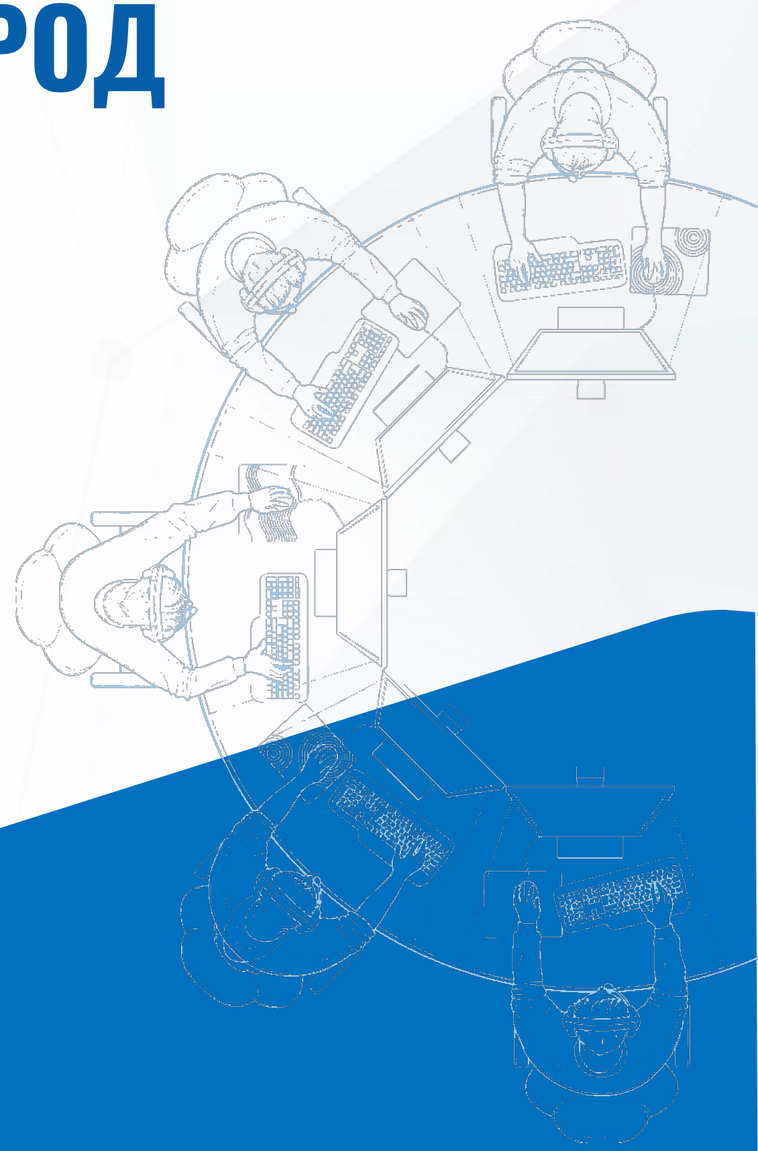
Добро пожаловать в Компьютерную Академию ШАГ

КОМПЬЮТЕРНАЯ АКАДЕМИЯ ШАГ ГОРОД

Директор филиала – Иваницкая Варвара

Менеджер учебного процесса – Земскова Яна



РАСПИСАНИЕ ЗАНЯТИЙ

Малая Компьютерная Академия

Длительность: 5 лет

Номер группы: С 2113

Первое занятие: 16 октября

Суббота: 15.00-17.50

КАНИКУЛЫ

Зима:
С 28.12 по 7.01

Весна:
С 1.05 по 9.05

Лето:
С 1.06 по 31.08

1 ГОД = 144 УЧ. ЧАСА

Виды платежей:

