



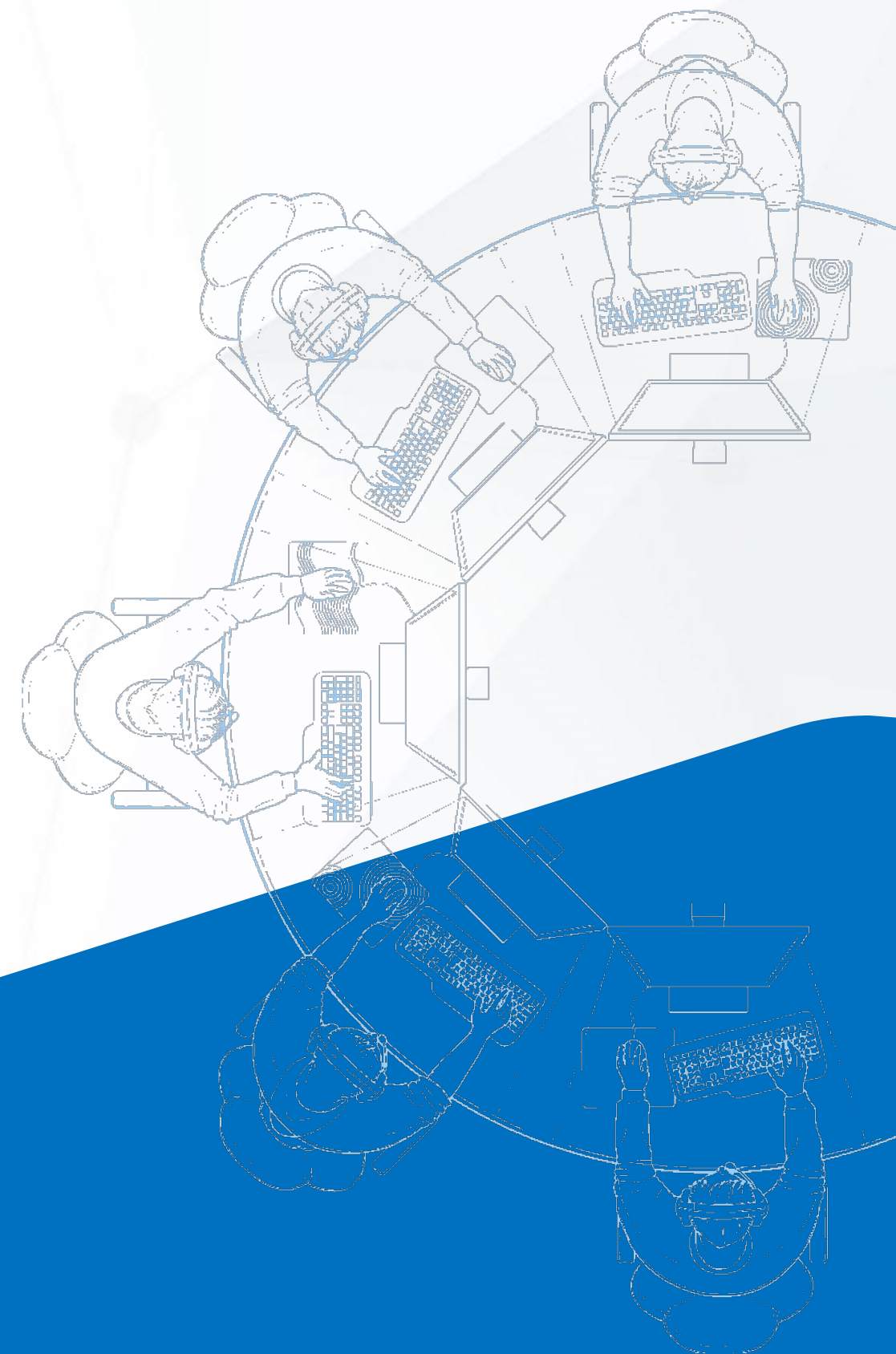
**Добро пожаловать в  
Компьютерную Академию TOP**

