

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА ДИЗАЙНА И ТЕХНОЛОГИЙ

Рабочая программа дисциплины (модуля)
ГЕНЕРАТИВНЫЙ АРТ

Направление и направленность (профиль)
54.03.01 Дизайн. Цифровой дизайн

Год набора на ОПОП
2025

Форма обучения
очная

Владивосток 2025

Рабочая программа дисциплины (модуля) «Генеративный арт» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 54.03.01 Дизайн (утв. приказом Минобрнауки России от 13.08.2020г. №1015) и Порядком организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утв. приказом Минобрнауки России от 06.04.2021 г. N245).

Составитель(и):

Погребняк Е.В., доцент, Кафедра дизайна и технологий, Pogrebnyak.EV@vvsu.ru

Утверждена на заседании кафедры дизайна и технологий от 16.09.2025 , протокол №

1

СОГЛАСОВАНО:

Заведующий кафедрой (разработчика)

Туговикова О.Ф.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат	1577199753
Номер транзакции	0000000000ECDAF3
Владелец	Туговикова О.Ф.

1 Цель, планируемые результаты обучения по дисциплине (модулю)

Цель дисциплины – сформировать у студентов навыки работы с современными инструментами для создания интерактивного аудио-визуального контента при помощи алгоритмов и языков программирования (python, javaScript) в специализированных средах. Получить опыт создания графического материала с помощью визуального программирования, а также применения этих навыков в профессиональной деятельности. Вовлечение обучающихся в процессы саморазвития и самореализации.

Задачи дисциплины:

- Познакомить студентов с современными инструментами создания интерактивного контента, в том числе с помощью адаптированного и/или визуального программирования, позволяющими создавать и редактировать не только конечные графические элементы, но и сами графические инструменты. Изучить различные способы проектирования собственных визуальных эффектов, применяя инструменты специализированных программных сред для визуального программирования для создания интерактивных стендов, в том числе для выставок и образовательных мастер-классов.

А также:

- Приобретение навыков формулирования целей и задач научного исследования, выбора и обоснования методики исследования; - Формирование навыков библиографической работы, самостоятельной работы с различными источниками информации; - Проведение анализа, систематизации и обобщение информации по теме исследований; - Развитие творческой активности и инициативы студентов.- Вовлечение студентов в социальную активность университета, развитие интереса к участию в социально-значимых проектах (спортивных, культурных, общественных, экологических мероприятиях);- Формирование желания активного участия в волонтерских мероприятиях, в оказании помощи нуждающимся;- Формирование эстетического вкуса при обустройстве внешней среды во время проведения культурных, общественных, экологических и других мероприятий;- Развитие навыков социализации в коллективе;

Планируемыми результатами обучения по дисциплине (модулю), являются знания, умения, навыки. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы, представлен в таблице 1.

Таблица 1 – Компетенции, формируемые в результате изучения дисциплины (модуля)

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине		
			Код результата	Формулировка результата	
54.03.01 «Дизайн» (Б-ДЗ)	ПКВ-3 : Способен проектировать корпоративную айдентику и системы визуальной коммуникации	ПКВ-3.1к : Проектирует корпоративную айдентику и адаптирует ее для нужд бизнеса в цифровой среде	РД1	Знание	Знает различные методы разработки интерактивных аудио-визуальных инсталляций
			РД2	Навык	Имеет опыт создания интерактивных инсталляций и визуальных эффектов
			РД3	Умение	Умеет работать в специализированном программном обеспечении для создания интерактивного контента

В процессе освоения дисциплины решаются задачи воспитания гармонично развитой, патриотичной и социально ответственной личности на основе традиционных российских духовно-нравственных и культурно-исторических ценностей, представленные в таблице 1.2.

