

Project Phase 3

Kernel Panic



Christopher Greinke, Doug VanAllen, James Butts, Johnathan Pelz

Table of Contents

1 Class Diagrams

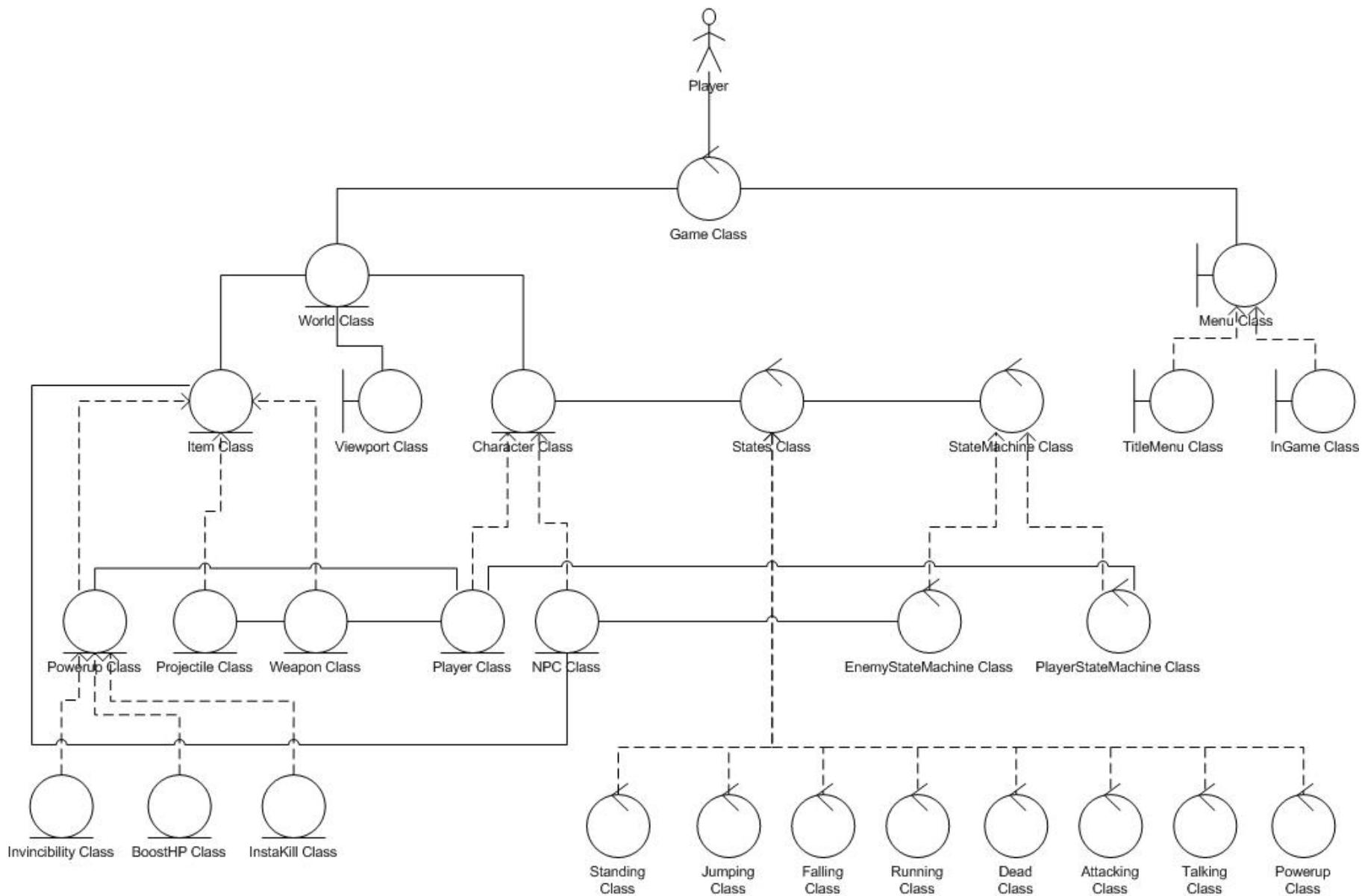
- 1.1 Overall Class Diagram
- 1.2 Entity Class Diagram
- 1.3 Control Class Diagram
- 1.4 Boundary Class Diagram

