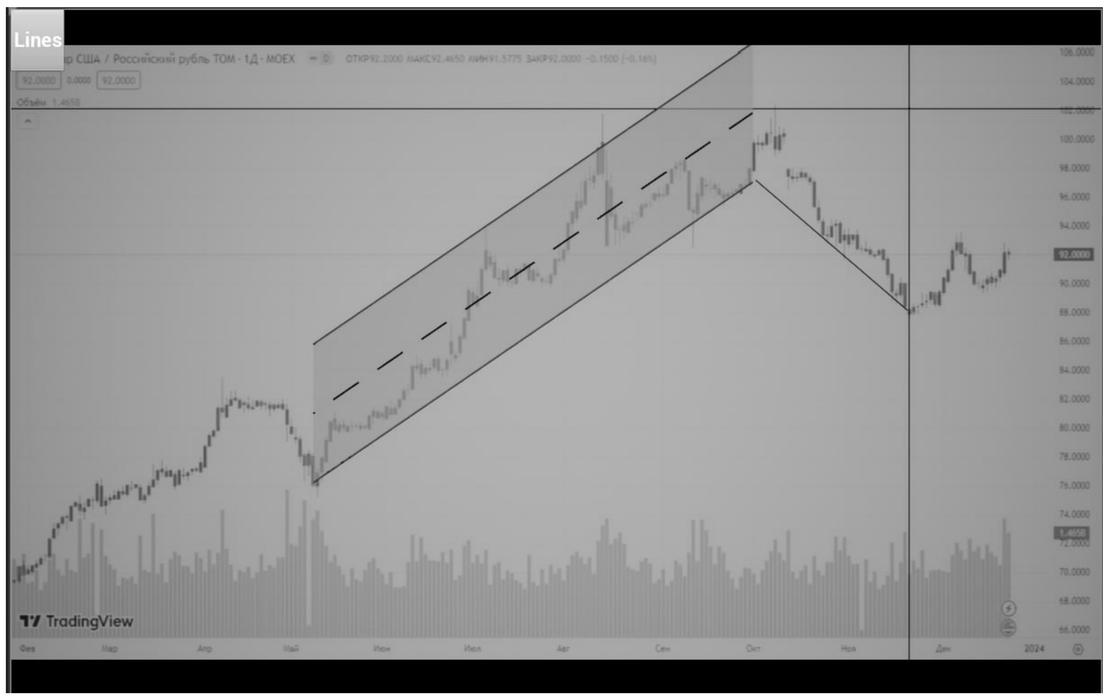
Меню выбора функциональных элементов

Пример изменения функциональных элементов



Изм.	Лист	№ докум.	Подп.	Дата
Разраб.		Афиногенов		
Пров.		Витюков		
Т.контр.				
Н.контр.				
Утв.				

Лит.	Масса	Масштаб
		1:1

Лист 1 Листов 4

МГТУ им. Н.Э. Баумана
кафедра РК6
группа РК6-71

Копировал

Формат А1

Перв. примен.

Справ. №

Подп. и дата

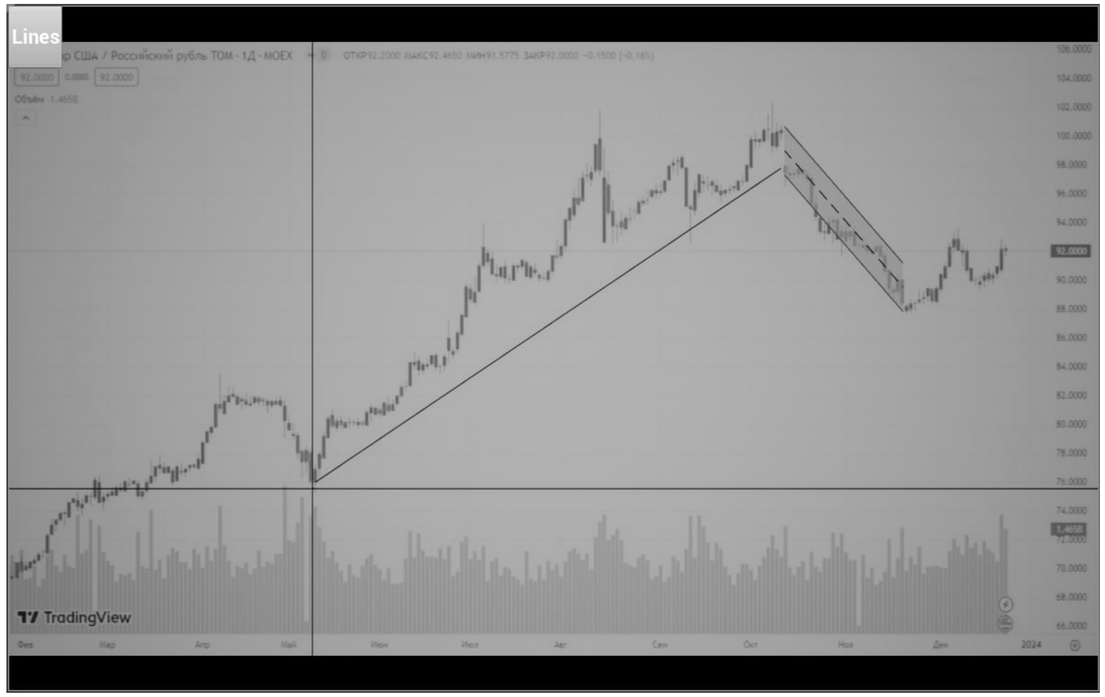
Инв. № подл.

