

LOW POLY

# epic pack

by [polyperfect](#)



*Have a Suggestion?*

[info@polyperfect.com](mailto:info@polyperfect.com)

# Thanks!

First of all, thank you for purchasing our pack, we really appreciate that! We are putting a lot of effort into this.

We are also planning to expand the list of the characters and their animations in the future with free updates of the pack. Check out our [Facebook Page](#) for any news.

# Handy Links ;)

## Other Low Poly Packs

[Low Poly Animated Animals](#)

[Low Poly Animated People](#)

[Low Poly Animated Dinosaurs](#)

[Low Poly Animated Prehistoric Animals](#)

[Low Poly Epic City](#)

[Low Poly Ultimate Pack](#)

[Low Poly War Pack](#)

## 2D Packs

[Low Poly Icon Pack](#)

[Fancy Icon Pack](#)

## Follow us

[Facebook](#)

[Instagram](#)

[Youtube](#)

[Polyperfect.com](#)

# Updates

**VERSION 1.0**

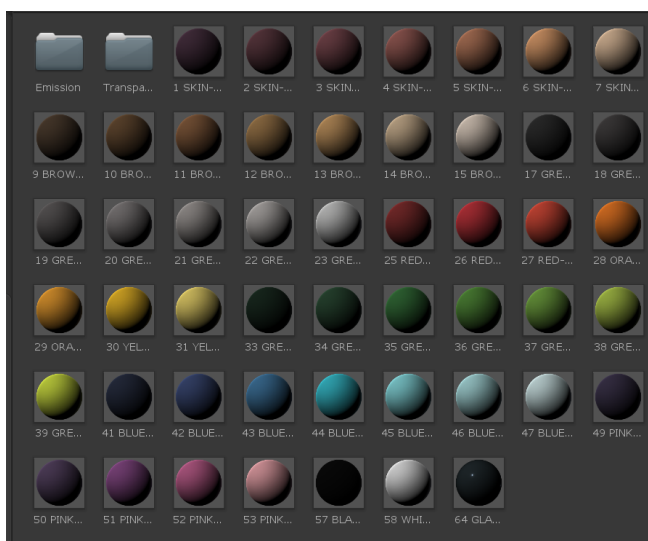
INITIAL RELEASE

# Material vs Textures

There are two versions of our models. Feel free to use ones that fit your workflow the best. Demo Scenes are made with material prefabs.

## “M” Models (Material)

- The color of an object is influenced by texture
- All prefabs share one atlas texture
- Good for speed and mobile



## “T” Models (Texture)

- The color of an object is influenced by texture
- All prefabs share one atlas texture
- Good for speed and mobile

***“This texture is not perfect, but it's ours.” We are using one big atlas texture and one material for all our meshes. It's great for speed - or so they say :))***



Left column is reserved for transparent colors.

							TRANSPARENT VALUES
1 SKIN DARKEST	2 SKIN DARK	3 SKIN MEDIUM	4 SKIN	5 SKIN LIGHT	6 SKIN LIGHTEST	7 SKIN WHITE	8 TRANSPARENT RED
9 BROWN DARKEST	10 BROWN DARK	11 BROWN MEDIUM	12 BROWN	13 BROWN LIGHT	14 BROWN LIGHTEST	15 BROWN WHITE	16 TRANSPARENT BROWN
17 GREY DARKEST	18 GREY DARK	19 GREY MEDIUM	20 GREY	21 GREY LIGHT	22 GREY LIGHTEST	23 GREY WHITE	24 TRANSPARENT GREY
25 RED DARKEST	26 RED DARK	27 RED MEDIUM	28 ORANGE	29 ORANGE LIGHT	30 YELLOW	31 YELLOW LIGHT	32 TRANSPARENT RED
33 GREEN DARKEST	34 GREEN DARK	35 GREEN MEDIUM	36 GREEN	37 GREEN LIGHT	38 GREEN LIGHTEST	39 GREEN WHITE	40 TRANSPARENT GREEN
41 BLUE DARKEST	42 BLUE DARK	43 BLUE MEDIUM	44 BLUE	45 BLUE LIGHT	46 BLUE LIGHTEST	47 BLUE WHITE	48 TRANSPARENT BLUE
49 PINK DARKEST	50 PINK DARK	51 PINK MEDIUM	52 PINK	53 PINK LIGHT	54 EMISSION BLUE	55 EMISSION PURPLE	56 TRANSPARENT WHITE
57 BLACK	58 WHITE	59 EMISSION WHITE	60 EMISSION RED	61 EMISSION ORANGE	62 EMISSION YELLOW	63 EMISSION GREEN	64 GLASS

1 SKIN DARKEST	2 SKIN DARK	3 SKIN MEDIUM	4 SKIN	5 SKIN LIGHT	6 SKIN LIGHTEST	7 SKIN WHITE	8 TRANSPARENT RED
9 BROWN DARKEST	10 BROWN DARK	11 BROWN MEDIUM	12 BROWN	13 BROWN LIGHT	14 BROWN LIGHTEST	15 BROWN WHITE	16 TRANSPARENT BROWN
17 GREY DARKEST	18 GREY DARK	19 GREY MEDIUM	20 GREY	21 GREY LIGHT	22 GREY LIGHTEST	23 GREY WHITE	24 TRANSPARENT GREY
25 RED DARKEST	26 RED DARK	27 RED MEDIUM	28 ORANGE	29 ORANGE LIGHT	30 YELLOW	31 YELLOW LIGHT	32 TRANSPARENT RED
33 GREEN DARKEST	34 GREEN DARK	35 GREEN MEDIUM	36 GREEN	37 GREEN LIGHT	38 GREEN LIGHTEST	39 GREEN WHITE	40 TRANSPARENT GREEN
41 BLUE DARKEST	42 BLUE DARK	43 BLUE MEDIUM	44 BLUE	45 BLUE LIGHT	46 BLUE LIGHTEST	47 BLUE WHITE	48 TRANSPARENT BLUE
49 PINK DARKEST	50 PINK DARK	51 PINK MEDIUM	52 PINK	53 PINK LIGHT	54 EMISSION BLUE	55 EMISSION PURPLE	56 TRANSPARENT WHITE
57 BLACK	58 WHITE	59 EMISSION WHITE	60 EMISSION RED	61 EMISSION ORANGE	62 EMISSION YELLOW	63 EMISSION GREEN	64 GLASS

**EMISSION  
VALUES**

Bottom part for emission ones.