По частям	За Семестр	За Год
9 платежей	2 платежа	1 платеж

+1 страховой/обеспечительный

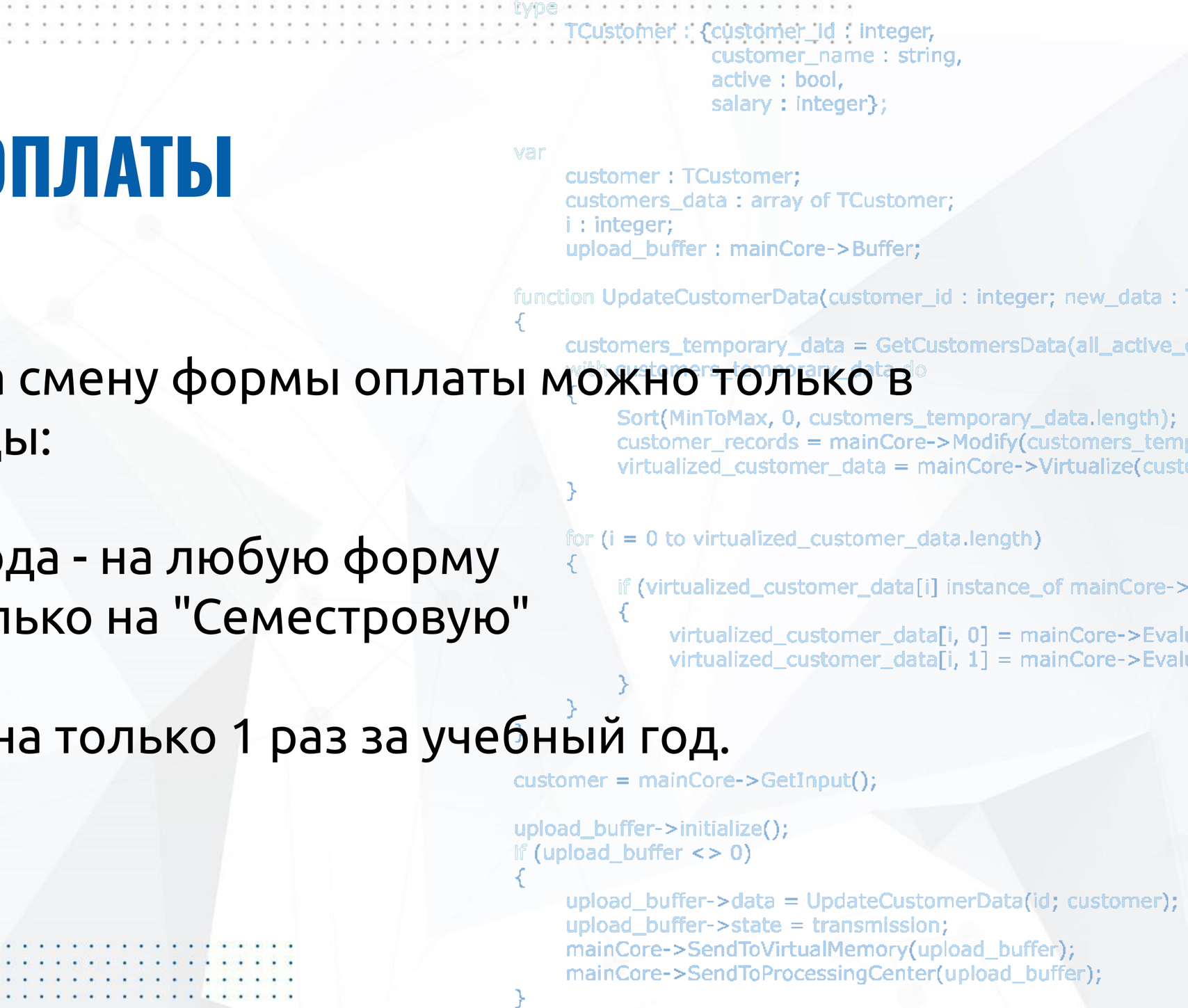
При отсутствии оплаты по графику, страховой платеж используется, как основной.
При отсутствии основного и страхового платежей студент отстраняется от занятий.

