In [50]:
```python
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader
from torchvision import datasets, transforms
import matplotlib.pyplot as plt
import numpy as np

latent_dim = 100
img_shape = (1, 28, 28)
lambda_gp = 10
n_critic = 5
batch_size = 64
lr = 1e-4
n_epochs = 200
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

In [51]:
```python
class Generator(nn.Module):

    def __init__(self):
        super(Generator, self).__init__()

        self.model = nn.Sequential(
            nn.Linear(latent_dim, 256),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(256, 512),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(512, 1024),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(1024, int(torch.prod(torch.tensor(img_shape)))),
            nn.Sigmoid()
        )

    def forward(self, z):
        img = self.model(z)
        img = img.view(img.size(0), *img_shape)
        return img
```

In [52]:
```python
class Discriminator(nn.Module):
    def __init__(self):

        super().__init__()

        self.model = nn.Sequential(
            nn.Flatten(),
            nn.Linear(784, 512),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(512, 256),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(256, 1)
        )

    def forward(self, img):
        img_flat = img.view(img.size(0), -1)
        validity = self.model(img_flat)
        return validity
```

In [53]:
```
def compute_gradient_penalty(discriminator, real_samples, fake_samples):
    alpha = torch.rand(real_samples.size(0), 1, 1, 1).to(device)
    interpolates = (alpha * real_samples + (1 - alpha) * fake_samples).re

    d_interpolates = discriminator(interpolates)
    fake = torch.ones(d_interpolates.size()).to(device)

    gradients = torch.autograd.grad(
        outputs = d_interpolates,
        inputs = interpolates,
        grad_outputs = fake,
        create_graph = True,
    )[0]

    gradients = gradients.view(gradients.size(0), -1)
    gradient_penalty = lambda_gp * ((gradients.norm(2, dim=1) - 1) ** 2).
    return gradient_penalty
```

In [54]:
```
generator = Generator().to(device)
discriminator = Discriminator().to(device)

#generator.load_state_dict(torch.load('generator_model.pth'))
#discriminator.load_state_dict(torch.load('discriminator_model.pth'))

print("Generator Architecture:\n", generator)
print("Discriminator Architecture:\n", discriminator)
```

```
Generator Architecture:
 Generator(
  (model): Sequential(
    (0): Linear(in_features=100, out_features=256, bias=True)
    (1): LeakyReLU(negative_slope=0.2, inplace=True)
    (2): Linear(in_features=256, out_features=512, bias=True)
    (3): LeakyReLU(negative_slope=0.2, inplace=True)
    (4): Linear(in_features=512, out_features=1024, bias=True)
    (5): LeakyReLU(negative_slope=0.2, inplace=True)
    (6): Linear(in_features=1024, out_features=784, bias=True)
    (7): Sigmoid()
  )
)
Discriminator Architecture:
 Discriminator(
  (model): Sequential(
    (0): Flatten(start_dim=1, end_dim=-1)
    (1): Linear(in_features=784, out_features=512, bias=True)
    (2): LeakyReLU(negative_slope=0.2, inplace=True)
    (3): Linear(in_features=512, out_features=256, bias=True)
    (4): LeakyReLU(negative_slope=0.2, inplace=True)
    (5): Linear(in_features=256, out_features=1, bias=True)
  )
)
```

In [55]:
```
optimizer_G = optim.Adam(generator.parameters(), lr=lr)
optimizer_D = optim.Adam(discriminator.parameters(), lr=lr)
```

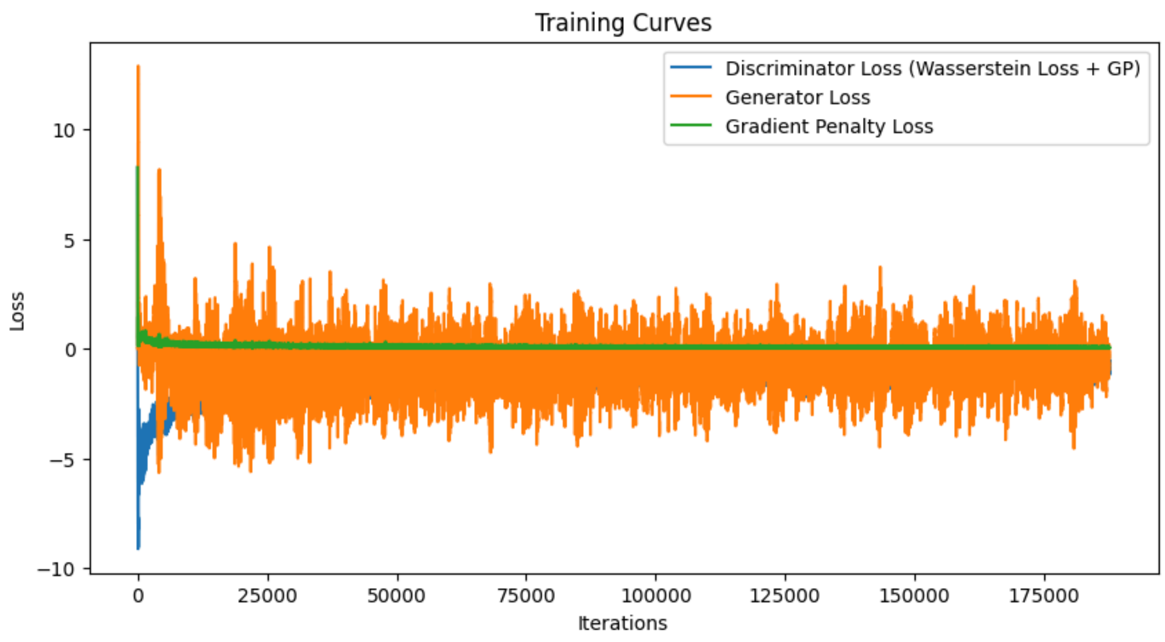
In [56]:
```
transform = transforms.Compose([
    transforms.ToTensor()
])
```