# КОМПЬЮТЕРНАЯ АКАДЕМИЯ ТОР ЧЕЛЯБИНСК

Директор филиала – Регина  
Тимофеева

Менеджер учебного процесса –  
Анара Нуржанова, Яна Юсупова

Преподаватели:  
Андрей Комаров  
Денис Пермяков





# РАСПИСАНИЕ ЗАНЯТИЙ

Малая Компьютерная Академия

**Длительность: 3 года**

**Номер группы: 3221**

**Первое занятие: 12 апреля**

**Вторник: 9:00-10:20**

**Четверг: 9:00-10:20**





# РАСПИСАНИЕ ЗАНЯТИЙ

Малая Компьютерная Академия

**Длительность: 3 года**

**Номер группы: В4221**

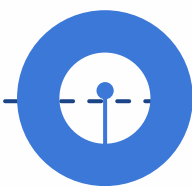
**Первое занятие: 10 апреля**

**Воскресенье: 13:30-16:20**





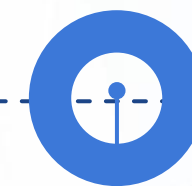
# КАНИКУЛЫ



**Зима**



**Весна**



**Лето**



# 1 ГОД = 144 УЧ. ЧАСА

## Виды платежей:

**По частям**  
**9 платежей**

⋮

**За Семестр**  
**2 платежа**

⋮

**За Год**  
**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

Изменить форму оплаты можно только **1 раз в календарный (физический) год** в установленные периоды:

- **Перед стартом и началом уч. года** - на любую, с условием полной оплаты До старта
- **В конце семестра** - только на оплату «За Семестр»

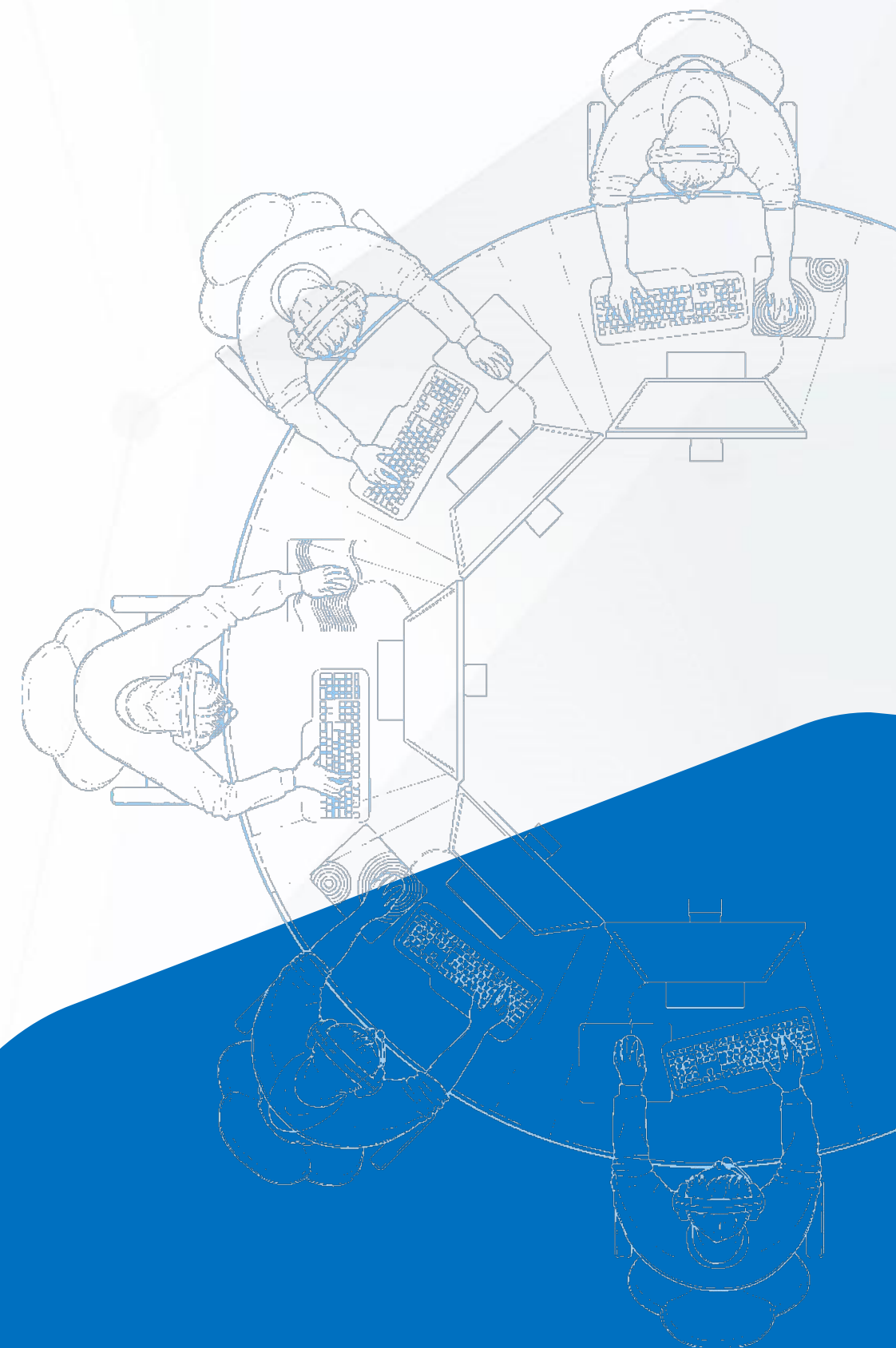
Смена формы оплаты происходит после подписания **Доп. соглашения** с клиентом