СМЕНА ФОРМЫ ОПЛАТЫ

Написать заявление на смену формы оплаты можно только в определенные периоды:

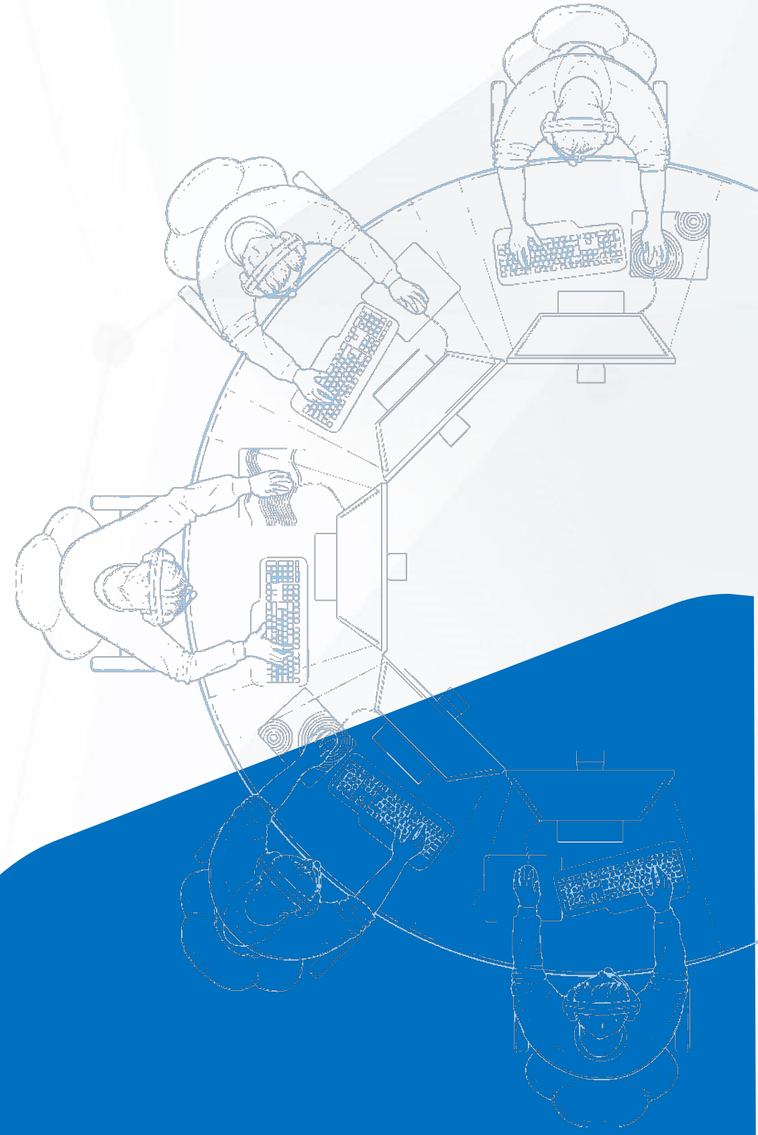
- В начале учебного года - на любую форму
- В конце семестра только на "Семестровую"

Смена формы возможна только 1 раз за учебный год.



ПРОПУСК ЗАНЯТИЙ

Если вы понимаете, что вынуждены пропустить занятие, обязательно сообщайте об этом в WhatsApp учебной части.



Отработка пропусков:

- Отработка по конспектам с возможностью связи с преподавателем;
- Отработка в параллельной группе (если будут места);
- Есть возможность организовать индивидуальную консультацию (платно).

ЧТО МЫ БУДЕМ ИЗУЧАТЬ

- Создание сайтов Wix, Canva 2.0
- Разработка игр на KODU
- 3D–моделирование и виртуальная реальность (Tinkercad и CoSpaces)
- Игровой дизайн
- Разработка игр – Junior (Construct 3)
- Робототехника и техника (LEGO)

```
type
      TCustomer : {customer_id : integer,
                   customer_name : string,
                   active : bool,
                   salary : integer};

var
      customer : TCustomer;
      customers_data : array of TCustomer;
      i : integer;
      upload_buffer : mainCore->Buffer;

function UpdateCustomerData(customer_id : integer; new_data : TCust
{
      customers_temporary_data = GetCustomersData(all_active_custon
      with customers_temporary_data do
      {
            Sort(MinToMax, 0, customers_temporary_data.length);
            customer_records = mainCore->Modify(customers_temporaty
            virtualized_customer_data = mainCore->Virtualize(customers
      }

      for (i = 0 to virtualized_customer_data.length)
      {
            if (virtualized_customer_data[i] instance_of mainCore->globa
            {
                  virtualized_customer_data[i, 0] = mainCore->Evaluate(s
                  virtualized_customer_data[i, 1] = mainCore->Evaluate(e
            }
      }
}

customer = mainCore->GetInput();

upload_buffer->initialize();
if (upload_buffer <> 0)
{
      upload_buffer->data = UpdateCustomerData(id; customer);
      upload_buffer->state = transmission;
      mainCore->SendToVirtualMemory(upload_buffer);
      mainCore->SendToProcessingCenter(upload_buffer);
}
```

Результат курса

Студент умеет создавать персональные сайты, используя базовый набор инструментов конструктора Wix, понимает структуру сайта, умеет использовать шаблоны Canva.