Таблица 1.2 – Целевые ориентиры воспитания

Воспитательные задачи	Формирование ценностей	Целевые ориентиры
Формирование гражданской позиции и патриотизма		
Воспитание уважения к истории и культуре России	Высокие нравственные идеалы	Активная жизненная позиция
Формирование духовно-нравственных ценностей		
Воспитание экологической культуры и ценностного отношения к окружающей среде	Гражданственность	Доброжелательность и открытость
Формирование научного мировоззрения и культуры мышления		
Развитие творческих способностей и умения решать нестандартные задачи	Единство народов России	Любовь к искусству
Формирование коммуникативных навыков и культуры общения		
Развитие умения эффективно общаться и сотрудничать	Созидательный труд	Уважение к другой культуре

2 Место дисциплины (модуля) в структуре ОПОП

Дисциплина «Генеративный арт» является элективной дисциплиной части учебного плана, формируемой участниками образовательных отношений у бакалавров направления «54.03.01 Дизайн» по профилю Цифровой дизайн. Совместно с другими дисциплинами участвует в формировании общепрофессиональных компетенций студента в контексте его будущей деятельности. Дисциплина разработана и читается студентам с целью общепрофессиональной подготовки и связана с формированием комплекса знаний о процессе создания интерактивного аудио-визуального контента с помощью специализированного программного обеспечения, в т.ч. для визуального или адаптированного программирования графики.

3. Объем дисциплины (модуля)

Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися (по видам учебных занятий) и на самостоятельную работу, приведен в таблице 2.

Таблица 2 – Общая трудоемкость дисциплины

Название ОПОП ВО	Форма обуче- ния	Часть УП	Семестр (ОФО) или курс (ЗФО, ОЗФО)	Трудо- емкость	Объем контактной работы (час)						СРС	Форма аттес- тации
				(З.Е.)	Всего	Аудиторная			Внеауди- торная			
						лек.	прак.	лаб.	ПА	КСР		
54.03.01 Дизайн	ОФО	Б1.ДВ.Б	3	3	55	18	36	0	1	0	53	Э

4 Структура и содержание дисциплины (модуля)

4.1 Структура дисциплины (модуля) для ОФО

Тематический план, отражающий содержание дисциплины (перечень разделов и тем), структурированное по видам учебных занятий с указанием их объемов в соответствии с учебным планом, приведен в таблице 3.1

Таблица 3.1 – Разделы дисциплины (модуля), виды учебной деятельности и формы текущего контроля для ОФО

№	Название темы	Код результата обучения	Кол-во часов, отведенное на				Форма текущего контроля
			Лек	Практ	Лаб	СРС	
1	Введение в генеративные технологии	РД1	3	6	0	10	практическая работа
2	Знакомство со специализированной программной средой и адаптированным языком python	РД2	3	0	6	10	практическая работа
3	Разработка визуальных эффектов с помощью обработки видеоконтента в реальном времени		4	0	8	10	практическая работа
4	Разработка собственного специализированного графического редактора под проект	РД2, РД3	4	8	0	10	практическая работа
5	Визуальное программирование и развивающаяся ниша веб-инструментов для визуальных эффектов	РД3	4	8	0	13	практическая работа
Итого по таблице			18	22	14	53	

4.2 Содержание разделов и тем дисциплины (модуля) для ОФО

Тема 1 Введение в генеративные технологии.

Содержание темы: Обзор программного обеспечения и возможностей современных технологий, в т.ч. Processing, Touch Designer и др.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: минилекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: повторение пройденного материала.

Тема 2 Знакомство со специализированной программной средой и адаптированным языком python.

Содержание темы: знакомство с возможностями языка python и программной средой Processing.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: минилекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: повторение пройденного материала.

Тема 3 Разработка визуальных эффектов с помощью обработки видеоконтента в реальном времени.

Содержание темы: Разработка собственных алгоритмов для создания интерактивных видеоэффектов.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: минилекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: повторение пройденного материала.

Тема 4 Разработка собственного специализированного графического редактора под проект.

Содержание темы: Изучение возможностей создания собственных графических инструментов под конкретный проект.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: минилекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: повторение пройденного материала.

Тема 5 Визуальное программирование и развивающаяся ниша веб-инструментов для визуальных эффектов.

Содержание темы: Изучение возможностей новых веб-инструментов для визуального программирования различных видео-эффектов, в т.ч. для 3Д-мэппинга.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: минилекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: повторение пройденного материала.