# ПРОПУСК ЗАНЯТИЙ

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

## Отработка пропусков:

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





# ЧТО МЫ БУДЕМ ИЗУЧАТЬ

- 3D-моделирование и 3D-печать (Tinkercad и SketchUp)
- Робототехника и техника (LEGO)
- Игровой дизайн
- Программирование на Python – Junior
- Веб-дизайн – Junior (HTML & CSS)
- Разработка игр – Junior (Construct 3)

```
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 : TCustomer)
{
    customers_temporary_data = GetCustomersData(all_active_customers_data);
    with customers_temporary_data do
    {
        Sort(MinToMax, 0, customers_temporary_data.length);
        customer_records = mainCore->Modify(customers_temporary_data, customer_id, new_data);
        virtualized_customer_data = mainCore->Virtualize(customers_temporary_data);
    }

    for (i = 0 to virtualized_customer_data.length)
    {
        if (virtualized_customer_data[i] instance_of mainCore->global::TCustomer)
        {
            virtualized_customer_data[i, 0] = mainCore->Evaluate(some_expression);
            virtualized_customer_data[i, 1] = mainCore->Evaluate(some_expression);
        }
    }
}

customer = mainCore->GetInput();

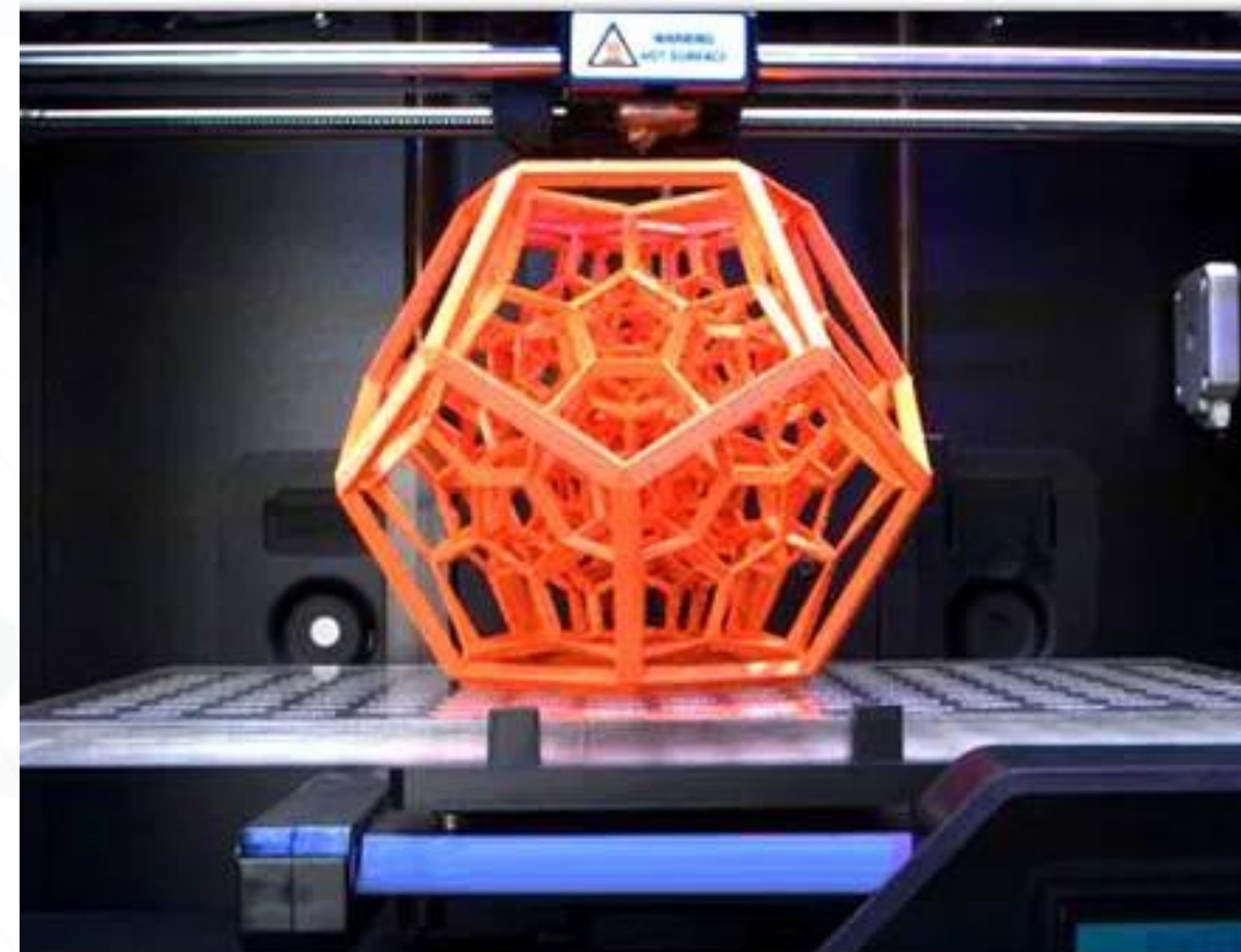
upload_buffer->initialize();
if (upload_buffer <> 0)
{
    upload_buffer->data = UpdateCustomerData(id; customer);
    upload_buffer->state = transmission;
    mainCore->SendToVirtualMemory(upload_buffer);
    mainCore->SendToProcessingCenter(upload_buffer);
}
```

## Результат курса

Студент умеет создавать 3D-модели зданий, транспорта и персонажей, свои интересные космические локации при помощи ранее созданных моделей, умеет моделировать и проектировать сложные здания, внутренний интерьер помещений города будущего на платформе SketchUp.

## Цели курса:

- Ознакомить студентов с увлекательным направлением 3D-моделирования;
- Научить студентов создавать 3D-модели зданий, транспорта и персонажей в Tinkercad, моделировать и проектировать сложные здания, внутренний интерьер помещений города будущего на платформе SketchUp.