Цели курса:

- Обучить базовым действиям, необходимым для работы в сети Интернет, Google-сервисах и на онлайн-платформах;
- Заинтересовать студентов направлением WEB-дизайн;
- Научить редактировать готовые шаблоны и создавать сайт с пустого шаблона;
- Научить создавать персональные сайты, используя базовый набор инструментов конструктора Wix и шаблоны Canva.

Создание сайтов Wix и Canva 2.0

Разработка игр на KODU

Результат курса

Создание и программирование своей собственной многоуровневой 3D-игры.

Цели курса:

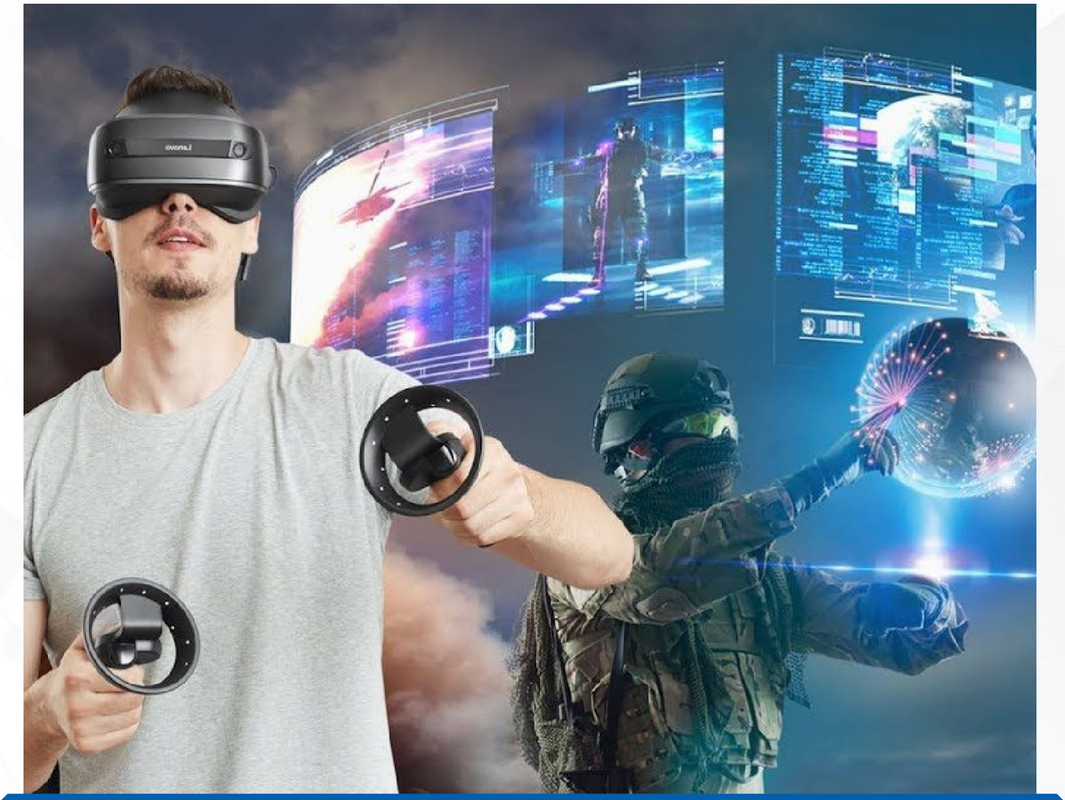
- Обучить логике и базису программирования без использования сложного синтаксиса;
- Продемонстрировать творческий аспект программирования;
- Развить навыки анализа поставленной задачи, структурирования и алгоритмизации решения.

