of ExampleApplication.h, is used to implement the overall system framework. It initializes the environment that is required for running this game and sets up a framelistener to track on user's operation. The BtooomManager.h is in charge of controlling game play and do management on every scene node (character, enemy, wall, etc.).

In this game, enemy is driven by a best route finding algorithm. 2d matrix _mValueInterest[ ][ ] and _mValueDanger[ ][ ] are kept to record the interesting value and the dangerous value for every point. Once a destination with high interesting value is decided, we use a Dijkstra algorithm to find the shorted path from current position to destination and store it in the stack _mFootWay. Enemy will check the whether the next position is safe or not before he moves. If the next position is not safe, the enemy will wait for a while or find a temporary place for shelter.


BtooomFrameListener
+ Ogre::SceneManager* _mSceneMgr
+ BtooomManager* _mManager
+ BtooomInterface* _mInterface
+ void keyPressed( )
+ void keyReleased( )
+ void frameStarted( )
+ void frameEnded( )
+ void mouseMoved( )
+ void mousePressed( )
+ void mouseReleased( )
+ void handleButton( )
BtooomApplication
- BtooomInterface* : _mInterface
- BtooomManager* : _mManager
- void chooseSceneManager( )
- void createScene( )
- void createCamera( )
- void createViewports( )
- void createFrameListener( )
- void loadResources( )
WallManager
- Ogre::SceneManager* _mSceneMgr
- Ogre::SceneNode* _mSn
- WallDe* _mWallDe
- Wall _mWall
+ void setupWall( )
BtooomInterface
+ CEGUI::Window* mEditorGuiSheet
+ CEGUI::OgreRenderer* mGUIRenderer
+ void cleanupSample( )
+ bool initialiseSample( )
BtooomManager
+ Ogre::SceneManager* _mSceneMgr
+ Character* _mCharacter
+ WallManager* _mWallMgr
+ Enemy* _mEnemy1
+ Enemy* _mEnemy2
+ Enemy* _mEnemy3
+ int update( )
+ void update_KeyUp( )
+ void update_KeyDown( )
Wall
- Ogre::SceneManager* _mSceneMgr
- Ogre::SceneNode* _mSn
+ void setupWall( )
- void Surrounding( )
Wall
- Ogre::SceneManager* _mSceneMgr
- Ogre::SceneNode* _mSn
+ void setupWall( )
Character
+ BombManager* _mBombList
+ int _mBombCurNumber
+ int _mBombNumber
+ float _mWalkSpeed
- Ogre::SceneManager* _mSceneMgr
- Ogre::Entity
- Ogre::SceneNode* _mSn
- StateID _mstate
- Vector3 _mDir
+ void updateCharacter( )
+ void injectKeyDown( )
+ void injectKeyUp( )
+ void setup( )
- void update( )
- void updateValueInt( )
- book checkcollision( )
Enemy
+ BombManager* _mBombList
+ int _mBombCurNumber
- Ogre::SceneManager* _mSceneMgr
- Ogre::Entity
- Ogre::SceneNode* _mSn
- StateID _mstate
- Vector3 _mDir
- int _mBombNumber
- float _mWalkSpeed
- int _mFootValue[ ][ ]
- float _mValueInterest[ ][ ];
- float _mValueDanger[ ][ ];
- std::stack<Vector2> _mFootValueStep
- std::queue<WalkStateID>
+ void updateEnemy( )
+ void setup( )
- void update( )
- void updateValueInt( )
- book checkcollision( )
- bool Deceide( )
- void CalculateDir( )
- bool IsSafe( );
- bool GoTemp( )
BombManager
- SceneManager* _mSceneMgr
- String _mID
+ int _mPower
+ Bomb _mBomb[ ]
+ bool update( )
+ bool setBomb( )
+ void addPower( )
- Check( )
Bomb
+ float time
+ int _mPower
+ String _mID
State _mState
- Ogre::SceneNode* _mSn
- Ogre::SceneManager* _mSceneMgr
- Ogre::Entity
- Bonus* _mBonus
+ void Start( )
+ void Find( )
+ void update( )
+ void Explode( )
+ void Destory( )
Bonus
- SceneManager* _mSceneMgr
- SceneNode* _mSn
- String _mID
- int _mPowerID
- int _mSpeedID
- int _mBombNumID
+ void CreateBonus( )
- void IncreasePower( )
- void IncreaseSpeed( )
- void IncreaseBombNum( )
- void updateValInt( )