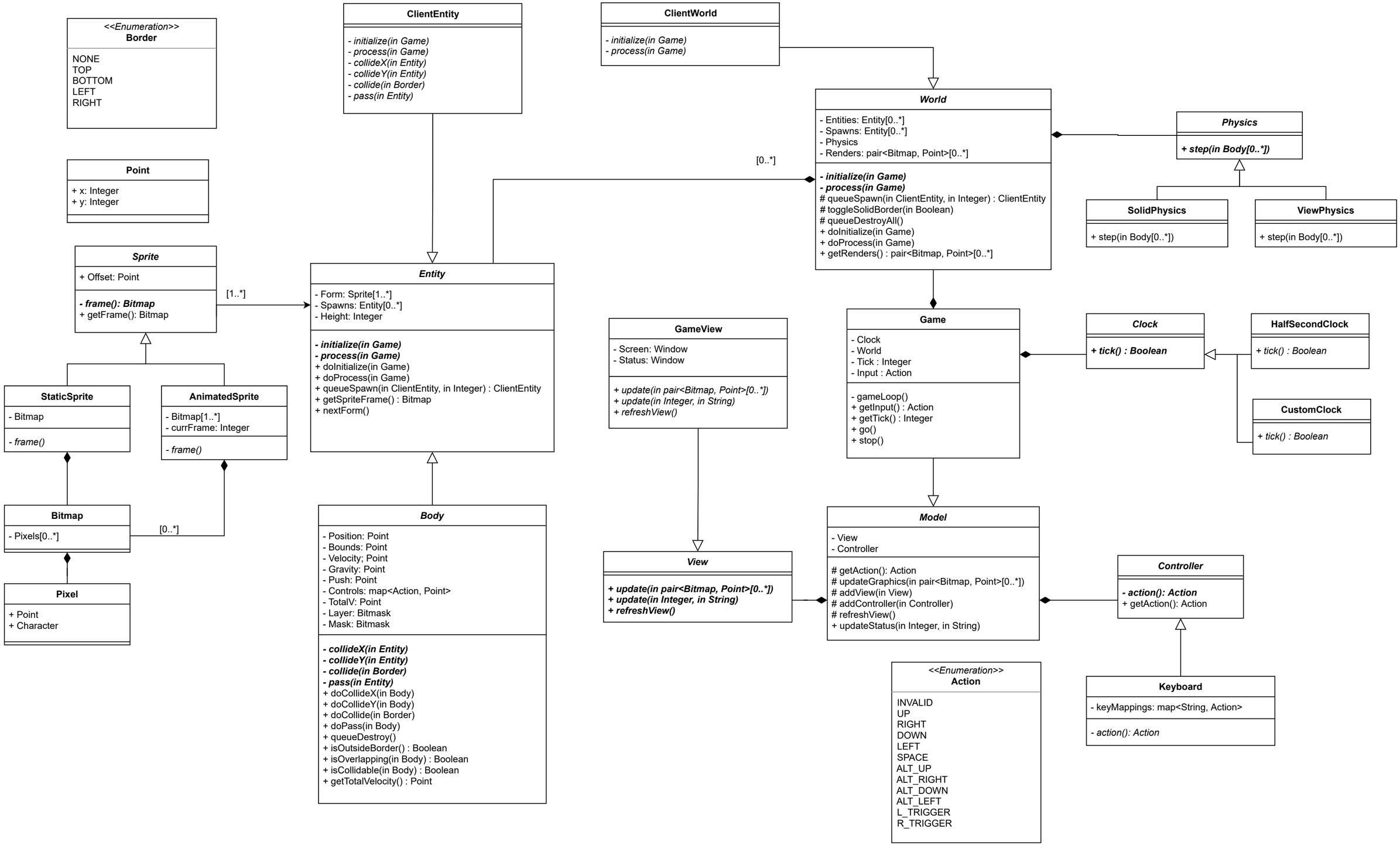
<<Enumeration>>
Border
NONE
TOP
BOTTOM
LEFT
RIGHT
ClientEntity
- initialize(in Game)
- process(in Game)
- collideX(in Entity)
- collideY(in Entity)
- collide(in Border)
- pass(in Entity)
ClientWorld
- initialize(in Game)
- process(in Game)
World
- Entities: Entity[0..*]
- Spawns: Entity[0..*]
- Physics
- Renders: pair<Bitmap, Point>[0..*]
- initialize(in Game)
- process(in Game)
queueSpawn(in ClientEntity, in Integer) : ClientEntity
toggleSolidBorder(in Boolean)
queueDestroyAll()
+ doInitialize(in Game)
+ doProcess(in Game)
+ getRenders() : pair<Bitmap, Point>[0..*]
Physics
+ step(in Body[0..*])
SolidPhysics
+ step(in Body[0..*])
ViewPhysics
+ step(in Body[0..*])
[0..*]
Point
+ x: Integer
+ y: Integer
Sprite
+ Offset: Point
- frame(): Bitmap
+ getFrame(): Bitmap
[1..*]
Entity
- Form: Sprite[1..*]
- Spawns: Entity[0..*]
- Height: Integer
- initialize(in Game)
- process(in Game)
+ doInitialize(in Game)
+ doProcess(in Game)
+ queueSpawn(in ClientEntity, in Integer) : ClientEntity
+ getSpriteFrame() : Bitmap
+ nextForm()
GameView
- Screen: Window
- Status: Window
+ update(in pair<Bitmap, Point>[0..*])
+ update(in Integer, in String)
+ refreshView()
Game
- Clock
- World
- Tick : Integer
- Input : Action
- gameLoop()
+ getInput() : Action
+ getTick() : Integer
+ go()
+ stop()
Clock
+ tick() : Boolean
HalfSecondClock
+ tick() : Boolean
CustomClock
+ tick() : Boolean
StaticSprite
- Bitmap
- frame()
AnimatedSprite
- Bitmap[1..*]
- currFrame: Integer
- frame()
Bitmap
- Pixels[0..*]
[0..*]
Pixel
+ Point
+ Character
Body
- Position: Point
- Bounds: Point
- Velocity; Point
- Gravity: Point
- Push: Point
- Controls: map<Action, Point>
- TotalV: Point
- Layer: Bitmask
- Mask: Bitmask
- collideX(in Entity)
- collideY(in Entity)
- collide(in Border)
- pass(in Entity)
+ doCollideX(in Body)
+ doCollideY(in Body)
+ doCollide(in Border)
+ doPass(in Body)
+ queueDestroy()
+ isOutsideBorder() : Boolean
+ isOverlapping(in Body) : Boolean
+ isCollidable(in Body) : Boolean
+ getTotalVelocity() : Point
View
+ update(in pair<Bitmap, Point>[0..*])
+ update(in Integer, in String)
+ refreshView()
Model
- View
- Controller
getAction(): Action
updateGraphics(in pair<Bitmap, Point>[0..*])
addView(in View)
addController(in Controller)
refreshView()
+ updateStatus(in Integer, in String)
Controller
- action(): Action
+ getAction(): Action
<<Enumeration>>
Action
INVALID
UP
RIGHT
DOWN
LEFT
SPACE
ALT_UP
ALT_RIGHT
ALT_DOWN
ALT_LEFT
L_TRIGGER
R_TRIGGER
Keyboard
- keyMappings: map<String, Action>
- action(): Action