Результат курса

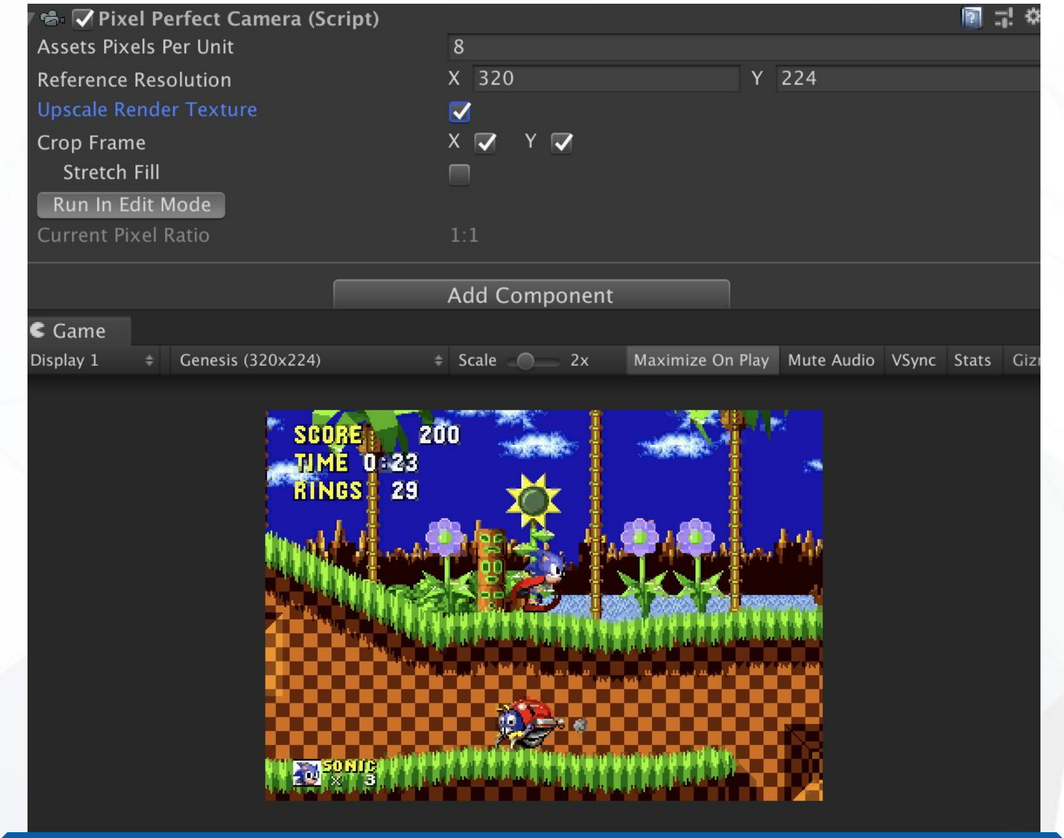
Студент умеет создавать 3D-модели зданий, транспорта и персонажей, космические локации при помощи ранее созданных моделей, трансформировать их в виртуальную экскурсию.

Цели курса:

- Ознакомить студентов с увлекательным направлением 3D-моделирования;
- Научить создавать свои интересные космические локации при помощи созданных 3D-моделей,
- Трансформировать модели в виртуальную экскурсию.



3D–моделирование и виртуальная реальность (Tinkercad и CoSpaces)



Игровой дизайн

Результат курса

Студент умеет создавать графические элементы для видеоигр.

Цели курса:

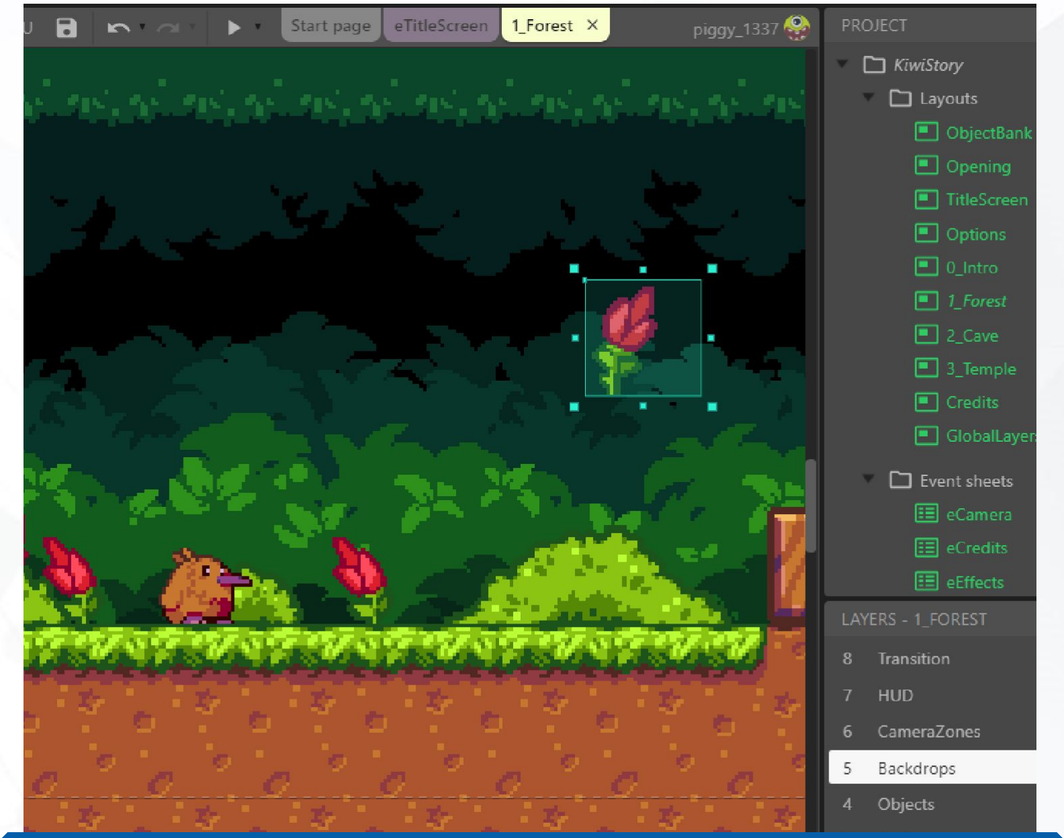
- Заложить основы дизайн-мышления;
- Заинтересовать студентов направлением концепт-арт и созданием игровых ассетов;
- Ознакомить студентов с основными направлениями дизайна игр: Pixel Art, Isometric Art, 2D-скетч, анимация;
- Научить создавать основные элементы игр: персонажи, меню, уровни, кнопки, загрузочный экран;
- Создать концепцию дизайна собственной игры.

Результат курса

Студент умеет создавать детальные сложные 2D-игры в жанре платформер на коммерческой платформе Construct 3.

Цели курса:

- Изучить программирование игр на движке Construct 3;
- Научить дизайну игровых элементов, созданию игровых сцен;
- Создать полноценную игру с несколькими уровнями, меню и финальным боссом.
- Научить программированию управляемых персонажей и компьютерных элементов;
- Изучить код, необходимый для создания игр в жанре платформер.



Разработка игр – Junior (Construct 3)

Робототехника и техника (LEGO)

Результат курса

Студент умеет конструировать и программировать различные модели роботов LEGO Mindstorms для соревнований и практических задач.

Цели курса:

- Увлечь студентов изучением технического дизайна и конструирования роботов;
- Научить конструировать и программировать роботов LEGO Mind-storms EV3;
- Научить применять датчики и создавать модели роботов в зависимости от поставленной задачи.

СЛЕДУЮЩИЕ СЕМЕСТРЫ

- Разработка мобильных приложений дополненной реальности
- Создание видео – YouTube
- LEGO Pro
- Программирование на Python –Junior
- Аркадные игры на Click Fusion 2.5
- Веб-дизайн – Junior (HTML & CSS)
- Программирование микроконтроллера BBC Micro:bit
- Фотолаборатория
- Разработка сайтов на WordPress
- Цифровая архитектура – SketchUp
- Разработка приложений на Python – Middle
- Анимация и мультипликация
- Диджитал арт
- Создание гаджетов на Arduino
- Искусственный интеллект и большие данные (Python – Senior)
- 3D–анимация
- Разработка игр – Middle (Unreal Engine)
- Видеолаборатория
- Создание виртуальных миров
- Инновационные технологии
- Стартап и фриланс
- Разработка игр – Senior (Unity)

