refactor notes

Monday, October 20, 2014 4:16 PM

Entity :

Vector2
X, Y

SpriteID
Position # V2
Width
height
Velocity # V2 (Speed)
direction # V2

Update (float deltaTime)
Draw()
IsCollided (Entity * other)

AttackStates
WAIT,
Move,
Circle,
Attack,
Return

Player sub Entity
moveLeftKey
moveRightKey
leftMoveExtreme
RightMoveExtreme
shootKey

Shoot()

Enemy sub Entity
activeEnemyCount
isAttacking
isLeader
attackExitChosen
attackSpeed
attackSlope
attackYIntercept
attackDirection

DegreeToRadians
RadiansToDegrees
GetSlopeOfLine

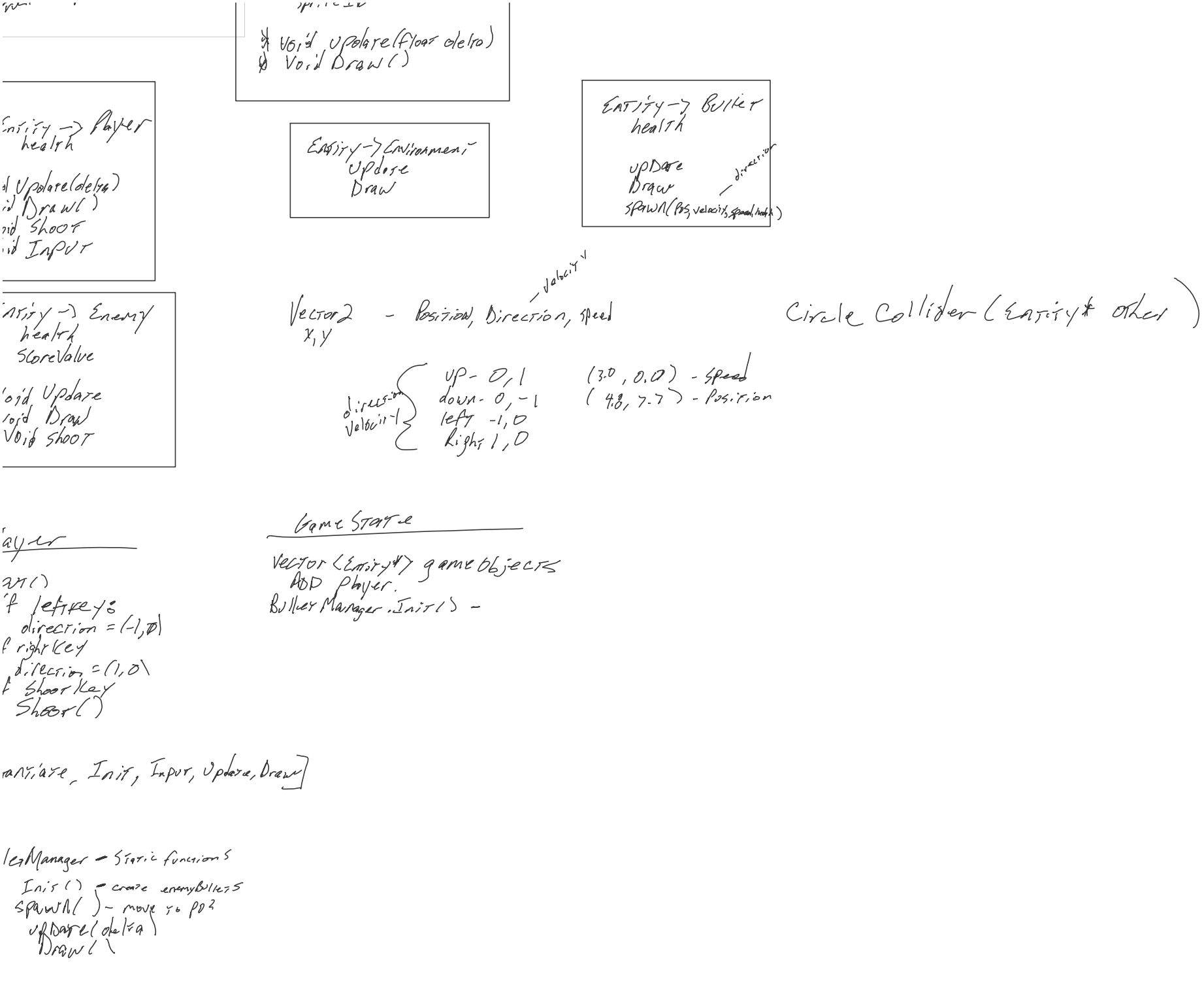
ScoreValue
isActive
returnPosition
attackAngle
attackRadius
attackExitPoint
attackState
shootTimer
ShootMaxTime

Bullet
isActive

Entity
Pos
width
height
speed
alive
SpriteID

Add Helper Methods

w/ Path, width, height


Void Update(float delta)
Void Draw()
Entity -> Player
health
Update(delta)
Draw()
Shoot
Input
Entity -> Environment
Update
Draw
Entity -> Bullet
health
update
Draw
direction
spawn(Pos, velocity, speed, health)
Entity -> Enemy
health
ScoreValue
Void Update
Void Draw
Void Shoot
Vector2
x, y
- Position, Direction, speed
Velocity
Circle Collider(Entity* other)
direction
Velocity
up - 0, 1
down - 0, -1
left -1, 0
Right 1, 0
(3.0, 0.0) - speed
(4.8, 7.7) - Position
Game State
Vector<Entity*> gameObjects
Add Player.
BulletManager.Init() -
ayer
nit()
f leftkey:
direction = (-1, 0)
f rightkey
direction = (1, 0)
f shootkey
Shoot()
nitialize, Init, Input, Update, Draw]
letManager - Static functions
Init() - create enemyBullets
spawn() - move to pos
update(delta)
Draw()