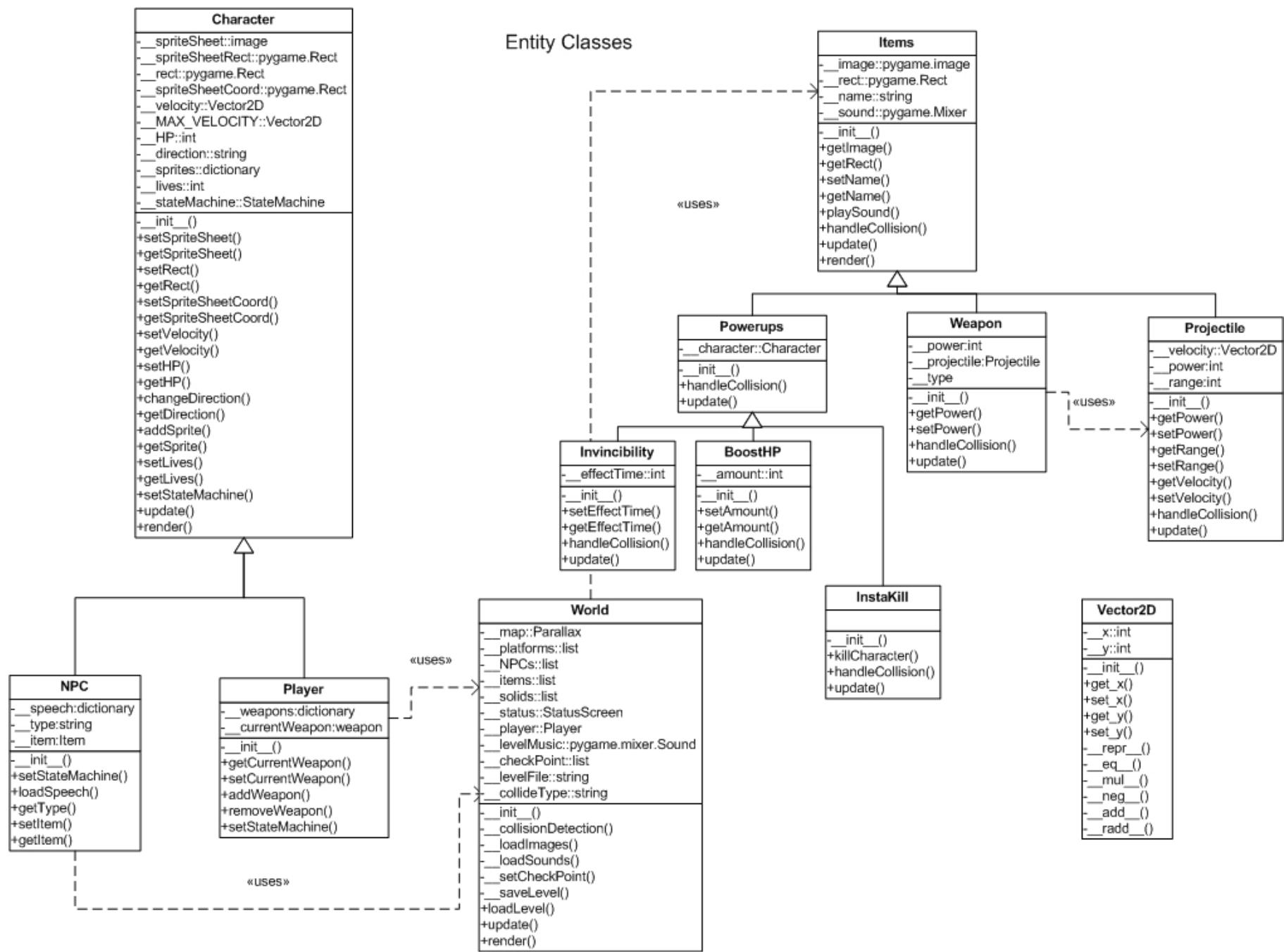
2 Tabular Representation

- 2.1 Entity Classes
- 2.2 Control Classes
- 2.3 Boundary Classes

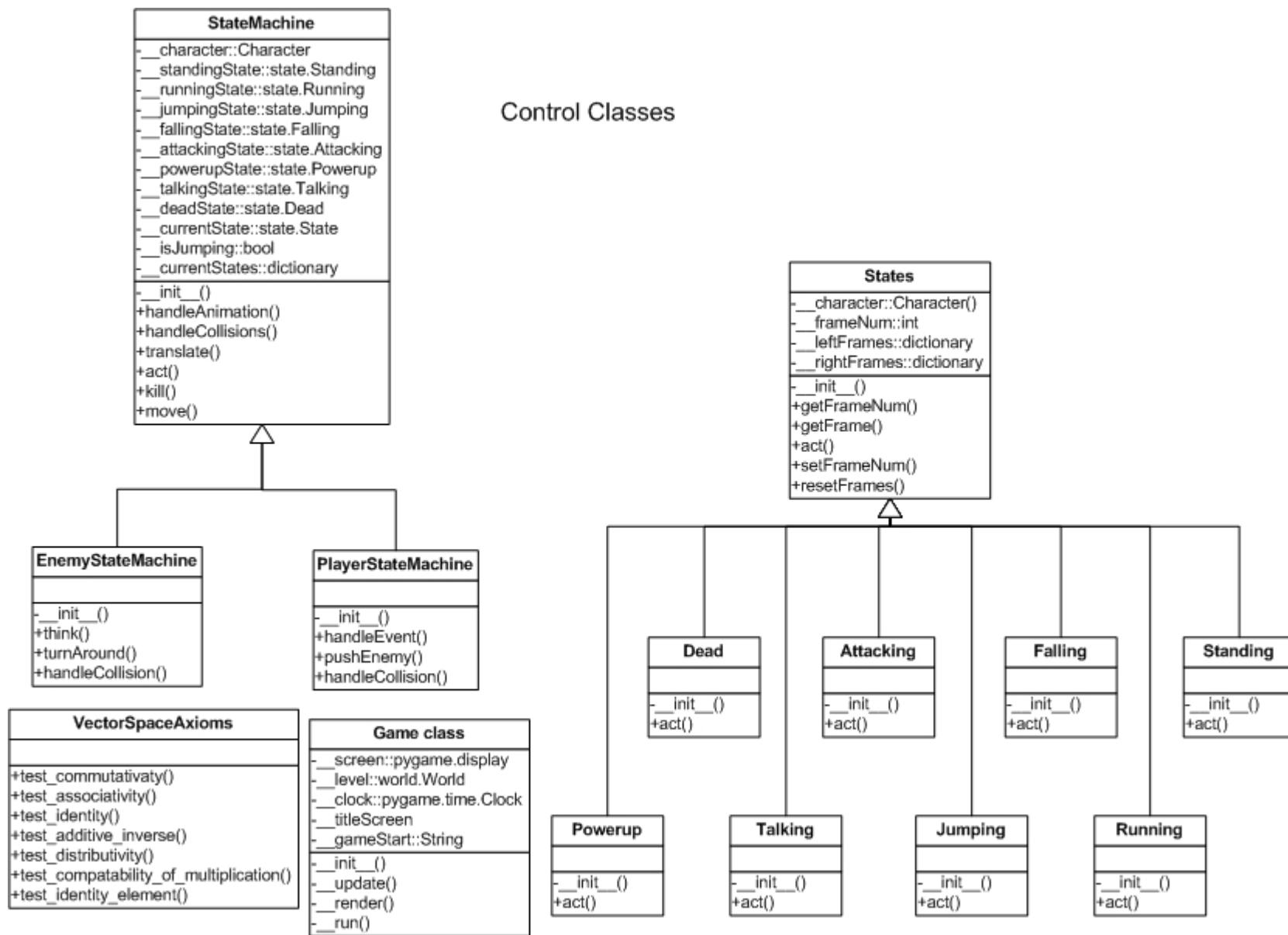
1 Class Diagrams

Class Diagram

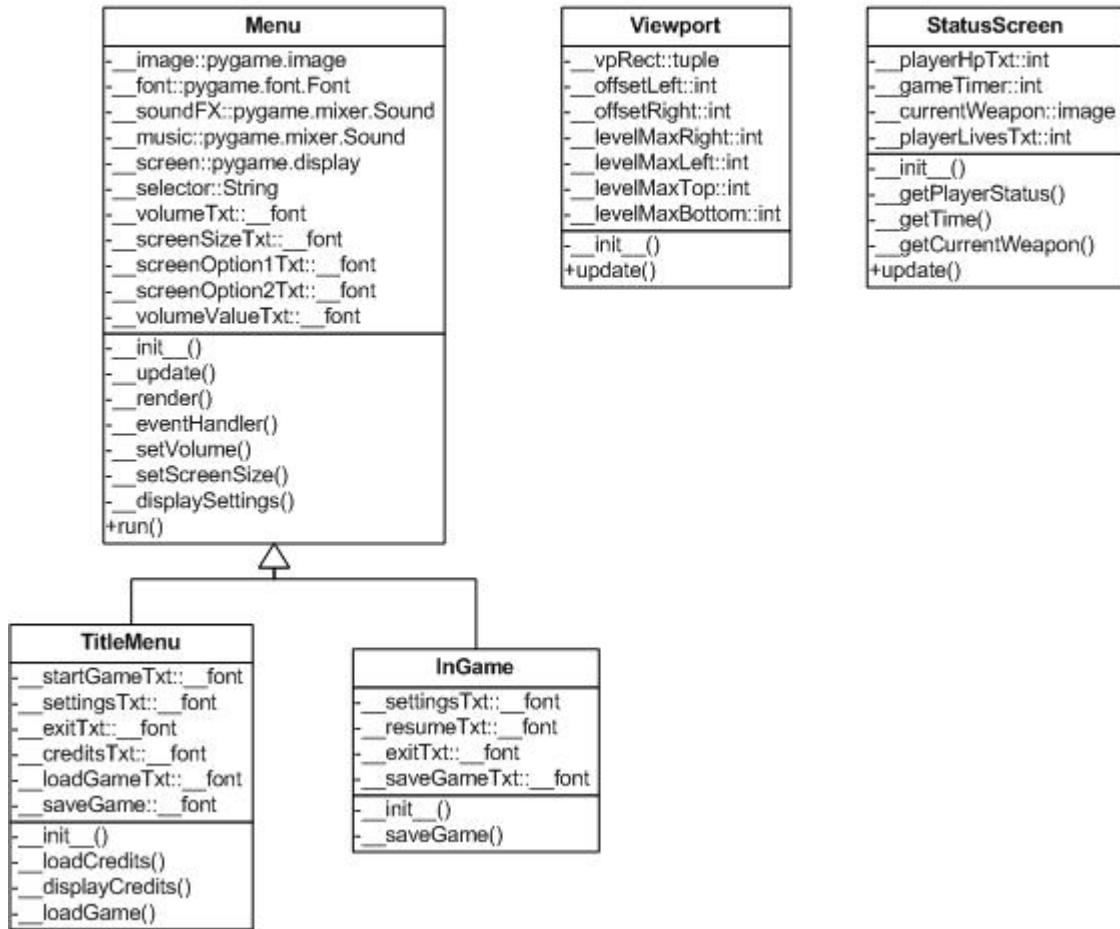




Control Classes



Boundary Classes



2 Tabular Representation of the Class Diagram

2.1 Entity Classes

- Item Class
- Weapon Class
- Projectile Class
- PowerUp Class
- Invincibility Class
- BoostHP Class
- InstaKill Class
- Character Class
- Player Class
- NPC Class
- World Class
- Vector2D

Item Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>rect::pygame.Rect</code> <code>name::string</code> <code>sound::pygame.mixer</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Assign <code>image</code> , <code>rect</code> , <code>name</code> , and <code>sound</code> argument variables to <code>_image</code> , <code>_rect</code> , <code>_name</code> , and <code>_sound</code> class variables respectively.
Module name	<code>getImage</code>
Module type	Function
Return type	<code>pygame.image</code>
Input arguments	none
Output arguments	<code>_image::pygame.image</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the appropriate image stored in the class variable <code>_image</code> .
Module name	<code>getRect</code>
Module type	Function
Return type	<code>pygame.Rect</code>
Input arguments	none
Output arguments	<code>_rect::pygame.Rect</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the appropriate rect stored in the class variable <code>_rect</code> .
Module name	<code>setName</code>
Module type	Function
Return type	none
Input arguments	<code>name::string</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_name</code> equal to the argument value <code>name</code> .

