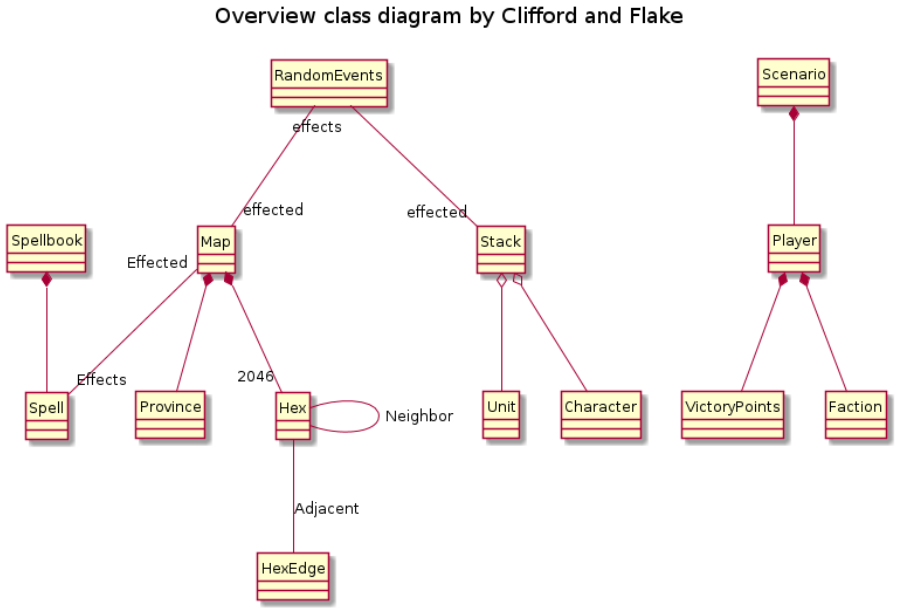
HW3 Team B Class Diagrams

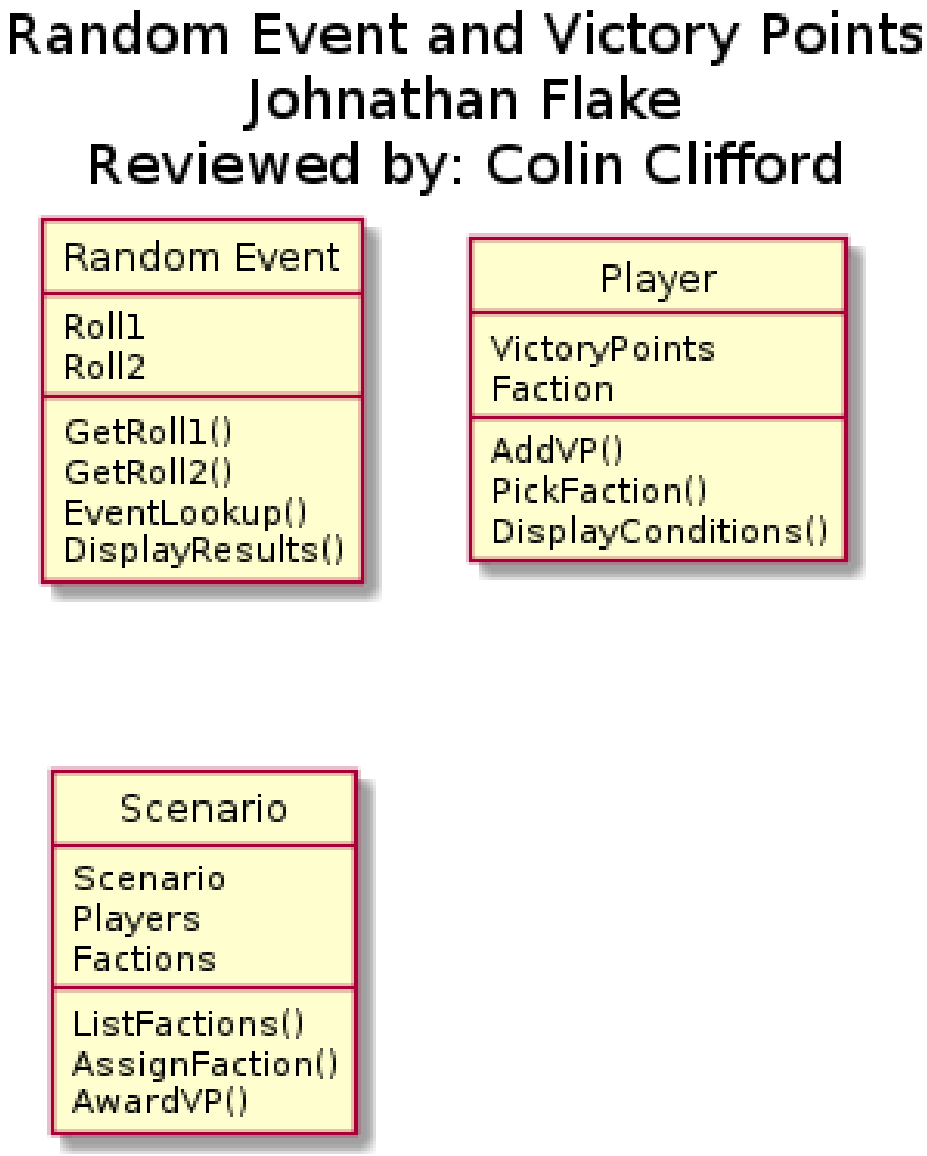
Tao John G Jay John F Matt Clifford

February 18, 2014

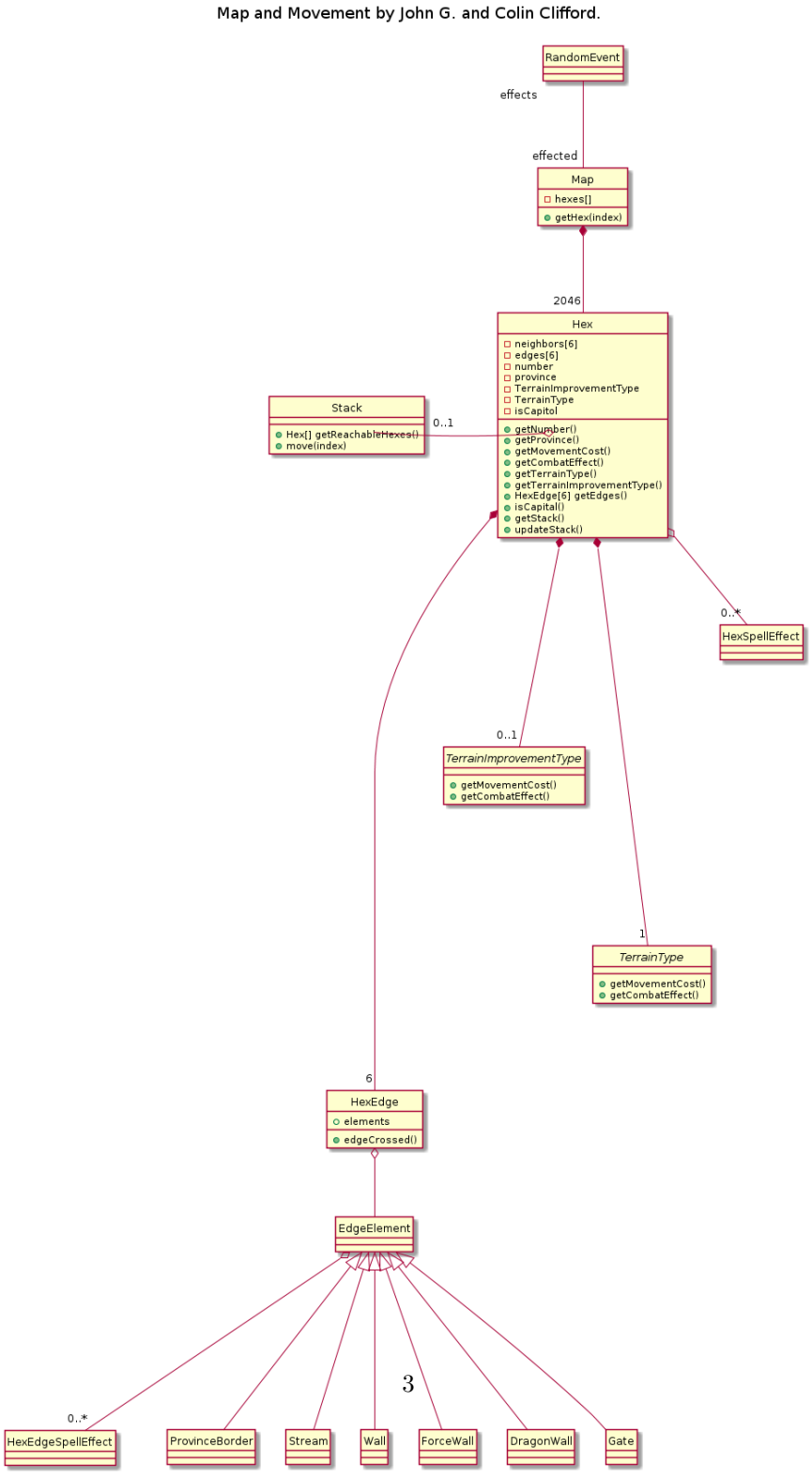
1 Overview



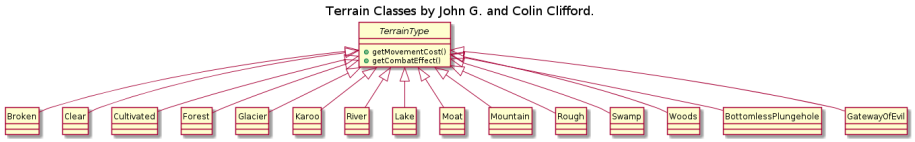
2 Random Events



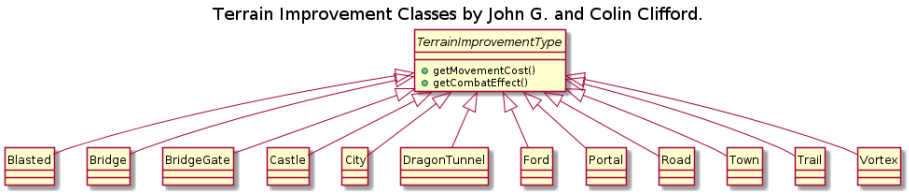
3 Map and Movement



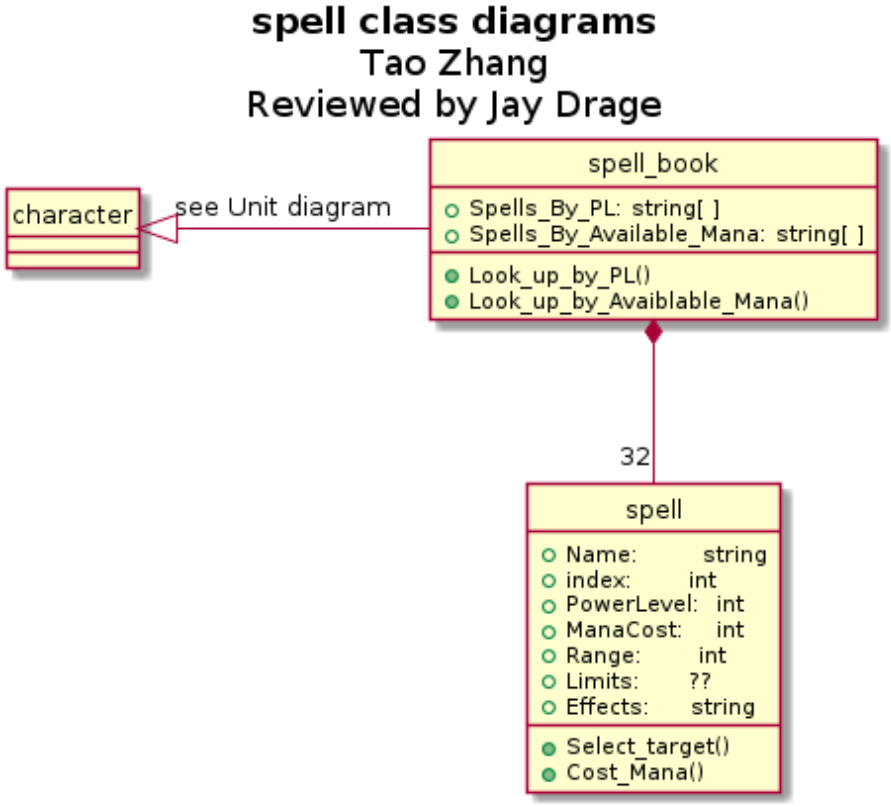
4 Terrain Types



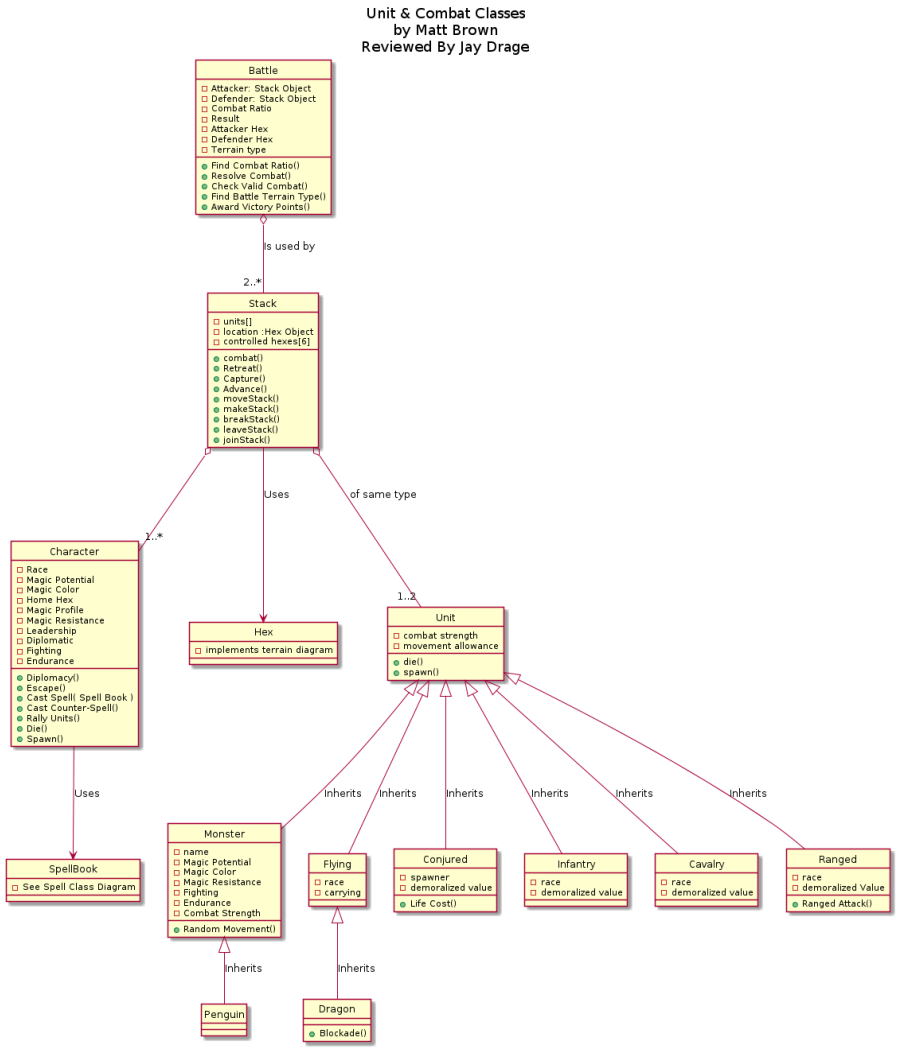
5 Terrain Improvement Types



6 Spells


spell class diagrams
Tao Zhang
Reviewed by Jay Drage
character
see Unit diagram
spell_book
Spells_By_PL: string[ ]
Spells_By_Available_Mana: string[ ]
Look_up_by_PL()
Look_up_by_Avaiblable_Mana()
32
spell
Name: string
index: int
PowerLevel: int
ManaCost: int
Range: int
Limits: ??
Effects: string
Select_target()
Cost_Mana()

7 Unit Classes



Combat and Unit classes dictionary

Jay Drage reviewed by Matt Brown