```
dataloader = DataLoader(
   datasets.FashionMNIST('./data', train=True, download=True, transform=t
    batch_size=batch_size,
    shuffle=True
)
```

In [ ]:

In [57]:
```
d_losses = []
g_losses = []
gp_losses = []

for epoch in range(n_epochs):
    for i, (imgs, _) in enumerate(dataloader):
        real_imgs = imgs.to(device)

        optimizer_D.zero_grad()

        z = torch.randn(imgs.size(0), latent_dim).to(device)
        fake_imgs = generator(z).detach()

        real_validity = discriminator(real_imgs)
        fake_validity = discriminator(fake_imgs)
        gradient_penalty = compute_gradient_penalty(discriminator, real_i
        d_loss = fake_validity.mean() - real_validity.mean() + gradient_p

        d_loss.backward()
        optimizer_D.step()

        if i % n_critic == 0:
            optimizer_G.zero_grad()

            z = torch.randn(imgs.size(0), latent_dim).to(device)
            gen_imgs = generator(z)
            g_loss = -discriminator(gen_imgs).mean()

            g_loss.backward()
            optimizer_G.step()

        d_losses.append(d_loss.item())
        g_losses.append(g_loss.item())
        gp_losses.append(gradient_penalty.item())

    print(f"Epoch [{epoch}/{n_epochs}] D loss: {d_loss.item()} | G loss:

plt.figure(figsize=(10, 5))
plt.plot(d_losses, label="Discriminator Loss (Wasserstein Loss + GP)")
plt.plot(g_losses, label="Generator Loss")
plt.plot(gp_losses, label="Gradient Penalty Loss")
plt.xlabel("Iterations")
plt.ylabel("Loss")
plt.title("Training Curves")
plt.legend()
plt.show()
```

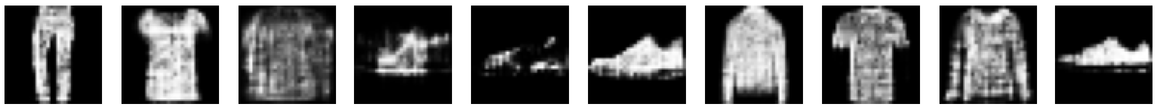
```
Epoch [0/200] D loss: -4.784727573394775 | G loss: 0.169526606798172
Epoch [1/200] D loss: -4.198870658874512 | G loss: -0.4279528260231018
Epoch [2/200] D loss: -3.8054778575897217 | G loss: -0.26286494731903076
Epoch [3/200] D loss: -3.0426764488220215 | G loss: -0.07763124257326126
Epoch [4/200] D loss: -2.7004265785217285 | G loss: -1.010871410369873
Epoch [5/200] D loss: -3.0623655319213867 | G loss: 0.5291200280189514
Epoch [6/200] D loss: -2.541698455810547 | G loss: 0.06524395942687988
Epoch [7/200] D loss: -2.2299346923828125 | G loss: -2.2297749519348145
Epoch [8/200] D loss: -2.4366817474365234 | G loss: -1.5529004335403442
Epoch [9/200] D loss: -2.21347975730896 | G loss: -0.1980552077293396
Epoch [10/200] D loss: -2.0448317527770996 | G loss: -2.7403006553649902
Epoch [11/200] D loss: -2.4805285930633545 | G loss: 0.019109107553958893
Epoch [12/200] D loss: -2.9548466205596924 | G loss: -1.8844891786575317
Epoch [13/200] D loss: -2.543830633163452 | G loss: -2.6593432426452637
Epoch [14/200] D loss: -2.127803087234497 | G loss: 0.045767754316329956
Epoch [15/200] D loss: -2.217164993286133 | G loss: -2.992729663848877
Epoch [16/200] D loss: -1.5044091939926147 | G loss: -2.6486356258392334
Epoch [17/200] D loss: -1.5105818510055542 | G loss: -1.5889580249786377
Epoch [18/200] D loss: -1.7259958982467651 | G loss: -0.5856677889823914
Epoch [19/200] D loss: -2.2793023586273193 | G loss: -2.086089611053467
Epoch [20/200] D loss: -2.3190035820007324 | G loss: -3.283159017562866
Epoch [21/200] D loss: -1.9477940797805786 | G loss: -1.078348159790039
Epoch [22/200] D loss: -1.6496578454971313 | G loss: -0.249592587351799
Epoch [23/200] D loss: -1.807557463645935 | G loss: -0.9590688943862915
Epoch [24/200] D loss: -1.8057098388671875 | G loss: -3.401531219482422
Epoch [25/200] D loss: -1.46560800075531 | G loss: -0.8102638125419617
Epoch [26/200] D loss: -1.8365799188613892 | G loss: -1.98586106300354
Epoch [27/200] D loss: -1.5203096866607666 | G loss: -2.7419533729553223
Epoch [28/200] D loss: -1.6732650995254517 | G loss: -1.7503268718719482
Epoch [29/200] D loss: -1.4505404233932495 | G loss: -1.4549610614776611
Epoch [30/200] D loss: -1.7219338417053223 | G loss: -1.4841046333312988
Epoch [31/200] D loss: -1.7025806903839111 | G loss: -1.1520391702651978
Epoch [32/200] D loss: -1.3481569290161133 | G loss: -3.1752326488494873
Epoch [33/200] D loss: -1.6308703422546387 | G loss: -2.167686700820923
Epoch [34/200] D loss: -1.6071159839630127 | G loss: -0.8672932982444763
Epoch [35/200] D loss: -1.3308722972869873 | G loss: -2.3233559131622314
Epoch [36/200] D loss: -1.5822030305862427 | G loss: 0.7071529030799866
Epoch [37/200] D loss: -1.3580435514450073 | G loss: -1.689042568206787
Epoch [38/200] D loss: -1.0146408081054688 | G loss: -1.0486106872558594
Epoch [39/200] D loss: -1.8164702653884888 | G loss: 0.33964934945106506
Epoch [40/200] D loss: -1.7001185417175293 | G loss: -1.4428422451019287
Epoch [41/200] D loss: -1.361594796180725 | G loss: -1.9842363595962524
Epoch [42/200] D loss: -1.3549630641937256 | G loss: 0.4150984287261963
Epoch [43/200] D loss: -1.5166816711425781 | G loss: -2.276649236679077
Epoch [44/200] D loss: -1.6139602661132812 | G loss: -1.0658272504806519
Epoch [45/200] D loss: -1.6700702905654907 | G loss: -1.7393922805786133
Epoch [46/200] D loss: -1.3482739925384521 | G loss: -1.0287418365478516
Epoch [47/200] D loss: -1.2738527059555054 | G loss: -0.24374142289161682