Module name	getName
Module type	Function
Return type	string
Input arguments	none
Output arguments	_name::string
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the appropriate name stored in the class variable <i>_name</i> .
Module name	playSound
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	_sound
Files changed	none
Modules called	none
Narrative	Play the appropriate sound located in the class variable <i>_sound</i> by calling <i>self._sound.play()</i> .
Module name	update
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	An virtual function who's only purpose is to dynamically bind it's derived classes update functions.
Module name	render
Module type	Function
Return type	none
Input arguments	screen::pygame.display
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Render will blit the <i>_image</i> in the <i>_rect</i> position.

Weapons Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>rect::pygame.Rect</code> <code>name::string</code> <code>sound::pygame.mixer</code> <code>power::int</code> <code>projectile::Projectile</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Item.__init__()</code>
Narrative	Call the base classes' constructor and pass in image, rect, name, and sound. Assign power and projectile argument variables to <code>_power</code> and <code>projectile</code> class variables respectively.
Module name	<code>getPower</code>
Module type	Function
Return type	int
Input arguments	none
Output arguments	<code>_power::int</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_power</code> .
Module name	<code>setPower</code>
Module type	Function
Return type	none
Input arguments	<code>power::int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_power</code> equal to the argument value power.

Module name	getProjectile
Module type	Function
Return type	Projectile
Input arguments	none
Output arguments	_projectile::Projectile
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <i>_power</i> .
Module name	setProjectile
Module type	Function
Return type	none
Input arguments	projectile::Projectile
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <i>_projectile</i> equal to the argument value <i>projectile</i> .

Projectile Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>rect::pygame.Rect</code> <code>name::string</code> <code>sound::pygame.mixer</code> <code>velocity::Vector2D</code> <code>power::int</code> <code>range::int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Item.__init__()</code>
Narrative	Call the base classes' constructor and pass in image, rect, name, and sound. Assign velocity, power, and range argument variables to <code>_velocity</code> , <code>_power</code> , and <code>_range</code> class variables respectively.
Module name	<code>getPower</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	<code>_power::int</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_power</code> .
Module name	<code>setPower</code>
Module type	Function
Return type	none
Input arguments	<code>power::int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_power</code> equal to the argument value <code>power</code> .

Module name	getRange
Module type	Function
Return type	int
Input arguments	none
Output arguments	_range::int
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <i>_range</i> .
Module name	setRange
Module type	Function
Return type	none
Input arguments	range::int
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <i>_range</i> equal to the argument value range.
Module name	getVelocity
Module type	Function
Return type	Vector2D
Input arguments	none
Output arguments	_velocity::Vector2D
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <i>_velocity</i> .
Module name	setVelocity
Module type	Function
Return type	none
Input arguments	velocity::Vector2D
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <i>_velocity</i> equal to the argument value velocity.

Module name	update
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Offset the Projectile Item's <code>_rect</code> by the current <code>_velocity</code> 's x and y values by calling move().

PowerUp Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>rect::pygame.Rect</code> <code>name::string</code> <code>sound::pygame.mixer</code> <code>character::Character</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Item.__init__()</code>
Narrative	Call the base classes' constructor and pass in <code>image</code> , <code>rect</code> , <code>name</code> , and <code>sound</code> . Assign <code>character</code> argument variable to <code>_character</code> class variable.

Invisibility Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>rect::pygame.Rect</code> <code>name::string</code> <code>sound::pygame.mixer</code> <code>effectTime::int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Item.__init__()</code>
Narrative	Call the base classes' constructor and pass in image, rect, name, and sound. Assign character argument variable to <code>_effectTime</code> class variable.
Module name	<code>getEffectTime</code>
Module type	Function
Return type	int
Input arguments	none
Output arguments	<code>_effectTime</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_effectTime</code> .
Module name	<code>setEffectTime</code>
Module type	Function
Return type	none
Input arguments	<code>effectTime::int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_effectTime</code> equal to the argument value <code>effectTime</code> .

BoostHP Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>rect::pygame.Rect</code> <code>name::string</code> <code>sound::pygame.mixer</code> <code>amount:int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Item.__init__()</code>
Narrative	Call the base classes' constructor and pass in image, rect, name, and sound. Assign amount argument variable to <code>_amount</code> class variable.
Module name	<code>setAmount</code>
Module type	Function
Return type	none
Input arguments	<code>amount:int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_amount</code> equal to the argument value <code>amount</code> .
Module name	<code>getAmount</code>
Module type	Function
Return type	<code>int</code>
Input arguments	none
Output arguments	<code>_amount:int</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_amount</code> .

InstaKill Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>rect::pygame.Rect</code> <code>name::string</code> <code>sound::pygame.mixer</code> <code>amount:int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Item.__init__()</code>
Narrative	Call the base classes' constructor and pass in image, rect, name, and sound. Assign amount argument variable to <code>_amount</code> class variable.

Module name	<code>killCharacter</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Subtract <code>_amount</code> of HP from the character's HP value.

Character Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>spriteSheet::Image</code> <code>sprites::dictionary</code> <code>MAX_VELOCITY::Vector2D</code> <code>stateMachine::stateMachine</code>
Output arguments	none
Error messages	none
Files accessed	<code>spriteSheet::Image</code>
Files changed	none
Modules called	none
Narrative	Assign <code>spriteSheet</code> , <code>sprites</code> , <code>MAX_VELOCITY</code> , and <code>stateMachine</code> argument variables to <code>_spriteSheet</code> , <code>_sprites</code> , <code>_MAX_VELOCITY</code> , and <code>_stateMachine</code> class variables respectively.
Module name	<code>setSpriteSheet</code>
Module type	Function
Return type	none
Input arguments	<code>spriteSheet::image</code>
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_spriteSheet</code> equal to the argument value <code>spriteSheet</code> .
Module name	<code>getSpriteSheet</code>
Module type	Function
Return type	Image
Input arguments	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_spriteSheet</code> .
Module name	<code>setRect</code>
Module type	Function
Return type	none
Input arguments	<code>levelPosition::pygame.Rect</code>
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_levelPosition</code> equal to the argument value <code>levelPosition</code> .