5 Методические указания для обучающихся по изучению и реализации дисциплины (модуля)

5.1 Методические рекомендации обучающимся по изучению дисциплины и по обеспечению самостоятельной работы

Процесс изучения данной дисциплины предполагает выполнение практических работ, сопровождающихся демонстрацией видео-уроков и других презентационных материалов. Знания, полученные студентами в аудитории, закрепляются и дополняются самостоятельно дома, в библиотеке, посредством использования ресурсов глобальной сети Интернет. Аудитория должна быть оснащена мультимедийным оборудованием и диапроектором. Классы для практических занятий должны быть оборудованы столами с горизонтальными столешницами. В качестве наглядных пособий на практических занятиях используется методический фонд кафедры.

5.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

6 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)

В соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине (модулю) созданы фонды оценочных средств. Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 1.

7 Учебно-методическое и информационное обеспечение дисциплины (модуля)

7.1 Основная литература

1. Войтов, А. Г. Наглядность, визуалистика, инфографика системного анализа : учебное пособие / А. Г. Войтов. - 7-е изд. - Москва : Дашков и К, 2022. - 212 с. - ISBN 978-5-394-05090-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1926412> (Дата обращения - 21.11.2025)

2. Инновации в сервисе: использование инфографии : учебное пособие / В. О. Чулков, Н. М. Комаров, Л. В. Сумзина [и др.]. - Москва : СОЛОН-ПРЕСС, 2020. - 124 с. - ISBN 978-5-91359-131-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1858774> (Дата обращения - 21.11.2025)

3. Титов, А.Н. Интерактивная визуализация данных. Работа с библиотекой Plotly : учеб.-метод. пособие / Р.Ф. Тазиева; Казан. нац. исслед. технol. ун-т; А.Н. Титов. — Казань : КНИТУ, 2023. — 136 с. — ISBN 978-5-7882-3387-1. — URL: <https://lib.rucont.ru/efd/870400> (дата обращения: 04.08.2025)

7.2 Дополнительная литература

1. Алпатов, А. Н. Визуальное программирование : методические указания / А. Н. Алпатов. — Москва : РТУ МИРЭА, 2020. — 49 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/167577> (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.

2. Гуревич, П. С., Эстетика : учебник / П. С. Гуревич. — Москва : КноРус, 2023. — 182 с. — ISBN 978-5-406-11278-6. — URL: <https://book.ru/book/948618> (дата обращения: 18.11.2025). — Текст : электронный.

3. Позднякова, Т. С. Пропедевтика графического дизайна : учебно-методическое пособие / Т. С. Позднякова. — Майкоп : АГУ, 2021. — 40 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/231407> (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.

7.3 Ресурсы информационно-телекоммуникационной сети "Интернет", включая профессиональные базы данных и информационно-справочные системы (при необходимости):

1. Электронно-библиотечная система "BOOK.ru"

2. Электронно-библиотечная система "ZNANIUM.COM"
3. Электронно-библиотечная система "ЛАНЬ"
4. Электронно-библиотечная система "РУКОНТ"
5. Open Academic Journals Index (ОАИ). Профессиональная база данных - Режим доступа: <http://oaji.net/>
6. Президентская библиотека им. Б.Н.Ельцина (база данных различных профессиональных областей) - Режим доступа: <https://www.prlib.ru/>
7. Информационно-справочная система "Консультант Плюс" - Режим доступа: <http://www.consultant.ru/>

8 Материально-техническое обеспечение дисциплины (модуля) и перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

Основное оборудование:

- Компьютеры
- Проектор

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА ДИЗАЙНА И ТЕХНОЛОГИЙ

Фонд оценочных средств
для проведения текущего контроля
и промежуточной аттестации по дисциплине (модулю)

ГЕНЕРАТИВНЫЙ АРТ

Направление и направленность (профиль)
54.03.01 Дизайн. Цифровой дизайн

Год набора на ОПОП
2025

Форма обучения
очная

Владивосток 2025

1 Перечень формируемых компетенций

Название ОПОП ВО, сокращенное	Код и формулировка компетенции и	Код и формулировка индикатора достижения компетенции
54.03.01 «Дизайн» (Б-ДЗ)	ПКВ-3 : Способен проектировать корпоративную айдентику и системы визуальной коммуникации	ПКВ-3.1к : Проектирует корпоративную айдентику и адаптирует ее для нужд бизнеса в цифровой среде

