AMMUNICORN

A 2D Run-and-Gun Action Game

CONCEPT

Ammunicorn is a 2D side-scrolling action game in which a unicorn uses rainbow-based weaponry to free her woodland friends from vicious monsters.

Graham, Clay. *The Badass Unicorn*. 2014. *ArtCorgi*. Web. 28 January 2017.

INSPIRATION

Metroid. Nintendo. 1986. Video Game.

Contra. Konami. 1987. Video Game.

PROJECT SCOPE

- Running, jumping, shooting player character
- Levels with ground, platforms, and pits
- Start with one weapon, add variety later
- Start with one enemy basic AI, add variety later
- Collision detection
- Heads-up display
- Sound effects and background music

POTENTIAL LIBRARIES

- Minim Library
  - Background music
  - Point sound

- AI for 2D Games Library
  - Creating AI profiles for the enemies
  - Includes a Vector2D class that uses the double data type instead of float.

- Hermes Library
  - Useful tutorials and possible resource for level-building
  - https://github.com/rdlester/hermes/wiki/Tutorial-Pt.-0:-Before-Getting-Started

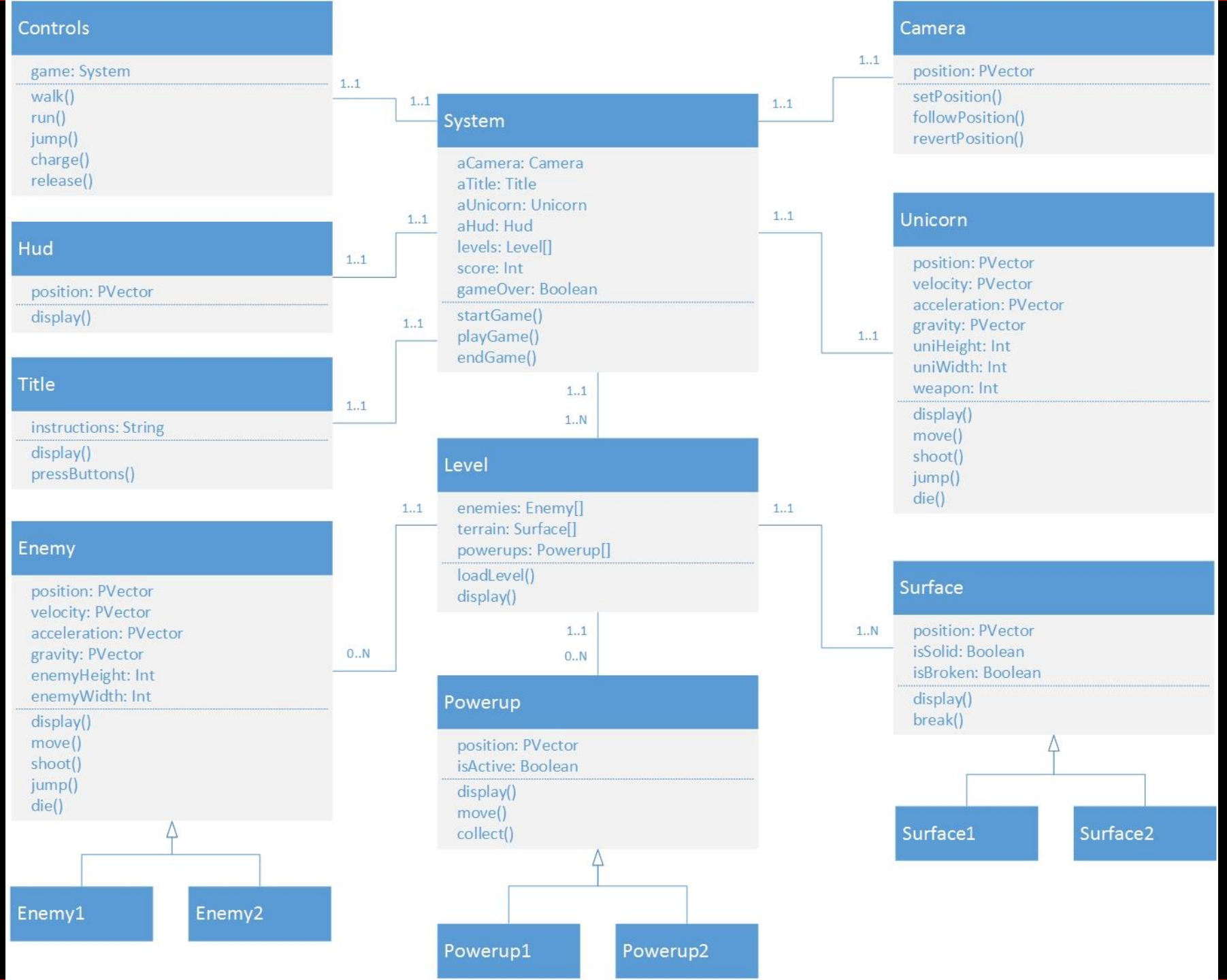
- Sprites Library?
  - Sprite control and animation

COMPUTATIONAL MODELING

- Break levels into blocks and represent them in 2D arrays
- Use variables to store key presses and releases for smoother motion
- Scroll the camera based on the player character's position by using the translate() function
- Use the player character's xSpeed and ySpeed to determine what types of collisions to test for
- Only check for collisions for blocks near the player character / enemy

COMPUTATIONAL RESOURCES

- Very simple platforming example:
  - https://www.openprocessing.org/sketch/119522
- Tutorial for a simple Mario level using a Processing.js game engine library:
  - http://processingjs.nihongoresources.com/test/PjsGameEngine/docs/tutorial/mario.html
- Very simple 2D camera tutorial
  - https://www.openprocessing.org/sketch/136702


Controls
game: System
walk()
run()
jump()
charge()
release()
Hud
position: PVector
display()
Title
instructions: String
display()
pressButtons()
Enemy
position: PVector
velocity: PVector
acceleration: PVector
gravity: PVector
enemyHeight: Int
enemyWidth: Int
display()
move()
shoot()
jump()
die()
Enemy1
Enemy2
System
aCamera: Camera
aTitle: Title
aUnicorn: Unicorn
aHud: Hud
levels: Level[]
score: Int
gameOver: Boolean
startGame()
playGame()
endGame()
Level
enemies: Enemy[]
terrain: Surface[]
powerups: Powerup[]
loadLevel()
display()
Powerup
position: PVector
isActive: Boolean
display()
move()
collect()
Powerup1
Powerup2
Camera
position: PVector
setPosition()
followPosition()
revertPosition()
Unicorn
position: PVector
velocity: PVector
acceleration: PVector
gravity: PVector
uniHeight: Int
uniWidth: Int
weapon: Int
display()
move()
shoot()
jump()
die()
Surface
position: PVector
isSolid: Boolean
isBroken: Boolean
display()
break()
Surface1
Surface2
1..1
1..N
0..N