Epoch [48/200] D loss: -1.883911371231079 | G loss: -2.266486167907715
Epoch [49/200] D loss: -1.4744877815246582 | G loss: -1.6296002864837646
Epoch [50/200] D loss: -1.3556045293807983 | G loss: -1.2702229022979736
Epoch [51/200] D loss: -0.8604758381843567 | G loss: 0.011688370257616043
Epoch [52/200] D loss: -1.436245322227478 | G loss: -2.253401279449463
Epoch [53/200] D loss: -1.4029138088226318 | G loss: -1.1343352794647217
Epoch [54/200] D loss: -1.4563809633255005 | G loss: 0.6730366945266724
Epoch [55/200] D loss: -1.3145806789398193 | G loss: 0.7450667023658752
Epoch [56/200] D loss: -1.35148024559021 | G loss: -1.211404800415039
Epoch [57/200] D loss: -1.3411881923675537 | G loss: -0.43886396288871765
Epoch [58/200] D loss: -1.0903950929641724 | G loss: -0.038929328322410583
Epoch [59/200] D loss: -0.9837324619293213 | G loss: -1.8245686292648315
Epoch [60/200] D loss: -0.40486061573028564 | G loss: -0.922829270362854
Epoch [61/200] D loss: -1.1692287921905518 | G loss: -1.0407068729400635
Epoch [62/200] D loss: -0.9088230729103088 | G loss: -1.0525753498077393
Epoch [63/200] D loss: -1.0458167791366577 | G loss: -0.06852507591247559
Epoch [64/200] D loss: -1.8309764862060547 | G loss: -2.627078056335449
Epoch [65/200] D loss: -0.7666274905204773 | G loss: -0.4337635338306427
Epoch [66/200] D loss: -1.0362086296081543 | G loss: 1.3452236652374268
Epoch [67/200] D loss: -1.1621387004852295 | G loss: -1.284398078918457
Epoch [68/200] D loss: -0.8498374223709106 | G loss: -1.8641475439071655
Epoch [69/200] D loss: -1.2109935283660889 | G loss: -0.8734230995178223
Epoch [70/200] D loss: -1.093865156173706 | G loss: -1.4405303001403809
Epoch [71/200] D loss: -1.0904138088226318 | G loss: -1.6389410495758057
Epoch [72/200] D loss: -1.275428056716919 | G loss: -0.5392375588417053
Epoch [73/200] D loss: -0.7897504568099976 | G loss: -0.11760638654232025
Epoch [74/200] D loss: -1.7635471820831299 | G loss: -0.278306782245636
Epoch [75/200] D loss: -1.0750583410263062 | G loss: -1.9015949964523315
Epoch [76/200] D loss: -1.002659797668457 | G loss: -1.8813329935073853
Epoch [77/200] D loss: -1.0767686367034912 | G loss: -1.95384681224823
Epoch [78/200] D loss: -1.3315331935882568 | G loss: -0.5057784914970398
Epoch [79/200] D loss: -0.8890280723571777 | G loss: 0.43942931294441223
Epoch [80/200] D loss: -1.5978734493255615 | G loss: 0.3259936571121216
Epoch [81/200] D loss: -1.2014572620391846 | G loss: -0.23073020577430725
Epoch [82/200] D loss: -1.0992411375045776 | G loss: -0.39433014392852783
Epoch [83/200] D loss: -0.8613370060920715 | G loss: -0.8600864410400391
Epoch [84/200] D loss: -1.6886788606643677 | G loss: -1.5181715488433838
Epoch [85/200] D loss: -1.208779215812683 | G loss: -1.3257280588150024
Epoch [86/200] D loss: -0.8798142671585083 | G loss: 0.364468514919281
Epoch [87/200] D loss: -1.3573882579803467 | G loss: -0.49752140045166016
Epoch [88/200] D loss: -1.4828734397888184 | G loss: -0.07625608891248703
Epoch [89/200] D loss: -1.8724108934402466 | G loss: -3.4049296379089355
Epoch [90/200] D loss: -0.7684583067893982 | G loss: 1.2735239267349243
Epoch [91/200] D loss: -0.9328217506408691 | G loss: -3.105137348175049
Epoch [92/200] D loss: -1.0116814374923706 | G loss: 0.9653522968292236
Epoch [93/200] D loss: -0.9905540347099304 | G loss: -0.03587200120091438
Epoch [94/200] D loss: -1.1753054857254028 | G loss: 0.10753403604030609
Epoch [95/200] D loss: -0.8875170946121216 | G loss: -2.3752031326293945
Epoch [96/200] D loss: -1.3512852191925049 | G loss: -0.05301424860954285
Epoch [97/200] D loss: -1.1102931499481201 | G loss: -1.8176383972167969
Epoch [98/200] D loss: -1.2619935274124146 | G loss: -0.6395589709281921
Epoch [99/200] D loss: -1.1186456680297852 | G loss: 0.5595718026161194