Компетенция считается сформированной на данном этапе в случае, если полученные результаты обучения по дисциплине оценены положительно (диапазон критериев оценивания результатов обучения «зачтено», «удовлетворительно», «хорошо», «отлично»). В случае отсутствия положительной оценки компетенция на данном этапе считается несформированной.

2 Показатели оценивания планируемых результатов обучения

Компетенция ПКВ-3 «Способен проектировать корпоративную айдентику и системы визуальной коммуникации»

Таблица 2.1 – Критерии оценки индикаторов достижения компетенции

Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине			Критерии оценивания результатов обучения
	Код результата	Тип результата	Результат	
ПКВ-3.1к : Проектирует корпоративную айдентику и адаптирует ее для нужд бизнеса в цифровой среде	РД 1	Знание	Знает различные методы разработки интерактивных аудиовизуальных инсталляций	устный ответ
	РД 2	Навык	Имеет опыт создания интерактивных инсталляций и визуальных эффектов	Выполнение работы
	РД 3	Умение	Умеет работать в специализированном программном обеспечении для создания интерактивного контента	Созданные объекты

Таблица заполняется в соответствии с разделом 1 Рабочей программы дисциплины (модуля).

3 Перечень оценочных средств

Таблица 3 – Перечень оценочных средств по дисциплине (модулю)

Контролируемые планируемые результаты обучения	Контролируемые темы дисциплины	Наименование оценочного средства и представление его в ФОС	
		Текущий контроль	Промежуточная аттестация
Очная форма обучения			

РД1	Знание : Знает различные методы разработки интерактивных аудио-визуальных инсталляций	1.1. Введение в генеративные технологии	Кейс-задача	Кейс-задача
РД2	Навык : Имеет опыт создания интерактивных инсталляций и визуальных эффектов	1.2. Знакомство со специализированной программной средой и адаптированным языком python	Кейс-задача	Кейс-задача
		1.4. Разработка собственного специализированного графического редактора под проект	Кейс-задача	Кейс-задача
РД3	Умение : Умеет работать в специализированном программном обеспечении для создания интерактивного контента	1.4. Разработка собственного специализированного графического редактора под проект	Индивидуальное домашнее задание	Индивидуальное домашнее задание
		1.5. Визуальное программирование и развивающаяся ниша веб-инструментов для визуальных эффектов	Индивидуальное домашнее задание	Индивидуальное домашнее задание

4 Описание процедуры оценивания

Качество сформированности компетенций на данном этапе оценивается по результатам текущих и промежуточных аттестаций при помощи количественной оценки, выраженной в баллах. Максимальная сумма баллов по дисциплине (модулю) равна 100 баллам.

Суммарно по дисциплине можно получить максимум 100 баллов за семестр.

Вид учебной деятельности	Кейс/задача	Индивидуальное домашнее задание	Итого
Текущая аттестация	40		40
Промежуточная аттестация		60	60
Итого	40	60	100

Сумма баллов, набранных студентом по всем видам учебной деятельности в рамках дисциплины, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика качества сформированности компетенции
от 91 до 100	«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций, обладает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями, умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым дисциплинарным компетенциям, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«не зачтено» /	У студента не сформированы дисциплинарные компетенции, проявляется недостаточность знаний, умений, навыков.

	«неудовлетворительно»	
от 0 до 40	«не зачтено» / «неудовлетворительно»	Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

5 Примерные оценочные средства

5.1 Задания для решения кейс-задачи

Создать собственные графические инструменты для плаката. Спроектировать три плаката, используя созданные инструменты.

Краткие методические указания

Программная среда для выполнения задания - Processing и язык Python. Воспользуйтесь лекциями и наработками с практических занятий.