Взам. инв. №

Инв. № дубл.

Подп. и дата

Пример размещения функциональных элементов



Участок кода, отвечающий за удаление выделенных объектов

```cpp
void ATradeviewPlayerController::DeleteObject()
{
    if (GEngine)
    {
        if (SelectedObject)
        {
            GEngine->AddOnScreenDebugMessage(-1, 15, FColor::Red, FString::Printf(TEXT("Selected Object is %s"), *SelectedObject->GetName()));
            TArray<USceneComponent*> ChildrenComponents;
            SelectedObject->GetChildrenComponents(true, ChildrenComponents);
            for (int i = 0; i < ChildrenComponents.Num(); i++)
            {
                ChildrenComponents[i]->DestroyComponent();
            }
            SelectedObject->DestroyComponent();
            SelectedObject = NULL;
            GetWorld()->GetFirstPlayerController()->CurrentMouseCursor = EMouseCursor::Default;
        }
        else
        {
            GEngine->AddOnScreenDebugMessage(-1, 15, FColor::Red, FString::Printf(TEXT("Select any object first")));
        }
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.		Афиногенов		
Пров.		Витюков		
Т.контр.				
Н.контр.				
Утв.				

Лит.	Масса	Масштаб
		1:1

Лист 2 Листов 4

МГТУ им. Н.Э. Баумана
кафедра РК6
группа РК6-71

Копировал

Формат А1

Перв. примен.
Справ. №
Подп. и дата
Инв. № дубл.
Взам. инв. №
Подп. и дата
Инв. № подл.

Участок кода, отвечающий за изменение вида курсора и отображения контролов

```cpp
void AParallelChannel::ChangeCursorInTouchZone(UPrimitiveComponent* TouchedComponent)
{
    TArray<USceneComponent*> AttachedLines;
    TArray<USceneComponent*> LineChildren;
    GetWorld()->GetFirstPlayerController()->CurrentMouseCursor = EMouseCursor::Hand;
    TouchedComponent->GetAttachParent()->GetChildrenComponents(false, AttachedLines);
    for (int i = 0; i < 2; i++)
    {
        AttachedLines[i]->GetChildrenComponents(false, LineChildren);
        for (int j = 0; j < LineChildren.Num(); j++)
        {
            EditZoneArray.Add(LineChildren[j]);
            EditZoneArray[i * 3 + j]->SetVisibility(true);
            EditZoneArray[i * 3 + j]->SetWorldScale3D(FVector(0.01f, 0.01f, 1.f));
        }
        LineChildren.Empty();
    }
}

void AParallelChannel::ChangeCursorToDefault(UPrimitiveComponent* TouchedComponent)
{
    GetWorld()->GetFirstPlayerController()->CurrentMouseCursor = EMouseCursor::Default;
    for (int i = 0; i < EditZoneArray.Num(); i++)
    {
        EditZoneArray[i]->SetVisibility(false);
    }
    EditZoneArray.Empty();
}
```

Участок кода, отвечающий за создание горизонтальных и вертикальных линий

```cpp
void ALine::CreateHorizontal(FVector Position)
{
    FVector LinePos = FVector(Position.X, Position.Y, 0.05f);
    FTransform LineTransform = FTransform(FRotator(0.f, 0.f, 0.f), LinePos, FVector(0.001f, 3.f, 1.f));
    FTransform DummyTransform = FTransform(LineTransform.GetRotation(), LineTransform.GetLocation(), LineTransform.GetScale3D() + FVector(0.0085f, 0.f, 0.f));
    FAttachmentTransformRules AttachmentRules = FAttachmentTransformRules(EAttachmentRule::SnapToTarget, EAttachmentRule::SnapToTarget, EAttachmentRule::KeepWorld, true);

    this->SetActorTransform(LineTransform);
    this->SetActorLocation(LinePos);