Epoch [100/200] D loss: -1.3285022974014282 | G loss: -1.295693039894104
Epoch [101/200] D loss: -0.6848299503326416 | G loss: -2.619511604309082
Epoch [102/200] D loss: -1.3818211555480957 | G loss: -0.018027808517217636
Epoch [103/200] D loss: -1.7697871923446655 | G loss: -0.02547229453921318
Epoch [104/200] D loss: -1.1044143438339233 | G loss: -1.1191753149032593
Epoch [105/200] D loss: -1.2921708822250366 | G loss: -0.20564165711402893
Epoch [106/200] D loss: -1.2111916542053223 | G loss: 0.06718191504478455
Epoch [107/200] D loss: -0.7805904150009155 | G loss: -1.0282882452011108
Epoch [108/200] D loss: -1.1173020601272583 | G loss: -1.949987530708313
Epoch [109/200] D loss: -0.9280787110328674 | G loss: -0.6859194040298462
Epoch [110/200] D loss: -0.5748558044433594 | G loss: -1.4872229099273682
Epoch [111/200] D loss: -0.7710491418838501 | G loss: -0.9147748947143555
Epoch [112/200] D loss: -0.9022006988525391 | G loss: -0.2356027513742447
Epoch [113/200] D loss: -0.7617656588554382 | G loss: -0.9996129274368286
Epoch [114/200] D loss: -0.961768627166748 | G loss: 0.6315566301345825
Epoch [115/200] D loss: -1.7094154357910156 | G loss: 0.11028887331485748
Epoch [116/200] D loss: -1.4102730751037598 | G loss: -1.2652016878128052
Epoch [117/200] D loss: -0.6177038550376892 | G loss: 0.8869935870170593
Epoch [118/200] D loss: -0.7268720269203186 | G loss: -1.4359148740768433
Epoch [119/200] D loss: -0.9853872656822205 | G loss: 0.5032544136047363
Epoch [120/200] D loss: -1.183333158493042 | G loss: 0.9076262712478638
Epoch [121/200] D loss: -0.6857655644416809 | G loss: -1.9075452089309692
Epoch [122/200] D loss: -1.2365915775299072 | G loss: 1.0210230350494385
Epoch [123/200] D loss: -0.8192325830459595 | G loss: -1.5182098150253296
Epoch [124/200] D loss: -1.0433673858642578 | G loss: -0.1290113925933838
Epoch [125/200] D loss: -0.862108588218689 | G loss: -2.0501906871795654
Epoch [126/200] D loss: -1.5771197080612183 | G loss: -2.2029800415039062
Epoch [127/200] D loss: -0.9113981127738953 | G loss: 0.2729465067386627
Epoch [128/200] D loss: -1.2888026237487793 | G loss: -1.326904058456421
Epoch [129/200] D loss: -1.3921443223953247 | G loss: 0.8829160928726196
Epoch [130/200] D loss: -1.0612295866012573 | G loss: 0.2706542909145355
Epoch [131/200] D loss: -0.8608335852622986 | G loss: -2.8234810829162598
Epoch [132/200] D loss: -1.0218472480773926 | G loss: 1.1521446704864502
Epoch [133/200] D loss: -0.7213597893714905 | G loss: 0.10528059303760529
Epoch [134/200] D loss: -0.7839843034744263 | G loss: -0.643567681312561
Epoch [135/200] D loss: -1.0342085361480713 | G loss: -1.163661003112793
Epoch [136/200] D loss: -1.3011951446533203 | G loss: 0.5092849731445312
Epoch [137/200] D loss: -0.9967387914657593 | G loss: -0.4304407238960266
Epoch [138/200] D loss: -1.1752887964248657 | G loss: 0.20007184147834778
Epoch [139/200] D loss: -0.9749467968940735 | G loss: -0.1665002554655075
Epoch [140/200] D loss: -0.8405683040618896 | G loss: -0.4821426272392273
Epoch [141/200] D loss: -1.139533519744873 | G loss: -0.18443486094474792
Epoch [142/200] D loss: -1.021334171295166 | G loss: -1.6884812116622925
Epoch [143/200] D loss: -1.0299252271652222 | G loss: 0.37690651416778564
Epoch [144/200] D loss: -1.0902111530303955 | G loss: -1.1950993537902832
Epoch [145/200] D loss: -0.9571722745895386 | G loss: -1.470567226409912
Epoch [146/200] D loss: -0.6771907210350037 | G loss: -1.8345415592193604
Epoch [147/200] D loss: -0.9570595026016235 | G loss: -0.4502165615558624
Epoch [148/200] D loss: -1.3931018114089966 | G loss: 1.6353775262832642
Epoch [149/200] D loss: -1.5264229774475098 | G loss: 1.7395942211151123
Epoch [150/200] D loss: -0.5074580311775208 | G loss: 0.27866291999816895