Шкала оценки

Максимально всего 40 баллов. Техническая сложность решения - до 15 баллов, Оригинальность решения - до 15 баллов, Корректность решения - до 10 баллов

5.2 Пример индивидуального домашнего задания

Создать интерактивный стол для рисования на мультитач экране

Краткие методические указания

Используйте наработки из практических занятий и возможности программной среды Processing

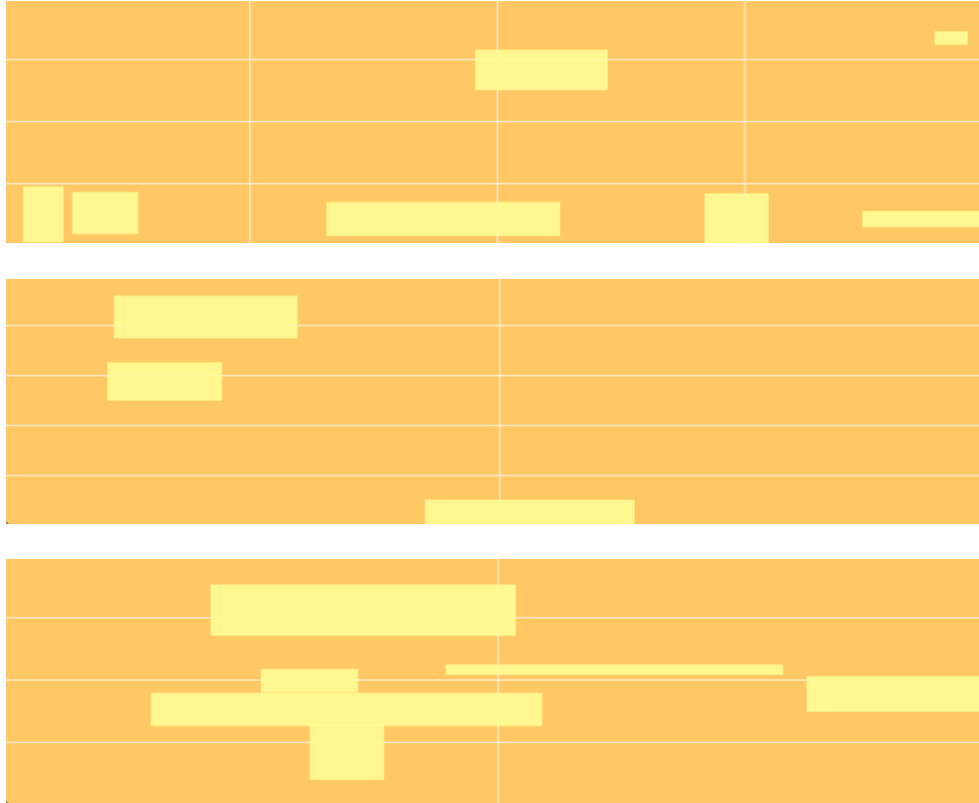
Шкала оценки

Максимально всего - 60. Техническая сложность решения - до 20 баллов, Оригинальность решения - до 20 баллов, Корректность решения - до 20 баллов

КЛЮЧИ ПО ДИСЦИПЛИНЕ
«ГЕНЕРАТИВНЫЙ АРТ (ID = 47390)»

Ключ к решению кейс-задачи «Создать собственные графические инструменты для плаката. Спроектировать три плаката, используя созданные инструменты»

Программная среда для выполнения задания - Processing и язык Python.
Воспользуйтесь лекциями и наработками с практических занятий.



