

**Controls:**

Weapon selection: Numeric row (1 – 0)

Aiming: Arrow Up / Down

Shooting: Space (Hold to select power, release to shoot)

Classes Overview:GameManager

- (Singleton)
- Controls the games flow via it's GameState implementation.
- Iterates over a list of all updatable objects and executes their update function
- Gives other classes reference to time (since last frame, since beginning of game, ...)
- Has access to the Winform Class. Can manipulate Windows UI elements and read their data.
- Owns all objects, which persist between turns (Tanks, Terrain, Active turn object)

RenderManager

- (Singleton)
- DrawLoop is automatically called when the Winform is invalidated.
- Iterates over a list of all drawable objects
- Calls various drawing functions based on the current GameState.
- Defines font sizes and brush colors during the drawing process.
- Defines a SpriteOrigin enum, used for anchroing drawn sprites at certain local coordinates.
- Implements and gives other classes access to higher draw wrappers (DrawSprite, DrawSpriteReference, DrawString, DrawDebugObject)

InputManager

- (Static Class)
- Gives other classes access to keyboard events
- Allows for simple enum based remapping of keyboard input.

SpriteAtlas

- (Singleton)

- Defines the Sprite and SpriteReference classes.
- Loads all sprite images referenced in the XML sprite atlas, and assigns a SpriteID to them.
- Allows other classes to access loaded sprites by referencing them via their SpriteID

Sprite

- Is initialized with a SpriteID, a path to an image file and defines the size in pixels.
- Loads the image file into the image object.

SpriteReference

- Allows to reference a loaded sprite from another class, without having to pass the entire sprite object to it.
- Implements a PixelOffset and Scale value to change the desired sprites rendering, without having to change the original sprites values.

Weaponry

- (Singleton)
- Loads all available weapon definitions from the weaponry XML file.
- Gives other classes access to weapons via their assigned WeaponID

Weapon

- Has values describing the functionality of the weapon (maxShots, firePerShot, startAngle, scatterAngle, mimMaxAngle).
- Has values about the behaviour of the weapons projectile (projectileType, impactRadius, strength, gravityMultiplier, terrainMultiplier).
- Implements the function FireShot, which performs the weapons shot at a given position, with a certain angle and a certain strength.

Projectile

- Object is created by any weapon upon firing. Constructor receive values from weapon.
- Executes different behaviour based on ProjectileState (Simulating, Impacting, End)
- Simulating: With velocity and gravity a simple trajectory is simulated, saved and displayed as a trail. Every frame a collision check occurs.
- Impacting: A growing explosion is drawn to the screen. On the climax of the explosion, damage is dealt to the players in radius and the terrain is effected.
- The projectiles type decides the behaviour after impacting. Type "Damage" deals dammages and pushes terrain down, while type "Attract" ads velocity to the players in radius and pushes the terrain upwards.

Tank

- Representation of the tanks in game.
- Calculates collision with terrain.
- Adjusts rotation based on angle of terrain.
- Owns a Canon as child object, which can is accessed and manipulated by the active Turn.
- The tank does not implement the shooting behaviour, as that is handled by the Turn object.

Terrain

- Calculates a random terrain for every new game, saves it in form of an Array of Points.
- First a smaller number of points is generated, divisible by 4. These 4-pairs are then used to generate bezier-curves of much denser points.
- These points are then saved into a new array, with one point entry for 3 horizontal pixels.
- Let's other classes inquiry the terrain height for any horizontal position.

Turn

- Counts down the 30 second timer.
- Manages the states of a given players turn (aiming, strength meter, end).
- Handles input for selecting of weapons.
- Handles input for aiming and firing of the weapons.
- Creates / allocates all fired projectiles and checks their status.
- Is replaced with a new turn object, given the next players index, when it's end state is reached.

PlayerStats

- Contains the players information (Name, Color, Health, ...)

IUpdateable

- A simple interface for all classes, which want to be called during the update loop.

IDrawable

- A simple interface for all classes, which want to be called during the draw loop.

ATransformable

- (Abstract Class)
- This class implements the tangible basics for all game objects.
- The values which this class contains are the most elementary for movement (position, rotation, shape, velocity, gravity).
- The "Simulate" function this class implements performs the movement for this transformable, given it's position, velocity and gravity. This is done respectfully to execution delta time, resulting in equal motion regardless of CPU-Speed or framerate.
- The parent and child principle, which is standard in computer graphics and game engines, is partially implemented here. Each ATransformable object has a list of children of the same class, which will move and rotate according to their relation to the parent ATransformables position.

AObject

- (Abstract Class)
- The basis for all "physical" objects in the game.
- Inherits from ATransformable and implements the IUpdateable and IDrawable interface.
- Besitzt eine 32-Bit GUID, eine Sprite Referenz und einen SpriteOrigin, sowie alle Werte von Atransformable
- Owns a 32-Bit GUID for identification, a sprite reference and a sprite origin, ontop of all values and function inherited from ATransformable.
- The "Update" and "Draw" functions of all classes which inherit from AObject are automatically added to the "updateable" and "drawable" lists of the game manager, within their constructor.
- Likewise, the "Update" and "Draw" entries are removed upon the classes destruction. This can also be done prematurely, via the public wrapper functions facilitating adding / removing from the lists.

Used Nuget Packages:

- EPP.PresentationCore-1.0.0
- WindowsBase-4.6