Epoch [151/200] D loss: -0.8777796626091003 | G loss: -0.1421634554862976
Epoch [152/200] D loss: -0.33353906869888306 | G loss: -2.019481897354126
Epoch [153/200] D loss: -1.2170199155807495 | G loss: -1.2944790124893188
Epoch [154/200] D loss: -0.7874226570129395 | G loss: -0.21257376670837402
Epoch [155/200] D loss: -0.42377969622612 | G loss: -2.8556714057922363
Epoch [156/200] D loss: -0.761292576789856 | G loss: -2.4128057956695557
Epoch [157/200] D loss: -1.0190764665603638 | G loss: -1.3903846740722656
Epoch [158/200] D loss: -0.7684799432754517 | G loss: -2.2051429748535156
Epoch [159/200] D loss: -0.7095032930374146 | G loss: -1.2796870470046997
Epoch [160/200] D loss: -0.976200520992279 | G loss: -0.11886359006166458
Epoch [161/200] D loss: -0.5933735966682434 | G loss: -1.3733537197113037
Epoch [162/200] D loss: -0.9767701029777527 | G loss: 0.025883685797452927
Epoch [163/200] D loss: -1.3172636032104492 | G loss: 0.1722470372915268
Epoch [164/200] D loss: -1.1843730211257935 | G loss: -0.5023186802864075
Epoch [165/200] D loss: -1.0267215967178345 | G loss: -1.0419063568115234
Epoch [166/200] D loss: -0.774192750453949 | G loss: 0.434415340423584
Epoch [167/200] D loss: -0.699401319026947 | G loss: -0.018332980573177338
Epoch [168/200] D loss: -0.8626505136489868 | G loss: 1.2914592027664185
Epoch [169/200] D loss: -0.8857458829879761 | G loss: 1.2062184810638428
Epoch [170/200] D loss: -1.0627950429916382 | G loss: 0.15468022227287292
Epoch [171/200] D loss: -0.509903609752655 | G loss: -0.7390735149383545
Epoch [172/200] D loss: -1.1068719625473022 | G loss: -1.14012610912323
Epoch [173/200] D loss: -1.0125969648361206 | G loss: -1.7228707075119019
Epoch [174/200] D loss: -0.844176173210144 | G loss: -0.7609506845474243
Epoch [175/200] D loss: -0.9819736480712891 | G loss: -1.5042914152145386
Epoch [176/200] D loss: -0.6360827684402466 | G loss: -0.3894903361797333
Epoch [177/200] D loss: -1.0542014837265015 | G loss: -1.276002049446106
Epoch [178/200] D loss: -1.5790331363677979 | G loss: -3.5067930221557617
Epoch [179/200] D loss: -0.5022937059402466 | G loss: -1.7712318897247314
Epoch [180/200] D loss: -0.3048878014087677 | G loss: 0.918065071105957
Epoch [181/200] D loss: -0.3648998737335205 | G loss: -0.06010467931628227
Epoch [182/200] D loss: -0.5496434569358826 | G loss: 0.050672534853219986
Epoch [183/200] D loss: -1.2488219738006592 | G loss: 0.2927255630493164
Epoch [184/200] D loss: -0.6463768482208252 | G loss: -0.970115602016449
Epoch [185/200] D loss: -0.8898677825927734 | G loss: -0.4845334589481354
Epoch [186/200] D loss: -0.659423828125 | G loss: 1.0711078643798828
Epoch [187/200] D loss: -1.205003023147583 | G loss: 0.6894303560256958
Epoch [188/200] D loss: -0.987583577632904 | G loss: -1.8946020603179932
Epoch [189/200] D loss: -0.5555692911148071 | G loss: -2.2378149032592773
Epoch [190/200] D loss: -0.6855987906455994 | G loss: -0.9001930356025696
Epoch [191/200] D loss: -0.964364767074585 | G loss: -2.028291702270508
Epoch [192/200] D loss: -1.2059669494628906 | G loss: -1.684090495109558
Epoch [193/200] D loss: -0.7392978668212891 | G loss: -0.316741943359375
Epoch [194/200] D loss: -0.9218664169311523 | G loss: 0.5124510526657104
Epoch [195/200] D loss: -0.8069612979888916 | G loss: 0.05819544568657875
Epoch [196/200] D loss: -1.0902960300445557 | G loss: -0.9484305381774902
Epoch [197/200] D loss: -0.3737024664878845 | G loss: -0.3585740625858307
Epoch [198/200] D loss: -0.8378769755363464 | G loss: -1.011535882949829
Epoch [199/200] D loss: -1.1609306335449219 | G loss: -0.6644271016120911
```



In [58]:
```python
def show_generated_images(generator, n_images=10):
    z = torch.randn(n_images, latent_dim).to(device)
    gen_imgs = generator(z).detach().cpu().numpy()
    gen_imgs = 0.5 * gen_imgs + 0.5