Class: Unit

Fields: combatStrength - the combat strength rating of unit

movementAllowance - movement points remaining

Methods: Die() - removes unit from play

Spawn() - creates unit and puts into play

Class: Flying inherits Unit

Fields: race - race type of unit

carrying - is unit carrying anything

Class: Dragon inherits Flying

Methods: Blockade[] - allows dragon to blockade tunnel entrance

Class: Monster extends Unit

Fields: name - name of unit

Magic Potential - maximum level of magic

Magic Color - associated magic color

Magic Resistance - magic resistance level

Fighting - fighting ability

Endurance - number of endurance points

Combat Strength - combat strength

Methods: Random Movement() - moves monster in a random manner

Class: Conjured extends Unit

Fields: spawner - character who spawned the unit

demoralized value - value once unit is demoralized

Class: Infantry extends Unit

Fields: spawner - character who spawned the unit

demoralized value - value once unit is demoralized

Class: Cavalry extends Unit

Fields: spawner - character who spawned the unit

demoralized value - value once unit is demoralized

Class: Cavalry extends Unit

Fields: spawner - character who spawned the unit

demoralized value - value once unit is demoralized

Methods: Ranged Attack() - allows ranged units to attack a non-adjacent hex

Class: Character

Fields: Race - race type of unit

Magic Potential - maximum allowed mana points

Magic Color - associated magic color

Home Hex - starting location of character

Magic Profile - magical visibilty of character

Endurance - number of endurance points

Magic Resistance - resistance level to magic

Leadership - leadership value

Diplomatic - diplomatic level

Fighting - fighting ability

Endurance - number of endurance points

Methods: Diplomacy() - conduct diplomacy with other nations

Escape() - attempt escape from capture

CastSpell() - cast spell if character has mana

CastCounterSpell() - cast a counter spell during different players turn

RallyUnits() - attempt to rally units in current hex

Die() - removes character from play

Spawn() - creates unit and puts into play

Class: Stack : dynamic aggregate of Units

Fields: Unit[] memberUnits - array of units that are in stack

location - Hex object: hex stack is on

controlledHexes[6] - hexes under zone of control

Methods: Retreat() - retreats after combat

Capture() - capture enemy units

Advance() - advance forward after combat

MoveStack() - move stack to selected hex

MakeStack() - create new stack for exiting units

breakStack() - destroys stack

UseMagic() - if magic units present then allow Unit.CastSpell()

