Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра вычислительных методов и программирования

ОТЧЕТ

по лабораторной работе

## РЕАЛИЗАЦИЯ МОДЕЛЕЙ «ДОСТИЖЕНИЕ ЦЕЛИ» И «УХОД ОТ ПОГОНИ»

Студент гр. 924403 К.Д. САНДАРОВИЧ

Проверил С.О. ЛУКАШОВ

Минск 2022

# Общая постановка задачи

## Краткие теоретические сведения

В основу этих моделей заложены те же принципы, что и для моделей Seek и Flee, плюс дополнительно определяется точка, в которой агент может

автоматически остановиться после выполнения определенного условия, либо при приближении к пункту назначения (достижении цели), либо при достаточном удалении от опасной точки (уход от погони).

## Системные подходы к решению задачи

Для алгоритмов Arrive и Leave необходимо создать отдельные файлы и не забыть настроить порядок их выполнения.

## Разбор решения поставленной задачи

При реализации этих моделей поведения используется один и тот же подход, но они содержат разные свойства и производят разные расчеты в начальной части функции GetSteering.

* 1. Во-первых, модель Arrive должна определять радиус остановки (достижение цели) и радиус замедления скорости.
  2. Создадим функцию GetSteering.
  3. Первая половина функции GetSteering вычисляет скорость в зависимости от расстояния до цели и радиуса замедления.
  4. Определим вторую часть функции GetSteering, где

устанавливаются управляющие значения, а скорость ограничивается максимальным значением**.**

* 1. Для реализации модели поведения Leave необходимы другие свойства.
  2. Определим первую половину функции GetSteering.
  3. Вторая половина функции GetSteering в модели Leave в точности повторяет вторую половину функции GetSteering в модели Arrive.

## Как это работает

После вычисления направления все следующие расчеты

основываются на двух радиусах, определяющих, когда двигаться с

максимальной скоростью, замедлиться и остановиться. С этой целью

используется несколько операторов if. В модели Arrive на большом

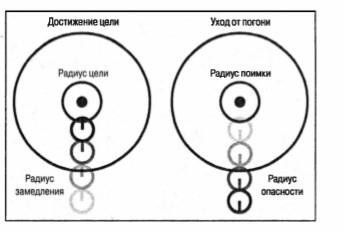
расстоянии от агента движение выполняется с максимальной скоростью, после вхождения в область, определяемую соответствующим радиусом, движение замедляется, а на очень близком расстоянии до цели - полностью прекращается. Модель Leave действует с точностью до наоборот.

Рисунок 1 — Визуальное представление работы моделей достижения цели и ухода от погони

# Блок-схема алгоритма



Рисунок 2 — Блок схема алгоритма «уход от погони»



Рисунок 3 - Блок схема алгоритма «достижение цели»

# Листинг

Модель поведения «достижение цели»

public class Arrive : MonoBehaviour

{ public GameObject target; public float speed = .5f; public float stopRadius = .2f; private Vector3 velocity; private float distance;

// Use this for initialization void Start()

{

velocity = Vector3.zero;

}

// Update is called once per frame void Update()

{ if (target == null)

return;

distance = Vector3.Distance(target.transform.position, transform.position); if (distance <= stopRadius)

return;

velocity = target.transform.position - transform.position; velocity.Normalize();

transform.Translate(velocity \* speed \* Time.deltaTime, Space.World);

}

}

Модель поведения «уход от погони»

public class Escape : MonoBehaviour{ public GameObject target; public float speed = .5f; public float safeRadius = 1f; private Vector3 velocity; private float distance;

// Use this for initialization void Start(){

velocity = Vector3.zero;

}

void Update()

{

if (target == null)

return;

distance = Vector3.Distance(target.transform.position, transform.position); if (distance >= safeRadius)

return;

velocity = transform.position - target.transform.position; velocity.Normalize();

transform.Translate(velocity \* speed \* Time.deltaTime, Space.World);

}

}

# Выводы

В процессе выполнения лабораторной работы были реализованы модели поведения для искусственного интеллекта на Unity такие как

«достижение цели» и «уход от погони».