    fig, axes = plt.subplots(1, n_images, figsize=(n_images, 1))
    for i in range(n_images):
        axes[i].imshow(np.squeeze(gen_imgs[i]), cmap='gray')
        axes[i].axis('off')
    plt.show()
```

In [59]:
```python
def show_generated_images(generator, n_images=10):
    z = torch.randn(n_images, latent_dim).to(device)  # 生成隨機 z
    gen_imgs = generator(z).detach().cpu().numpy()  # 生成圖像，轉為 numpy
    gen_imgs = 0.5 * gen_imgs + 0.5  # 將圖像從 [-1,1] 轉回 [0,1] 範圍

    fig, axes = plt.subplots(1, n_images, figsize=(n_images, 1))  # 設置圖
    for i in range(n_images):
        axes[i].imshow(np.squeeze(gen_imgs[i]), cmap='gray')  # 顯示每張生
        axes[i].axis('off')  # 隱藏坐標軸
    plt.show()
```

In [60]:
```
show_generated_images(generator)

generator_save_path = "./generator_model.pth"
discriminator_save_path = "./discriminator_model.pth"

torch.save(generator.state_dict(), generator_save_path)
torch.save(discriminator.state_dict(), discriminator_save_path)
```



In [61]:
```
z = torch.randn(100, latent_dim).to(device)
gen_imgs = generator(z).detach().cpu()

fig, axs = plt.subplots(10, 10, figsize=(10, 10))
for i in range(10):
    for j in range(10):
        axs[i, j].imshow(gen_imgs[i * 10 + j].squeeze(), cmap='gray')
        axs[i, j].axis('off')
plt.show()
```