Module name	getRect
Module type	Function
Return type	pygame.Rect
Input arguments	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <i>_rect</i> .
Module name	setSpriteSheetCoord
Module type	Function
Return type	none
Input arguments	rect::pygame.Rect
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <i>_spriteSheetCoord</i> equal to the argument value rect.
Module name	getSpriteSheetCoord
Module type	Function
Return type	pygame.Rect
Input arguments	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <i>_spriteSheetCoord</i> .
Module name	setVelocity
Module type	Function
Return type	none
Input arguments	velocity::Vector2D
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <i>_velocity</i> equal to the argument value velocity.
Module name	getVelocity
Module type	Function
Return type	Vector2D
Input arguments	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <i>_velocity</i> .

Module name	setHP
Module type	Function
Return type	none
Input arguments	life::int
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_life</code> equal to the argument value <code>life</code> .
Module name	getHP
Module type	Function
Return type	Int
Input arguments	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>__HP</code> .
Module name	changeDirection
Module type	Function
Return type	none
Input arguments	direction::String
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_direction</code> equal to the argument value <code>direction</code> .
Module name	getDirection
Module type	Function
Return type	String
Input arguments	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_direction</code> .
Module name	addSprite
Module type	Function
Return type	none
Input arguments	key::String rect::pygame.Rect
Files accessed	none
Files changed	none
Modules called	none
Narrative	Append a new key-value pair to the <code>_sprites</code> list using the <code>key</code> and <code>rect</code> argument variables respectively.

Module name	getSprite
Module type	Function
Return type	pygame.Rect
Input arguments	key::String
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns a rect by indexing the <code>_sprites</code> dictionary with the key argument.
Module name	setLives
Module type	Function
Return type	none
Input arguments	numLives::int
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_lives</code> equal to the argument value <code>numLives</code> .
Module name	getLives
Module type	Function
Return type	int
Input arguments	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_lives</code> .
Module name	setStateMachine
Module type	Function
Return type	none
Input arguments	stateMachine::StateMachine
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_stateMachine</code> equal to the argument value <code>stateMachine</code> .
Module name	update
Module type	Abstract Method
Return type	none
Input arguments	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	An virtual function who's only purpose is to dynamically bind it's derived classes update functions.

Player Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Character.__init__()</code>
Narrative	Call the base classes' constructor and pass in default values for spriteSheet, sprites, MAX_VELOCITY, stateMachine.
Module name	<code>getCurrentWeapon</code>
Module type	Function
Return type	Weapon
Input arguments	none
Output arguments	returns <code>_currentWeapon</code> .
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_currentWeapon</code> .
Module name	<code>setCurrentWeapon</code>
Module type	Function
Return type	none
Input arguments	<code>keyString:String</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set class variable <code>_currentWeapon</code> to the value corresponding to <code>keyString</code> in the <code>_weapons</code> dictionary.
Module name	<code>addWeapon</code>
Module type	Function
Return type	none
Input arguments	<code>weapon:Weapon</code> <code>keyString:String</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Append a new key-value pair to the <code>_weapons</code> dictionary using the <code>keyString</code> and <code>weapon</code> argument variables respectively.

Module name	removeWeapon
Module type	Function
Return type	none
Input arguments	keyString::String
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Delete a weapon from the <code>_weapons</code> dictionary that is retrieved by indexing with the <code>keyString</code> argument variable.
Module name	setStateMachine
Module type	Function
Return type	none
Input arguments	stateMachine::StateMachine
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_stateMachine</code> equal to the argument variable <code>stateMachine</code> .

NPC Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>type::String</code> <code>speechFile::String</code> <code>Item::Item</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Character.__init__()</code> , <code>loadSpeech()</code>
Narrative	Call base class constructor. Set class variables <code>_type</code> and <code>_item</code> to type and item argument variables respectively. Call <code>loadSpeech(speechFile)</code> .
Module name	<code>setStateMachine</code>
Module type	Function
Return type	none
Input arguments	<code>stateMachine::StateMachine</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <code>_stateMachine</code> equal to the argument value <code>stateMachine</code> .
Module name	<code>loadSpeech</code>
Module type	Function
Return type	none
Input arguments	<code>speechFile::String</code>
Output arguments	none
Error messages	If file fails to load, throw an exception.
Files accessed	# <code>speechFile#.txt</code>
Files changed	none
Modules called	<code>open()</code>
Narrative	Load textual information from given argument file name.
Module name	<code>getType</code>
Module type	Function
Return type	String
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <code>_type</code> .

Module name	setItem
Module type	Function
Return type	none
Input arguments	item::Item
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the class variable <i>_item</i> equal to the argument value item.
Module name	getItem
Module type	Function
Return type	Item
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the class variable <i>_item</i> .

World Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>levelFile</code>
Output arguments	none
Error messages	none
Files accessed	<code>checkpoint.ckp</code> , <code>level.lvl</code> , <code>level.ogg</code> ,
Files changed	<code>checkPoint.ckp</code> , <code>level.lvl</code>
Modules called	<code>pygame.mixer.sound()</code> , <code>pygame.image.load()</code> , <code>image.get_rect()</code> , <code>loadLevel()</code>
Narrative	This module initializes the world class. Sets all class variables to null. Call <code>loadLevel</code> based upon the argument from <code>__init__()</code> .
Module name	<code>_loadLevel</code>
Module type	Function
Return type	none
Input argument	<code>levelFile::String</code>
Output arguments	none
Error messages	Cannot access file.
Files accessed	<code>levelFile</code>
Files changed	none
Modules called	none
Narrative	Loads the file based on the argument <code>levelFile</code> . Appropriately sets all the World class variables to the values denoted by contents of <code>levelFile</code> .
Module name	<code>update</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>collisionDetection()</code>
Narrative	Call <code>collisionDetection()</code> to update all objects in the World.
Module name	<code>render</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>pygame.flip()</code> , <code>pygame.blit()</code> , <code>render()</code>
Narrative	Calls <code>render()</code> for all objects in the World. Calls <code>pygame.flip()</code> and <code>pygame.blit()</code> .