LeaveStack() - removes unit from memberUnits[], update characterPresent

JoinStack() - adds unit to memberUnits[], update characterPresent if unit is character

Class: Battle

Fields: attacker - stacks of attacking units

defender - stack of defending units

combatRatio - calculated ratio from combat table

result - result of battle

attackerHex - hex numbers of attackers before battle

defenderHex - hex number of defender before battle

terrainType - holds int code for terrain type of battle hex

Methods: FindCombatRatio() - calculates combatRatio

ResolveCombat( ) - resolves combat and sets retreat or advance status of stacks

CheckValidCombat( ) - performs check to see if battle can procede

FindBattleTerrainType() - uses defenderHex to determine terrainType

AwardVictoryPoints() - adds victory points to victory conditions

Maps and Movement diagram dictionary

Class: Map

The Map class represents the army game word, and all the hex tiles it contains.

Fields: hexes - array of hexes in the world

Methods: getHex(index) - get hex from numeric index

HexEdge

A HexEdge is simply an edge between two hexes. Every Hex has 6 HexEdges, while HexEdges that aren't on the edge of the map have two Hexes it lays between.

Fields: elements - Everything currently on the hex edge

Methods: edgeCrossed - called when edge is crossed, to send message to things that care about this

EdgeElement

An EdgeElement is anything that lives on a Hex Edge

Wall - normal wall, such as what might border a city

ForceWall - A magical wall, cast by a spell

DragonWall - A dragon created wall at a dragon tunnel

ProvinceBorder - The boundary between two provinces

Gate - A gate

TerrainType / TerrainImprovementType

A TerrainType is just that: the sort of Terrain that is in a Hex. Examples include Forest, Karoo, Mountain.

A TerrainImprovementType is something that lies on top of a TerrainType, such as a road.

Methods: getMovementCost - Gets the movement cost modifier

getCombatEffect() - gets any combat modifiers from fighting on the terrain

Hex

The Hex class represents a single hex inside the game Map.

A single Hex can have several properties in the game, notably it has one or more terrain types, 6 or less neighbor hexes, and 6 hex edges.

Additionally a Hex can be effected by magic or random events.

Fields: neighbors - the neighboring hexes

edges - the neighboring edges

province - the province that the hex is part of

TerrainType - the terrain type of the hex

TerrainImprovementType - the terrain improvement hex

Methods: isCapital - returns whether the hex is a capital

getNumber - return the number of the hex

getProvince - return the province of the hex

getMovementCost - return the movement cost of moving a unit onto the hex

getCombatEffect - return the combat modifiers for fighting on the hex

getEdges - return the hex edges

getStack - return any stack occupying the hex

updateStack - modify the stack

Movement parts of Stack

The stack class needs a couple of methods to facilitate Movement

Methods: getReachableHexes - performs a floodfill and looks at the varying terrains to determine what hexes all the units in the stack can reach

move(index) - move to a destination hex

Spell diagram dictionary

Class: Spell_book

A class that sorts all available spells for the character

Fields: Spells_By_PL - a string array to store all the name of spells that is below, equal, or one level higher than the Magic PL of the character.

Spells_By_Available_Mana - A string array to store all the name of spells that the character has enough mana to cast FROM Spells_By_PL.

Methods:

Look_up_by_PL() - Function to sort search Spells_By_PL.

Look_up_by_Available_Mana() - Function to search Spells_By_Available_Mana from Spells_By_PL.

Class: Spell

We will have 32 spell classes for each spell

Fields: Name - String Name of the spell

index - We can put index on each spell so that is easy to deal with.

PowerLevel - Just the Magic PL of this spell

Mana Cost - Integer Mana for this spell to cost

Range - Range of the spell in hexes

Limits - There are varies of limitations. We will have many booleans.

Effects - Just the description of the spell that can be shown to players

Methods: Select_target() - Function to select a target that the player want to cast his spell on.

Cost_Mana() - Simple calculation to reduce the current mana points for that character.