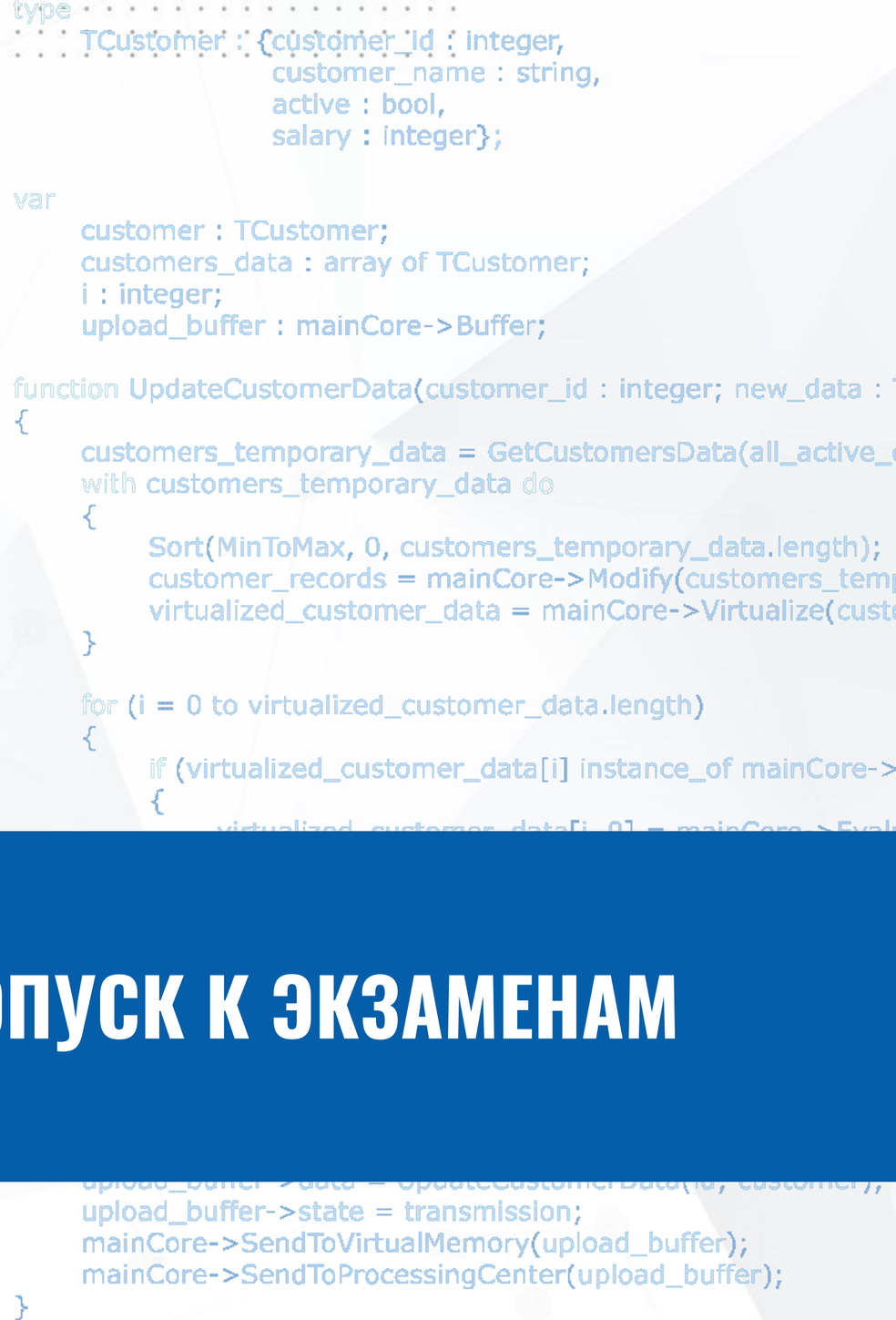
```
type
    TCustomer : {customer_id : integer,
                 customer_name : string,
                 active : bool,
                 salary : integer};

var
    customer : TCustomer;
    customers_data : array of TCustomer;
    i : integer;
    upload_buffer : mainCore->Buffer;

function UpdateCustomerData(customer_id : integer; new_data : TCust
{
    customers_temporary_data = GetCustomersData(all_active_custon
    with customers_temporary_data do
    {
        Sort(MinToMax, 0, customers_temporary_data.length);
        customer_records = mainCore->Modify(customers_temporaty
        virtualized_customer_data = mainCore->Virtualize(customers
    }

    for (i = 0 to virtualized_customer_data.length)
    {
        if (virtualized_customer_data[i] instance_of mainCore->globa
        {
            virtualized_customer_data[i, 0] = mainCore->Evaluate(s
            virtualized_customer_data[i, 1] = mainCore->Evaluate(e
        }
    }
}

customer = mainCore->GetInput();

upload_buffer->initialize();
if (upload_buffer <> 0)
{
    upload_buffer->data = UpdateCustomerData(id; customer);
    upload_buffer->state = transmission;
    mainCore->SendToVirtualMemory(upload_buffer);
    mainCore->SendToProcessingCenter(upload_buffer);
}
```

