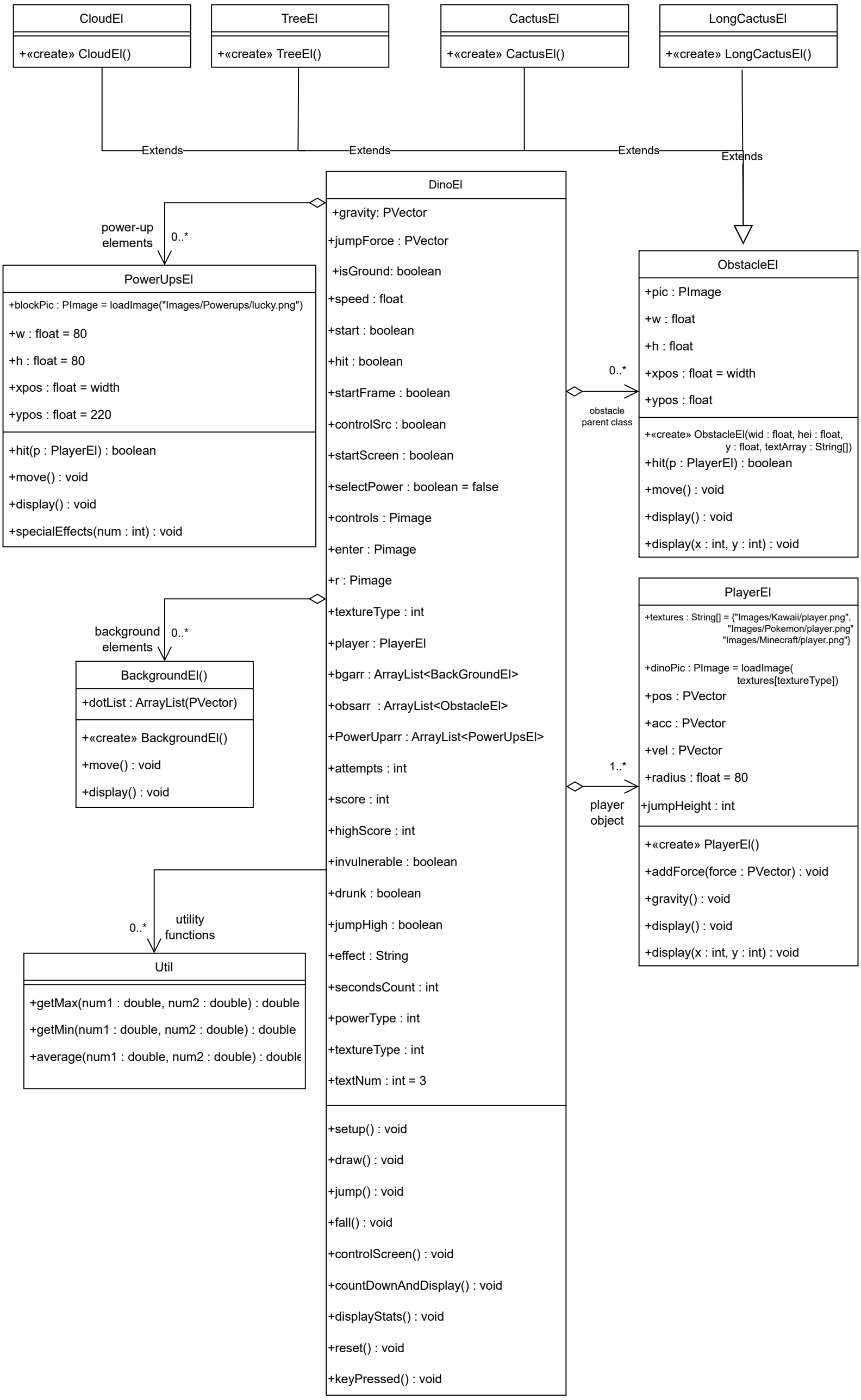
CloudEl
+«create» CloudEl()
TreeEl
+«create» TreeEl()
CactusEl
+«create» CactusEl()
LongCactusEl
+«create» LongCactusEl()
Extends
Extends
Extends
Extends
DinoEl
+gravity: PVector
+jumpForce : PVector
+isGround: boolean
+speed : float
+start : boolean
+hit : boolean
+startFrame : boolean
+controlSrc : boolean
+startScreen : boolean
+selectPower : boolean = false
+controls : Pimage
+enter : Pimage
+r : Pimage
+textureType : int
+player : PlayerEl
+bgarr : ArrayList<BackGroundEl>
+obsarr : ArrayList<ObstacleEl>
+PowerUparr : ArrayList<PowerUpsEl>
+attempts : int
+score : int
+highScore : int
+invulnerable : boolean
+drunk : boolean
+jumpHigh : boolean
+effect : String
+secondsCount : int
+powerType : int
+textureType : int
+textNum : int = 3
+setup() : void
+draw() : void
+jump() : void
+fall() : void
+controlScreen() : void
+countDownAndDisplay() : void
+displayStats() : void
+reset() : void
+keyPressed() : void
power-up elements
0..*
PowerUpsEl
+blockPic : PImage = loadImage("Images/Powerups/lucky.png")
+w : float = 80
+h : float = 80
+xpos : float = width
+ypos : float = 220
+hit(p : PlayerEl) : boolean
+move() : void
+display() : void
+specialEffects(num : int) : void
0..*
obstacle parent class
ObstacleEl
+pic : PImage
+w : float
+h : float
+xpos : float = width
+ypos : float
+«create» ObstacleEl(wid : float, hei : float, y : float, textArray : String[])
+hit(p : PlayerEl) : boolean
+move() : void
+display() : void
+display(x : int, y : int) : void
background elements
0..*
BackgroundEl()
+dotList : ArrayList(PVector)
+«create» BackgroundEl()
+move() : void
+display() : void
1..*
player object
PlayerEl
+textures : String[] = {"Images/Kawaii/player.png", "Images/Pokemon/player.png" "Images/Minecraft/player.png"}
+dinoPic : PImage = loadImage(textures[textureType])
+pos : PVector
+acc : PVector
+vel : PVector
+radius : float = 80
+jumpHeight : int
+«create» PlayerEl()
+addForce(force : PVector) : void
+gravity() : void
+display() : void
+display(x : int, y : int) : void
0..*
utility functions
Util
+getMax(num1 : double, num2 : double) : double
+getMin(num1 : double, num2 : double) : double
+average(num1 : double, num2 : double) : double