    DummyMesh->AttachToComponent(Line, AttachmentRules);
    DummyMesh->SetWorldTransform(DummyTransform);

    Line->SetVisibility(true);
}

void ALine::CreateVertical(FVector Position)
{
    FVector LinePos = FVector(Position.X, Position.Y, 0.05f);
    FTransform LineTransform = FTransform(FRotator(0.f, 0.f, 0.f), LinePos, FVector(3.0f, 0.001f, 1.f));
    FTransform DummyTransform = FTransform(LineTransform.GetRotation(), LineTransform.GetLocation(), LineTransform.GetScale3D() + FVector(0.f, 0.0085f, 0.f));
    FAttachmentTransformRules AttachmentRules = FAttachmentTransformRules(EAttachmentRule::SnapToTarget, EAttachmentRule::SnapToTarget, EAttachmentRule::KeepWorld, true);

    this->SetActorTransform(LineTransform);
    this->SetActorLocation(LinePos);

    DummyMesh->AttachToComponent(Line, AttachmentRules);
    DummyMesh->SetWorldTransform(DummyTransform);

    Line->SetVisibility(true);
}
```

Изм.	Лист	№ докум.	Подп.	Дата		Лит.	Масса	Масштаб
Разраб.		Афиногенов						1:1
Пров.		Витюков				Лист 3	Листов 4	
Т.контр.						МГТУ им. Н.Э. Баумана кафедра РК6 группа РК6-71		
Н.контр.								
Утв.								

Копировал

Формат А1

Перв. примен.

Справ. №

Подп. и дата

Инв. № дубл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Участок кода, отвечающий за редактирование линий с произвольным углом наклона

```
263  void ATradeviewPlayerController::Tick(float DeltaSeconds)
264  {
265      if (GrabMode)
266      {
267          if (GrabbedComponent)
268          {
269              FVector WorldLocation;
270              FVector WorldDirection;
271              this->DeprojectMousePositionToWorld(WorldLocation, WorldDirection);
272              FVector StartTrace = this->PlayerCameraManager->GetCameraLocation();
273              FVector EndTrace = StartTrace + WorldDirection * 100.f;
274              FVector OriginalEndTrace = WorldLocation + WorldDirection * OriginalDistance;
275              if (LineEditing)
276              {
277                  FHitResult HitResult;
278                  FCollisionQueryParams CollisionParams;
279                  FCollisionResponseParams Response;
280                  GetWorld()->LineTraceSingleByChannel(HitResult, StartTrace, EndTrace, ECollisionChannel::ECC_WorldStatic, CollisionParams, Response);
281
282                  FVector Direction = FVector(HitResult.ImpactPoint.X - AncorPoint.X, HitResult.ImpactPoint.Y - AncorPoint.Y, 0.f);
283                  FVector TransformLocation = FVector((HitResult.ImpactPoint.X - AncorPoint.X) / 2.f, (HitResult.ImpactPoint.Y - AncorPoint.Y) / 2.f, 0.f);
284                  FRotator Rotation = Direction.Rotation();
285                  float Length = Direction.Size();
286                  FVector Scale = FVector(Length / 100.f, 0.001f, 1.f);
287                  FVector LinePosition = FVector(AncorPoint.X + TransformLocation.X, AncorPoint.Y + TransformLocation.Y, 0.04f);
288                  LineTransform = FTransform(Rotation, LinePosition, Scale);
289                  GrabbedComponent->GetAttachParent()->SetWorldTransform(LineTransform);
290
291                  GrabbedComponent->GetAttachParent()->GetChildComponent(1)->SetWorldScale3D(FVector(0.01f, 0.01f, 1.f));
292                  GrabbedComponent->GetAttachParent()->GetChildComponent(1)->SetVisibility(true);
293
294                  GrabbedComponent->GetAttachParent()->GetChildComponent(2)->SetWorldScale3D(FVector(0.01f, 0.01f, 1.f));
295                  GrabbedComponent->GetAttachParent()->GetChildComponent(2)->SetVisibility(true);
296              }
```


Перв. примен.
Справ. №
Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.		Афиногенов		
Пров.		Витюков		
Т.контр.				
Н.контр.				
Утв.				

Лит.	Масса	Масштаб
		1:1

Лист 4 Листов 4

МГТУ им. Н.Э. Баумана
кафедра РК6
группа РК6-71

Копировал Формат А1