Module name	<code>_saveLevel</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	checkpoint.ckp
Files changed	checkpoint.ckp
Modules called	<code>open()</code> , <code>close()</code>
Narrative	Take current object variable values and write to checkpoint.ckp in a specified delimited format.

Module name	<code>_collisionDetection</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>StateMachine.handleCollision()</code> , <code>pygame.colliderect()</code>
Narrative	Check intersections of rectangles between characters, World objects, and projectiles by calling <code>pygame.colliderect()</code> . Calls the StateMachines <code>handleCollision()</code> on the character.

Module name	<code>_loadImage</code>
Module type	Function
Return type	none
Input arguments	<code>fileName::String</code>
Output arguments	none
Error messages	none
Files accessed	Appropriate image file (png, jpg, or).
Files changed	none
Modules called	<code>pygame.image.load()</code>
Narrative	Load an image with a name equal to argument variable filename.

Module name	<code>_loadSound</code>
Module type	Function
Return type	none
Input arguments	<code>fileName::String</code>
Output arguments	none
Error messages	none
Files accessed	Appropriate sound file (ogg).
Files changed	none
Modules called	<code>pygame.mixer.Sound()</code>
Narrative	Load a sound with a name equal to argument variable filename.

Module name	_reachedCheckPoint
Module type	Function
Return type	none
Input arguments	characterRect::rect
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	_saveLevel()
Narrative	Checkes characterRect against class variable <i>_checkPoint</i> to determine if the characters position has either reached or passed <i>_checkPoint</i> .

Vector2D Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>values::list</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	set local class variables <code>__x</code> to the first element of values and <code>__y</code> to second element in values.
Module name	<code>get_x</code>
Module type	Function
Return type	int
Input arguments	none
Output arguments	<code>__x</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	returns the class variable <code>__x</code> .
Module name	<code>set_x</code>
Module type	Function
Return type	none
Input arguments	<code>value::int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Sets the class variable <code>__x</code> to the input argument value.
Module name	<code>get_y</code>
Module type	Function
Return type	int
Input arguments	none
Output arguments	<code>__y</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	returns the class variable <code>__y</code> .

Module name	<code>set_y</code>
Module type	Function
Return type	none
Input arguments	<code>value::int</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Sets the class variable <code>_y</code> to the input argument value.
Module name	<code>__repr__</code>
Module type	Function
Return type	String
Input arguments	none
Output arguments	string representing the vector
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Converts the vector to a string, and returns it.
Module name	<code>__eq__</code>
Module type	Function
Return type	Boolean
Input arguments	<code>other::Vector2D</code>
Output arguments	True if x and y values of the two vectors are equal.
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the Boolean representation of if x and y values of the two vectors are equal.
Module name	<code>__mul__</code>
Module type	Function
Return type	<code>Vector2D</code>
Input arguments	<code>scalar::int</code>
Output arguments	<code>Vector2D</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Multiplying the vector by the input argument scalar.

Module name `neg`
Module type Function
Return type Vector2D
Input arguments none
Output arguments Vector2D
Error messages none
Files accessed none
Files changed none
Modules called none
Narrative Returns the negated vector.

Module name `add`
Module type Function
Return type Vector2D
Input arguments other::Vector2D
Output arguments vector2D
Error messages none
Files accessed none
Files changed none
Modules called none
Narrative returns the sum of two vectors.

Module name `radd`
Module type Function
Return type Vector2D
Input arguments other::Vector2D
Output arguments vector2D
Error messages none
Files accessed none
Files changed none
Modules called none
Narrative returns the sum of two vectors.

2.2 Control Classes

- Game Class
- StateMachine Class
- EnemyStateMachine Class
- PlayerStateMachine Class
- State Class
- Standing Class
- Standing Class
- Jumping Class
- Falling Class
- Running Class
- Dead Class
- Attacking Class
- Talking Class
- PowerUp Class
- VectorSpaceAxioms

Game Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Sets class variables to their initial values. <code>_screen</code> to <code>640x480</code> , <code>_level</code> to level one, <code>_clock</code> to <code>pygame.time.Clock()</code> , <code>_titleScreen</code> to current, <code>_gameStart</code> to false.
Module name	<code>update</code>
Module type	Function
Return type	none
Input arguments	<code>updateFlag::String</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>_titleScreen.update()</code> , <code>_level.update()</code>
Narrative	Based on the updateFlag call either <code>_titleScreen.update()</code> , or <code>_level.update()</code> .
Module name	<code>render</code>
Module type	Function
Return type	none
Input arguments	<code>updateFlag::String</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>_titleScreen.render()</code> , <code>_level.render()</code>
Narrative	Based on the updateFlag call either <code>_titleScreen.render()</code> , or <code>_level.render()</code> .
Module name	<code>run</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>update()</code> , <code>render()</code>
Narrative	This is the main game loop. Pass a string as an argument to tell update and render what to update or render.