Contributors: Joseph Mei, Timothy Young Period: 1
Group Name: Make a Wish Foundation Co.

APCS - Final Project: Scuffed Dino Game

Project Description:

This game emulates the Google Dino Run game but with extra functions to spice it up. There are the basic elements of a player located on the left side of the screen and obstacles randomly spawning on the right side that move towards the player, which the player jumps over. For our Scuffed Dino Game, however, we included 4 different obstacles for the player to avoid: regular cactuses, long cactuses, trees, and clouds. In addition to jumping, the player can also drop, enabling them to fall down quicker. This in turn allows the user to have more control over their gameplay.

Like the Google Dino Run game, there is a score and highscore feature that is displayed on the top right of the screen, and the obstacles also speed up as the game progresses to make the game more challenging. To make the game more fun and interesting, we added random power-ups that spawn and give buffs and debuffs to the player, such as high-jumps, invulnerability, and drunkenness. We also have 3 different texture packs for the user to select from that will change the skins of the players and the obstacles, which include a Kawaii pack, a Pokemon pack, and a Minecraft pack. There is also a control screen that displays the controls and obstacles for the game.

Functionalities:

- Basic Functions of the Google Dino Run Game:
  - Kinematics:
    - gravity() PVector that constantly applies a downward force to the character.
    - jump() function that is binded to the up arrow.
      - Applies an upward jump force to the player.
      - Follows kinematics laws as the downward gravitational force will cause the player to accelerate downwards as it jumps up and falls down.
      - Only works when the player is grounded. (isGround boolean)
      - Double jump feature is included.
    - fall() function that is binded to the down arrow.
      - Accelerates the fall of the player.
      - Only works when the player is not in the air.
  - Scoring:
    - Score and highscore are displayed on the top right.
    - The score updates as the game progresses (increments by 1 every 10 frames).
    - The high score updates after every death.
  - Game Elements:
    - An ArrayList of background elements (rocks) are cycled through, displayed, and moved as the game progresses → Creates the illusion of movement
    - An ArrayList of obstacle elements are cycled through, displayed, and moved as the game progresses.
      - There are four obstacle elements, each with their own unique size and hitboxes: Cactus, Long Cactus, Tree, and Cloud.
        - Cactus: regular-sized and can be avoided with a single jump.
        - Long Cactus: wider object but can also be avoided with a single jump.
        - Tree: taller object that requires a double jump to avoid.
        - Cloud: hovers above the ground and can be avoided by not jumping.
      - The obstacle elements are spawned in at random time intervals and are spaced out enough for the game to be playable.
      - Each elements' hitbox follows its x- and y-coordinates.

        - The speed of the obstacles and background elements increases as the game progresses. (increments by 0.1 every 50 frames)
        - A player object is static on the left side of the screen.
            - Has its own hitbox that follows its x- and y-coordinates.
        - hit() function that returns true when the hitbox of the obstacle elements coincide with that of the player object → will call the reset() function that resets values and brings the user to the restart screen.

- Display:
    - Start Screen:
        - Allows the user to select texture pack and game mode
            - A player icon will appear over its respective texture.
            - An arrow is used to indicate whether the user is selecting from textures or game modes.
        - Instructs the user to press spacebar for controls menu and enter to start the game
    - Controls Menu: (Spacebar)
        - Pauses the game and brings up instructions.
        - Lists the different power ups featured in the game on the bottom right of the screen.
        - Displays the different obstacle elements on the bottom left corner of the screen
    - Reset Screen: (R key)
        - Displays the number of attempts tried.
        - Displays the score and the previous high score.
            - A message will appear if a new high score is achieved.
        - Allows the user to select the texture pack and game mode.
            - A player icon will appear over its respective texture.
            - An arrow is used to indicate whether the user is selecting from textures or game modes.

- Power-ups:
    - Four different power-ups: Drunk, Invulnerability, Shrink, and High Jump.
    - Drunk:
        - Inverts the jump() and fall() key binds.
        - The screen will be green for the duration of the effect.
    - Invulnerability:
        - Protects the player from oncoming obstacles (unable to be hit).

        - A purple shield/force field will envelope the player for the duration of the effect.
    - Shrink:
        - The player's size and hitbox will decrease for the duration of the effect.
    - High Jump:
        - The player's jump height will drastically increase for the duration of the effect.
    - At the top left corner of the screen, the name of the power-up will be displayed along with a timer of the duration of the effects (5 seconds).
    - Lucky blocks containing the power-ups are spawned randomly and can be picked up when the player hits the block.
- Textures:
    - Three different texture packs that the user can choose from: Kawaii, Pokemon, and Minecraft.
    - Imported images of the player, obstacles, and background elements from the internet to be displayed (each texture pack has a different set of player and obstacle images).
    - The texture packs can only be selected during the start and reset screens
        - A box with an arrow will appear to show the selected texture pack
        - The right and left arrows are used to cycle through the texture packs
        - The up and down arrows are used to switch between selecting textures and selecting game mode.
        - The texture pack cannot be swapped while the game is running.

Disclaimer:

- All required libraries needed to run the game are already within the the repository, so users do not have to download anything after cloning the repository.
- The libraries used in this project are located in the data folder, which includes Papyrus-Regular-48.vlw and ColonnaMT-48.vlw.

Group Work Log:

- **Joseph**:
    - Brainstormed and implemented power-ups
        - Implemented invulnerability power-up with the shield effect
        - Implemented high jump power-up
    - Created the function for the power-ups to last a limited duration
        - Also added the display for the countdown of the power-up's effect and its name.