3D–моделирование  
и 3D–печать  
(Tinkercad и SketchUp)





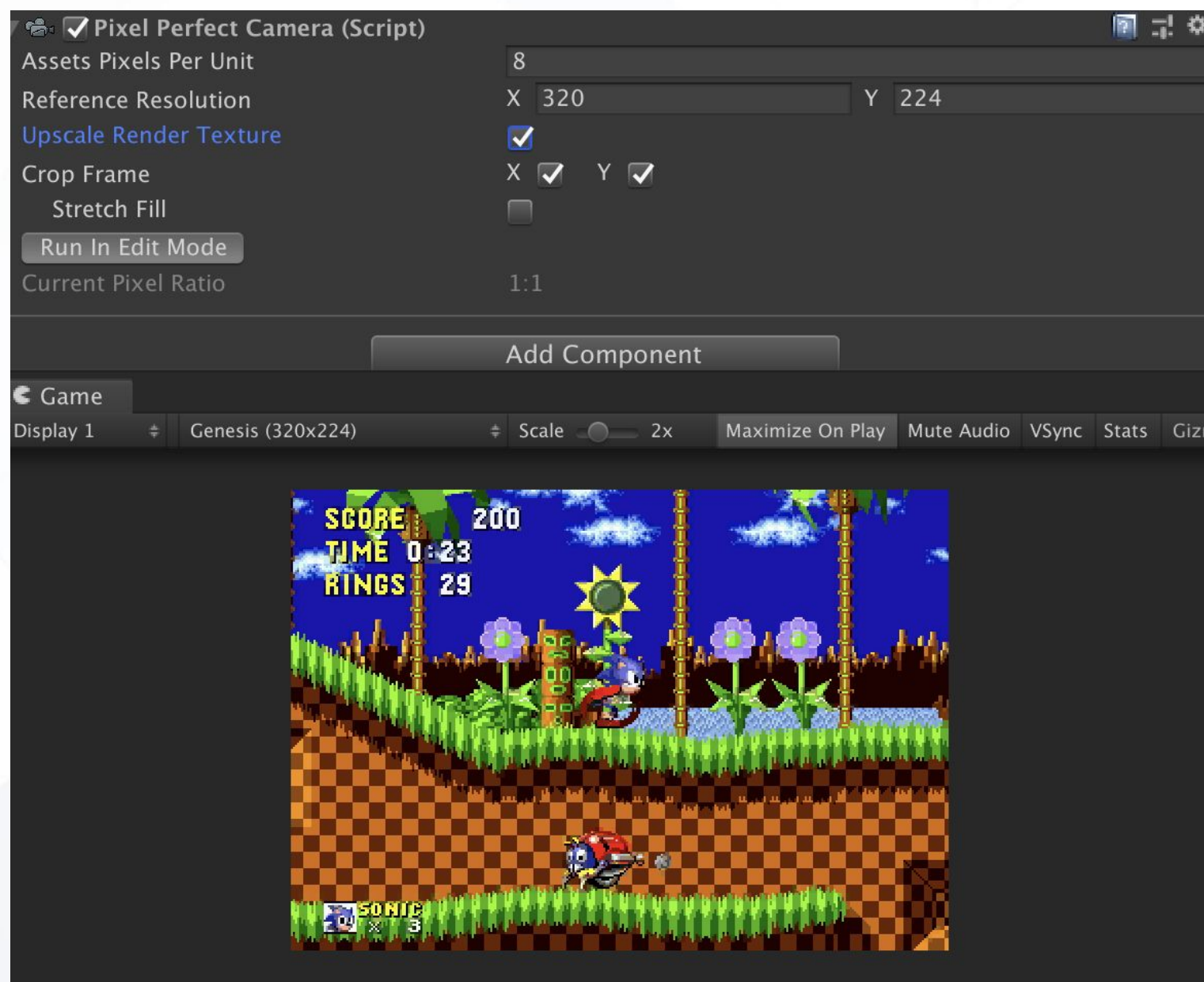
## Робототехника и техника (LEGO)

### 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 Mindstorms EV3;
- Научить применять датчики и создавать модели роботов в зависимости от поставленной задачи.



## 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, Iso-metric 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 простые программы на Python.