Пример кода для Processing

```
def setup():
    size(800,200)
    background(200,0,0)
    stroke(0)
    building_block(0,0,width,height,color(255,200,100),True)

def building_block(x, y, w, h, fill_color, detail=True):
    """
    Рисует прямоугольный блок (здание/окно) со случайными деталями.
    x, y: координаты левого верхнего угла
    w, h: ширина и высота
    fill_color: цвет заливки
    detail: добавлять ли случайные детали (окна)
    """
    # Основной фасад
    fill(fill_color)
    rect(x, y, w, h)

    if detail and w > 10 and h > 10: # Добавляем детали, если блок достаточно большой
        stroke(255, 200) # Светлые линии окон
        strokeWeight(1)
```

```

# Случайное количество горизонтальных и вертикальных линий
h_lines = int(random(2, 6))
v_lines = int(random(2, 6))

# Горизонтальные разделители (этажи)
for i in range(1, h_lines):
    line_y = y + (h / h_lines) * i
    line(x, line_y, x + w, line_y)

# Вертикальные разделители (подъезды/секции)
for i in range(1, v_lines):
    line_x = x + (w / v_lines) * i
    line(line_x, y, line_x, y + h)

# Случайные "горящие" окна
noStroke()
for _ in range(int(random(2, 8))):
    wx = x + random(5, w-5)
    wy = y + random(5, h-5)
    ww = random(3, w / v_lines - 3)
    wh = random(3, h / h_lines - 3)
    fill(255, 255, 150, 220) # Теплый желтый свет
    rect(wx, wy, ww, wh)

```

Ключ к решению индивидуального домашнего задания «Интерактивный стол для рисования на мультитач экране»

Используйте наработки из практических занятий и возможности программной среды Processing

Концепция

Мультитач холст с различными кистями, цветовой палитрой, ластиком и возможностью сохранения рисунков. Интерфейс оптимизирован для управления пальцами.

Пример кода для Processing (Полный код решения)

```

# Мультитач стол для рисования
# Управление: пальцы для рисования, жесты для инструментов

# Глобальные переменные
drawing = True # Режим рисования
current_color = color(0) # Текущий цвет
brush_size = 20 # Размер кисти
brush_type = 0 # Тип кисти: 0-круглая, 1-квадратная, 2-звездная
background_color = color(255) # Цвет фона
stroke_history = [] # История штрихов для отмены
max_history = 50 # Максимальное количество шагов в истории

# Цветовая палитра
palette = [
    color(0), # Черный
    color(255, 0, 0), # Красный
    color(0, 255, 0), # Зеленый
    color(0, 0, 255), # Синий
    color(255, 255, 0), # Желтый
    color(255, 0, 255), # Пурпурный
    color(0, 255, 255), # Голубой
    color(255, 165, 0), # Оранжевый
    color(128, 0, 128), # Фиолетовый
    color(255) # Белый (ластик)
]

```

```

def setup():
    # Используем fullScreen для мультитач экрана
    fullScreen(P2D)
    background(background_color)

    # Настройка сглаживания
    smooth()

    # Инициализация истории
    stroke_history.append(get())

    # Отображение инструкций
    display_instructions()

def draw():
    # Рисуем только при касании
    if drawing and len(touches) > 0:
        for touch in touches:
            draw_with_touch(touch.id, touch.x, touch.y, touch.pressure)

    # Рисуем интерфейс поверх всего
    draw_interface()

def draw_with_touch(touch_id, x, y, pressure):
    """Рисование в зависимости от типа кисти и давления"""

    # Определяем размер кисти с учетом давления
    current_size = brush_size * (0.5 + pressure * 1.5)

    # Выбираем тип кисти
    if brush_type == 0:
        # Круглая кисть
        noStroke()
        fill(current_color)
        ellipse(x, y, current_size, current_size)

    elif brush_type == 1:
        # Квадратная кисть
        noStroke()
        fill(current_color)
        pushMatrix()
        translate(x, y)
        rotate(frameCount * 0.05) # Легкое вращение
        rectMode(CENTER)
        rect(0, 0, current_size, current_size)
        popMatrix()

    elif brush_type == 2:
        # Звездная кисть
        noStroke()
        fill(current_color)
        draw_star(x, y, current_size/2, current_size/4, 5)

    # Сохраняем в историю каждые 10 кадров для одного касания
    if frameCount % 10 == touch_id % 10:
        save_to_history()

def draw_star(x, y, radius1, radius2, npoints):
    """Рисует звезду"""
    angle = TWO_PI / npoints
    half_angle = angle / 2.0
    beginShape()