StateMachine Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>character::character, sprite::dictionary</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Sets class variables <code>_character</code> and <code>_sprites</code> to the passed in arguments character and sprite. Create instances of all the states and assign the states to class variables. Set the class variable <code>_isJumping</code> to false. Add the falling state to the class dictionary <code>_currentStates</code> .
Module name	<code>handleAnimation</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>self._character.setSpriteSheetCoord()</code>
Narrative	Checks character's dictionary of sprites and cycles through the dictionary with each call. There is a priority to which sprites to display. Attacking, running, then standing respectively. If switching from a new state's sprites then reset the frame number of the previous state.
Module name	<code>translate</code>
Module type	Function
Return type	String
Input arguments	<code>rect::pygame.Rect()</code>
Output argument	<code>typeOfColl</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>self._character.getRect()</code>
Narrative	This method will calculate the minimum translation distance. This distance is the smallest of the distances between: The character's bottom with the rect's top, The character's top with the rect's bottom, The character's left with the rect's right, The character's right with the rect's left. Takes the shortest distance of these and translates the character's position accordingly, based on the collision. Also set the character's X or Y Velocity to zero, based upon the collision.

Module name	act
Module type	Function
Return type	none
Input argument	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	self._currentStates.has_key(), state.act()
Narrative	For each state in the current state dictionary call the state's act(). Then check to see if the current state dictionary contains the jumping state, if so then remove from the dictionary.
Module name	kill
Module type	Function
Return type	none
Input argument	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	self._currentStates.has_key()
Narrative	Check to see if the running state are in the state's dictionary, if so then remove from the dictionary. Then add the dead state to the dictionary.
Module name	move
Module type	Function
Return type	none
Input argument	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Module called	self._character.getRect()
Narrative	Move the character's position based upon the character's velocity.

EnemyStateMachine Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, sprites::dictionary, playerRect::pygame.Rect, topographyRects::pygame.Rect
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	self.tRects.append(), super().__init__()
Narrative	Calls base class's <u>__init__()</u> , passing character and sprites as parameters. Add the running state and falling state to the <u>_currentStates</u> 's dictionary. Set class variable <i>counter</i> to zero. For every rectangle in topographyRects append it to the class variable <i>tRects</i> .
Module name	think
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<u>_currentStates[state].act()</u> , <u>_currentStates.has_key()</u> , self._character.getRect().collideRect(), handleCollision()
Narrative	For every state in currentStates call the state's act(). Then if jumping state is in the currentState then delete it. Call handleCollision for every rect in tRect. In short based on the character's type, currentState, and level topography change currentState to a different state.
Module name	turnAround
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<u>_character.getDirection()</u> , <u>_character.setDirection()</u> , <u>_currentState.has_key()</u>
Narrative	Based upon the character's direction, change the direction to the opposite direction. Set the character's X Velocity to zero.

Module name	handleCollision
Module type	Function
Return type	none
Input arguments	type::String, rect::pygame.Rect
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	translate(), turnAround()
Narrative	Based on the type of collision, which is passed as the input argument type, either turn around if collides with solid, or decrements the HP if collides with enemy.

PlayerStateMachine Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, sprites::Dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	super().__init__()
Narrative	Call the parent class's constructors, passing both input parameters.
Module name	handleEvent ()
Module type	Function
Return type	none
Input arguments	events::list
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	sys.exit(), _currentState.has_key(), _character.setDirection(), _character.getDirection(), act()
Narrative	Add states to the currentStates, based upon the key down events. Remove states based upon key up events. Based on event set the currentStates and remove states. Then call act().
Module name	pushEnemy
Module type	Function
Return type	none
Input arguments	enemy::pygame.Rect, direction::String
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Based on the direction the player pushes the enemy, translate the enemy position by 150 pixels.
Module name	handleCollision
Module type	Function
Return type	String
Input arguments	type::String, rect::pygame.Rect
Output arguments	coll
Error messages	none
Files accessed	none
Files changed	none
Modules called	translate()
Narrative	Based upon the input argument type, call translate to adjust the character's position.

State Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	character::character, rightFrames::dictionary, leftFrames::dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set all class variables to their corresponding init arguments. With the exception of frame number which gets set to zero.
Module name	<code>getFrame()</code>
Module type	Function
Return type	Integer
Input arguments	frameSet:String
Output arguments	String that represents Sprite.
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Return the string that corresponds to a key in the character's sprite dictionary to retrieve a frame for animation. <code>_frameNum</code> is used to cycle through the appropriate frame set for animation. The frameSet is a collection of strings which will be used as keys in the sprite dictionary.
Module name	<code>getFrameNum()</code>
Module type	Function
Return type	Integer
Input arguments	none
Output arguments	<code>_frameNum:int</code>
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Returns the current frame.

Module name	setFrameNum()
Module type	Function
Return type	none
Input arguments	num::int
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Sets the current frame number to input argument.
Module name	resetFrames
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Sets the <i>_frameNum</i> to zero.
Module name	act()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	This method is a pure virtual method to be overridden by derived classes. Each state has its own act method which will be called to perform some action on the character.

Standing Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, rightFrames::Dictionary, leftFrames::Dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	super().__init__()
Narrative	Call the parent class's constructors.
Module name	act()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Will set the x and y velocity values to zero.

Jumping Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, rightFrames::Dictionary, leftFrames::Dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	super().__init__()
Narrative	Call the parent class's constructors.
Module name	act()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Set the Y Velocity to the negative MAX_VELOCITY.

Falling Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>character::character, rightFrames::Dictionary, leftFrames::Dictionary</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>super().__init__()</code>
Narrative	Call the parent class's constructors.
Module name	<code>act()</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Increment y velocity with each call until velocity is equal to MAX_VELOCITY. If the velocity equals MAX_VELOCITY, no calculations need to be performed so do nothing.

Running Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, rightFrames::Dictionary, leftFrames::Dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	super().__init__()
Narrative	Call the parent class's constructors.

Module name	act()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Based on the character's direction either increment or decrements the x velocity until it reaches MAX_VELOCITY.