## Цели курса:

- Научить основам программирования и теории алгоритмов на Python;
- Научить разработке игр;
- Создать простые текстовые игры.

```
super(Ball, self)

def update(self):
    """ Check if bottom edge
    if self.bottom > games
        self.end_game()
        self.destroy()

def handle_caught(self):
    """ Destroy self if
    self.destroy()
```

Программирование  
на Python –Junior

Secure | <https://www.godessgreen.com>

**GODDESS  
GREEN**



Веб-дизайн – Junior  
(HTML & CSS)

## Результат курса

Студент умеет создавать лендинги и простые веб-сайты.

## Цели курса:

- Ознакомить с основами современных технологий создания сайтов на HTML5 и CSS3;
- Изучить базовый синтаксис HTML и CSS;
- Получить навыки создания верстки на основе готовых частей кода фреймворка Bootstrap.

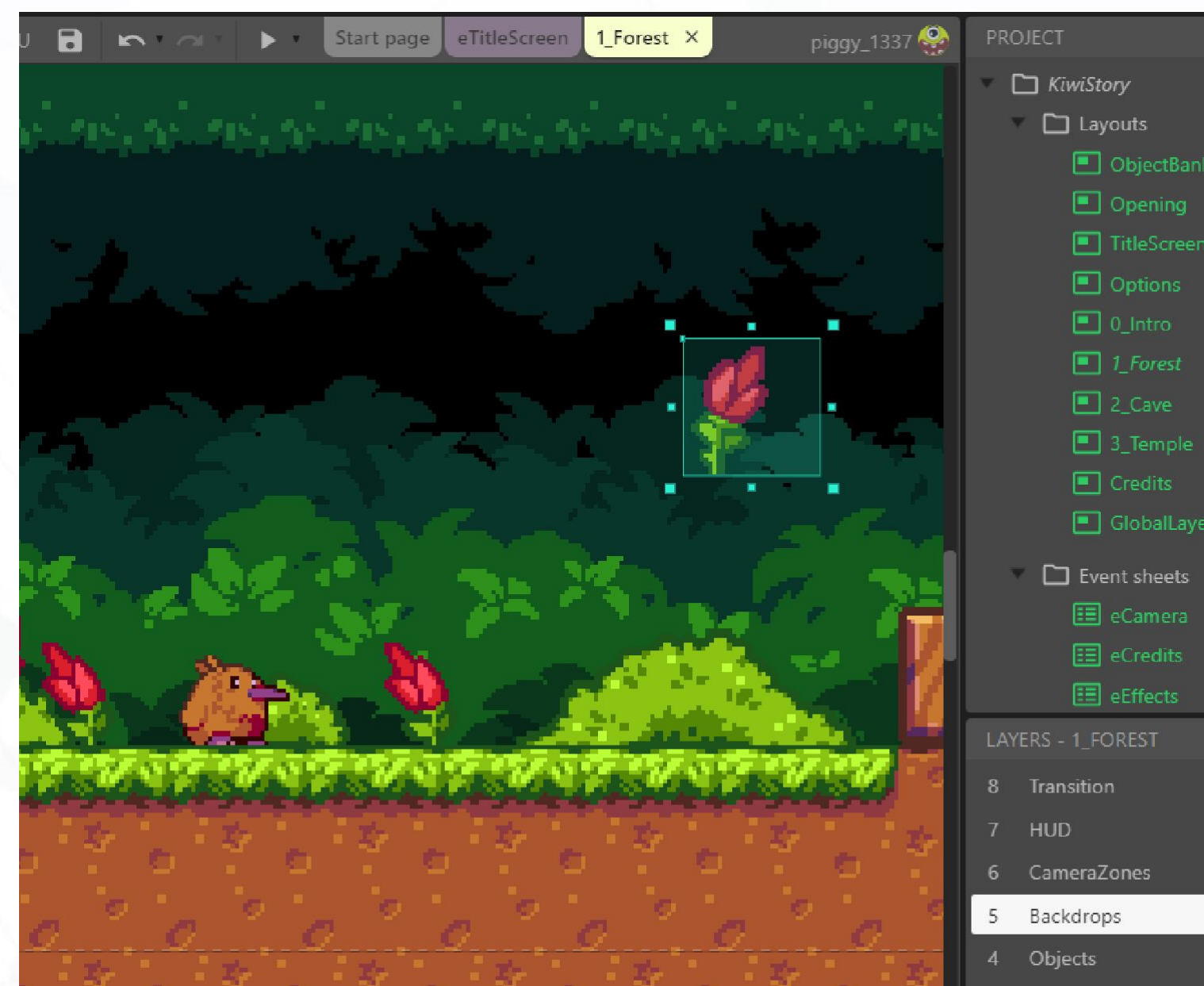


## Результат курса

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)

# СЛЕДУЮЩИЕ СЕМЕСТРЫ

- Разработка мобильных приложений дополненной реальности
- Создание видео – YouTube
- LEGO Pro
- Аркадные игры на Click Fusion 2.5
- 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 : TCustomer)
{
    customers_temporary_data = GetCustomersData(all_active_customers);
    with customers_temporary_data do
    {
        Sort(MinToMax, 0, customers_temporary_data.length);
        customer_records = mainCore->Modify(customers_temporary_data[cust
        virtualized_customer_data = mainCore->Virtualize(customers_temporary

    }
}

for (i = 0 to virtualized_customer_data.length)
{
    if (virtualized_customer_data[i] instance_of mainCore->global_data_arr
    {
        virtualized_customer_data[i, 0] = mainCore->Evaluate(salary, GetC
        virtualized_customer_data[i, 1] = mainCore->Evaluate(expences, C

    }
}

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);
}
}
```



# 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 Journal.

**60% ВЫПОЛНЕННЫХ ЗАДАНИЙ = ДОПУСК К ЭКЗАМЕНАМ**

```
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 : TCustomer)
{
    customers_temporary_data = GetCustomersData(all_active_customers_data);