```

```

for i in range(npoints * 2):
    r = radius1 if i % 2 == 0 else radius2
    px = x + cos(i * half_angle) * r
    py = y + sin(i * half_angle) * r
    vertex(px, py)
endShape(CLOSE)

def draw_interface():
    """Рисование интерфейса управления"""
    # Полупрозрачная панель внизу
    fill(50, 50, 50, 200)
    noStroke()
    rect(0, height - 100, width, 100)

    # Палитра цветов
    palette_width = min(len(palette) * 60, width - 200)
    start_x = (width - palette_width) / 2

    for i, col in enumerate(palette):
        x = start_x + i * 60
        y = height - 70

        # Рамка для текущего цвета
        if col == current_color:
            stroke(255)
            strokeWeight(3)
        else:
            noStroke()

        fill(col)
        ellipse(x, y, 40, 40)

    # Индикатор кисти
    fill(current_color)
    stroke(255)
    strokeWeight(2)

    brush_x = width - 150
    brush_y = height - 70

    if brush_type == 0:
        ellipse(brush_x, brush_y, brush_size, brush_size)
    elif brush_type == 1:
        rectMode(CENTER)
        rect(brush_x, brush_y, brush_size, brush_size)
    elif brush_type == 2:
        draw_star(brush_x, brush_y, brush_size/2, brush_size/4, 5)

    # Информация
    fill(255)
    noStroke()
    textSize(16)
    textAlign(CENTER)
    text("Кисть: " + str(brush_size), brush_x, height - 30)

    # Кнопки управления слева
    draw_control_buttons()

def draw_control_buttons():
    """Рисует кнопки управления"""
    button_size = 60

    # Кнопка очистки

```

```

draw_button(50, height - 70, button_size, "C", color(255, 100, 100))

# Кнопка отмены
draw_button(120, height - 70, button_size, "←", color(100, 100, 255))

# Кнопка сохранения
draw_button(190, height - 70, button_size, "S", color(100, 255, 100))

# Кнопка смены кисти
draw_button(260, height - 70, button_size, "B", color(255, 255, 100))

def draw_button(x, y, size, label, col):
    """Рисует круглую кнопку"""
    fill(col)
    stroke(255)
    strokeWeight(2)
    ellipse(x, y, size, size)

    fill(0)
    noStroke()
    textAlign(CENTER, CENTER)
    textSize(20)
    text(label, x, y)

def mousePressed():
    """Обработка кликов для интерфейса (для тестирования на ПК)"""
    check_interface_click(mouseX, mouseY)

def touchStarted():
    """Обработка мультитач касаний"""
    for touch in touches:
        check_interface_click(touch.x, touch.y)
    return False

def check_interface_click(x, y):
    """Проверяет клик по элементам интерфейса"""
    global current_color, brush_type, brush_size, drawing

    # Проверка палитры
    palette_width = min(len(palette) * 60, width - 200)
    start_x = (width - palette_width) / 2

    for i, col in enumerate(palette):
        px = start_x + i * 60
        py = height - 70

        if dist(x, y, px, py) < 20:
            current_color = col
            # Если выбрали белый - включаем режим ластика
            drawing = (col != color(255))
            return True

    # Проверка кнопок управления
    button_size = 30 # Радиус кнопки

    # Очистка
    if dist(x, y, 50, height - 70) < button_size:
        clear_canvas()
        return True

    # Отмена
    if dist(x, y, 120, height - 70) < button_size:
        undo_last()