Dead Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, rightFrames::Dictionary, leftFrames::Dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	super().__init__()
Narrative	Call the parent class's constructors.
Module name	act()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Take away a character life. Set x and y velocity to zero.

Attacking Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, rightFrames::Dictionary, leftFrames::Dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	super().__init__()
Narrative	Call the parent class's constructors.
Module name	act()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	character.getCurrentWeapon(), Projectile()
Narrative	Inject projectile into the world, based upon the current weapon type.

Talking Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, rightFrames::Dictionary, leftFrames::Dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	super().__init__()
Narrative	Call the parent class's constructors.
Module name	act()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	dialogFile.txt
Files changed	none
Modules called	none
Narrative	Will set the x and y velocity values to zero. Initiate dialog with string from dialogFile.txt.

PowerUp Class

Module name	<u>__init__</u>
Module type	Function
Return type	none
Input arguments	character::character, rightFrames::Dictionary, leftFrames::Dictionary
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	super().__init__()
Narrative	Call the parent class's constructors.
Module name	act()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	none
Narrative	Apply the effects of the powerUp to the character.

VectorSpaceAxioms Class

Module name	test_commutativity
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	assertEqual
Narrative	Tests for commutativity of addition.
Module name	test_associativity
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	assertEqual
Narrative	Tests for associativity of addition.
Module name	test_identity
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	assertEqual
Narrative	Test vector v plus zero is zero.
Module name	test_additive_inverse
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	assertEqual
Narrative	Tests that vector v and the negation of vector v's sum equals zero.

Module name	test_distributivity
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	assertEqual
Narrative	Tests for distributivity of scalar multiplication.
Module name	test_compatibility_of_multiplication
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	assertEqual
Narrative	Tests for compatibility of scalar and field multiplication.
Module name	test_identity_element
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	assertEqual
Narrative	Tests for distributivity of scalar multiplication.

2.3 Boundary Classes

- Menu Class
- TitleMenu Class
- InGameMenu Class
- Viewport Class

Menu Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>soundFX::pygame.mixer.Sound</code> <code>music::pygame.mixer.Sound</code> <code>screen::pygame.display</code>
Output arguments	none
Error messages	none
Files accessed	<code>soundFX.ogg</code> <code>image.png</code> <code>music.ogg</code>
Files changed	none
Modules called	none
Narrative	Assign <code>image</code> , <code>soundFX</code> , <code>music</code> , and <code>screen</code> argument variables to <code>_image</code> , <code>_soundFX</code> , <code>_music</code> , and <code>_screen</code> class variables respectively.

Module name	<code>update</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>_eventHandler()</code>
Narrative	Call <code>_eventHandler()</code> to handle inputs from user and adjust class variables accordingly.

Module name	<code>render</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>pygame.Surface.blit()</code>
Narrative	Blit text and images to a <code>pygame.surface</code> class variable <code>_screen</code> .

Module name	_eventHandler
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	pygame.event.get()
Narrative	Handles events from the event queue by calling pygame.event.get() and processes input from user.

Module name	setVolume
Module type	Function
Return type	none
Input arguments	volume::int
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	pygame.mixer.Sound.set_volume()
Narrative	Adjust the music volume for the entire game based on the input argument volume by calling pygame.mixer.Sound.set_volume().

Module name	setScreenSize
Module type	Function
Return type	none
Input arguments	size::tuple
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	pygame.Display.set_mode()
Narrative	Sets class variable <i>_screen</i> to the new size as determined by the input argument size by calling pygame.Display.set_mode().

Module name	displaySettings()
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	pygame.Surface.blit()
Narrative	Blits the settings text to the screen by calling pygame.Surface.blit().

Module name	run
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	update() render()
Narrative	Utilizes a loop for the menu screen calling update() and render() until the player makes a selection.

TitleMenu Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>soundFX::pygame.mixer.Sound</code> <code>music::pygame.mixer.Sound</code> <code>screen::pygame.display</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Menu.__init__()</code>
Narrative	Call the base class's <code>__init__()</code> and pass in image, soundFX, music, and screen argument variables.
Module name	<code>_loadCredits</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	<code>credits.txt</code>
Files changed	none
Modules called	<code>open()</code> , <code>close()</code>
Narrative	Loads game credits by calling python's <code>open()</code> and <code>close()</code> .
Module name	<code>_displayCredits</code>
Module type	Function
Return type	none
Input arguments	none
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>pygame.Surface.blit()</code>
Narrative	Display game credits and blit to the screen by calling <code>pygame.Surface.blit()</code> .
Module name	<code>_loadGame</code>
Module type	Function
Return type	none
Input arguments	<code>levelName</code>
Output arguments	none
Error messages	none
Files accessed	<code>level.lvl</code>
Files changed	none
Modules called	none
Narrative	Loads a level from a delimited file <code>level.lvl</code> and sets appropriate class variables based on the information obtained.

InGameMenu Class

Module name	<code>__init__</code>
Module type	Function
Return type	none
Input arguments	<code>image::pygame.image</code> <code>soundFX::pygame.mixer.Sound</code> <code>music::pygame.mixer.Sound</code> <code>screen::pygame.display</code>
Output arguments	none
Error messages	none
Files accessed	none
Files changed	none
Modules called	<code>Menu.__init__()</code>
Narrative	Call the base class's <code>__init__()</code> and pass in image, soundFX, music, and screen argument variables.

Module name	<code>_saveGame</code>
Module type	Function
Return type	none
Input arguments	<code>level::World</code>
Output arguments	none
Error messages	none
Files accessed	<code>checkpoint.ckp</code>
Files changed	<code>checkpoint.ckp</code>
Modules called	none
Narrative	Write appropriate class variables to a file <code>checkpoint.ckp</code> in a specified delimited format.