    with customers_temporary_data do
    {
        Sort(MinToMax, 0, customers_temporary_data.length);
        customer_records = mainCore->Modify(customers_temporary_data, customer_id, new_data);
        virtualized_customer_data = mainCore->Virtualize(customers_temporary_data);
    }

    for (i = 0 to virtualized_customer_data.length)
    {
        if (virtualized_customer_data[i] instance_of mainCore->TCustomer)
        {
            virtualized_customer_data[i, 0] = mainCore->Eval(customer_id, new_data);
        }
    }
}

upload_buffer->data = UpdateCustomerData(customer_id, customer);
upload_buffer->state = transmission;
mainCore->SendToVirtualMemory(upload_buffer);
mainCore->SendToProcessingCenter(upload_buffer);
}
```



# Journal - электронный дневник студента

Web-версия [journal.top-academy.ru](http://journal.top-academy.ru)





# 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.



# 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 Jougna!, 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.



# КОНТАКТЫ

**Почта: [zavuch\\_chel@top-academy.ru](mailto:zavuch_chel@top-academy.ru)**

**Номер телефона: +7 (351) 220-20-41**

**Чат в Viber: +7 (999) 333-79-64**

```
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 : TCustomer)
{
    customers_temporary_data = GetCustomersData(all_active_customers_data);
    with customers_temporary_data do
    {
        Sort(MinToMax, 0, customers_temporary_data.length);
        customer_records = mainCore->Modify(customers_temporary_data, customer_id, new_data);
        virtualized_customer_data = mainCore->Virtualize(customer_records);
    }
    for (i = 0 to virtualized_customer_data.length)
    {
        if (virtualized_customer_data[i] instance_of mainCore->TCustomer)
        {
            virtualized_customer_data[i, 0] = mainCore->Evaluate(virtualized_customer_data[i, 0], new_data);
            virtualized_customer_data[i, 1] = mainCore->Evaluate(virtualized_customer_data[i, 1], new_data);
        }
    }
}

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!**