АКТИВНОСТИ

- Испытания на желтые браслеты;
- Праздники в Академии;
- Экскурсии и интерактивы.

ДОМАШНИЕ ЗАДАНИЯ

- Домашние задания обязательны к выполнению;
- В среднем, на выполнение домашних заданий уходит не более 1 часа в неделю.
- Все задания можно найти в MyStat.



80% ВЫПОЛНЕННЫХ ЗАДАНИЙ = ДОПУСК К ЭКЗАМЕНАМ

MyStat - электронный дневник студента

Web-версия **mystat.itstep.org**

GET IT ON Google Play

Download on the App Store

ПРАВИЛА ПОВЕДЕНИЯ НА ЗАНЯТИЯХ

1. Приходим на занятия за 10-15 мин до начала урока.
2. Не разрешается употреблять напитки и еду во время занятий.
3. Не разрешается пользоваться телефонами и планшетами, если это не предусмотрено заданием от учителя.
4. Если что-то не понятно или что-то пропустили, смело говорите об этом преподавателю.
5. При обнаружении любых проблем с компьютером или MyStat, важно сразу сказать об этом преподавателю или сообщать в учебную часть.
6. Мы уважаем себя и коллег, поэтому на занятиях не мешаем друг другу и поддерживаем.
7. Бережно относимся к технике и остальному имуществу. Заказчик обучения несет финансовую ответственность.

ЧТО ПОНАДОБИТСЯ НА ЗАНЯТИЯХ

- Флешка от 4 ГБ;
- Сменная обувь;
- Ручка, тетрадка - по желанию.

На первое занятие:

- Соглашение на то, что ребёнок может сам уходить домой.
- Соглашение на размещение фото учебного процесса в соцсетях.

Соглашения нужно отдать МУП или оставить на ресепшене.

КОНТАКТЫ

Почта: zemskova_y@itstep.org

Номер телефона: +7 (343) 239-57-55

Чат в WhatsApp: +7 (912) 654-07-72

```
type
      TCustomer : {customer_id : integer,
                   customer_name : string,
                   active : bool,
                   salary : integer};

var
      customer : TCustomer;
      customers_data : array of TCustomer;
      i : integer;
      upload_buffer : mainCore->Buffer;

function UpdateCustomerData(customer_id : integer; new_data :
{
      customers_temporary_data = GetCustomersData(all_active_
      with customers_temporary_data do
      {
            Sort(MinToMax, 0, customers_temporary_data.length);
            customer_records = mainCore->Modify(customers_temp
            virtualized_customer_data = mainCore->Virtualize(custo
      }

      for (i = 0 to virtualized_customer_data.length)
      {
            if (virtualized_customer_data[i] instance_of mainCore->
            {
                  virtualized_customer_data[i, 0] = mainCore->Evalu
                  virtualized_customer_data[i, 1] = mainCore->Evalu
            }
      }
}

customer = mainCore->GetInput();

upload_buffer->initialize();
if (upload_buffer <> 0)
{
      upload_buffer->data = UpdateCustomerData(id; customer);
      upload_buffer->state = transmission;
      mainCore->SendToVirtualMemory(upload_buffer);
      mainCore->SendToProcessingCenter(upload_buffer);
}
```

Спасибо за внимание!
Желаем успехов в учебе!