- Implemented the kinematics function of the player
    - Implemented the gravity PVector to constantly apply a downward force on the player
    - Implemented jumping and double jumping by applying an upward jump force when the up arrow is pressed (under the condition that the player is grounded)
    - Implemented the falling function by increasing the downward force when the down arrow is pressed.
- Compiled and cropped images for the various texture packs
- Created the start/death screen selection tool to select the texture pack and game mode.
- Implemented spawning in random obstacles at random time intervals
- Removed lag by removing obstacle elements and background elements that are out of view of the screen.
- Updated the reset() function to reset game values to default values.
- Added a feature to increment and display the score as the game is running
- Added a way to display and move the images of the player, obstacles, and lucky blocks
- Created a way to pause the game when the controls menu is brought up as well as a way to differentiate keybinds between the start/death screen and the run screen.
- Implemented a way to select the texture pack and change the skin of the obstacles and the player object at a click of a button.
- Created the hit() function that will return true when any part of the player's hitbox coincides with that of an obstacle or power-up.
- Tested the game and debugged errors involved with the start/death screens and the running of the game.

➢ **Timothy**:

- Compiled skeleton of the various classes to streamline the readability and implementation of methods
- Brainstormed and compiled images for the various texture packs and powerUps
- Brainstormed and implemented power-ups
    - Implemented drunk power-up with the green screen effect and shrink with minimized player size.
- Improved visuals for start/death screen (reset function)
    - Created the attempts counter, which is displayed in the death screen
    - Added images to texture selection tool
    - Created text for death screen and improved color scheme of start/death screens

        - Displayed the tracked score and highScore in the death screen
        - Comparing scores to determine a new high score and implemented text notification for a new High Score in the final screen
        - Added new fonts to increase the overall visual aesthetic of the game
    - Implemented the Control screen
        - Displays the keybinds involved in playing the game
        - Images of sprites included in the selected texture pack
    - Creating the background
        - Visual distinction between the ground and sky, coupled with the creation of ground particles to realize the illusion of forward movement
    - Brainstormed and implemented the idea of having obstacles extend from the Obstacle class, and differentiating between obstacle/power up hit functions
    - Tested the game to find and resolve errors
    - Improving the perspicuity of the code by commenting functionalities and removing extraneous elements/methods
    - Added the feature to speed up the game as the game progresses.

Game Manual:

- Objective: Last as long as possible and achieve the highest score by avoiding oncoming obstacles.
    - To avoid obstacles, the player must time their jumps using the up and down arrows to avoid a collision.
        - This includes the ability to fall quicker while pressing down, and the ability to double jump over obstacles by pressing up twice.
    - As time passes, the game speed increases proportionally, so the game progressively gets harder.
    - The run ends if the player hits an object, but users have an infinite amount of tries to beat their previous records.
        - Scores are displayed in the top right corner while playing, and in the end screen.
        - Players will get a “New High Score” notification when they beat their previous record.
    - If the power-ups gamemode is selected, the player can also hit lucky blocks to gain an ability that will either help or hinder them in their run.
        - The duration and type of powerup will be displayed in the upper left corner of the screen.

        - Drunk: The player will be under the influence of alcohol and cannot run properly! Jumping and falling are switched for the duration of this effect.
        - Invulnerability: A purple shield/force field will envelope the player for the duration, protecting the player from oncoming obstacles.
        - Shrink: Someone raided Mario's secret stash! The player's size and hitbox will decrease for the duration of the effect.
        - High Jump: RedBull gives you wings! The player's jump height will drastically increase for the duration of the effect.

- Key Binds:
    - Up Arrow:
        - In Start/Death Screens: Cycle between selecting texture pack and game mode
        - During the Run: Jump (Double press the key to double jump)
    - Down Arrow:
        - In Start/Death Screens: Cycle between selecting texture pack and game mode
        - During the Run: Fall
    - Left + Right Arrow: Cycle between the options for texture pack and game mode
    - Enter Key: Start/Resume the game.
    - R key: Resets the game and brings the user to the death screen.
    - Spacebar: Pauses the game and brings up the controls menu.