```

```

        return True

# Сохранение
if dist(x, y, 190, height - 70) < button_size:
    save_drawing()
    return True

# Смена кисти
if dist(x, y, 260, height - 70) < button_size:
    change_brush()
    return True

# Изменение размера кисти жестом (два пальца)
if len(touches) == 2:
    # Вычисляем расстояние между двумя пальцами
    touch1 = touches[0]
    touch2 = touches[1]
    distance = dist(touch1.x, touch1.y, touch2.x, touch2.y)

    # Масштабируем размер кисти
    brush_size = constrain(distance / 5, 5, 100)
    return True

return False

def clear_canvas():
    """Очищает холст"""
    background(background_color)
    stroke_history.clear()
    stroke_history.append(get())

def undo_last():
    """Отменяет последнее действие"""
    if len(stroke_history) > 1:
        stroke_history.pop() # Удаляем текущий
        # Восстанавливаем предыдущий
        image(stroke_history[-1], 0, 0)
    else:
        clear_canvas()

def save_to_history():
    """Сохраняет текущее состояние в историю"""
    if len(stroke_history) >= max_history:
        stroke_history.pop(0) # Удаляем самый старый
    stroke_history.append(get())

def save_drawing():
    """Сохраняет рисунок в файл"""
    from datetime import datetime
    filename = "drawing_" + datetime.now().strftime("%Y%m%d_%H%M%S") + ".png"
    save(filename)
    print("Рисунок сохранен как:", filename)

    # Визуальное подтверждение
    fill(100, 255, 100, 200)
    noStroke()
    rect(width/2 - 100, height/2 - 25, 200, 50)
    fill(0)
    textAlign(CENTER, CENTER)
    text("Сохранено!", width/2, height/2)

def change_brush():
    """Меняет тип кисти"""

```

```

global brush_type
brush_type = (brush_type + 1) % 3

# Визуальная обратная связь
fill(255, 255, 100, 200)
noStroke()
rect(width/2 - 100, 50, 200, 50)
fill(0)
textAlign(CENTER, CENTER)
brush_names = ["Круглая", "Квадратная", "Звездная"]
text("Кисть: " + brush_names[brush_type], width/2, 75)

def display_instructions():
    """Отображает инструкции при запуске"""
    fill(0, 0, 0, 150)
    noStroke()
    rect(0, 0, width, height)

    fill(255)
    textAlign(CENTER, CENTER)
    textSize(32)
    text("МУЛЬТИТАЧ ХОЛСТ", width/2, height/2 - 50)

    textSize(20)
    text("• Касание одним пальцем - рисование", width/2, height/2)
    text("• Два пальца - изменение размера кисти", width/2, height/2 + 30)
    text("• Нижняя панель - цвета и инструменты", width/2, height/2 + 60)
    text("Коснитесь экрана для начала", width/2, height/2 + 120)

def keyPressed():
    """Обработка клавиатуры (для отладки)"""
    if key == 'c' or key == 'C':
        clear_canvas()
    elif key == 's' or key == 'S':
        save_drawing()
    elif key == 'z' or key == 'Z':
        undo_last()
    elif key == 'b' or key == 'B':
        change_brush()
    elif key == ' ':
        # Пробел - случайный цвет
        global current_color
        current_color = color(random(255), random(255), random(255))

# Жесты для мультитач (опционально - для расширения)
def touchMoved():
    """Обработка движения касаний"""
    # Можно добавить специальные жесты
    return False

def touchEnded():
    """Обработка окончания касания"""
    # Сохраняем в историю при окончании штриха
    save_to_history()
    return False

```

Особенности реализации для мультитач:

1. **touches массив** - встроенная переменная Processing для работы с мультитач
2. **Давление (pressure)** - используется для динамического изменения размера кисти
3. **Жесты:**

- Один палец - рисование
 - Два пальца - изменение размера кисти (pinch/zoom жест)
4. **Интерфейс для пальцев:**
- Крупные элементы управления
 - Минимальное расстояние между кнопками
 - Визуальная обратная связь
-

Расширения и улучшения:

1. **Добавить больше жестов:**
 - Три пальца - панорамирование холста
 - Четыре пальца - поворот кисти
 2. **Дополнительные инструменты:**
 - Градиентная заливка
 - Текст
 - Геометрические фигуры
 - Штампы
 3. **Сетевые функции:**
 - Совместное рисование по сети
 - Экспорт в социальные сети
 4. **Анимация:**
 - Анимированные кисти
 - Эффекты частиц
 5. **Настройки:**
 - Сохранение пользовательских палитр
 - Настройка чувствительности
-

Тестирование на ПК:

Если мультитач экран недоступен, программа работает с мышью:

- ЛКМ - рисование
- Клавиши C, S, Z, B, Space - управление

Это решение полностью функционально и может быть развернуто на любом мультитач экране с поддержкой Processing!