# Path Finding

Basic implementation of the A\* algorithm with the use of a sorted queue.

This script is required for path finding for trains, cars and humans.

Heuristic function is implemented as the Manhattan distance.

Pathfinding tree is created with nodes of type [path](#).

# Path

This script is used as a basic node in the pathfinding tree.

## Path type

Determines for what path can be used (Road - Road vehicles, Rail - Trains, Sidewalk - Humans).

## Speed

The maximum speed vehicle can move on this path. On sidewalks this variable doesn't matter.

## Next Paths

Holds all paths that are immediately connected to the end of this path. When part of the tile is set up automatically by Tile script.

## Path Checkpoints

Ordered list of path checkpoints through which path goes.

When the Edit toggle is on, the selected checkpoint can be edited by the Position handle in the scene or set to the exact local position above this list. Checkpoints can be rearranged by dragging.

New checkpoints can be added by + button on bottom right.

Currently selected checkpoint can be deleted by clicking on - button next to + button.

# Trains

Locomotives and all wagons have their tags set to “Train”.

## Wagons

Trains can have from zero up to 64 wagons.

Each wagon gameobject must have attached:

- Wagon script with reference to box collider.
- Box collider - must be set to match the length of the mesh.

## Max speed

Determines the maximum speed ( in km/h) a locomotive can achieve.

## Acceleration

Acceleration of the train.

## Checkpoints

Checkpoints are world positions through which the [pathfinder](#) tries to find a path for the train.

At the start the train tries to find a path to the first checkpoint.

When the train reaches the end it goes back through the same checkpoints.

# Road Vehicles

Each vehicle has pathfinding and CarBehavior script as well as Rigidbody component set to kinematic and gravity off. Of course the box collider and box collider are set to trigger in front of the vehicle to detect vehicles/traffic lights/crosswalk in front. Tag of the gameobject that script and collider is attached to must be set to Car.

## Random destination

When turned on finds a path to a random accessible tile from the current position.  
When turned off finds the path through setup checkpoints.

## Closed Circuit

Only visible if Random destination is off.  
When true, the path created from checkpoints is sharing start and end.  
When false, the path goes through checkpoints. When the vehicle reaches the end it goes back through the same checkpoints.

## Max speed

Max speed of this vehicle.

## Checkpoints

Only visible if Random destination is off.  
Checkpoints are world positions through which the [pathfinder](#) tries to find a path for the vehicle.  
At the start it tries to find a path to the first checkpoint.

# **Ships**

Ships don't follow paths created by pathfinder rather follow predetermined paths.

## **Trajectory**

Path that ship will follow. When it reaches end of path

## **Max speed**

Determines the maximum speed ( in km/h) a ship can achieve.

## **Acceleration**

Acceleration of the ship.

## **Ship Tipping**

Maximum angle in degrees that ship can tipp.

# Planes

Planes same as ships don't follow paths created by pathfinder rather follow predetermined paths.

## **Trajectory**

Path that plane will follow. When it reaches the end of the path it continues from the start.

## **Max speed**

Determines the maximum speed ( in km/h) a plane can achieve.

## **Acceleration**

Acceleration of the plane.

# Tiles

Each drivable/walkable tile must have Tile script attached to it.

At the awake function it gets all of its connected neighbors tiles. Gap between tiles to be connected must be less than 0,5m. Next it searches through all child objects for Path script and adds them to the list. Then the Start function connects all paths saved in the list there's next paths.

## Tile Shape

Shape of tile paths.

## Vertical Tile

Determines if tile is tunnel/ hill(ramp) / bridge or basic plain.

## Tile Type

What path types are on this tile. Road includes pathwalks.

# Level Crossing

**LevelCrossingController** script requires a box collider at the gameobject and needs to be set as trigger as well as rigidbody set to kinematic. When a train enters the collider all road vehicles will stop if they enter. If they are already on the level crossing they will continue driving.

## Level Crossings

List of individual LevelCrossing scripts.

The LevelCrossing script controls individual barriers and lights.

# Crosswalks

Crosswalk script requires a box collider at the gameobject and needs to be set as trigger. When a human steps into a collider all road vehicles will stop if they enter. If they are already on the crosswalk they will continue driving.



# Traffic Lights

## Traffic Light Control

This script's job is switching between two sets of traffic lights in every preset seconds.

### First Lights

First set of traffic lights which will be green at the same time.

### Second Lights

Second set of traffic lights which will be green at the same time.

### Time Interval

Time in second that traffic lights switch.

## Traffic Light script

This script requires a box collider at the gameobject and needs to be set as trigger.

This trigger job is to stop vehicles at red lights. Gameobjects tag needs to be set to "TrafficLight".

### Crosswalk

Reference to crosswalk script which will stop humans from crossing when they can't.

Gameobject that the crosswalk script is attached to needs its tag set to "TrafficLightCrosswalk".

